



Ingenieur fakultät Bau Geo Umwelt
Lehrstuhl für Computation in Engineering

Kollaborative Modellierung und simulationsgestützte Evaluierung
trassenbasierter Infrastrukturbauwerke

Matthias Heinrich Josef Flurl

Vollständiger Abdruck der von der Ingenieur fakultät Bau Geo Umwelt der
Technischen Universität München zur Erlangung des akademischen Grades
eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. André Borrmann

Prüfer der Dissertation:

1. Prof. Dr. rer. nat. Ernst Rank
2. Prof. Dr. rer. nat. Martin Breunig
Karlsruher Institut für Technologie
3. Prof. Dr.-Ing Stephan Freudenstein

Die Dissertation wurde am 24.05.2016 bei der Technischen Universität Mün-
chen eingereicht und durch die Ingenieur fakultät Bau Geo Umwelt am
26.07.2016 angenommen.

Zusammenfassung

In der vorliegenden Arbeit werden Konzepte und Methoden zur Verbesserung der Kooperation im Planungsprozess von trassenbasierten Infrastrukturbauwerken vorgestellt. Dabei ist die Konzeption einer Kollaborationsplattform wesentlich, die im Zentrum dieses Prozesses steht. Hier ermöglicht diese zunächst die synchrone geometrische Modellierung durch verschiedene Experten, die bei der Erstellung eines räumlichen Bauwerkmodells prinzipiell die CAD-Werkzeuge ihrer Wahl verwenden dürfen. Durch die Einführung verschiedener Levels of Detail werden sie gemäß der Abstraktionsstufe ihrer Planungsaufgabe unterstützt. Die Plattform selbst orchestriert diesen kooperativen Prozess und sichert die Konsistenz zwischen dem zentralen Modell und seinen lokalen Replikaten auf der jeweiligen Benutzerseite. Die in der Planung benötigten und heute häufig durch *Geo Web Services* bereitgestellten Geoinformationen können mit Hilfe der Plattform generisch in den Planungsprozess und die entsprechenden Werkzeuge integriert werden. Ob ein Bauwerk bestimmten, etwa durch gesetzliche Vorgaben oder Richtlinien gegebenen Kriterien genügt, obliegt dem Urteil eines entsprechenden Experten. In vielen Fällen können bei dieser Beurteilung benötigte Kenntnisse mit Hilfe von simulationsgestützten Berechnungen erzielt werden. Eine enge Verzahnung zwischen Modellierungs- und Simulationswerkzeugen ist somit wünschenswert und kann etwa durch eine direkte Integration der Simulationen in die Plattform erreicht werden. Exemplarisch soll dazu die Integration zweier Simulationstypen vorgestellt werden. In beiden Fällen ist die Integration dabei so konzipiert, dass Resultate der Berechnungen wieder direkt in den Modellierungsprozess einfließen können: Zum einen wird die Integration einer Energiesimulation von U-Bahnzügen beschrieben. Diese berechnet automatisiert die Änderungen im Energieverbrauch, die infolge von Änderungen in der Trassierung entstehen, die dem Bauwerk zugrunde liegt. Zum anderen wird eine Strategie zur Integration von Personenstromsimulatoren diskutiert. Diese sind bei der Bewertung von Bahnhofsbauwerken sowohl hinsichtlich des Alltagsbetriebs als auch möglicher Evakuierungen von enormer Bedeutung und arbeiten im vorgestellten Ansatz direkt auf den räumlichen Geometriemodellen, die über die zu beschreibende Plattform modelliert und auf dieser hinterlegt wurden.

Abstract

In this thesis, concepts and methodologies will be presented that provide possibilities to improve the cooperation in the planning process of alignment-based infrastructure facilities. An essential element of this research is the conception of a collaboration platform, which orchestrates this planning process from a central position. This platform enables a synchronous geometric modelling process. Thereby, it allows the various experts to use the CAD-modelling tools of their choice while creating a spatial model of an infrastructure facility or building. Here, they are supported by the definition of different Levels of Detail, which allow them to

view a model according to the abstraction level of their specific modelling task. The platform itself orchestrates this collaborative planning process and ensures the consistency between the central hosted model and its local replicas on the client side. Often today, the geoinformation needed in this planning process are provided by *Geo Web Services*. Thus, a strategy will be shown that facilitates a generic integration of these *Services* into the planning process via the collaboration platform. Whether a building satisfies given legal standards and guidelines or it does not is the task of the specific expert to decide. In many cases, simulation based calculation results can support these experts in their tasks. Thus, a strong interleaving between modelling and simulation tools is desirable and can be achieved by a direct integration of the simulation tools into the collaboration platform. Here, two examples for the integration of simulation tools will be presented. In both cases, the integration is realized in a way that calculation results can be directly (re-)used within the modelling process: On the one hand, the integration of an energy simulation of subway trains will be presented, which automatically recalculates the energy consumption if the underlying track alignment changes; on the other hand, a concept to integrate pedestrian flow simulators will be shown. These simulators are of major importance in the evaluation of station buildings regarding the daily operations as well as possible evacuation scenarios. In the given approach, they directly use the spatial geometry models, created via and stored into the collaboration platform.

Vorwort

Die vorliegende Arbeit entstand im Rahmen meiner Abordnung durch das Bayerische Staatsministerium für Bildung und Kultus, Wissenschaft und Kunst vom Wilhelmsgymnasium in München an die Technische Universität München. Von August 2011 bis Juli 2016 durfte ich mich dort am Lehrstuhl für Computation in Engineering im Rahmen eines DFG-Projekts als Wissenschaftlicher Mitarbeiter in Forschung und Lehre engagieren. An dieser Stelle möchte ich mich zunächst bei allen herzlich bedanken, die mich während dieser Tätigkeit und beim Verfassen dieser Arbeit unterstützt haben.

Zuallererst möchte ich Herrn Prof. Dr. rer. nat. Ernst Rank ganz besonders danken, der zunächst durch sein enormes Engagement gegen alle bürokratischen Hindernisse meinen vorübergehenden Wechsel vom Wilhelmsgymnasium an die Technische Universität ermöglichte. Die hervorragenden Rahmenbedingungen an seinem Lehrstuhl, insbesondere das menschlich wie fachlich exzellente Team entstanden und entstehen durch seinen unendlichen Einsatz und trugen ganz entscheidend zur fortwährenden Freude an meiner Tätigkeit bei. Sein stetes Interesse an meiner Forschung und die zahlreichen intensiven Diskussionen über den aktuellen Fortschritt und die erzielten Ergebnisse trugen in ganz erheblichem Maße zur Fertigstellung dieser Arbeit bei.

Mein Dank gilt Herrn Prof. Dr. rer. nat. Martin Breunig für die rasche und unkomplizierte Übernahme des Zweitgutachtens. Die vielen intensiven Gespräche sowohl im Rahmen der DFG-Forschergruppe *3DTracks*, in welcher er als Sprecher fungierte, als auch insbesondere in der Abschlussphase meiner Promotion waren mir stets ein großer Ansporn.

Für die spontane und unkomplizierte Übernahme des Drittgutachtens bedanke ich mich herzlich bei Herrn Prof. Dr.-Ing. Stephan Freudenstein.

Innerhalb des Lehrstuhls war ich der Gruppe „Efficient Algorithms“ zugeordnet, die von Herrn Dr. (habil) Ralf-Peter Mundani geleitet wird. Ihm gilt mein aufrechter Dank für seinen immensen Einsatz bei meiner Betreuung. In den knapp fünf Jahren meiner Tätigkeit verging wohl keine Woche (außerhalb der Urlaubs- und Konferenzzeit), in der wir uns nicht zumindest ein- oder zweimal getroffen haben, um aktuellen Strategien zu diskutieren und zu optimieren.

Herzlich bedanke ich mich auch bei Herrn Prof. Dr.-Ing. André Borrmann, der als Co-Sprecher der DFG-Forschergruppe stets auf den Erfolg aller Gruppen geachtet hat. Die wöchentlichen Jours Fixes und die häufigen intensiven persönlichen Diskussionen waren eine bedeutsame Hilfe für diese Arbeit.

Auch bei allen meinen Kollegen und Kolleginnen am Lehrstuhl für Computation in Engineering sowie am Lehrstuhl für Computergestützte Modellierung und Simulation möchte ich mich ganz herzlich bedanken. Mein besonderer Dank gilt hier meinen Bürokollegen und meinem vierjährigen direkten Gegenüber Herrn Dipl.-Ing. Javier Ramos Jubierre M. Sc. Die vielen offenen Gespräche und Diskussionen waren stets hilf- und lehrreich.

Zu guter Letzt möchte ich mich bei meiner Familie herzlich bedanken, hier zunächst bei meinen Eltern, die sicherlich den Grundstein für meine Interessen und vor allem mein Interesse an akademischen Inhalten gelegt haben. Meinem Vater danke ich für das sehr aufmerksame Lesen dieser Arbeit und die wertvollen Tipps zur Korrektur, die er mir dabei gegeben hat. Von ganzem Herzen danke ich auch meiner lieben Frau Germaine, die mir steten Rückhalt während meiner Tätigkeit an der TUM und innerhalb der Entstehungsphase dieser Arbeit gewährte und stets auch darauf bedacht war, meinen Fokus während dieser hoch interessanten, aber ebenso anstrengenden Lebensphase auch auf andere Aspekte des Alltags und der Lebenswirklichkeit zu richten.

Der Deutschen Forschungsgemeinschaft (DFG) danke ich für ihre finanzielle Unterstützung durch die Forschergruppe 1546 „Rechnergestützte kooperative Trassenplanung in mehrskaligen 3D-Stadt- und Bauwerksmodellen“, in deren Rahmen ein großer Teil dieser Arbeit entstanden ist.

Abschließend möchte ich mich bei der Metro Bilbao bedanken, die mir grundlegende Daten und Informationen ihres U-Bahnnetzes zur Validierung des in dieser Arbeit zu präsentierenden Energiemodells zur Verfügung stellte, sowie bei der Obermeyer Planen + Beraten GmbH und hier insbesondere bei Herrn Dipl.-Ing. Markus Hochmuth, der stets bereit war, mich mit Erfahrungen und Tipps aus der Planungspraxis zu unterstützen.

Inhaltsverzeichnis

Zusammenfassung	2
Abstract	2
Vorwort	4
1. Einführung	9
1.1. Motivation und Ausgangspunkt	9
1.1.1. Synchroner kollaborativer geometrischer Modellierung.....	10
1.1.2. Integration von Geoinformationen	12
1.1.3. Integration verschiedener Simulationsmethoden	12
1.2. Aufbau der Arbeit	13
2. Computergestützte kooperative Arbeit und Modellierung.....	15
2.1. Computergestützte kooperative Arbeit	15
2.2. Arten der computergestützten kooperativen Arbeit	17
2.3. Computergestützte geometrische Modellierung	19
2.4. Computergestützte kooperative Arbeit in der geometrischen Modellierung	23
2.5. Zusammenfassung.....	30
3. Die Kollaborationsplattform Kollabserver.....	31
3.1. Grundlegende Zielsetzung	31
3.2. Die grundlegende Architektur der Kollaborationsplattform	32
3.3. Prozedurale geometrische Modellierung	39
3.4. Ein multi-skaliges prozedurales Geometriemodell	45
3.5. Der Austausch von Geometriemodellen	48
3.6. CAD-Modellierung über die Plattform Kollabserver	50
3.6.1. Benutzerverwaltung	50
3.6.2. Zugriff auf die am Server hinterlegten Modelle.....	51
3.6.3. Transformationen neutraler Modelle in proprietäre	52
3.6.4. Architektur der Modellierungsclients.....	53
3.6.5. Der konkrete Modellierungsprozess.....	54
3.7. Nebenläufigkeitskontrolle durch die Kollaborationsplattform	56
3.7.1. Begriffsdefinitionen, Arten der Nebenläufigkeitskontrolle	56

3.7.2.	Der prozedurale Modellierungsprozess als bipartiter Graph.....	60
3.7.3.	Ableitung des Sperr-Abhängigkeitsgraphen	64
3.7.4.	Ein graph-basierter Sperr-Algorithmus.....	66
3.7.5.	Sperrungen über verschiedene Levels von Detail hinweg	69
3.8.	Praxistauglichkeit des prozeduralen Modells	70
3.9.	Regelbasierte prozedurale Modellierung mit Unterstützung der Kollaborationsplattform	72
3.9.1.	Problembeschreibung anhand zweier Beispiele	72
3.9.2.	Regelbasierte Systeme.....	74
3.9.3.	Integration von Regeln in die Kollaborationsplattform	75
3.9.4.	Die Kombination verschiedener Regeln	77
3.10.	Zusammenfassung.....	78
4.	Integration von Geoinformationen – Geo Web Services	80
4.1.	Geoinformationen im Planungsprozess von Infrastrukturbauwerken.....	80
4.2.	Geo Web Services – Web Map, Web Feature und Web Processing Service.....	81
4.3.	Kollaborative Nutzung von Geo Web Services	83
4.4.	Integration von Geo Web Services in CAD-Werkzeuge	84
4.5.	Web Service Context Documents – eine generische Beschreibung von Geo Web Services	85
4.6.	Eine Containerstruktur zur Integration von Geo Web Services.....	86
4.7.	Ein Beispiel für die Integration eines Web Map Service.....	92
4.8.	Zusammenfassung.....	93
5.	Das Alignmentmodell – Grundlage der Infrastrukturplanung	95
5.1.	Mathematische Grundlagen	96
5.1.1.	Der Lageplan	96
5.1.2.	Die Klothoide als Übergangskurve im Lageplan	98
5.1.3.	Der Höhenplan	99
5.1.4.	Die Parabel als Übergangselement im Höhenplan.....	100
5.2.	Formate zum Austausch von Alignmentmodellen.....	102
5.3.	Integration von Alignmentmodellen in die Kollaborationsplattform	103
5.4.	Teilmodellerzeugung basierend auf dem Alignmentmodell.....	108

5.5.	Zusammenfassung.....	112
6.	Einbindung von Simulationssoftware	114
6.1.	Energiesimulation auf Basis des Alignmentmodells	114
6.1.1.	Modelle für die Berechnung der Energieeffizienz von Schienenfahrzeugen...	115
6.1.2.	Ein physikalisches Modell zur Energiesimulation eines U-Bahnzuges	117
6.1.3.	Eine Architektur zur Integration des physikalischen Modells	128
6.1.4.	Darstellung der Simulationsresultate	129
6.1.5.	Validierung des Modells anhand von Messdaten der Metro Bilbao	133
6.1.6.	Manuelle Optimierung der Energieeffizienz einer Trasse	137
6.1.7.	Automatisierte Optimierung der Energieeffizienz einer Trasse.....	141
6.1.8.	Zusammenfassung Energiesimulation.....	151
6.2.	Integration von Personenstromsimulatoren	153
6.2.1.	Prozedurale Modellierung von Bahnhofsbauwerken	155
6.2.2.	Eingangsformate von Personenstromsimulatoren	157
6.2.3.	Transformation von prozeduralen Bahnhofsmoellen.....	158
6.2.4.	Exemplarische Plausibilisierung des Algorithmus.....	164
6.2.5.	Zusammenfassung zur Integration von Personenstromsimulatoren.....	170
7.	Zusammenfassung und Ausblick	171
	Synchrone Modellierung über die Plattform Kollabserver.....	171
	Einbindung von Simulationssoftware.....	172
	Ausblick.....	173
Anhang	175
	Konstruktionsbeispiel für einen typischen Trassenabschnitt.....	175
	Ein Algorithmus zum Auffinden von Zyklen im Skizzengraphen	179
Literatur	180

1. Einführung

1.1. Motivation und Ausgangspunkt

Tunnel, Brücken und Fernstraßen sind die wohl bekanntesten Beispiele für trassenbasierte Infrastrukturbauwerke. Ihre Planung ist hoch komplex und erfordert die Kooperation von Experten unterschiedlichster Fachdisziplinen. Diese Experten verwenden bei der Planung eine Vielzahl von computergestützten Werkzeugen, insbesondere bei komplizierten Berechnungs-, Modellierungs- und Simulationsaufgaben. Diese Werkzeuge sind jedoch häufig primär domänenspezifisch, weshalb sich die computergestützte Zusammenarbeit vielfach auf den Austausch der mit diesen erzeugten Ergebnisse beschränkt. Oft liegen diese in Form von gedruckten Zeichnungen, Plänen und Dateien vor, wobei letztere meist ganz spezielle Formate benutzen. Zudem spielen bei der Modellierung von Infrastrukturbauwerken nach wie vor zweidimensionale Modelle und Pläne eine entscheidende Rolle. Die dreidimensionale Modellierung hingegen setzt sich nur zaghaf durch¹.

Bei der Planung dieser Bauwerke sind Geoinformationen von grundlegender Bedeutung. Ihre unmittelbare Verfügbarkeit hat durch die Einführung und Standardisierung sogenannter *Geo Web Services* stark zugenommen. Die Integration der durch sie bereitgestellten Informationen in einen kooperativen Planungsprozess und in die entsprechenden Werkzeuge wird aber bis heute nur unzureichend unterstützt.

Bei der Bewertung eines Bauwerks hinsichtlich gesetzlicher Vorgaben und Richtlinien werden in zunehmendem Maße Erkenntnisse herangezogen, die der Experte aus simulationsgestützten Berechnungen gewinnt. Typische Simulationswerkzeuge sind häufig als mehr oder minder alleinstehende Anwendungen konzipiert, weshalb es oft enorm aufwändig ist, die Modelldaten für die verschiedenen Werkzeuge in ein geeignetes Format zu transformieren.

Die vorliegende Arbeit soll Möglichkeiten zur Verbesserung der Zusammenarbeit der verschiedenen Experten aufzeigen. Dabei werden primär drei Kernproblemstellungen diskutiert und dazu konkrete Lösungsvorschläge gegeben:

- Die Unterstützung der gemeinsamen Modellierung durch eine zentrale Kollaborationsplattform, wobei von vorneherein ein dreidimensionales Modell als Planungsbasis verwendet wird. Über unterschiedliche Levels of Detail (LoD) wird in diesem Modell berücksichtigt, dass mehrere Experten das gemeinsame Modell auf verschiedenen Abstraktionsebenen betrachten.

¹ Dies wird auch durch Herrn Dipl.-Ing. Markus Hochmuth, Fachbereichsleiter BIM bei der OBERMEYER Planen + Beraten GmbH in München, bestätigt.

- Die Möglichkeit, über die Plattform Geoinformationen, die von *Geo Web Services* bereitgestellt werden, einheitlich, dynamisch und generisch in den Planungsprozess zu integrieren.
- Die Einbindung verschiedener Simulationen in die Plattform, die den Spezialisten bei der Bewertung des Modells unterstützen. Im Rahmen dieser Arbeit werden dafür exemplarisch die Integration von Fußgängersimulationen sowie einer Energiesimulation als Anwendungen diskutiert. Entscheidend dabei ist, dass die mit den Simulationen gewonnenen Erkenntnisse über die Plattform unmittelbar im Modellierungsprozess weiterverwendet werden können.

Zunächst wird kurz auf diese drei Kernziele und die dahinterstehenden Fragestellungen eingegangen. In den darauf folgenden Kapiteln werden dann sukzessive Konzepte zur Lösung dieser Fragen vorgestellt, detailliert die im Zentrum dieser Arbeit stehende Kollaborationsplattform diskutiert und sich daraus ergebende Perspektiven und Verbesserungen in einem kollaborativen Planungsprozess präsentiert.

1.1.1. Synchrone kollaborative geometrische Modellierung

Im Zentrum eines kollaborativen² Planungsprozesses im Bauwesen steht das gemeinsame Modell des Bauwerks oder Gebäudes (Eastmann 1999, Borrmann 2007). In unserem Kontext ist dies ein dreidimensionales Geometriemodell eines trassenbasierten Infrastrukturbauwerks. Im klassischen Planungsprozess wurde ausschließlich mit zweidimensionalen Plänen im Grund- und im Aufriss gearbeitet. Dies hat sich auch durch die Einführung von computergestützten Planungswerkzeugen zunächst nicht wesentlich verändert; denn diese nahmen mehr oder minder die Rolle eines elektronischen Zeichenbretts ein, so dass die Pläne weitgehend zweidimensional blieben (Trautwein 1990, Borrmann 2007, Abulawi 2012). Neben der fehlenden Möglichkeit, das resultierende Bauwerk vollständig dreidimensional zu visualisieren, ist diese Strategie bei Planungsänderungen fehleranfällig. So ist eine Prüfung korrespondierender 2D-Pläne auf Konsistenz nicht möglich, Modellfehler können schwer oder gar nicht erkannt werden, während zugleich Änderungen in einem Plan manuell in korrespondierenden Plänen und Ansichten nachgezogen werden müssen. Bei einer von Beginn an dreidimensionalen Modellierung hingegen können die zweidimensionalen Pläne einfach als Schnitte aus dem dreidimensionalen Modell gewonnen werden (Kuhr 1991, Vajina et al. 1994, Borrmann 2007). Zudem ist die zweidimensionale Modellierung für eine spätere Einbeziehung von Simulationswerkzeugen zur Bewertung des Bauwerks völlig ungeeignet. Hier werden möglichst exakte räumliche Modelle benötigt. Typische Simulationsszenarien im Bauwesen befassen

² Die Begriffe Kooperation und Kollaboration werden innerhalb dieser Arbeit synonym verwendet, auch wenn einige Autoren hier feinsinnige Unterschiede feststellen können, die aber sicherlich keinen fundamentalen Einfluss auf die Problematik der gemeinsamen Arbeit an und für sich haben.

sich dabei mit 3D-Struktur-Berechnungen (Romberg 2005), 3D-Strömungssimulationen (van Treeck 2005, Borrmann 2007), 3D-Fluid-Simulationen (Scholz et al. 2004, Varduhn und Mundani 2014), aber auch etwa mit vibroakustischen Simulationen (Frischmann et al. 2013, Müller und Buchschmid 2014) oder der Simulation von Personenströmen (Klingsch et al. 2010, Kneidl 2013). Um diese Überlegungen hinsichtlich der dreidimensionalen Modellierung zu berücksichtigen, wurde innerhalb dieser Arbeit von vorneherein konsequent ein dreidimensionaler Modellierungsansatz verfolgt.

Im Bereich der computergestützten dreidimensionalen geometrischen Modellierung haben sich in den vergangenen Jahren und Jahrzehnten einige CAD-Werkzeuge (*Computer Aided Design*) in der industriellen Anwendung, insbesondere im Bereich des Maschinenwesens stark verbreitet. Zu den bekanntesten Werkzeugen zählen etwa Siemens NX³, PTC Creo⁴ (ehemals ProEngineer) oder Dassault Catia⁵. Einen Überblick über verschiedene Systeme und ihre wesentlichen Charakteristika findet man etwa bei (Abulawi 2012). Diese Werkzeuge sind jedoch im Wesentlichen als Stand-Alone-Produkte konzipiert worden, womit eine kollaborative Modellierung – wenn überhaupt – nur innerhalb einer Produktlinie möglich ist.

Um den verschiedenen Experten eine synchron kooperative Modellierung zu ermöglichen, wurde im Rahmen der Forschergruppe *3DTracks*⁶ zunächst ein prozedurales, multi-skaliges Geometriemodell entworfen (Jubierre und Borrmann 2013, Borrmann et al. 2014). Dieses Geometriemodell bildet die Grundlage für einen kollaborativen Modellierungsprozess über die Plattform *Kollabserver*, welche im Rahmen der Forschung des Autors dieser Arbeit konzipiert und in dieser Arbeit ausführlich diskutiert wird. Instanzen dieser Modelle werden von dieser Plattform verwaltet und den einzelnen Planern im Rahmen einer gemeinsamen Planungssitzung zur Verfügung gestellt (Flurl et al. 2012, 2014). Dabei ist diese Plattform weit mehr als ein reines Data-Repository mit Check-In- und Check-Out-Möglichkeiten, wie man es häufig in der kollaborativen CAD-Modellierung findet (Lee et al. 2010). Über öffentliche Schnittstellen gewährt sie den verschiedenen Planern einen synchronen Zugriff auf das zentral verwaltete Modell und erlaubt so ein simultanes Arbeiten. Dabei ist es den Benutzern im Gegensatz zu bisherigen Ansätzen grundsätzlich erlaubt, die Modellierungswerkzeuge ihrer Wahl zu verwenden. Durch Mechanismen zur Nebenläufigkeitskontrolle übernimmt die Platt-

³ http://www.plm.automation.siemens.com/de_de/products/nx/index.shtml

⁴ <http://de.ptc.com/product/creo>

⁵ <http://www.3ds.com/de/produkte-und-services/catia>

⁶ Die Forschergruppe *3DTracks* wurde im Frühjahr 2011 von der Deutschen Forschungsgemeinschaft (DFG) ins Leben gerufen und setzt sich allgemein mit der Verbesserung der Kooperation bei der Planung von Infrastrukturbauwerken auseinander. Das Projekt umfasst fünf verschiedene Teilprojekte, welche diese Problematik aus verschiedenen Blickwinkeln gemeinsam untersuchen. Für nähere Informationen sei auf die Projekthomepage: <http://www.3dtracks.kit.edu/index.php> verwiesen. Werden in dieser Arbeit Ergebnisse präsentiert, die in gemeinsamer Forschung erzielt wurden, so wird dies stets explizit erwähnt werden. Alle anderen Ergebnisse gehen direkt auf den Autor dieser Arbeit zurück.

form die zentrale Aufgabe der Konsistenzsicherung des zentral verwalteten Modells und seiner lokalen Replikate auf der Clientseite (Mundani 2006, Tanenbaum und van Steen 2006). Konzepte für einen synchron kooperativen Modellierungsprozess über diese Plattform sowie Strategien zur Konsistenzsicherung werden ausführlich in Kapitel 3 erläutert.

1.1.2. Integration von Geoinformationen

Für die Planung von Infrastrukturbauwerken spielen Geoinformationen offensichtlich eine grundlegende Rolle, etwa die Pläne von (unterirdischen) Bauwerken, Daten zu Boden- und Untergrundbeschaffenheit, Informationen über elektrische Leitungsnetze, Kanalrohrnetze, Liegenschaften, Eigentümer des Plan- oder Baugrundes (privat, öffentlich etc.). Diese Informationen werden heute in zunehmendem Maße durch *Geo Web Services* angeboten (Donaubauer 2004, Jones et al. 2014, Steuer et al. 2014).

Dabei kommen im Rahmen der Forschung des Autors primär drei verschiedene Typen von *Geo Web Services* zum Einsatz: *Web Map*, *Web Feature* und *Web Processing Services*. Die Kollaborationsplattform stellt den verschiedenen Planern Schnittstellen bereit, die einen einheitlichen, dynamischen und generischen Zugriff auf diese Services und die von ihnen bereitgestellten Daten erlauben. Im Rahmen eines am *3DTracks*-Projekt teilnehmenden Teilprojekts wurden Möglichkeiten für die universelle Beschreibung derartiger *Geo Web Services* erforscht. Diese Untersuchungen führten zum Entwurf sogenannter *Web Service Context Documents* (Steuer et al. 2013, 2014), welche die konkrete Grundlage für diese Schnittstellen bilden und auf der Kollaborationsplattform für die Planer zugänglich hinterlegt werden. Der eingehenden Auseinandersetzung mit dieser Thematik ist Kapitel 4 gewidmet.

1.1.3. Integration verschiedener Simulationsmethoden

In der Bauplanung hat sich in den letzten Jahren eine Vielzahl von Bereichen etabliert, in denen Simulationen⁷ Anwendung finden, etwa klassische Tragwerksberechnungen, Strömungs- und Bauablaufsimulationen sowie thermische und energetische Simulationen. Neue Szenarien und Werkzeuge werden in Forschung und Industrie entwickelt und erweitern das Portfolio dieser Werkzeuge ständig. Im Rahmen dieser Arbeit sollen passend zum gegebenen Kontext Konzepte zur Einbindung zweier Simulationstypen vorgestellt werden: zum einen die Integration einer Energiesimulation zur Berechnung der Änderung des Energieverbrauchs eines U-Bahnzuges infolge von Trassierungsänderungen, zum anderen die Integration von Personenstromsimulatoren mit dem Ziel, sowohl den Alltagsbetrieb als auch Evakuierungsszenarien in U-Bahnhöfen abzubilden.

⁷ Eine prägnante Erläuterung des Begriffs Simulation findet man etwa auf Wikipedia unter <https://de.wikipedia.org/wiki/Simulation>. Eine detaillierte Einführung findet man etwa bei (Bungartz et al. 2013).

Das Alignment- oder Trassierungsmodell⁸ bildet die Grundlage für den Entwurf von Tunneln, Brücken und Fernstraßen und beschreibt den grundlegenden Verlauf dieser Bauwerke (Kaczmarczyk 2010, Amann et al. 2014, Ji 2014). Somit bildet dieses Modell im Bereich der U-Bahn-Tunnelmodellierung eine Grundlage für den Energieverbrauch der Züge und Fahrzeuge im späteren Betrieb (Flurl et al. 2014 II, 2016). Hier können kleine Änderungen in der Trassierung über einen langen Betriebszeitraum zu signifikanten Änderungen in den Betriebskosten sowie zu Änderungen hinsichtlich der Schonung oder des Verbrauchs von Energieressourcen führen. Um den planenden Ingenieuren den Vergleich verschiedener Varianten hinsichtlich der Energieeffizienz und der resultierenden Kosten zu ermöglichen, wurde ein physikalisches Modell zur Berechnung des Energieverbrauchs entwickelt und die auf diesem Modell basierende Berechnungskomponente in die Plattform integriert. Nach einer kurzen Einführung in die Grundlagen der Alignmentmodellierung in Kapitel 5 werden dieses Modell und die damit erzielten Ergebnisse in Kapitel 6.1 vorgestellt. Insbesondere wird aufgezeigt, wie man das Höhenprofil eines U-Bahn-Tunnels hinsichtlich der Energieeffizienz verbessern oder sogar (automatisch) optimieren kann.

In der Bauplanung spielen zunehmend auch Evakuierungsszenarien eine Rolle (Kneidl 2013). Die durch Feuer, Rauch oder Überflutung bedingte Evakuierung von Bauwerken soll dafür möglichst realitätsnah simuliert werden. Ein Ziel bei der Konzeption der Kollaborationsplattform war die Integration von Personenstromsimulatoren in den Planungsprozess, die gerade diese Evakuierungsszenarien, aber auch den Alltagsbetrieb realitätsnah abbilden können. Dabei sollen diese Simulationen automatisch oder zumindest halbautomatisch angesteuert und die geometrischen Eingangsdaten direkt aus dem zentralen Geometriemodell gewonnen werden können (Flurl 2015).

1.2. Aufbau der Arbeit

In Kapitel 2 werden Grundlagen der kooperativen Arbeit zunächst allgemein dargestellt, bevor die kooperative Arbeit im Bereich der Modellierung in den Fokus gerückt wird. Dabei wird gezeigt, dass ein synchroner Modellierungsprozess unter Einbeziehung verschiedener Modellierungswerkzeuge viel Freiraum für neue Konzepte und Strategien bietet. Insbesondere gilt dies im Hinblick auf die Integration der Werkzeuge verschiedener Hersteller sowie die Verbindung verschiedener Detaillierungsstufen.

Kapitel 3 widmet sich Konzepten für den Entwurf einer Kollaborationsplattform, die den oben beschriebenen Anforderungen genügt. Hier werden notwendige Komponenten identifiziert und ihre Bedeutung im Detail diskutiert. Zudem werden Strategien für eine kooperative Mo-

⁸ Die Begriffe Alignment und Trasse sowie Alignmentmodell und Trassierungsmodell werden in dieser Arbeit synonym verwendet.

dellierung über diese Plattform aufgezeigt. Dabei werden Möglichkeiten der Nebenläufigkeitskontrolle in einem synchronen Modellierungsprozess erörtert und insbesondere ein Algorithmus präsentiert, der einen derartigen Prozess flexibel unterstützt. Als Ausblick wird abschließend ein Konzept zur Integration sogenannter kollaborativer Regeln präsentiert. Diese ermöglichen es, weiterführendes Ingenieurwissen in das zentral hinterlegte Modell zu integrieren.

Die Integration von Geoinformationen in den Modellierungsprozess bildet den Kern des Kapitels 4. Nachdem das Konzept sogenannter *Web Context View Documents* dargestellt wurde, wird gezeigt, wie diese wirkungsvoll zur Lösung der oben angeschnittenen Problemstellung eingesetzt werden können.

In Kapitel 5 wird die Alignmentmodellierung als Grundlage der Planung von Tunneln, Brücken und Fernstraßen erläutert und der Fokus zunehmend auf diesen Typ der Modellierung gesetzt. Durch eine Segmentierung des Modells gemäß der natürlichen Unterteilung im zugrunde liegenden Trassenmodell können die Möglichkeiten für eine synchrone Modellierung eines derartigen Bauwerks verbessert werden.

Zum Abschluss werden in Kapitel 6 Konzepte zur Integration der oben angesprochenen zwei Simulationstypen vorgestellt. Exemplarisch, aber wirkungsvoll belegen diese, wie man mit Hilfe der Kollaborationsplattform Möglichkeiten zur Unterstützung des Experten bei der Evaluierung des Bauwerks hinsichtlich der Energieeffizienz und des Alltagsbetriebs bezüglich des dortigen Fußgängeraufkommens sowie Evakuierungsszenarios bereitstellen kann.

Die Arbeit endet mit einer kurzen Zusammenfassung und einem Ausblick für weitere Möglichkeiten, mit Hilfe der Plattform die Kooperation einzelner Experten zu verbessern.

2. Computergestützte kooperative Arbeit und Modellierung

In diesem Kapitel befassen wir uns mit dem Begriff der computergestützten kooperativen Arbeit CSCW (*Computer Supported Cooperative Work*). Wir betrachten dabei kurz die historische Entstehung und Entwicklung des Begriffs CSCW, bevor wir verschiedene Arten von CSCW selbst klassifizieren. Insbesondere untersuchen wir in Kapitel 2.4 CSCW-Konzepte im Bereich der geometrischen Modellierung, da diese hinsichtlich der Konzeption der Plattform *Kollabserver* von grundlegender Bedeutung ist.

2.1. Computergestützte kooperative Arbeit

Den Begriff CSCW prägten Irene Greif und Paul Cashman im Jahr 1984 bei der Organisation eines Workshops, der gerade diesen Titel trug (Penichet et al. 2007). Ziel dieses Workshops war es, Experten verschiedener Fachdisziplinen zusammenzubringen und ihnen einen Erfahrungsaustausch im Bereich der Zusammenarbeit im computergestützten Arbeitsumfeld zu ermöglichen (Grudin und Poltrock 2014). Heute bezeichnet dieser Begriff allgemein ein interdisziplinäres Forschungsgebiet, das sich mit computergestützter Gruppenarbeit selbst und darüber hinaus mit Techniken und Technologien der Informatik zu ihrer Unterstützung auseinandersetzt. Dabei werden aus verschiedensten Forschungsdisziplinen Anleihen entnommen. Die wichtigsten sind nach Schwabe die Informatik, die Soziologie, die Psychologie, die Anthropologie und die Wirtschafts- und Medienwissenschaften (Schwabe 2001). Das Zusammenwirken verschiedener Disziplinen nach (Teufel et al. 1995) ist in Abbildung 1 dargestellt.

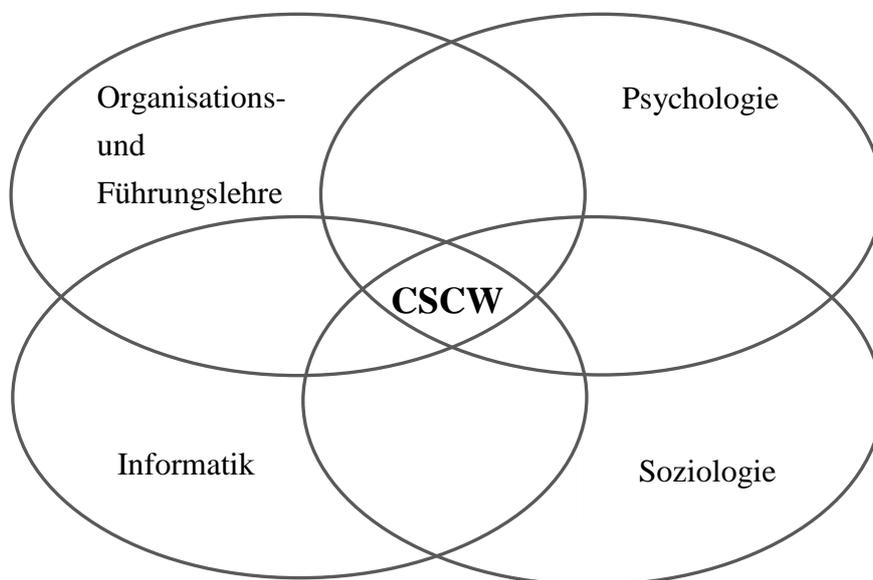


Abbildung 1: Interdisziplinarität von CSCW nach (Teufel et al. 1995)

Der Begriff CSCW selbst umfasst offensichtlich zwei Teilaspekte: zum einen den Aspekt der Computerunterstützung (*computer supported*), zum anderen den der Gruppenarbeit (*cooperative work*). Nach (Zimbardo 1992) versteht man in diesem Zusammenhang unter einer Gruppe

„zwei oder mehrere Personen, die interagieren und dabei eine gegenseitige Beeinflussung ausüben“, während die Mitglieder eines Teams an einer gemeinsamen Aufgabe arbeiten und dabei durch den Willen gekennzeichnet sind, ein gemeinsames Ziel zu erreichen (Teufel et al. 1995). Daraus leiten sie den Begriff der Gruppenarbeit ab als „die Summe aller aufgabenbezogenen Tätigkeiten, die von Gruppenmitgliedern ausgeführt werden, um zielbezogene Aufgaben zu erfüllen und somit Gruppenziele zu erreichen.“

Somit umfasst das Kerngebiet der CSCW-Forschung zum einen die Untersuchung der Gruppenarbeit an und für sich und dabei insbesondere die Interaktion der einzelnen Gruppenmitglieder, zum anderen aber natürlich die Erforschung der computergestützten Techniken und Technologien, die diese Gruppenprozesse unterstützen. Auch (Ellis et al. 1991) betonen diese zwei verschiedenen Aspekte: „CSCW looks at how groups work and seeks to discover how technology (especially computers) can help them work.“

Nahtlos fügt sich in diese Begriffsklärung der Term *Groupware* ein, der ursprünglich sogar schon kurz vor dem Begriff CSCW von (P. Johnson-Lenz und T. Johnson-Lenz 1982) als „intentional group processes plus software to support them“ definiert wurde. Heute versteht man unter Groupware allgemein Software oder aus spezifischer Hardware bestehende Systeme, welche die Gruppenarbeit unterstützen (Teufel et al. 1995). (Ellis et al. 1991) definieren Groupware als „computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment.“

Klassische Beispiele für Groupware sind elektronische Konferenzsysteme, Projektmanagementwerkzeuge oder auch Mehrbenutzer-Texteditoren. Eine ausführliche Diskussion verschiedener derartiger Systeme und Systemtypen findet man etwa bei (Teufel et al. 1995) oder (Schwabe et al. 2001). Eine Einteilung in die vier Hauptanwendungsgebiete Kommunikationssysteme, Gemeinsame Informationsräume, Workflow Managementsysteme und Workflow Computing Systeme findet sich bei (Borrmann 2007).

Entscheidend in der Definition von Groupware ist die Betonung sowohl der gemeinsamen Aufgabe als auch des gemeinsamen Ziels. Dies unterscheidet Groupwaresysteme von klassischen Mehrbenutzersystemen wie etwa dem Betriebssystem UNIX. Diese klassischen Mehrbenutzersysteme unterstützen die verschiedenen Benutzer zwar in der Erfüllung ihrer verschiedenen Aufgaben, aber die Aufgaben der einzelnen Benutzer sind separiert und werden unabhängig voneinander bearbeitet (Ellis et al. 1991). Im Alltagsgebrauch verwendet man die Begriffe CSCW und Groupware häufig synonym, während die formale Unterscheidung hinsichtlich theoretischer Grundlage und anwendungsorientierten Softwarewerkzeugen zusehends verschwindet.

	Same time	Different time
Same place	Face to face interaction	Asynchronous interaction
Different place	Synchronous distributed interaction	Asynchronous distributed interaction

Tabelle 1: Raum-Zeit Matrix nach (Johansen 1988) zur Klassifizierung kooperativer Arbeit

penarbeit, also Groupwarewerkzeuge, so können diese gerade danach klassifiziert werden, welche der obigen Kategorien sie unterstützen, also ob sie allein die Kommunikation, die Koordination und die Kooperation oder mehrere Kategorien zugleich fördern. Hinsichtlich der vorliegenden Arbeit kann die anschließend zu beschreibende Kollaborationsplattform wohl am ehesten in den Bereich der Kooperationsunterstützung eingeordnet werden, wie in der später folgenden Diskussion deutlich wird.

Eine weitere gerade im Hinblick auf die vorliegende Arbeit wichtige Unterscheidung verschiedener Groupwaresysteme ergibt sich durch die Einführung einer Raum-Zeit Matrix (Johansen 1988, Ellis et al. 1991, Hauschild 2003). In dieser Matrix (siehe Tabelle 1) führt man zunächst eine räumliche Dimension ein, in der man unterscheidet, ob verschiedene Gruppenmitglieder am selben Ort zusammenarbeiten oder nicht (verteilte Arbeit). Anschließend definiert man eine zeitliche Dimension, welche die zwei Varianten der synchronen und asynchronen Arbeit charakterisiert. Im ersten Fall findet die Arbeit (bewusst) gleichzeitig, im zweiten Fall zu unterschiedlichen Zeiten statt.

Diese Dimension charakterisieren (Ellis et al. 1991) folgendermaßen: „*Furthermore a groupware system can be conceived to enhance communication and collaboration within a real-time interaction, or an asynchronous, non-real-time interaction.*“ Dieser Zusammenhang wird üblicherweise in einer 2x2 Matrix nach (Johansen 1988) wie in Tabelle 1 dargestellt. Borrmann merkt zur räumlichen Dimension an, dass es sinnvoll sei, bereits von verteilter Arbeit zu sprechen, wenn zwei Personen zwar in demselben Raum, aber an verschiedenen Rechnern arbeiten (Borrmann 2007). Eine kritische Betrachtung der Klassifizierung anhand der Raum-Zeit Matrix findet man bei (Penichet et al. 2007), die konkrete Beispiele dafür anführen, dass viele heutige Softwarewerkzeuge für eine derartige Kategorisierung zu komplex sind.

Maher und Rutherford passen die von Johansen entworfene Matrix dem Entwurfsprozess von Bauwerken und dem Einsatz von CAD-Werkzeugen an, wie es in Tabelle 2 dargestellt ist (Maher und Rutherford 1997). Insbesondere führt dies zur Charakterisierung eines kollaborativen CAD-Werkzeugs, wie sie auch bei (Li W.D. et al. 2004) zu finden ist. Diese Charakteri-

	Same time	Different time
Same place	Single user CAD	CAD with data management
Different place	Collaborative design	Distributed CAD

Tabelle 2: Raum-Zeit Matrix nach (Maher und Rutherford 1997) zur Klassifizierung kooperativer Arbeit in der CAD-Modellierung

sierung ist im Rahmen dieser Arbeit als ein Prozess, der gleichzeitig an verschiedenen Stellen stattfindet, von großer Bedeutung (Tabelle 2 – links unten).

Firmenich und Rank führen im Falle der synchronen Kooperation zusätzlich die Unterscheidung zwischen *wechselseitiger Kooperation* und *paralleler Kooperation* ein (Firmenich und Rank 2007). Dabei betonen sie im Fall der *wechselseitigen Kooperation*, dass mehrere Bearbeiter gleichzeitig am gleichen Material arbeiten. Im Fall *paralleler Kooperation* hingegen arbeiten mehrere Bearbeiter zwar gleichzeitig (an der gleichen Problemstellung), aber an unterschiedlichen Materialien. Die an die zu beschreibende Kollaborationsplattform gestellten Anforderungen machen schnell deutlich, dass sie hinsichtlich der hier vorliegenden Klassifikationsmatrix in die Kategorie „Same time / Different place“ einzuordnen sind. Denn das erklärte Ziel ist ja gerade, die synchrone Zusammenarbeit verschiedener Experten zu unterstützen, die an verschiedenen Orten arbeiten. Weiterhin machen ihre Eigenschaften rasch klar, dass die von Firmenich und Rank geprägten Begriffe der *wechselseitigen* und der *parallelen Kooperation* beide je nach der konkreten Arbeitssituation unterstützt werden.

2.3. Computergestützte geometrische Modellierung

Die dreidimensionale Beschreibung von Körpern bildet einen Schwerpunkt des Forschungsgebiets des Solid Modelling (Mäntylä 1988), welches selbst die Grundlage für die Entwicklung der entsprechenden computergestützten CAD-Modellierungswerkzeuge ist. Die CAD-Modellierung wiederum liefert im Rahmen dieser Arbeit die Grundlage für den kollaborativen geometrischen Modellierungsprozess.

Im Bereich des Solid Modelling haben sich mehrere Beschreibungsformen von Körpern etabliert, die sich gegenseitig ergänzen. Requicha gibt die Notwendigkeit für eine Reihe von Kriterien an, die ein Beschreibungsschema für Volumenkörper erfüllen sollte, wie Vollständigkeit und Eindeutigkeit, und unterscheidet sieben grundlegende Darstellungsmethoden (Requicha 1980). Davon sind im Kontext der CAD-Modellierung die wichtigsten Methoden die BRep-Darstellung (*Boundary Representation*), die CSG-Modellierung (*Constructive Solid Geometry*) und das *Sweeping*.

Wie der Name bereits andeutet, ist die grundlegende Idee in der BRep-Darstellung die Beschreibung eines Körpers durch die ihn begrenzenden Oberflächen. Im einfachsten Fall werden diese Oberflächen als Polygone angenommen, deren Eckpunkte – genauer deren Koordinaten – in der sogenannten Knotenliste gespeichert werden. Die Polygonkanten werden über ihre Endpunkte identifiziert und in der Kantenliste gespeichert, während die zu einer Fläche gehörenden Kanten in der Flächenliste hinterlegt werden. Diese Listen ergeben einen sogenannten *vef*-Graph (englisch *vertices, edges, faces*). Die Knotenliste enthält dann die benötigten geometrischen Informationen, die Kanten- und Flächenliste hingegen die topologische Struktur des Körpers. Für ein Tetraeder ist eine (einfache) BRep-Darstellung bestehend aus Flächen-, Kanten- und Knotenliste exemplarisch in Abbildung 3 dargestellt. In der heutigen

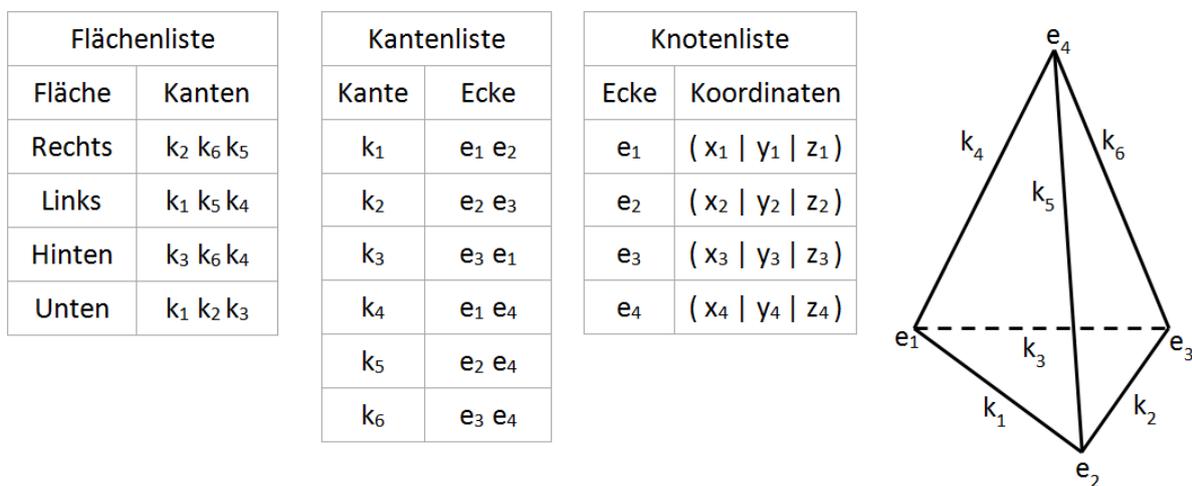


Abbildung 3: In einer BRep-Darstellung werden Körper durch ihre Oberfläche beschrieben. Grundlegend sind dabei die Flächen-, Kanten- und Knotenliste, die in dieser Abbildung konkret in den drei Tabellen für das rechts dargestellte Tetraeder angegeben sind.

Anwendung typisch für die konkrete Speicherung der Daten sind etwa die *winged-edge* oder die *half-edge* Struktur; für deren detaillierte Erklärung und die Erläuterung ihrer spezifischen Vor- und Nachteile verweisen wir auf (Baumgart 1975) bzw. (Mäntylä 1988); einen kurzen Überblick über weitere gängige Strukturen findet man bei (Stouffs 1996).

In der CSG-Modellierung dagegen wird ein Körper aus einer Menge von Grundprimitiven (etwa Würfel, Quader, Zylinder, Kugel etc.) mittels der drei Booleschen Operationen Schnitt, Subtraktion und Vereinigung erzeugt. Die elementare Funktionalität bei der CSG-Modellierung bildet der sogenannte CSG- oder Konstruktions-Baum, welcher die Schritte zur Erzeugung der Grundkörper und die einzelnen Booleschen Operationen speichert, die zum resultierenden Gesamtkörper führen. Beispielhaft sind die Operationen Vereinigung und Subtraktion für die zwei Grundprimitive Quader und Kugel in Abbildung 4 dargestellt. Da die einzelnen Operationen chronologisch im CSG-Baum gespeichert werden, werden bei der CSG-Modellierung die (hierarchische) Entstehungsgeschichte und Logik abgebildet, die den

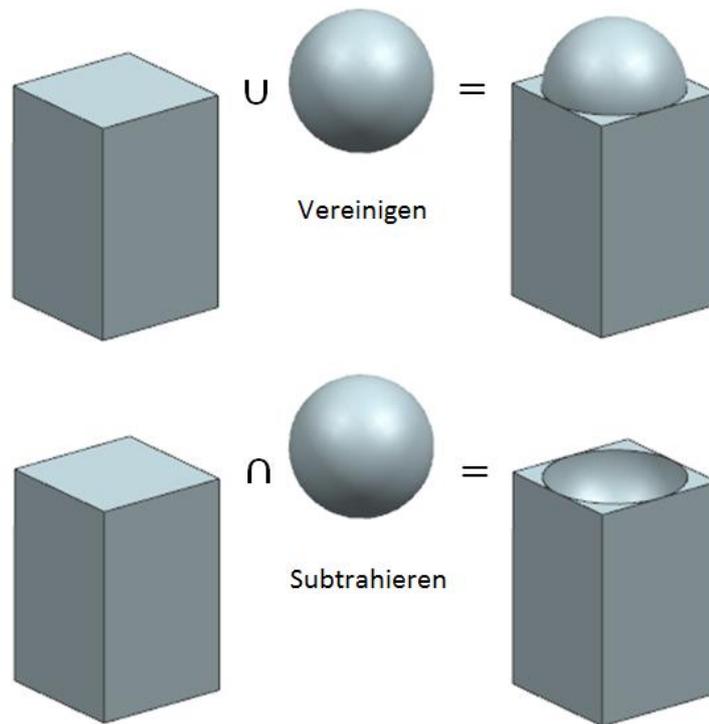


Abbildung 4: Grundlegend in der CSG-Modellierung sind die drei Booleschen Operationen: Vereinigung, Subtraktion und Schnitt. Hier sind beispielhaft Vereinigung und Subtraktion für einen Quader und eine Kugel dargestellt.

resultierenden Körper erzeugt. Somit wird im Gegensatz zur BRep-Darstellung bei der CSG-Modellierung die Entstehungsgeschichte eines Modells fundamental in die Beschreibung integriert, während die konkrete geometrische Lage erst durch die Interpretation dieser Informationen berechnet wird.

Eine dritte grundlegende und bei Modellierung von Tunneln wichtige Methode ist die Technik des *Sweeping*. Die Grundidee ist es, eine Fläche oder einen Körper entlang einer Kurve (Trajektorie) durch den Raum zu bewegen (Requicha 1980), so dass die überstrichenen Flächen oder Volumina einen Endvolumenkörper definieren. So kann man sich einen Quader auch als eine senkrechte Übereinanderschichtung infinitesimal dünner Rechtecke ähnlich wie bei der Volumenberechnung mit den Mitteln der Integralrechnung vorstellen, wobei sich die volle Stärke der Methode sicherlich erst bei komplexeren Flächen oder Trajektorien zeigt. So entstehen die im Rahmen dieser Forschung entwickelten Tunnelmodelle prinzipiell durch das *Sweeping* des Tunnelquerschnitts entlang einer dreidimensionalen Kurve (repräsentiert durch einen Spline). Ein Beispiel für das *Sweeping* eines sogenannten Maultunnelprofils zu einem einfachen Tunnelmodell ist in Abbildung 5 dargestellt.

Eine ausführliche Beschreibung dieser Konzepte und auch der dahinterstehenden mathematischen Modelle findet man etwa bei (Requicha 1980), (Mäntylä 1988) oder (Foley et al. 1997). Wir werden uns hier mit dieser knappen Beschreibung der drei wesentlichen Konzepte zufried-

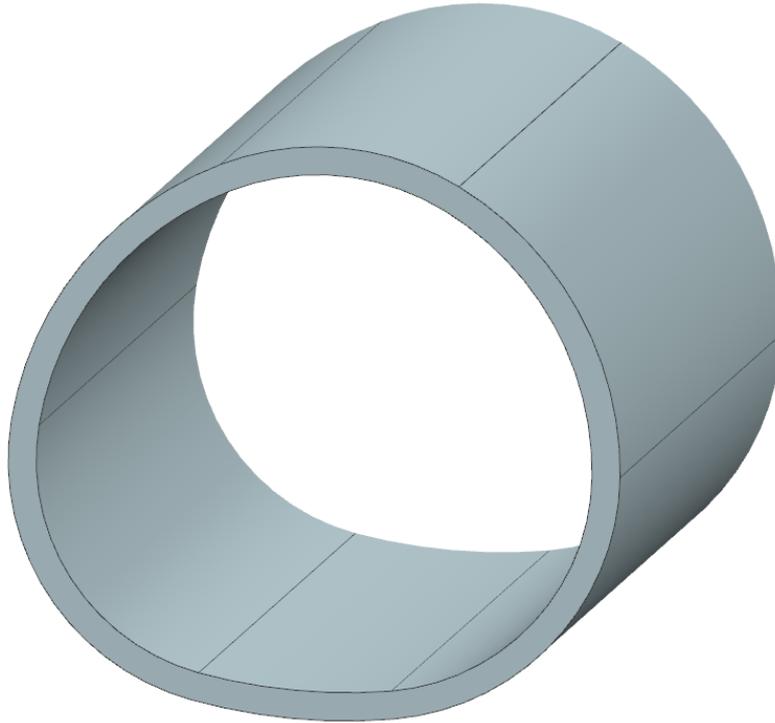


Abbildung 5: Ein einfaches Modell eines Tunnelstücks entsteht hier durch das *Sweeping* eines sogenannten Maulprofils.

dengeben. Abschließend kann hier noch angemerkt werden, dass die im Rahmen dieser Arbeit untersuchten und verwendeten CAD-Modellierungswerkzeuge wie Siemens NX und Autodesk Inventor auf einer Kombination der vorgestellten drei Darstellungsformen BRep, CSG und *Sweeping* basieren (Abulawi 2012). Andere nach (Requicha 1980) grundlegende Methoden, wie das Normzellen-Aufzählungsschema (Mäntylä 1988), spielen bei der CAD-Modellierung eine untergeordnete Rolle, sind aber als Grundlage für Simulationen von entscheidender Bedeutung. Im Allgemeinen unterstützen die im Rahmen dieser Forschung verwendeten CAD-Werkzeuge eine Konstruktionshistorie (ähnlich einem CSG-Baum), gleichzeitig benutzen sie auch BRep-Elemente wie Kanten oder Ecken von Körpern als Eingangsgrößen für weitere Konstruktionsschritte. Das *Sweeping*-Verfahren findet Eingang in Form von Extrusionsoperationen, in welchen eine zweidimensionale Skizze entlang einer Führung gezogen und so ein Volumenkörper erzeugt wird. Eine ausführliche Beschreibung konkreter CAD-Werkzeuge und der zugrunde liegenden Technologien findet man etwa bei (Abulawi 2012).

Nachdem in diesem Kapitel bisher einige grundlegende Begriffe im Bereich der computergestützten kooperativen Arbeit im Allgemeinen dargelegt wurden und kurz entscheidende Konzepte der geometrischen Modellierung angesprochen wurden, werden nun kooperative Aspekte im Bereich der geometrischen Modellierung und der Entwicklungsstand von kooperativen Modellierungswerkzeugen in dieser Domäne untersucht.

2.4. Computergestützte kooperative Arbeit in der geometrischen Modellierung

Auch im Bereich der geometrischen Modellierung⁹ spielt der Aspekt der Kooperation eine entscheidende Rolle. Dabei führen die gegen Ende des letzten Jahrhunderts immer stärker zunehmenden Tendenzen zur Globalisierung sowohl zu völlig veränderten Produktentwicklungs- als auch Produktionsprozessen. So wurden und werden der Produktentwicklungsprozess und damit auch insbesondere der Modellierungsprozess zunehmend zu verteilten kooperativen Aufgaben (Kao und Lin 1998, Ma et al. 2008). Diese Tendenz beobachtet auch (Ramani 2003): *„There have been tremendous changes in product development processes in recent years. A significant increase in the outsourcing of components necessitates increased coordination and collaboration within and across enterprises.“*

Auch (Li Min et al. 2007) teilen diese Beobachtung der Tendenz zu einem verteilten Entwicklungsprozess und gehen speziell auf die sich ändernden Anforderungen im geometrischen Modellierungsprozess ein: *„The paradigm of product development is changing with the increasing globalization of the economy and the rapid development of information technology. In recent years, more and more complex products need to be collaboratively developed by multiple departments or groups geographically dispersed. It is well recognized that this new product development paradigm requires new computer-aided design CAD approaches and tools that effectively support collaborative design activities.“*

In ihrer Forschung berichten (Li W.D. et al. 2004) über die sich verändernden Anforderungen im Bereich der geometrischen Modellierung: *„The most recent R&D cycle starting from the end of last century is to renovate CAD systems to be distributed and collaborative by using the quickly developed information technology (IT). It aims to meet the increasing demands of globally collaborative design and the outsourcing trends in manufacturing.“*

In den folgenden Untersuchungen ist für uns hauptsächlich der Stand der Technik und Forschung hinsichtlich der Kollaboration im Bereich der Modellierung von Interesse. Die Entwicklung der ersten CAD-Systeme geht in die späten Sechziger- bis frühen Siebzigerjahre des letzten Jahrhunderts zurück. Zunächst wurden CAD-Systeme als klassische Stand-Alone-Produkte konzipiert und entwickelt (Maher und Rutherford 1997, Kao und Lin 1998). Möglichkeiten für einen kooperativen computergestützten Entwicklungsprozess waren hier zunächst nicht vorgesehen. Der Ingenieur war somit als „Einzelkämpfer“ an seiner Workstation mit der Lösung der Probleme beschäftigt. Durch (Kao und Lin 1998) wird dies folgender-

⁹ In dieser Arbeit ist ausschließlich die dreidimensionale geometrische Modellierung von Interesse, so dass wir stets implizit Dreidimensionalität voraussetzen, wenn von Modellierung gesprochen wird.

maßen beschrieben: „*The design philosophy of existing CAD/CAM applications is based on the computer environments, developed in the 1970, which supported individual work in resolving the increased complexity in the engineering of new products and production facilities. Within this constraint, users cannot interact and collaborate on editing CAD models through their CAD/CAM system.*“

Ähnliches beobachten (Bidarra et al. 2001) auch noch mehr als zehn Jahre später: „*A common characteristic of most current CAD systems is that they run on powerful workstations or personal computers. Interaction with the system is usually only possible if the user is directly working at the CAD station, although remote interaction is sometimes possible through a high-bandwidth local area network.*“

Neben der historisch bedingten Konzeption der CAD-Systeme als Stand-Alone-Produkte ist zudem ein grundlegendes Problem dadurch gegeben, dass die verschiedenen Technologiemarktführer im Bereich der CAD-Modellierung von Beginn der Entwicklung an ihre proprietären Datenformate als Grundlage ihrer Systeme verwendeten und immer noch verwenden. Deshalb kann ein in einem System entworfenes Modell im Allgemeinen in einem anderen System nicht ohne Verlust von Metadaten und Detailinformationen geöffnet (Abulawi 2012), höchstwahrscheinlich nicht modifiziert oder sogar häufig überhaupt nicht geöffnet werden. Dieser Umstand schränkt die Möglichkeiten der Kollaboration zwischen Experten oder Firmen, die unterschiedliche Systeme verwenden, offensichtlich zusätzlich soweit ein, dass sie noch nicht einmal ihre Entwürfe untereinander austauschen können. Um diesem Problem beizukommen, wurde massiver Aufwand in die Erforschung und Entwicklung neutraler Datenformate gesteckt, die zumindest einen Austausch von expliziten Modelldaten zwischen verschiedenen Systemen zulassen (Choi et al. 2002). Dass dieser Datenaustausch ein Kernkonzept zur kollaborativen Produktentwicklung darstellt, wurde früh erkannt und bleibt unbestritten (Bianconi et al. 2006). Andererseits wurden neue Konzepte und Architekturen kollaborativer Modellierungssysteme entworfen, welche sich primär mit einer Abkehr von klassischen Stand-Alone-Produkten befassen. Im Folgenden werden wir beide Gesichtspunkte erläutern, uns aber auf den zweiten Punkt konzentrieren.

Hinsichtlich des Austauschs von CAD-Modellen werden häufig die Begriffe horizontaler und vertikaler Datenaustausch verwendet (Li W.D. et al. 2004, Bianconi et al. 2006). Dabei wird unter einem horizontalen Austausch, auf den wir uns fokussieren werden, der Datenaustausch zwischen verschiedenen CAD-Systemen verstanden. Als vertikal wird hingegen der Datenaustausch zwischen einem CAD-System und anwendenden CAx¹⁰-System bezeichnet (Chu C.-H. et al. 2008, Gujarathi und Ma 2011). Bereits in den Achtzigerjahren des vorherigen

¹⁰ CAx steht für Computer Aided x, wobei x als Platzhalter für verschiedene Buchstaben dient. Etwas CAE: Computer Aided Engineering oder CAP: Computer Aided Planning.

Jahrhunderts entstand mit dem Datenformat IGES (Initial Graphics Exchange Specification) ein bis heute international anerkannter und vielbeachteter Standard zum Austausch von Produktmodelldaten¹¹ zwischen verschiedenen CAD- und CAx-Systemen. Ein weiterer Meilenstein in der Entwicklung eines neutralen Austauschformats war sicherlich der IGES Nachfolger ISO 10303, besser bekannt als STEP – Standard for the Exchange of Product model data (Choi et al. 2002, Pratt und Kim 2006, Kim et al. 2007). Die Entwicklung von STEP begann Mitte der Achtzigerjahre des letzten Jahrhunderts. Ein erster Meilenstein war hier die Veröffentlichung des First Release 1994/1995 durch die ISO (International Organization for Standardization), ein zweiter die Veröffentlichung des Second Release 2002. Alle bedeutenden Hersteller dreidimensionaler CAD-Systeme bieten heute auf diesen Standards basierende Import- wie Exportfunktionalitäten an (Maier und Stumptner 2007). Diese ermöglichen zumindest einen statischen Austausch von Modellen zwischen verschiedenen Systemen (Bianconi 2005, Lee et al. 2010). Dabei verstehen wir unter einem statischen Datenaustausch die Möglichkeit, die ausgetauschten Modelle zumindest zu visualisieren, während ihre Modifikation in diesem Fall nicht oder genauer nicht hinsichtlich der Konstruktionshistorie möglich ist.

Dem Format IGES, den ersten beiden Versionen von STEP wie auch anderen weit verbreiteten Austauschformaten (VRML, STL etc.) ist jedoch gemeinsam, dass ausschließlich explizite Geometriedaten ausgetauscht werden. Denn mit diesen werden die Modelle auf der Basis von Punkten, Linien, Flächen ausgetauscht. Verloren geht so die Intention des Modellierers oder Ingenieurs, der sogenannte *Design Intent*. Dieses Manko führt dann vielfach dazu, dass ein ausgetauschtes Modell möglicherweise missverstanden wird und sicherlich schwierig oder überhaupt nicht zu modifizieren ist (Bianconi 2005, Bianconi et al. 2006). Kim et al. attestieren den beiden Austauschformaten STEP (in der Form des Second Release) und IGES nur die Fähigkeit, einfache Modelle übertragen zu können, wobei alle Informationen über den Modellierungsprozess an und für sich komplett verloren gehen (Kim et al. 2008). Diese Informationen sind gerade Teile des *Design Intent*, der nach Anderson und Ansaldi als „*some functional requirements provided by customers, that is, a set of geometric and functional rules which the final product have to satisfy*“ definiert wird (Anderson und Ansaldi 1998). Nach Kim et al. wird dieser vom Konstrukteur durch folgende Schlüsseleigenschaften in das Modell integriert: die Konstruktionshistorie, die Constraints, die Parameter und die Features. Hier umfasst die Konstruktionshistorie die einzelnen Konstruktionsschritte des Modellierers und deren Abhängigkeiten untereinander (Kim et al. 2008). Ihre Speicherung ist die Grundlage sogenannter *Prozeduraler Geometriemodelle*, die in letzterer Zeit vermehrt als Austauschgrundlage vorgeschlagen werden (Bianconi et al. 2006, Kim et al. 2008, Borrmann et al. 2014). In Kapitel 2.4 werden wir diese Modelltypen ausführlich diskutieren. Die Parameter beschreiben dabei di-

¹¹ Ein Produktmodell umfasst dabei üblicherweise nicht nur geometrie-spezifische Daten, sondern alle weiteren produkt-spezifischen.

mensionale Größen eines Modells, für die Änderungen zulässig sind. Constraints erlauben (möglicherweise parametrisierte) Beziehungen zwischen bestimmten topologischen Entitäten oder dimensional Größen. Typische Beispiele für Constraints sind neben bestimmten Längenangaben (dimensionale Constraints) Eigenschaften wie Parallelität, Senkrechtstehen, Vertikalität oder Horizontalität (topologische Constraints). Der Austausch dieser Kernkomponenten wird jedoch weder von IGES noch von STEP (in den Versionen bis zum Jahr 2005) unterstützt (Choi et al. 2002, Pratt und Kim 2006). Ziel der neuesten STEP-Version aus dem Jahr 2005 war es, sich gerade dieser Probleme anzunehmen. Das enorm komplexe Datenmodell und die schwierige Aufgabe, dafür Übersetzer zu schreiben, ist wohl die Hauptursache dafür, dass von Seiten der Industrie bisher kaum Unterstützung im Hinblick auf Import- und Exportmechanismen für STEP 2005 zu finden ist. Der Vollständigkeit halber soll hier noch das EREP (Editable Representation) Datenformat erwähnt werden, das in den Neunzigerjahren von Hoffmann entwickelt wurde (Hoffmann 1992). Hierin wurde bereits damals ein feature- oder prozedurbasierter und vom zugrunde liegenden Geometrie kern unabhängiger Austausch vorgeschlagen. Dieser Ansatz konnte sich aber niemals innerhalb der Forschungsgemeinde und Industrie durchsetzen.

Abschließend lässt sich sagen, dass bis zum heutigen Tag kein Dateiformat für den Austausch von CAD-Modellen weder in der Industrie noch in der Forschung Verbreitung gefunden hat, das neben den reinen Geometrieinformationen auch den *Design Intent* und die hierfür nötigen Informationen zufriedenstellend abbildet. Weiterhin ist der Entwurf eines allgemeingültigen Dateiformats für Geometriemodelle sicherlich ein Weg, eine engere Zusammenarbeit zwischen verschiedenen Ingenieuren zu ermöglichen. Es liegt aber in der Natur des Austausches von Dateien, dass auf diese Art und Weise zunächst nur eine asynchrone Form der Kollaboration unterstützt werden kann.

Somit werden wir uns nun auf Systeme konzentrieren, deren Ziel es von vorneherein ist, eine synchrone geometrische Modellierung zu unterstützen. Die Erforschung und Entwicklung derartiger Systeme setzte in den Neunzigerjahren des letzten Jahrhunderts ein und wurde um die Jahrtausendwende und in den Jahren danach intensiv untersucht, während es in den letzten Jahren in diesem Forschungsgebiet etwas ruhiger wurde. Der Grund für ihre Entwicklung ist der gleiche wie bei der Entwicklung von Dateiformaten zum Datenaustausch, nämlich der Wunsch, die Zusammenarbeit zwischen verschiedenen Experten zu ermöglichen und zu verbessern (Bidarra et al. 2001). Dass ihre Erforschung gerade in den Neunzigerjahren und um die Jahrtausendwende eine Hochphase erlebte, ist wohl in der enormen Steigerung der Leistungsfähigkeit von Personal Computern während dieser Zeit zu sehen, die es erlaubte, professionelle CAD-Programme mit alltagstypischen Geräten auszuführen. Gleichzeitig trugen die Steigerung der Bandbreite von Netzwerken und die immer bessere Verfügbarkeit des Internets dazu bei, dass ein Einsatz synchron kollaborativer CAD-Systeme möglich wurde. Im Folgen-

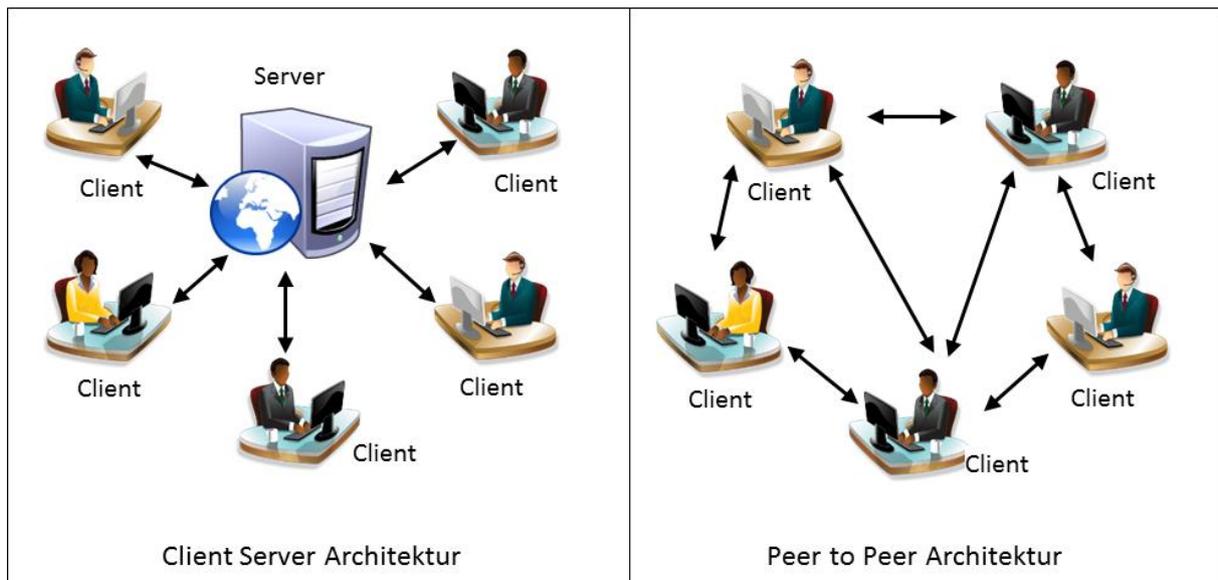


Abbildung 6: Schematische Darstellung von Client Server Architektur und Peer to Peer Architektur

den wird ein kurzer Einblick in die historische Entwicklung dieser Systeme, über die ihnen zugrunde liegenden Strategien, ihre Stärken und Schwächen gegeben.

Bei der Klassifizierung von synchronen kollaborativen CAD-Systemen ist es üblich, zwischen zwei verschiedenen Systemtypen zu unterscheiden. Die eine Gruppe bietet lediglich eine synchrone Visualisierung (*visualization based*) des gemeinsamen Modells, während die zweite eine echte gemeinsame synchrone Modellierung (*co-design*) ermöglicht (Li W.D. et al. 2004). Erstere stellt prinzipielle Möglichkeiten zur Verfügung, gemeinsam von verteilten Orten aus Modelle zu betrachten, zu beurteilen und mit Anmerkungen (*Annotations*) zu versehen. Häufig werden derartige Systeme als *Add-Ons* von bereits bestehenden Systemen oder als Web-Applikation realisiert, die mit gängigen Webbrowsern ausgeführt werden können. Somit stellen sie bereits wichtige Funktionalitäten für eine kollaborative Modellierung bereit. Die zweite Kategorie von Systemen geht über diese Möglichkeiten aber deutlich hinaus. Denn den einzelnen Teilnehmern einer kollaborativen Sitzung werden Möglichkeiten angeboten, das Modell tatsächlich von ihrem Rechner aus aktiv zu verändern. Wir fokussieren uns hier auf die zweite Kategorie von Systemen; einen Überblick über klassische Systeme der ersten Art, die auch die Grundlage für derartige heutzutage eingesetzten Systeme bilden, geben (Li Min et al. 2004).

Die echten Co-Design-Systeme lassen sich wiederum hinsichtlich ihrer grundlegenden Software- und Netzwerkarchitektur klassifizieren. So findet man einzelne Systeme, die auf einer klassischen Peer to Peer Architektur basieren (siehe Abbildung 6); häufiger sind jedoch Systeme anzutreffen, die sich auf eine Client Server Architektur stützen. Eine prägnante Beschreibung dieser beiden Architekturen findet man etwa bei (Tanenbaum und van Steen 2006). Dazu merken (Bidarra et al. 2001) an: „*The requirements for concurrency and syn-*

chronization in a collaborative modeling context lead almost inevitably to the adoption of a client-server architecture, in which the server provides the participants in a collaborative modeling session with the indispensable communication, coordination and data consistency tools, in addition to the necessary basic modeling facilities.“ So wird im Rahmen dieser Arbeit der Ansatz einer Client Server Architektur verfolgt (Kapitel 3.6). In diesem Fall wird weiter zwischen sogenannten Thin Client – Strong Server (TCSS) und Strong Client – Thin Server Systemen (SCTS) differenziert (Bidarra et al. 2001, Li W.D. et al. 2004, Li Min et al. 2004).

In einem klassischen Client Server System stellt der Server den Teilnehmern Möglichkeiten zur Kommunikation, Koordination und zur Konsistenzsicherung zur Verfügung (Bidarra et al. 2001, Mundani 2006). Die Klassifizierung der Client Server Systeme im Bereich der kollaborativen geometrischen Modellierung ergibt sich dann im Wesentlichen daraus, wo sich der eigentliche Geometriekern befindet. Im Fall einer TCSS Architektur werden die eigentlichen geometrischen Berechnungen zentral an einem Server ausgeführt, während die Client Rechner im Wesentlichen Visualisierungsaufgaben übernehmen. Diese Charakteristik beschreiben (Li Min et al. 2007) wie folgt: *„The clients are lightweight and primarily support visualization and interactive function, such as selection, transformation, changing visualization properties of displayed parts, etc. The main modeling activities are taken in a common workspace in the server side.“* Vorteile eines derartigen Systems sind sicherlich die leicht zu realisierende Synchronisierung und Nebenläufigkeitskontrolle der Clients, während ein Nachteil in der hohen Netzwerklast gesehen werden kann, sobald die Häufigkeit von Updates zunimmt.

Im Gegensatz dazu sind im Fall einer SCTS Architektur die Client-Rechner mit einem vollwertigen CAD-System ausgestattet. Der Server hat im Wesentlichen die Aufgabe, die Synchronisation der verschiedenen Clients zu orchestrieren und Kommunikationsmöglichkeiten für die einzelnen Clients zur Verfügung zu stellen. In dieser Architektur ist es prinzipiell möglich, dass verschiedene Clients auf verschiedenen CAD-Systemen operieren. Diese Eigenschaft wird ebenfalls zur Klassifizierung von Co-Design-Systemen verwendet. So bezeichnen Li Min et al. und auch Li Xiaoxia. et al. Architekturen als heterogen, in denen verschiedene Clients unterschiedliche CAD-Systeme verwenden können, während Systeme homogen genannt werden, falls alle ein und dasselbe CAD-System verwenden (Li Min et al. 2004/2007, Li Xiaoxia et al. 2009). Im Fall einer TCSS Architektur handelt es sich also prinzipiell immer um eine homogene Architektur.

Ein sehr bekanntes und viel diskutiertes Beispiel für eine TCSS Architektur ist die *WebSPIFF* Plattform, welche an der Delft University of Technology von Rafael Bidarra, Edleco van der Berg und Willem F. Broonsvoort entworfen wurde. Das Ziel dieser Plattform beschreiben deren Urheber (Bidarra et al. 2002) wie folgt: *„... is to develop a collaborative modeling system that offers all facilities of advanced modeling systems to its users, while at the same time*

providing them with the necessary coordination mechanisms that guarantee an effective collaboration.“ Die *WebSPIFF* Plattform umfasst einen sogenannten Session-Manager und einen zentralen Geometrieserver, der auf dem ebenfalls an der Delft University entwickelten *SPIFF* System basiert und einen Feature-Modellierungs-Geometriekern darstellt. Dieser Geometrieserver übernimmt zentral alle geometrischen Berechnungsaufgaben und leitet deren Ergebnisse an die Sitzungsteilnehmer weiter, während der Session-Manager die Koordinationsaufgaben übernimmt. So verwaltet er explizit alle Informationen über aktuelle Sitzungen und ist für die Nebenläufigkeitskontrolle verantwortlich. Die Clients sind als Webbrowser-Applikationen realisiert und verfügen über ausgeklügelte Visualisierungsmethoden für die über das Netzwerk transferierten Geometriedaten. Diese Clients können prinzipiell konkurrierend an ein und demselben Modell arbeiten, wobei die angebotene Nebenläufigkeitskontrolle rudimentär ausfällt: Ein oranges oder ein rotes Ampelsymbol zeigt dem Benutzer an, dass gerade ein anderer Benutzer mit einem Modellierungsschritt beginnt bzw. der Server diesen neuen Modellierungsschritt ausführt. Ein grünes Symbol gibt dagegen an, dass kein anderer Benutzer aktiv am Modell arbeitet. Daneben werden den Teilnehmern Chat-Möglichkeiten angeboten, um Ihre Schritte aktiv zu koordinieren.

Ein intensiv diskutiertes Beispiel einer SCTS Architektur stammt von (Li Min et al. 2007). Die Grundidee in ihrem Ansatz ist es, alle an einem CAD-System ausgeführten Modellierungsschritte, die als systemspezifische Operationen bezeichnet werden, in neutrale Kommandos zu übersetzen. Führt ein Benutzer einen derartigen systemspezifischen Schritt durch, so wird dieser automatisiert durch eine über die API (*Application Programming Interface*) seines CAD-Systems integrierte Bibliothek in ein neutrales Kommando übersetzt und an den zentralen Server geschickt. Dieser wiederum leitet das neutrale Kommando an alle anderen registrierten Clients weiter, die dieses unverzüglich in ein für ihr System spezifisches Kommando wiederum mit Hilfe einer geeigneten Bibliothek übersetzen und über die API ihres CAD-Systems konkret ausführen. Der große Vorteil dieses Ansatzes ist es, dass die Benutzer mit dem für sie gewohnten CAD-System arbeiten können, vorausgesetzt es wurden Übersetzerbibliotheken dafür entwickelt. Konkret wurden die Systeme SolidWorks 2003, Autodesk MDT 6.0 und ProENGINEER Wildfire 2 unterstützt. Wiederum rudimentär ist die Umsetzung einer Nebenläufigkeitskontrolle im vorgeschlagenen System, die einfach auf einem klassischen Token bzw. Floor Control Mechanismus basiert (Borghoff und Schlichter 1995, Munson 1996, Hauschild 2003). Hier wird vom Server als zentraler Instanz einem ausgezeichneten Benutzer das sogenannte Token übergeben, so dass dieser Benutzer aktiv Modellierungsschritte durchführen kann, während die anderen Benutzer nur passiv beobachtend an einer kollaborativen Sitzung teilnehmen können. Dieser vorgeschlagene Ansatz ähnelt prinzipiell dem in dieser Arbeit vorgeschlagenen Konzept, unterscheidet sich aber wesentlich hinsichtlich der Möglichkeiten der Nebenläufigkeit und der Integration der Multi-Skaligkeit bei den zugrunde liegenden geometrischen Modellen.

2.5. Zusammenfassung

Die Grundlage für die Konzeption einer Kollaborationsplattform, die den in Kapitel 1.1 geschilderten Anforderungen genügt, bilden Begriffe, Techniken und Konzepte aus dem Bereich der computergestützten kooperativen Arbeit CSCW. Die wichtigsten hier benötigten Grundlagen wurden diskutiert und kurz ihre historische Entwicklung angedeutet. Intensiv haben wir uns dabei mit der Kooperation im Bereich der geometrischen Modellierung auseinandergesetzt und ausführlich bereits existierende Strategien und Konzepte vorgestellt. Den Austausch von geometrischen Modellen haben wir erläutert und zudem einige für die Arbeit zentrale Begriffe wie den des prozeduralen Geometriemodells und des *Design Intent* eingeführt.

Im folgenden Kapitel sollen nun die wesentlichen Grundkonzepte und -komponenten der im Rahmen dieser Forschung konzipierten Kollaborationsplattform *Kollabserver* präsentiert werden. Anschließend werden wir uns auf die kollaborative Modellierung über diese Plattform fokussieren.

3. Die Kollaborationsplattform Kollabserver

3.1. Grundlegende Zielsetzung

Für die moderne Produktentwicklung ist das verteilte Arbeiten charakteristisch. Zunehmend übernehmen kleine Teams aus Spezialisten über Büro-, Gebäude- und Firmengrenzen hinweg, häufig über den gesamten Globus verteilt, einzelne spezialisierte Teilaufgaben. Ihre Resultate werden später zum großen Ganzen zusammengefügt. Eine treibende Kraft für diese Entwicklung war und ist sicherlich der Maschinenbau, insbesondere mit der klassischen Teildisziplin des Fahrzeugbaus. Aber auch das Bauwesen und darin speziell die Bauplanung werden zunehmend durch das verteilte Arbeiten geprägt. Die Arbeitsteilung im Bauwesen ist zunächst historisch bedingt. So ist nach Hauschild grundsätzlich ein Szenario typisch, in dem Experten verschiedener Fachdisziplinen gemeinsam an einem Planungsgegenstand arbeiten (Hauschild 2003). Diese verschiedenen Disziplinen umfassen etwa Architekten, Bodengutachter, HLS-Planer, Tragwerks- oder Elektroplaner. Borrmann stellt fest, dass die zunehmende Komplexität der Bauprojekte zu einer immer größeren Spezialisierung der beteiligten Planer führt (Borrmann 2007). Hieraus resultieren für ihn neue Notwendigkeiten der Arbeitsteilung, wobei eine größere Arbeitsteilung dann auch einen höheren Grad an Kooperation verlangt. Popescu et al. erkennen in der interdisziplinären Organisation von Bauprojekten und der daraus resultierenden Spezialisierung einzelner Fachplaner die Ursache für die in Kapitel 1.1 angesprochene Entwicklung fachspezifischer Softwareprodukte (Popescu et al. 2014). Diesen Entwicklungen und Problemen muss von Seiten der Bauinformatik durch die Entwicklung geeigneter Konzepte und konkreter Werkzeuge Rechnung getragen werden (Hauschild 2003). Auch an der Planung trassenbasierter Infrastrukturbauwerke sind Experten der verschiedensten Domänen beteiligt. Ihre Zusammenarbeit kann momentan als eher lose gekoppelt angesehen werden. Diese Tatsache ist mit hoher Wahrscheinlichkeit auch auf das Fehlen einer geeigneten Unterstützung in Form passender IT-Infrastrukturen zurückzuführen (Borrmann 2014 et al.). Um die Möglichkeiten der Zusammenarbeit der verschiedenen Planer zu verbessern, werden wir nun das Konzept der Kollaborationsplattform *Kollabserver* vorstellen. Diese Plattform trägt zur Lösung der zuvor angesprochenen Probleme bei. Zunächst seien nochmals die ursprünglich in Kapitel 1.1 angegebenen und grundlegenden Ziele dieser Forschung in Erinnerung gerufen:

- Die Unterstützung eines kollaborativen Modellierungsprozesses durch eine Kollaborationsplattform, welche die Geometriemodelle zentral verwaltet und diesen Prozess orchestriert.
- Die generische Integration planungsrelevanter, durch *Geo Web Services* bereitgestellter Geoinformationen in den Planungs- und Modellierungsprozess.

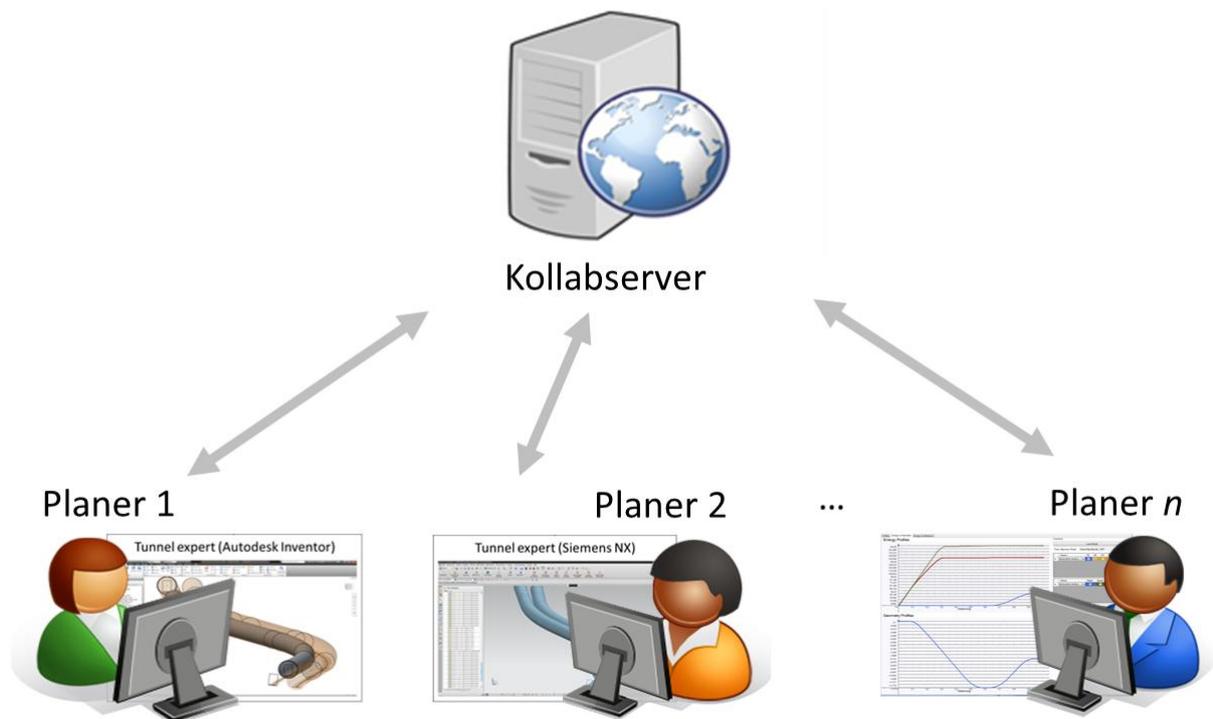


Abbildung 7: Die Plattform *Kollabservers* steht im Zentrum eines kollaborativen und synchronen Planungs- und Modellierungsprozesses. Die verschiedenen Planer dürfen dabei die Werkzeuge ihrer Wahl wie etwa Siemens NX oder Autodesk Inventor verwenden, während die Plattform diesen Modellierungsprozess orchestriert.

- Die Einbindung verschiedener Simulationen, um die Evaluation des Modells durch entsprechende Spezialisten zu unterstützen.

Selbst die vollständige Verwirklichung dieser drei Ziele löst gewiss nicht annähernd alle in diesem Zusammenhang auftretenden Probleme. Sie trägt aber sicher dazu bei, neue Perspektiven und Möglichkeiten aufzuzeigen.

3.2. Die grundlegende Architektur der Kollaborationsplattform

Die Plattform *Kollabservers* bildet den zentralen Knotenpunkt in einem Computerverbund, der im Wesentlichen gemäß einer Client Server Architektur aufgebaut ist (siehe Abbildung 7, Tanenbaum und van Steen 2006, Starke 2015). Bormann diskutiert für die von ihm entwickelte COCOS-Plattform (COCOS – Collaborative Computational Steering) ausführlich die Abweichungen ihrer Architektur von einer klassischen Client Server Architektur (Bormann 2007). So ist sich in der COCOS Architektur der Server stets über die teilnehmenden Clients bewusst und kann diesen im Laufe einer Sitzung immer wieder Nachrichten zukommen lassen. Dagegen sind in einer klassischen Client Server Architektur die Aktionen immer clientseitig bedingt, d.h. der Client stellt eine Anfrage, die der Server beantwortet, woraufhin die Verbindung zwischen beiden wieder komplett getrennt wird. Diese Abweichungen treffen auch hier weitgehend zu und sollen deshalb nicht diskutiert werden. Für eine kurze, aber

prägnante Beschreibung sowohl der Client Server als auch der Peer to Peer Architektur verweisen wir auf (Tanenbaum und van Steen 2006). Schon in Kapitel 2.4 wurde diskutiert, dass Bidarra et al. bei Anforderungen, wie sie für die Kollaborationsplattform *Kollabservers* bestehen, eine Client Server Architektur für die optimale Lösung halten (Bidarra et al. 2001). Bei der Begründung für die Verwendung einer Client Server Architektur in unserem Kontext wollen wir nur noch folgenden entscheidenden Gedanken hinzufügen: Das am häufigsten vorgebrachte Argument gegen eine Client Server Architektur besagt, dass der Server in einer derartigen Architektur infolge des Sendens und Empfangens großer Datenmengen zu einem Flaschenhals wird. Dieses Argument trifft aber für unseren Ansatz nicht zu. Denn ein schlankes prozedurales Geometriemodell (Borrmann et al. 2012/I, Borrmann et al. 2012/II, siehe auch Kapitel 3.4) sowie die Verwendung von *Web Context View Dokumenten* (Steuer et al. 2014, siehe auch Kapitel 4) für die Integration von Geoinformationen reduzieren den Datendurchsatz zwischen Client und Server stark; gleichzeitig hat die Bandbreite von Netzwerken in den letzten Jahren stark zugenommen.

Die über die Plattform interagierenden Clients¹² sind typischerweise CAD-Werkzeuge. Eine herausragende Stärke dieser Plattform besteht darin, dass die verschiedenen Planer prinzipiell als Clients die CAD-Werkzeuge ihrer Wahl verwenden können. In diese werden über deren API spezielle Bibliotheken integriert, so dass die Benutzer der Client-Werkzeuge über die Plattform kommunizieren und kooperieren können. Die Kollaborationsplattform bietet den verschiedenen Planungs- und Modellierungsexperten dazu die Möglichkeit, sich über die Client-Werkzeuge an der Plattform anzumelden und an einer kollaborativen Planungssitzung teilzunehmen. Typischerweise umfasst eine derartige (verteilte) Planungssitzung die gemeinsame Bearbeitung des zentral von der Plattform verwalteten Geometriemodells. Diese Bearbeitung erfolgt unter Zuhilfenahme von Geoinformationen, um die Planung in den gegebenen äußeren Kontext einzupassen. Zusätzlich unterstützen integrierte Simulationswerkzeuge den Planer bei der Beurteilung der Resultate einer Planungssitzung. Beim Entwurf der Plattform wurde von vorneherein auf ein modulares Konzept Wert gelegt. So können einzelne Komponenten unabhängig voneinander entwickelt, hinzugefügt und später auf einfache Art und Weise durch verbesserte Komponenten ausgetauscht werden. Die zentralen Komponenten (Flurl et al. 2012)

- Kommunikation,
- Interpreter,
- Benutzer,
- Gruppenbewusstsein (Awareness),
- Verwaltung von prozeduralen Modellen,

¹² Als Client oder Client-Werkzeug wird in dieser Arbeit immer eine Softwarekomponente, die sie verwendende Person als Benutzer, Experte oder ähnlich bezeichnet.

- Nebenläufigkeitskontrolle,
- regelbasierte prozedurale Modellierung,
- Integration von Geoinformationen,
- Verwaltung von Alignmentmodellen,
- Energiesimulation,
- Personenstromsimulation,

sowie deren Funktionalität und Beitrag zum Gesamtprozess werden im Folgenden kurz vorgestellt und beleuchtet.

Kommunikation

Das Kommunikationsmodul regelt die prinzipielle Kommunikation zwischen Client und Server. Aus technischer Sicht basiert das Modul auf einer Socket to Socket Kommunikation über das TCP/IP-Protokoll (Stevens 2004). Dabei werden prinzipiell (alpha-numerische) Zeichenketten über das Netzwerk ausgetauscht und auf der jeweiligen Seite mit speziell entwickelten Interpretern verarbeitet. Die Verwendung derartiger low-level Konzepte mag zunächst unnötig schwierig erscheinen, garantiert dem Entwickler aber die volle Kontrolle über die Schnittstellen. Dabei stellt das Kommunikationsmodul den Clients und damit den Anwendern über eine Schnittstelle einen ganz konkreten Satz von Befehlen für die Kommunikation mit dem Server zur Verfügung. Dies bedeutet zunächst etwa die Möglichkeit, sich am Server für eine kollaborative Sitzung zu registrieren („*Login*“) oder sich wieder abzumelden („*Logout*“). Hinzukommen Befehle zum Laden der verschiedenen Modelle und Modellversionen. Von besonderer Bedeutung sind die Befehle zur Bearbeitung des zentral verwalteten Geometriemodells: das Hinzufügen, das Modifizieren und das Löschen von Konstruktionsschritten („*AddOperation*“, „*ModifyOperation*“, „*RemoveOperation*“).

Interpreter

Die Interpretation der Benutzeranfragen, konkret der eingehenden textuellen Anfragen, übernimmt das Interpretermodul. Dieses wertet die Anfragen zunächst aus und delegiert sie dann an die für ihre Ausführung verantwortlichen Komponenten. Über das reine Interpretieren von einzelnen Befehlen hinaus können aber auch Befehlsfolgen und kleine Programme interpretiert werden. Insbesondere können beliebige arithmetische Ausdrücke ausgewertet werden. Diese Auswertung ist speziell bei der Verarbeitung von Parametern, welche selbst wieder Parameter in arithmetischen Ausdrücken referenzieren, von entscheidender Bedeutung. Die konkret verwendeten Techniken entstammen dem Gebiet des klassischen Compilerbaus und werden hier nicht näher erläutert. Hierfür verweisen wir den Leser etwa auf (Aho et al. 1999). Besonderer Wert wurde bei der Konzeption dieser Komponente auf die einfache Erweiterbarkeit hinsichtlich des Einbaus neuer Befehle gelegt. So entspricht in der resultierenden Software Architektur jedem Befehl genau eine sogenannte Interpreter Klasse, die eine gemeinsa-

me Interpreter Schnittstelle implementiert. Das Hinzufügen eines neuen Befehls ergibt dann gerade das Hinzufügen einer neuen Klasse, die diese Schnittstelle implementiert. Diese Erweiterung ist ohne jegliche weitere Registrierung oder manuelle Anpassung möglich. So erlaubt diese Architektur eine sehr einfache Erweiterung der Plattform und hat sich zugleich in der Praxis als sehr stabil erwiesen.

Benutzer

Um sich am Server anmelden und an einer kollaborativen Planungssitzung teilnehmen zu können, muss man über ein Benutzerkonto und die zugehörigen Zugangsdaten verfügen. Die Benutzerkomponente verwaltet diese Daten und regelt darüber hinaus über ein Rollen- und Rechte-basiertes System die Interaktionen, die ein Benutzer mit dem Server durchführen kann. Eine typische Situation dafür ist etwa die Frage, ob ein bestimmter Benutzer aktiv oder nur beobachtend an einer kollaborativen Modellierungssitzung teilnehmen darf. Dabei verwenden wir Standardtechniken zur Benutzerverwaltung, wie sie auch häufig im Bereich von Datenbank- oder Betriebssystemen zu finden sind (Sandhu et al. 1995). Deshalb werden wir den Aufbau dieses Moduls hier nicht detaillierter untersuchen.

Gruppenbewusstsein (Awareness)

Wie in Kapitel 2 diskutiert, ist es in CSCW-Werkzeugen von entscheidender Bedeutung, dass sich die teilnehmenden Benutzer der gemeinsamen Arbeit bewusst (*Awareness*) sind (Gutwin und Greenberg 1997, Kirsch-Pinheiro et al. 2003). Dazu wurden in die Plattform Funktionalitäten integriert, die den Benutzern die Aktionen anderer Benutzer aufzeigen. So werden diese über die graphische Benutzerschnittstelle über die An- und Abmeldung anderer Benutzer genauso wie über die von diesen durchgeführte Modifikationen am geometrischen Modell informiert. Explizit werden Bereiche des geometrischen Modells, die infolge der Bearbeitung durch andere Benutzer gesperrt sind, in besonderer farblicher Darstellung visualisiert. Ein Beispiel ist in Abbildung 8 dargestellt. Zusätzlich wurden Chat-Funktionalitäten integriert. Diese erlauben es den verschiedenen Benutzern etwa, sich über bestimmte Varianten und Modellierungsmöglichkeiten über die Plattform auszutauschen. Wie bei der Benutzerkomponente wurden Standardtechniken verwendet, um das System gemäß den Anforderungen eines CSCW-Systems zu komplementieren. Somit wird auch auf die Awarenesskomponente nicht detaillierter eingegangen; dazu können auch klassische Beiträge etwa von (Ellis et al. 1991) oder (Stefik et al. 1987) und hieraus resultierende Arbeiten für tiefergehende Diskussionen dieser Techniken herangezogen werden.

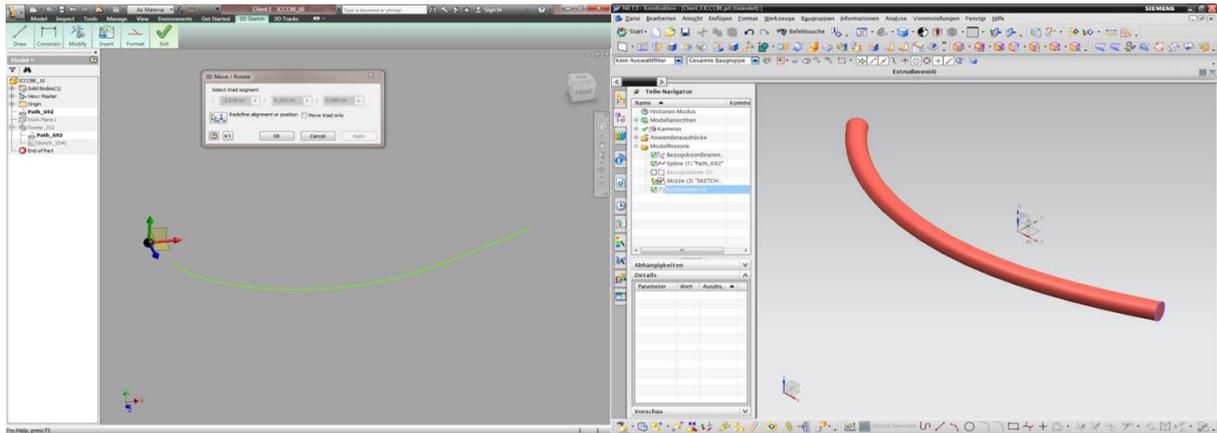


Abbildung 8: Bei der Bearbeitung eines bestimmten Elements werden dieses und die davon abhängigen Elemente für andere Benutzer gesperrt und entsprechend visualisiert.

Verwaltung von prozeduralen Modellen

Die verschiedenen prozeduralen Modelle werden in der Kollaborationsplattform hinterlegt und bilden die Basis für den kollaborativen Modellierungsprozess. Die dafür verantwortliche *Prozeduralgeometrie-Modell-Management-Komponente* (PMMK) werden wir später noch detaillierter untersuchen. Sobald ein Benutzer ein bestimmtes Modell anfordert, wird dieses beim ersten Aufruf durch einen Benutzer aus der Datenbank in den Hauptspeicher geladen, an den Benutzer weitergeleitet und im Folgenden von der PMMK administriert. Änderungen am (lokalen) Modell durch den Benutzer werden in das zentrale Modell integriert und unverzüglich in Kooperation mit dem Benutzermodul an alle anderen Bearbeiter des Modells weitergeleitet. In Zusammenarbeit mit der Komponente zur Nebenläufigkeitskontrolle wird dabei die Konsistenz zwischen dem zentralen Modell und den lokalen Replikaten auf den Client-Seiten gewahrt. In Kapitel 3.6 und speziell in Kapitel 3.7 werden die verantwortlichen Konzepte und Strategien und Mechanismen detailliert erläutert.

Nebenläufigkeitskontrolle

In einem verteilten Benutzersystem muss durch Mechanismen, die man üblicherweise mit dem Begriff der Nebenläufigkeitskontrolle zusammenfasst, sichergestellt werden, dass die Eingaben eines Benutzers nicht denen eines anderen zuwiderlaufen. In der gewählten Architektur verwenden wir einen pessimistischen Ansatz: Mit Hilfe von Sperrmechanismen werden sich widersprechende Benutzereingaben grundsätzlich unterbunden. Somit übernimmt die Kooperationsplattform die für CSCW-Systeme fundamentale Aufgabe der Konsistenzsicherung (Tannenbaum und van Steen 2006). Eine differenzierte Erläuterung der verwendeten Algorithmen und eine Erläuterung der zugrunde liegenden Theorien wird in Kapitel 3.7 gegeben.

Regelbasierte prozedurale Modellierung

Constraints unterstützen das Ziel, den *Design Intent* des modellierenden Ingenieurs in das geometrische Modell zu integrieren (Kapitel 2.3). Doch bereits bei einfachen Ungleichungen reicht dieses Prinzip nicht mehr aus. Man denke etwa an die Forderung, dass der Radius eines Kreises größer als der Radius eines anderen Kreises sein soll. Diese Forderung kann nicht mit Hilfe von gängigen Constraints abgebildet werden. Um dieses Defizit und ähnliche Probleme abzumildern, wurde ein Konzept entworfen, das es den Benutzern ermöglicht, dynamisch zur Laufzeit Regeln zu entwerfen, auf der Plattform zu hinterlegen und so in den Modellierungsprozess zu integrieren. Wir diskutieren dies als Abschluss dieses Kapitel in Abschnitt 3.9.

Integration von Geoinformationen

Geoinformationen sind grundlegend für die Planung trassenbasierter Infrastrukturbauwerke. Das Modul zur Integration der *Geo Web Services* stellt einerseits dem GIS-Experten (GIS – GeoInformationsSysteme) Möglichkeiten zur Verfügung, Dokumente zu hinterlegen, welche diese Services generisch beschreiben. Andererseits bietet es den Planern eine Schnittstelle, um hinterlegte Dokumente zu durchsuchen und mit ihrer Hilfe an die gewünschten Informationen selbst zu gelangen. Mit dieser Problematik befassen wir uns in Kapitel 4.

Verwaltung von Alignmentmodellen

Die Grundlage für den Entwurf von Tunnel- und Brückenbauwerken wie auch von Fernstraßen bildet das sogenannte Alignment- oder Trassenmodell, das unter anderem den prinzipiellen Verlauf dieser Bauwerke repräsentiert (Amann et al. 2014). Über eine geschickte Entkopplung des geometrischen Modells vom abstrakten Trassenmodell wird es dem Benutzer ermöglicht, noch in späten Entwurfsphasen Änderungen an der Trassierung vorzunehmen, die dann automatisiert an das davon abhängige Bauwerksmodell weitergereicht werden. Dies schafft grundsätzlich neue Möglichkeiten hinsichtlich späterer Veränderungen an der geplanten Trasse, wie sie in der heutigen Planungspraxis bisher nicht erlaubt sind. Der Grund für dieses Defizit heute ist, dass in der gängigen Planungspraxis das Alignmentmodell nach Fertigstellung mehr oder minder eingefroren wird, sobald mit der Modellierung von darauf basierenden Bauwerken begonnen wird. Das Konzept zur Verwaltung der Alignmentmodelle und dessen Integration in eine Kollaborationsplattform inklusive der Schnittstelle zum Geometriemodell werden in Kapitel 5 ausführlich besprochen.

Energiesimulation

Der Verlauf der Tunneltrasse hat maßgeblichen Einfluss auf den Energieverbrauch der Züge im operativen Alltagsbetrieb. Um diesen zu berechnen, wurde ein modulares Energiemodell konzipiert und eine entsprechende Berechnungskomponente in die Plattform integriert. Diese Komponente wurde an das oben beschriebene Alignmentmodul gekoppelt und reagiert autonom auf Änderungen im Trassenverlauf: Automatisch wird bei Trassierungsänderungen im Hintergrund eine Neuberechnung des Energieverbrauchs durchgeführt, und die Ergebnisse

werden an die Benutzer zurückgemeldet. Zur intuitiven Interpretation der Ergebnisse wurden eine graphische Oberfläche und verschiedene Visualisierungsmöglichkeiten entwickelt, welche den Trassierungsexperten unterstützen. Zusätzlich wurden Strategien entwickelt und Möglichkeiten in die Plattform integriert, eine Optimierung des Höhenprofils im Hinblick auf den Energieverbrauch sowohl manuell als auch automatisiert zu erreichen. Dieser Themenkomplex wird ausführlich im Kapitel 6.1 diskutiert werden.

Personenstromsimulation

Die Kollaborationsplattform wurde mit dem Fokus auf die Modellierung von Tunnelbauwerken entworfen. Es konnte aber gezeigt werden, dass sie sich auch zur Modellierung von Bauwerken hervorragend eignet (siehe Kapitel 3.8). Bei der Evaluation von Bahnhofsbauwerken spielen Evakuierungsszenarien sowie der Alltagsbetrieb von Fußgängern eine zunehmende Rolle (Kneidl 2013). Dazu wurden in Forschung und Industrie sehr ausgereifte Werkzeuge entwickelt, die häufig auf ähnlichen Prinzipien basieren und dabei insbesondere ähnliche Eingangsdatenformate erwarten. Um diese Werkzeuge bei der Evaluation von Bauwerken, die mit Hilfe der Plattform entworfen wurden, verwenden zu können, wurde eine Transformationskomponente in die Plattform integriert. Diese unterstützt den Modellierungsexperten durch eine halb-automatisierte Transformation der dreidimensionalen Bauwerksmodelle in die von typischen Personenstromsimulatoren verwendeten Datenformate, welche üblicherweise (etagenweise) zweidimensionale Daten erwarten. Somit kann der Planer seine Modelle mit einem geringen Anpassungsaufwand direkt in diese Werkzeuge exportieren und wird somit bei der Beurteilung der Bauwerke in Bezug auf Personenströme unterstützt. Ausführlich werden wir diese Komponente in Kapitel 6.2 diskutieren.

3.3. Prozedurale geometrische Modellierung

Nachdem wir im letzten Abschnitt einen Überblick über die wichtigsten Module gegeben haben, die in die Plattform *Kollabserver* integriert wurden, werden nun zentrale Konzepte dieser Plattform detailliert beleuchtet und ihre Neuartigkeit und Stärken in einem kollaborativen Modellierungs- und Planungsprozess, insbesondere bei der Planung von trassenbasierten Infrastrukturbauwerken, herausgearbeitet.

Die Plattform *Kollabserver* bildet den zentralen Zugriffspunkt in einem kollaborativen Modellierungs- und Planungsprozess. Die Basis dieses Modellierungsprozesses ist ein innerhalb des *3DTracks* Forschungsprojekts entwickeltes multi-skaliges prozedurales Geometriemodell (Borrmann et al. 2014), dessen Instanzen von der Plattform *Kollabserver* während einer kollaborativen Sitzung verwaltet werden. Zunächst werden einige allgemeine Grundbegriffe im Bereich der prozeduralen Modellierung erläutert, also der geometrischen Modellierung, die auf einem sogenannten prozeduralen Modell beruht.

Prozedurale Geometriemodelle

Prozedurale Geometriemodelle sind generell dadurch charakterisiert, dass in diesen die sogenannte Konstruktionshistorie gespeichert wird (Pratt et al. 2006, Bianconi et al. 2006). Diese umfasst die einzelnen Konstruktionsschritte und ihre Abhängigkeiten untereinander. In mo-

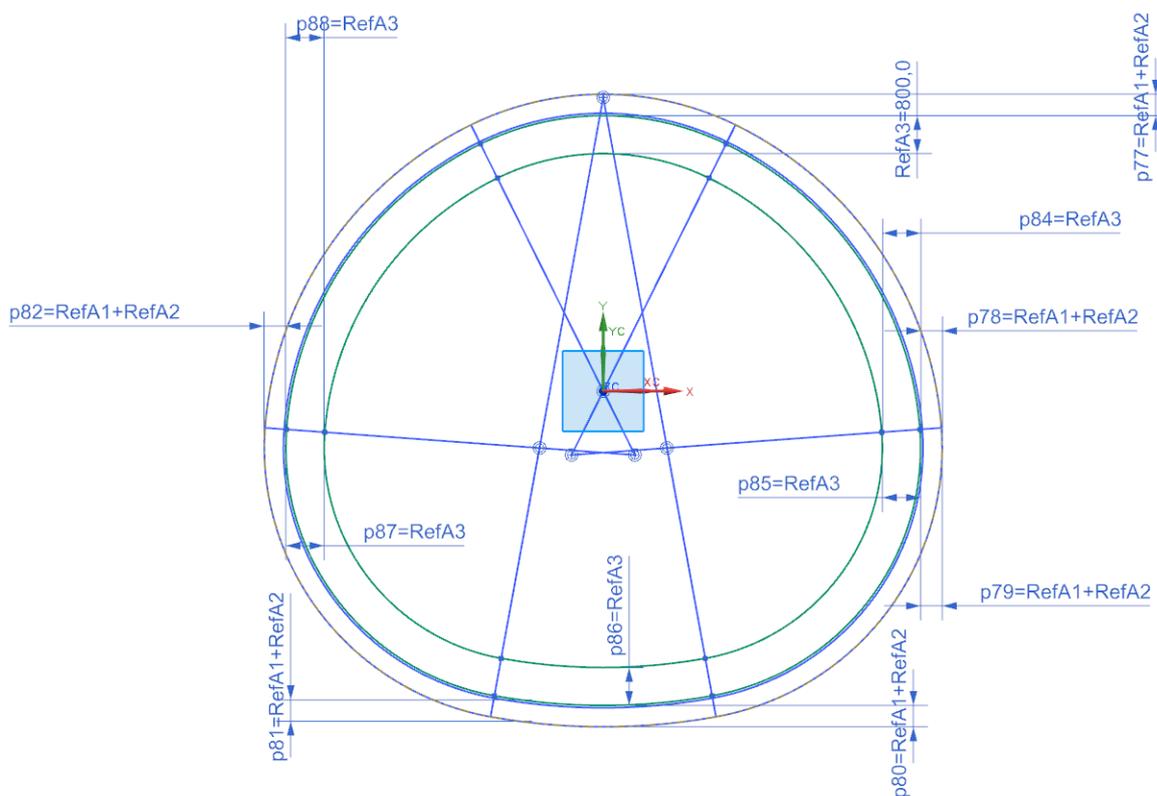
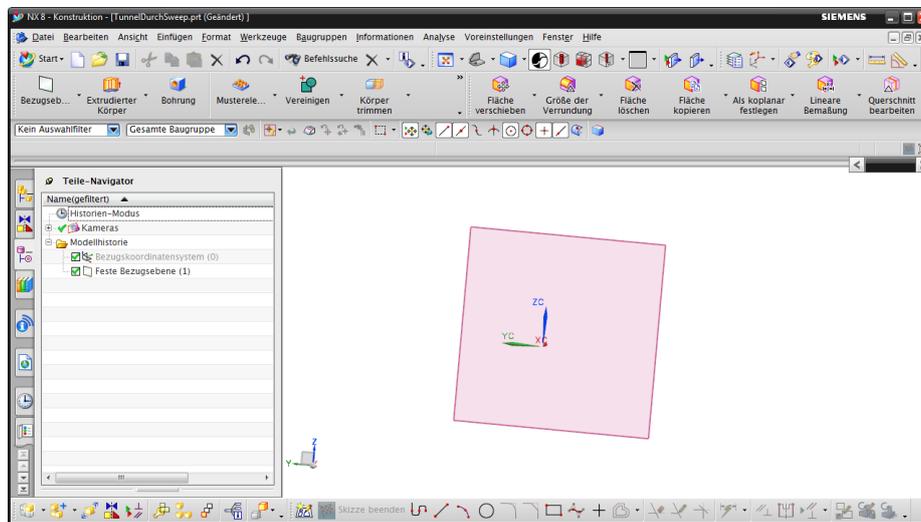


Abbildung 9: Eine parametrische Skizze zur Beschreibung des Querschnitts eines U-Bahn-Tunnels

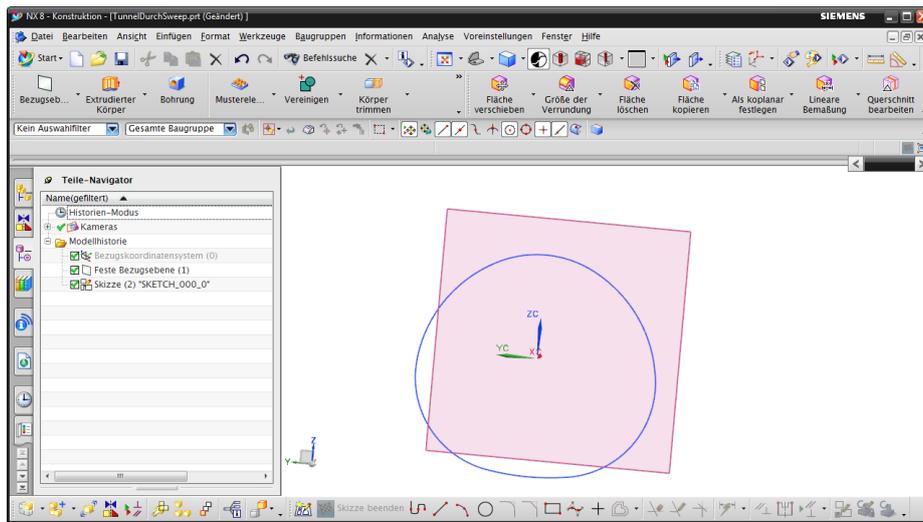
dernen CAD-Werkzeugen repräsentieren die Konstruktionsschritte – oft auch als prozedurale Operationen oder Features bezeichnet – hochwertige Geometrieoperationen wie das Extrudieren oder Verschneiden von Körpern im Gegensatz zu rudimentären Konstruktionsschritten wie dem Zeichnen von einzelnen Punkten oder Linien. Die expliziten geometrischen Informationen werden dann durch den im CAD-System integrierten Geometriekern aus den gespeicherten Operationen berechnet.

Rudimentäre Konstruktionsschritte wie das Zeichnen von Kreisen und Linien werden in sogenannten Skizzen gekapselt, welche dann wiederum als Basis für weitere Operationen wie Extrusionen dienen können. Ein Beispiel für eine zweidimensionale Skizze zur Beschreibung eines Maul-Tunnelprofils ist in Abbildung 9 dargestellt; die Verwendung dieses Maul-Tunnelprofils zur Erzeugung eines einfachen prozeduralen Tunnelmodells (Abbildung 5) wird auf den folgenden Seiten Schritt für Schritt erläutert und kann ein erstes Verständnis für den Prozess der prozeduralen Modellierung liefern. Das konkrete Modell wurde mit dem CAD-System Siemens NX erstellt und umfasst sieben Konstruktionsschritte, die in den folgenden sieben Abbildungen einzeln dargestellt sind. Im linken Teil der Abbildungen kann dabei jeweils das Anwachsen der Konstruktionshistorie im sogenannten Konstruktionsbaum beobachtet werden.

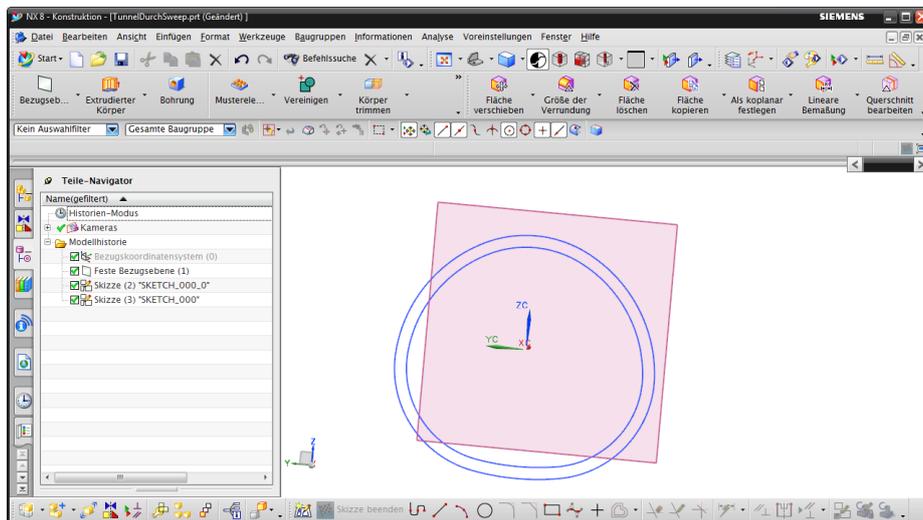
1. Eine Arbeitsebene als (räumliche) Grundlage für die folgenden Skizzen,



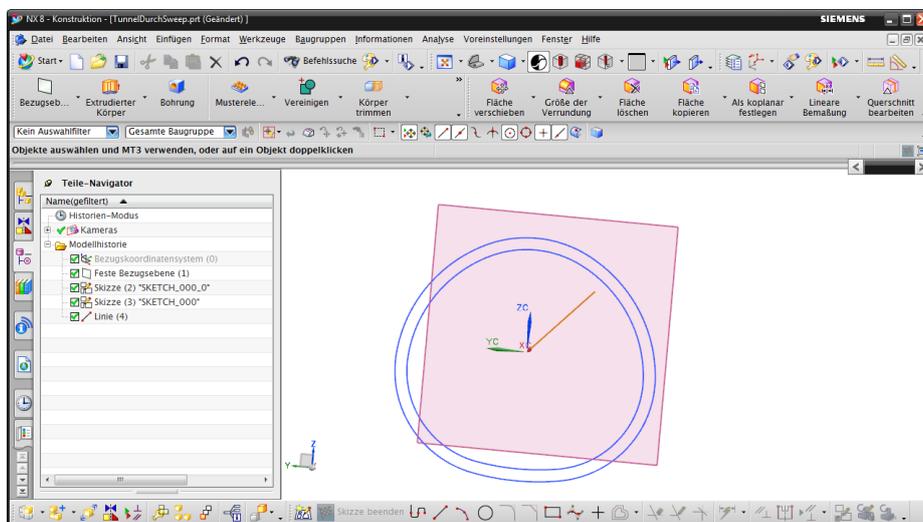
2. eine ebene Skizze zur Definition der inneren Tunnelhülle in der Arbeitsebene aus (1.),



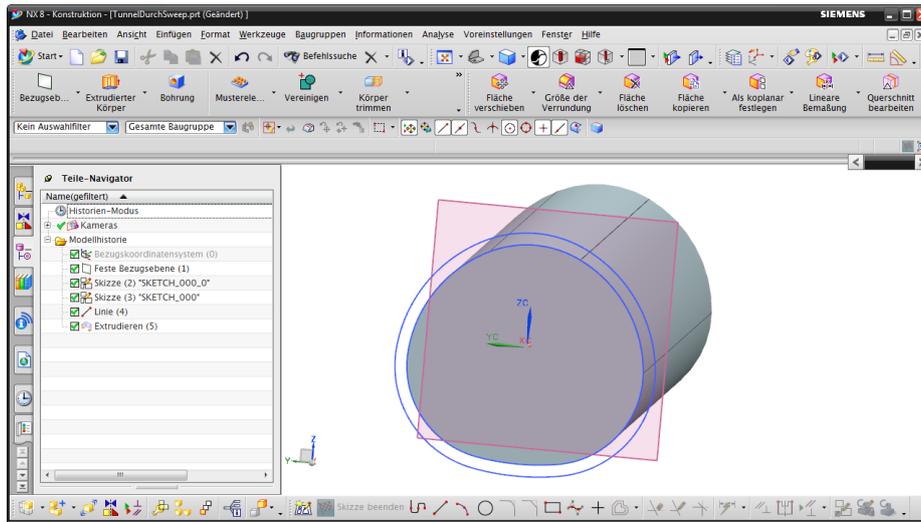
3. eine ebene Skizze zur Definition der äußeren Tunnelhülle in der Arbeitsebene aus (1.),



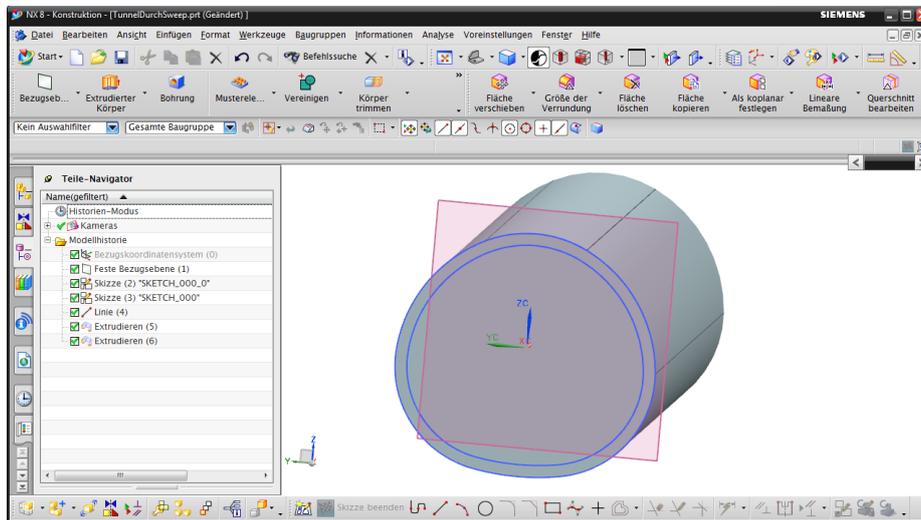
4. eine (räumliche) Linie als Richtungsvektor für die folgenden Extrusionen,



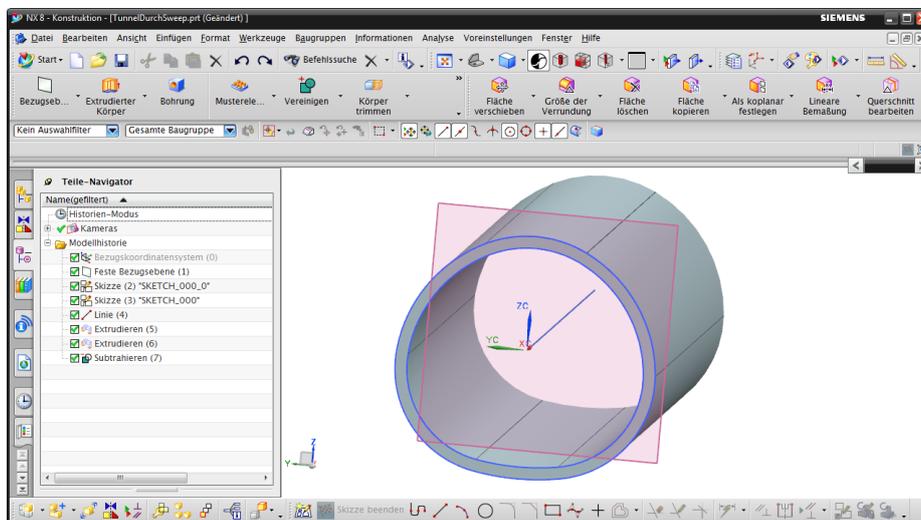
5. eine Extrusion unter Verwendung der inneren Skizze (2.) entlang der Linie aus (4.),



6. eine Extrusion unter Verwendung der äußeren Skizze (3.) entlang der Linie aus (4.),



7. eine Boolesche Subtraktion der beiden aus den Extrusionen resultierenden Volumenkörper, die in den Schritten (5.) und (6.) erzeugt wurden.



Der Begriff Feature

Wie zuvor erklärt, sind für moderne CAD-Werkzeuge hochwertige Operationen typisch, welche häufig auch als *Features* bezeichnet werden. Der Begriff *Feature* ist in diesem Kontext häufig auch mit einem semantischen Aspekt belegt, weshalb wir die Art seiner Verwendung innerhalb dieser Arbeit nun kurz diskutieren und erklären. Ein Feature definieren (Mun et al. 2003) folgendermaßen:

„Features are high-level shape constructs that make it possible for a shape designer to avoid having to work from the low level or individual curve and surface elements.“

Hingegen spricht (Bidarra 1999) von einem *Feature* als:

„A representation of shape aspects of a product that are mappable to a generic shape and are functionally significant for some product life-cycle phase. ... An essential element of a feature is thus that it has a well-defined meaning, or semantics, in a particular context or life-cycle activity.“

Somit geht Bidarra hier über eine rein geometrische Definition eines *Features* hinaus und fügt dem Begriff *Feature* bewusst eine semantische Komponente hinzu. Hierbei beruft sich Bidarra auf (Dixon et al. 1990), die genau diesen Aspekt anmerken. Um dieser Forderung des semantischen Aspekts Rechnung zu tragen, führt er den Begriff des semantischen *Features* und der semantischen *Feature*-Modellierung ein. Auch (Pratt et al. 2005) integrieren diesen Gesichtspunkt der Semantik in der Definition eines *Features*:

„High-level constructs used during the design process to create shape configurations in the model that are usually related to the intended functionality of the designed product.“

Im Folgenden werden wir die klassisch geometrische und auch heute noch gängige Definition verwenden und den Begriff *Feature* mit der geometrischen Definition assoziieren. Typische *Features* sind dann Arbeitsebenen, Skizzen, Splines, Extrusionen, Sweeps, Lofts und Boolesche Operationen wie das Vereinigen, Schneiden oder Subtrahieren von bereits existierenden Körpern. Eine ausführliche Erläuterung dieser grundlegenden Operationen und Modellierungsschritte findet man etwa auf der Plattform des freien CAD-Werkzeugs *FreeCAD*¹³.

Die Granularität von prozeduralen Operationen

Diese typischen Features werden dem Modellierer von den gängigen CAD-Werkzeugen in mehr oder minder identischer oder zumindest ähnlicher Form zur Verfügung gestellt. Neben trivialen Unterschieden in der reinen Benennung bestimmter Features, wie etwa der Sweep- und Extrusions-Operation in Siemens NX und Autodesk Inventor, findet man Unterschiede in

¹³ http://www.freecadweb.org/wiki/index.php?title=About_FreeCAD.

der sogenannten Granularität. Den Begriff der Granularität definieren (Pratt und Kim 2006) folgendermaßen: „*CAD systems differ in the complexity of the model substructures that are created by a single operation.*“

Grob gesprochen korreliert die Granularität mit den verschiedenen Varianten einer bestimmten prozeduralen Operation und spiegelt sich (im konkreten CAD-Werkzeug) in der Anzahl möglicher Parameter dieser Operation wider, wobei eine höhere Anzahl verschiedener Parameter die unterschiedlichen Varianten prinzipiell erhöht. Als Beispiel kann hier im Fall der Extrusion im System Siemens NX die Operation „Extrusion im Abstand“ genannt werden, welche im System Autodesk Inventor aus zwei Extrusionen und einer Booleschen Subtraktion zusammengesetzt werden muss. In unserem Kontext war die Granularität bei der Entwicklung des neutralen prozeduralen Geometriemodells von Bedeutung: Je feiner man im neutralen Modell die Granularität wählt, d.h. je mehr Optionen man bei einer prozeduralen Operation zulässt, desto komplexer wird im Allgemeinen der Vorgang der Übersetzung einer neutralen Operation in eine oder mehrere systemspezifische Operationen, wie auch das gerade gegebene Beispiel der Extrusion in den beiden System Siemens NX und Autodesk Inventor verdeutlicht. In unserem Ansatz sind wir deshalb generell von einer groben Granularität ausgegangen, was keinen fundamentalen Unterschied hinsichtlich der prinzipiellen Modellierungsmöglichkeiten ergibt, aber zu mehr Operationen in der konkreten Konstruktionshistorie führen kann.

Parameter und Constraints

Der hier und im Folgenden verwendete Begriff des Parameters entspricht seinem natürlichen Verständnis. So definieren auch (Pratt et al. 2005) Parameter als: „*Values of quantities in the model (typically dimensions) that may be regarded as variable for purposes of editing the model.*“ So erlauben etwa Parameter eine später modifizierbare Definition von Längen, Winkeln oder Entfernungen in Skizzen genauso wie etwa der Höhe von Extrusionen oder den Abständen von Arbeitsebenen. Abbildung 9 verdeutlicht die Verwendung von Parametern bei der Gestaltung einer Skizze.

Neben Features und Parametern zeichnen sich moderne prozedurale Geometriemodelle und die darauf basierenden CAD-Systeme durch die Integration sogenannter Constraints aus (Brüderlin und Roller 1998). Constraints erlauben es dem Modellierer, Zwangsbedingungen für bestimmte Features oder zwischen verschiedenen Features festzulegen. So umfassen Zwangsbedingungen alle Parameter zur Beschreibung (Parametrisierung) von Streckenlängen oder Winkeln (Mun et al. 2003, Li M. et al. 2007), sichern aber zusätzlich geometrische oder topologische Beziehungen zwischen verschiedenen Features und geometrischen Entitäten zu, etwa die Parallelität von Linien und Flächen oder das Zusammenfallen von Punkten. Dazu schreiben (Pratt et al. 2005): „*Constraints: Relationships between geometric entities in the*

model (e.g. parallelism, tangency, symmetry) that are required to be maintained if the model is edited.“

Zur Lösung der Menge der durch Constraints gegebenen Zwangsbedingungen werden in heutige CAD-Werkzeuge leistungsstarke Constraintsolver integriert (Brüderlin und Roller 1998), die automatisiert die Parameter gemäß einer Lösung der Menge dieser Zwangsbedingungen anpassen und den Benutzer bei der Wahl der Constraints so unterstützen, dass sich eindeutige rekonstruierbare Objekte, insbesondere eindeutige rekonstruierbare Skizzen ergeben.

Zusammenfassend lässt sich sagen, dass die vier wichtigsten und somit charakteristischen Eigenschaften eines prozeduralen Geometriemodells durch die Verwendung einer Konstruktionshistorie, von Features, Parametern und Constraints gegeben sind.

3.4. Ein multi-skaliges prozedurales Geometriemodell

Die in den gängigen Modellierungswerkzeugen gemeinsam verfügbaren oder ähnlichen Konstruktionsschritte führten im Rahmen des *3DTracks* Projekts zur Idee, ein hersteller- und systemneutrales prozedurales Geometriemodell zu entwerfen, das nur diese typischen Operationen auf abstrakter Ebene verwendet und als prozedurales Modell speichert. Eine konkrete Instanz eines proprietären prozeduralen Modells soll dann in diesem neutralen Modell dargestellt und von dort aus wieder in ein beliebiges anderes proprietäres Format übertragen werden können. Unter Verwendung eines derartigen Modells findet dann der Austausch von geometrischen Modellen nicht auf Basis der resultierenden expliziten Geometrie, sondern auf Basis der vom Designer verwendeten Konstruktionsschritte statt. Die Auswahl der konkret in diesem neutralen Modell abgebildeten Operationen ist dabei von der Anwendung getrieben, in unserem Fall von der Modellierung von trassenbasierten Infrastrukturbauwerken. Bei (Mun et al. 2003) findet man einen ähnlichen Ansatz mit einem aus dem Maschinenbau getriebenen Auswahlprozess für den oben erwähnten makrobasierten Austausch von parametrischen Modellen. Die Besonderheit des im Rahmen des *3DTracks*-Projekts geschaffenen Geometriemodells (Borrmann et al. 2014) ist einerseits seine Systemneutralität und andererseits die fundamentale Integration der verschiedenen Levels of Detail (LoD).

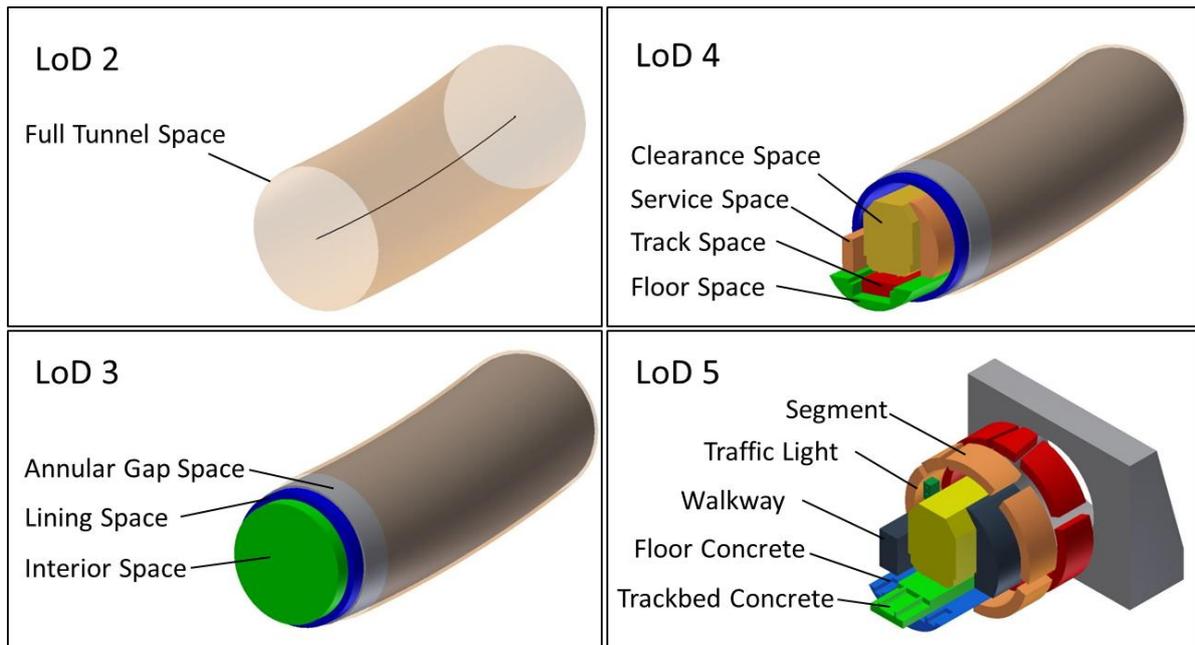


Abbildung 10: Die Levels of Detail zwei bis fünf für ein Tunnelmodell nach (Borrmann et al. 2014); der hier nicht abgebildete erste Level of Detail stellt eine linienförmige Repräsentation der Achse der Trasse dar.

Diese LoDs spiegeln die verschiedenen Abstraktionsstufen der an der Modellierung beteiligten Experten wider und können flexibel von dem an der Modellierung beteiligten Team je nach Anwendung definiert und angepasst werden (Borrmann et al. 2014). Eine explizite Definition fünf verschiedener LoDs im Bereich der Modellierung von U-Bahn-Tunnels findet man etwa bei (Borrmann et al. 2012) oder (Borrmann et al. 2014). Schematisch sind die LoDs zwei bis fünf für die Modellierung eines U-Bahn-Tunnels nach (Borrmann et al. 2014) in Abbildung 10 dargestellt : Level zwei definiert die äußere Tunnelfassung, im dritten und vierten Level werden die innere bzw. äußere Tunnelhülle und einzelne Räume (*Spaces*) wie etwa der *Track Space* und der *Floor Space* zur Festlegung des Gleisbetts bzw. der Gleise selbst festgelegt. Im fünften und letzten LoD werden diese Räume dann in den feinst möglichen Details

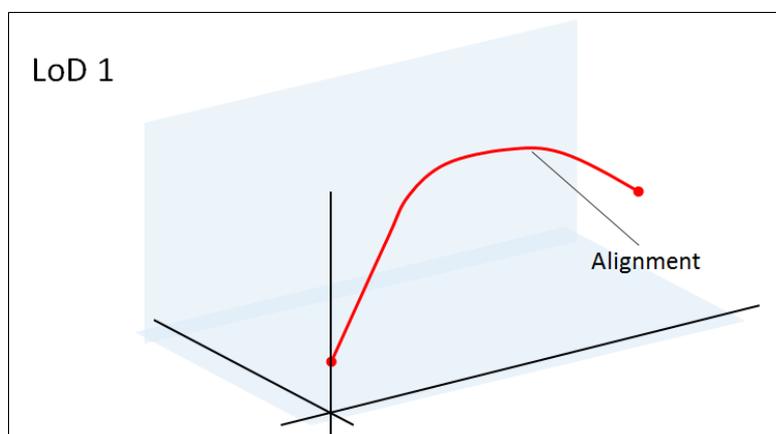


Abbildung 11: Im Level of Detail 1 wird der linienförmige Verlauf der Trasse beschrieben.

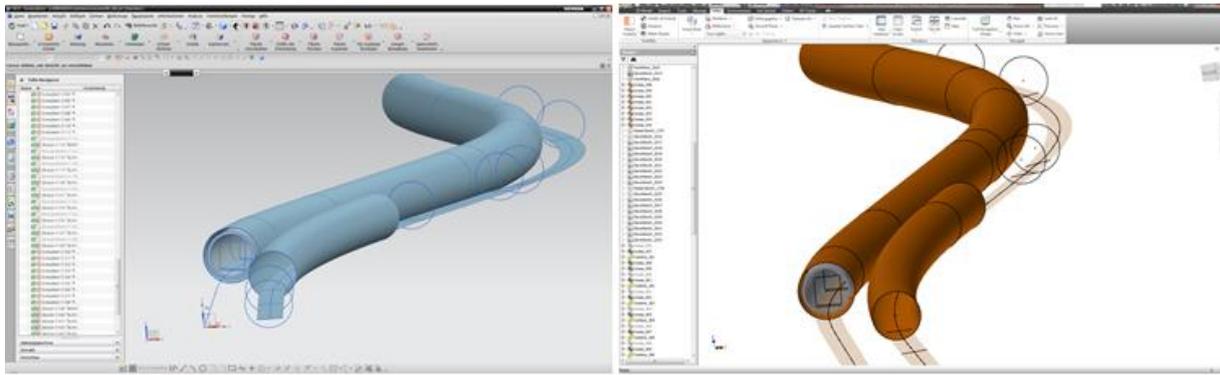


Abbildung 12: Prozedurales Modell eines Tunnels, dargestellt mit Hilfe der prototypischen Bibliotheken für die CAD-Systeme Siemens NX und Autodesk Inventor

aufgelöst. Der in Abbildung 11 dargestellte Level 1 repräsentiert ausschließlich die Achse des Tunnel-Bauwerks.

In vorangegangenen Kapitel wurde kurz der Begriff der Granularität prozeduraler Operationen angesprochen. Um die gewünschte Systemneutralität zu wahren, wurde in unserem Modell von einer groben Granularität ausgegangen, welche sich sehr gut in das gegebene Modellierungsumfeld von Tunneln einpasst und mit dem vorgestellten Konzept verschiedener LoDs kompatibel ist. So werden etwa bei der Modellierung einer sehr einfachen Tunnelröhre zunächst eine Skizze, welche die äußere Hülle beschreibt, und anschließend eine weitere Skizze festgelegt, welche die innere Hülle definiert. Beide Skizzen werden extrudiert und die entstehenden Volumenkörper subtrahiert (siehe auch das Beispiel in Kapitel 3.3). Durch diese Aufteilung in mehrere separate Schritte wird etwa auf die (möglicherweise nicht in allen CAD-Werkzeugen vorhandene) Möglichkeit verzichtet, beliebig komplexe Profile extrudieren zu können. Verallgemeinert man diese Vorgehensweise, so ergibt sich im unterstützten Modellierungsparadigma der Grundsatz, pro Skizze nur ein Profil zuzulassen, das durch einen geschlossenen Linienzug repräsentiert wird. Ein derartiges Profil kann dann als Grundlage für eine Extrusionsoperation verwendet werden. Dieses Vorgehen ist für die Modellierung eines rudimentären Tunnelmodells in Abbildung 13 schematisch dargestellt, insbesondere sind die verschiedenen LoDs der einzelnen prozeduralen Operationen gekennzeichnet. Ein auf diese Weise entstandenes Tunnelmodell wurde in Kapitel 3.3 ausführlich diskutiert, ein weiteres ist in Abbildung 12 dargestellt, einmal unter Verwendung des CAD-Systems Siemens NX, das andere Mal unter Verwendung von Autodesk Inventor.

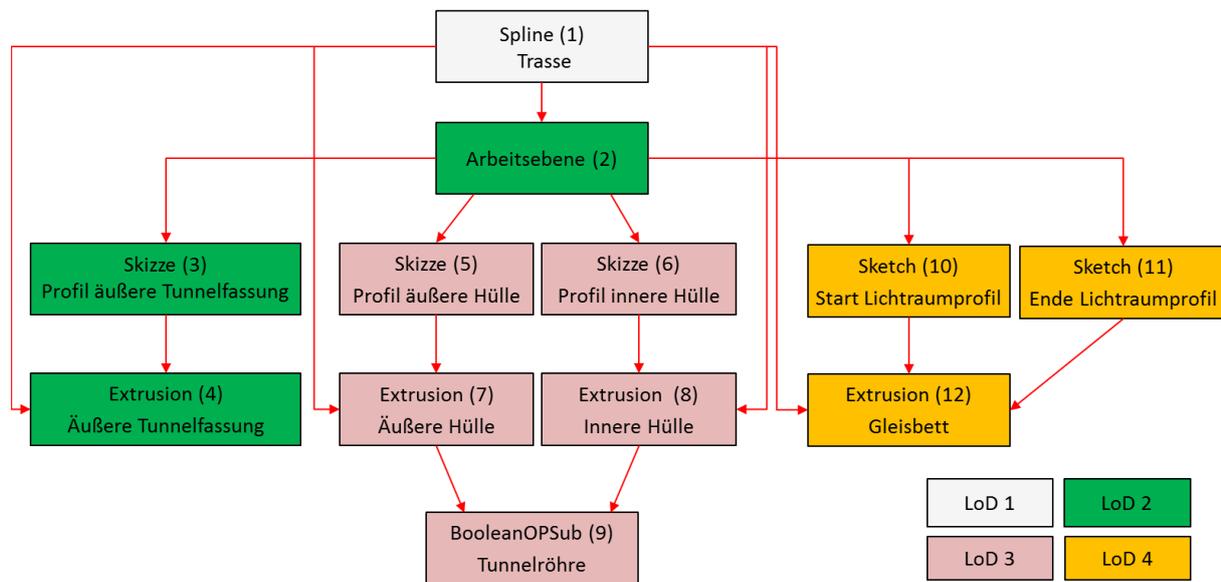


Abbildung 13: Ein einfaches (prozedurales) Tunnelmodell, welches insbesondere die verschiedenen Levels of Detail der einzelnen prozeduralen Operationen angibt.

3.5. Der Austausch von Geometriemodellen

Der Austauschprozess auf der Basis von prozeduralen Modellen ist dem Austausch rein expliziter Geometriemodelle dadurch überlegen, dass der sogenannte *Design Intent* des Modellierers oder zumindest ein Teil davon neben den reinen geometrischen Informationen übertragen werden kann. Einerseits spiegelt sich dieser bereits durch die Übertragung der verschiedenen Konstruktionsschritte und insbesondere der Reihenfolge wider, in der diese ausgeführt wurden (d.h. in der Konstruktionshistorie), andererseits werden die Abhängigkeiten zwischen den verschiedenen Schritten abgebildet. Diese Abhängigkeiten können prinzipiell in zwei Kategorien unterteilt werden. Zum einen werden bestimmte Konstruktionsschritte, genauer deren geometrisches Resultat als Eingangsgröße für weitere Konstruktionsschritte verwendet, zum anderen werden Beziehungen zwischen diesen Konstruktionsschritten, genauer den resultierenden Objekten ganz explizit durch die oben beschriebenen Zwangsbedingungen (Constraints) abgebildet. Diese Eigenschaften prozeduraler Modelle machen besonders bei den Modifikationen von Parametern die Stärke dieser Art von CAD-Werkzeugen aus. Dazu merken (Pratt und Anderson 2001) an: „*The designer’s choice of parameters, features, and constraints constitutes an important part of what is known as design intent.*“

Die konkrete Umsetzung eines Teils des multi-skaligen prozeduralen Geometriemodells, das im *3DTracks*-Projekt – im Wesentlichen innerhalb des von André Borrmann geleiteten Teilprojekts B – entworfenen wurde, ist in Abbildung 14 als Klassendiagramm dargestellt. Dabei ist die zentrale Grundlage für alle Feature-Klassen die gemeinsame Basisklasse *PMOperation*, die gemäß dem objektorientierten Paradigma (Gamma et al. 1995) all die Informationen um-

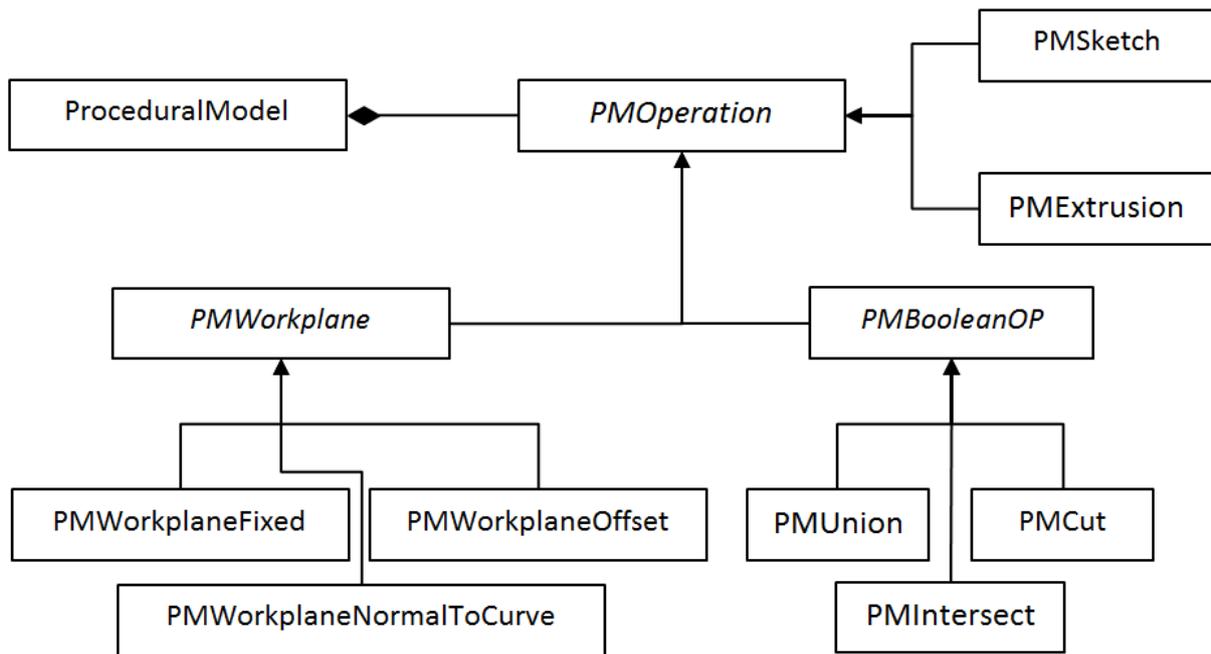


Abbildung 14: Ein Ausschnitt aus der Struktur des prozeduralen Modells

fasst, die den abgeleiteten Klassen – wie etwa *PMSketch*, *PMSweep*, *PMExtrusion* – gemeinsam sind, während die abgeleiteten Klassen selbst konkret existierende Features repräsentieren. Diese Gemeinsamkeiten sind hier etwa eine innerhalb des Modells eindeutige ID, der zur Operation zugehörige LoD und eine Liste von abhängigen Operationen. Neben den Klassen, welche die verschiedenen Features beschreiben, wurden Strukturen für die Beschreibung von Constraints, lokalen und globalen Parametern sowie zur Beschreibung von Einheiten ebenfalls gemäß dem objektorientierten Paradigma konzipiert.

Die prototypische Implementierung dieses Klassenmodells erfolgte in der Sprache C# innerhalb des .Net-Frameworks, wobei man prinzipiell auch jede andere ausgereifte objektorientierte Programmiersprache hätte wählen können. Das .Net-Framework bietet darüber hinaus einen umfangreichen Katalog von Klassen und Klassenbibliotheken zur Lösung klassischer Softwareprobleme wie etwa dem Datenbankzugriff, dem Dateihandling und auch der Serialisierung¹⁴ von Objektstrukturen in XML-Dokumenten (Albahari und Albahari 2012). So kann ein existierendes prozedurales Geometriemodell ohne weiteren besonderen Eingriff mit Hilfe dieser Bibliotheken automatisiert in ein XML-Dokument serialisiert werden, das dann die physikalische Grundlage für einen Austausch von einzelnen Operationen und Modellen bildet. Die hohe Akzeptanz von XML als Datenaustauschformat wird in der breiten Unterstützung für derartige Dokumente, insbesondere durch die Vielzahl vorhandener Editoren deutlich und

¹⁴ Unter Serialisierung verstehen wir hier die Umwandlung strukturierter Daten in eine austauschbare Zeichenkettensequenz mit dem Ziel, ein Objekt und seinen Status persistent zu machen, unter Deserialisierung den gerade umgekehrten Vorgang.

Siehe etwa auch [https://msdn.microsoft.com/de-de/library/7ay27kt9\(v=vs.120\).aspx](https://msdn.microsoft.com/de-de/library/7ay27kt9(v=vs.120).aspx)

ist im Kontext der Prototypenentwicklung und der Validierung von nicht zu überschätzendem Wert. Abschließend möchten wir noch hervorheben, dass ein weiterer entscheidender Vorteil der Verwendung von prozeduralen Geometriemodellen im gegebenen Kontext sicherlich darin zu sehen ist, dass die auszutauschende Datenmenge gering ist verglichen mit dem Austausch der kompletten, rein geometrischen Informationen bezüglich der enthaltenen Punkte, Linien und Flächen. Diese Tatsache ist von entscheidender Bedeutung bei der Wahl der zugrunde liegenden Architektur und rechtfertigt daher sicherlich aus dieser Perspektive den Einsatz einer Client Server Architektur (siehe auch Kapitel 3.2). Bei Tests zur Praxistauglichkeit der Plattform *Kollabserver*, die wir im Folgenden vorstellen, konnte diese Wahl wieder bestätigt werden.

3.6. CAD-Modellierung über die Plattform Kollabserver

Im Folgenden wird dargestellt, wie man das in Kapitel 3.4 besprochene prozedurale Modell in einem synchron kollaborativen Modellierungsprozess, in dessen Mittelpunkt die Plattform *Kollabserver* steht, wirkungsvoll einsetzen kann. Zuallererst dient diese als Repository für die verschiedenen prozeduralen Modelle und bietet den Teilnehmern einer kollaborativen Sitzung über eine genau definierte Schnittstelle Zugriff auf diese Modelle. Diese Schnittstelle und die daraus resultierenden Möglichkeiten für eine kollaborative Modellierung sollen zunächst diskutiert werden, bevor im nächsten Kapitel die Problematik der Nebenläufigkeitskontrolle behandelt und ein mathematisches Modell für den Vorgang der prozeduralen Modellierung sowie ein konkreter Algorithmus zur Nebenläufigkeitskontrolle präsentiert wird. Generell wurde diese Schnittstelle so konzipiert, dass sie für den Anwender transparent bleibt, da er ausschließlich über graphische Benutzeroberflächen mit der Plattform kommuniziert.

3.6.1. Benutzerverwaltung

Um an einer kollaborativen Sitzung teilnehmen zu können, muss sich ein Benutzer zunächst über die besagte Schnittstelle anmelden. Der Server delegiert die vom Benutzer bereitgestellten Anmeldungsinformationen an die *Benutzer*-Komponente, die nach Kapitel 3.2 generell alle benutzerspezifischen Aufgaben übernimmt. So identifiziert diese nun den konkreten Benutzer aus der Menge der registrierten Benutzer und fügt ihn der Menge der aktiven Benutzer hinzu. Insbesondere speichert sie die IP-Adresse und den zugehörigen Port, über den sich der Benutzer angemeldet hat. Diese Registrierung ist neben dem bloßen Login nötig, um dem Teilnehmer zu einem späteren Zeitpunkt modellspezifische Nachrichten, wie durch andere Teilnehmer angeforderte (lokale) Sperren oder Aktualisierungen, zu übermitteln (siehe auch Borrmann 2007). Der Benutzer selbst erhält als Rückgabe eine eindeutige Identifikationsnummer, die allen weiteren Aktionen als Parameter automatisch hinzugefügt wird. Dies ermöglicht später nachzuvollziehen, welche Teile der Konstruktionshistorie auf welchen Benutzer zurückgehen. Die einzelnen Teilnehmer werden dabei generell nach Gruppen und Rollen

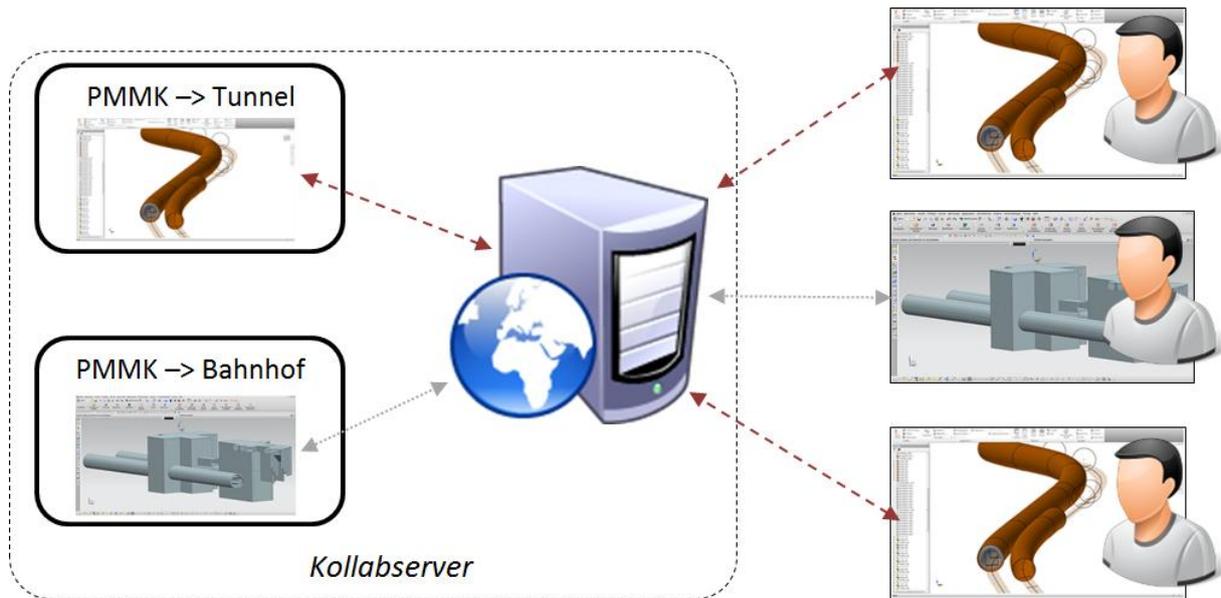


Abbildung 15: Verschiedene Benutzer arbeiten an zwei verschiedenen am Server hinterlegten Modellen. Die Verwaltung dieser zwei Modelle übernehmen zwei verschiedene Instanzen der PMMK.

klassifiziert, welche diese mit bestimmten Rechten für bestimmte Aktionen ausstatten (Sandhu et al. 1995).

3.6.2. Zugriff auf die am Server hinterlegten Modelle

Der Server gewährt den Benutzern über die genannte Schnittstelle ebenfalls Zugriff auf die Instanzen der hinterlegten Modelle. Ein bestimmtes Modell verfügt dabei über eine eindeutige Identifikationsnummer und die verschiedenen Versionen eines bestimmten Modells über eindeutige Versionsnummern. Somit wird eine konkrete Instanz eines Modells anhand einer eindeutigen Modell- und Versionsnummer identifiziert. Die aktive Bearbeitung erfolgt momentan nur auf der neuesten Modellversion, andere Versionen dienen nur zu (historischen) Vergleichszwecken. Wann eine neue Modellversion angelegt wird, wird explizit durch die Benutzer vorgegeben. Ausdrücklich wurden keine Mechanismen zur Zusammenführung verschiedener Versionen eines Modells implementiert, denn das grundlegende Forschungsziel war es, einen synchronen Modellierungsprozess zu untersuchen. Geeignete Strategien und Mechanismen im asynchronen Fall werden bei (Firmenich 2002) und (Koch 2009) ausführlich diskutiert.

Nach der Benutzeranfrage für das Laden eines konkreten Modells überprüft die Plattform, ob dieser Benutzer überhaupt berechtigt ist, die angefragte Modellversion zu laden, und anschließend, ob bereits ein anderer Benutzer diese konkrete Version geladen hat. Wenn dies nicht der Fall ist, startet der Server zunächst eine neue Instanz der sogenannten *Prozeduralgeometrie-Modell-Management-Komponente* (PMMK – Abbildung 15). Diese ruft eine Instanz aus der zugrunde liegenden Modelldatenbank ab, lädt sie in den Hauptspeicher und regelt

fortan alle Zugriffe auf diese Modellinstanz. Das PMMK-Konzept trennt die Verwaltung verschiedener prozeduraler Modelle strikt, indem immer genau eine Instanz dieser Komponente für die Verwaltung genau einer Modellversion verantwortlich ist. Alle Instanzen einer bestimmten PMMK laufen jeweils in einem eigenen Thread, der bei der Erzeugung der Komponente gestartet wird. Dies garantiert, dass der Server weiterhin gut erreichbar ist und nicht durch Modellierungsoperationen von Benutzern blockiert werden kann, die gleichzeitig andere Modelle bearbeiten. Weiterhin erleichtert die Verwendung genau eines Threads pro Instanz einer PMMK die Synchronisation zwischen dem zentralen Modell am Server und den lokalen Replikaten auf den Client-Seiten. Abbildung 15 verdeutlicht den Einsatz der verschiedenen Instanzen einer PMMK. Die konkreten Daten, genauer die diese Daten beschreibenden Dateien eines prozeduralen Modells werden in einer Datenbank im XML-Format abgelegt. Um dies zu ermöglichen, wurde die Objektstruktur des prozeduralen Modells von vornherein so angelegt, dass eine automatisierte Serialisierung mit Hilfe der vom zugrunde liegenden .Net-Framework bereitgestellten Funktionalitäten möglich ist (Albahari und Albahari 2012). Dieses Vorgehen hat sich während der Entwicklungs- und Testphase als stabil und benutzerfreundlich erwiesen.

3.6.3. Transformationen neutraler Modelle in proprietäre

Als Antwort auf eine Modellanfrage wird einem Benutzer konkret eine XML-Datei zurückgegeben, welche die Daten, die das Modell beschreiben, in serialisierter Form enthält. Ein so empfangenes XML-Dokument wird auf der Client-Seite zunächst über eine dort integrierte Bibliothek (automatisiert) wieder in eine Klassenstruktur deserialisiert, was in einer im Hauptspeicher gehaltenen Objektstruktur resultiert. Diese umfasst neben globalen Parametern und Einheiten alle bisher ausgeführten Konstruktionsschritte in Form von sogenannten neutralen prozeduralen Operationen (NOP). Dabei sind die neutralen Operationen genau in der Reihenfolge gespeichert, in der sie ausgeführt wurden. Die Beibehaltung der Reihenfolge ermöglicht eine problemlose schrittweise Rekonstruierbarkeit des Modells und erübrigt insbesondere ein komplexes Durchsuchen des Abhängigkeitsgraphen. Neben allgemeingültigen Funktionen etwa zur Kommunikation enthalten die clientseitigen Bibliotheken auch systemspezifische Funktionalitäten. Eine spezielle Interpreter-Komponente ist dabei für die Übersetzung neutraler Operationen in systemspezifische Operationen (SOP) integriert und muss je nach CAD-Zielsystem ausgewählt werden. Nachdem das prozedurale Modell durch die Deserialisierung auf der Client-Seite in den Speicher geladen wurde, werden alle NOPs sukzessive in SOPs übersetzt und in der gegebenen Reihenfolge nacheinander über die API des zugrunde liegenden CAD-Systems ausgeführt (siehe auch Abbildung 16). Neben dem Laden des gesamten Modells ist es für den Benutzer möglich, ein bestimmtes Modell nur genau bis zu einem bestimmten LoD oder bei einem trassenbasierten Modell nur bestimmte Teilmodelle zu laden. So werden im ersten Fall nur die Konstruktionsschritte integriert, deren Level of Detail kleiner oder gleich einem (vom Benutzer) vorgegebenen Wert ist, was anschaulich zur Dar-

stellung des Modells auf einer größeren Abstraktionsebene führt. Im zweiten Fall hingegen werden nur die Konstruktionsschritte integriert, die zu einem festgelegten Trassenabschnitt in Bezug stehen, was zu einer Unterteilung des Modells in kleinere Abschnitte führt und die Modellierung kleinerer und somit übersichtlicherer Entitäten ermöglicht. Auch in diesen Fällen garantiert die Plattform die Konsistenz sowohl dieser abstrakteren Modelldarstellungen als auch von Modellausschnitten, wie wir in Kapitel 3.7.5 sehen werden.

3.6.4. Architektur der Modellierungsclients

Wie vorab diskutiert, müssen für einen kollaborativen Modellierungsprozess über die Kollaborationsplattform spezielle Bibliotheken in die einzelnen CAD-Werkzeuge integriert werden. Prototypisch wurden solche für die kommerziellen CAD-Systeme Siemens NX und Autodesk Inventor sowie für das am Lehrstuhl für Computation in Engineering entwickelte Geometrie Framework TUM-Geoframe entworfen (Sorger et al. 2014). Neben der angesprochenen Übersetzung von NOPs in SOPs und von SOPs in NOPs sind diese Bibliotheken für die Kommunikation zwischen Benutzer und Kollaborationsplattform und der Interaktion zwischen CAD-Programm und Benutzer verantwortlich.

Dies bedeutet einerseits, dass die für die Modellierung bedeutsamen Benutzereingaben registriert, andererseits vom Server eingehende Benachrichtigungen etwa über Modellmodifikationen durch andere Benutzer entgegengenommen werden müssen. Um Benutzereingaben wahrzunehmen, registriert sich die Client-Bibliothek für die entsprechenden Ereignisse (*Events*) der API des CAD-Werkzeugs, während sie zum Empfangen von Nachrichten des Servers permanent auf dem bei der Anmeldung angegebenen Port hört und somit selbst als Server fungiert. Auf diese Ereignisse müssen die Client-Bibliotheken angemessen reagieren; insbesondere müssen selbst-aufrufende Ereignisse vermieden werden: Diese entstehen, wenn bei der Abarbeitung einer Servernachricht von der API selbst Ereignisse ausgelöst werden, die sonst gerade durch Benutzereingaben hervorgerufen werden. So wird etwa beim Hinzufügen einer Operation über die API das gleiche Ereignis ausgelöst wie beim Hinzufügen über die graphische Oberfläche des CAD-Werkzeugs. Doch die daraus resultierenden Reaktionen sind völlig unterschiedlich. Bei einer Modifikation infolge einer Servernachricht ist nichts zu unternehmen. Denn die Modifikation betrifft nur lokal diese Client-Seite und wurde zum Ereigniszeitpunkt bereits vollständig in das zentrale Modell eingebaut. Bei einer Benutzereingabe hingegen muss ein komplizierter Prozess zur Integration der neuen Operation in das (zentrale) Gesamtmodell durchlaufen werden. Diesen Prozess untersuchen wir detailliert im anschließenden Abschnitt Kapitel 3.6.5.

Unterscheidet man nicht sauber zwischen diesen beiden Fällen, kann etwa die Integration einer Änderung, die aus einer Servernachricht resultiert, dazu führen, dass die Client-Bibliothek diese fehlerhaft als Benutzereingabe interpretiert und diese ihrerseits an den Server meldet. Die Folge ist dann ein nicht abbrechender Zyklus von Nachrichten, die zwischen dem Server und den verschiedenen Clients hin- und hergeschickt werden. Dieses Problem der sich selbst-

aufzufindenden Ereignissen wurde gelöst, indem die Client-Bibliothek grundsätzlich in zwei verschiedenen Modi, dem Server- und dem Benutzermodus, arbeitet. Grundsätzlich befindet sich die Bibliothek im Benutzermodus und reagiert dementsprechend. Nach dem Eingang einer Servernachricht wird die Bibliothek während der Abarbeitung der eingegangenen Nachricht in den Servermodus versetzt, so dass auf dabei geworfene Ereignisse situationsgerecht (eventuell gerade gar nicht) reagiert werden kann. Diese Strategie ist eine Anwendung des sogenannten Status-Musters (Gamma et al. 1995) und garantiert eine situationsabhängige Reaktion auf bestimmte Ereignisse. Bei der Validierung des Modellierungsprozesses hat sich ihre Verwendung als vorteilhaft und stabil erwiesen.

3.6.5. Der konkrete Modellierungsprozess

Für die konkrete Modellierung stellt die Benutzerschnittstelle folgende mögliche Aktionen bereit: das Hinzufügen *AddOperation*<BenutzerID, *NewOperation*>, das Modifizieren *ModifyOperation*<BenutzerID, *ModifiedOperation*> und das Löschen *RemoveOperation*<BenutzerID, *OperationID*> von Konstruktionsschritten. Dieser sehr einfache, nur drei Aktionen umfassende Befehlssatz, um das Modell und auch die darin enthaltenen Operationen direkt zu modifizieren, hat sich in der Praxis als ausreichend erwiesen und entspricht in natürlicher Weise den Benutzerinteraktionen, die tatsächlich über die graphische Benutzeroberfläche der CAD-Systeme durchgeführt werden können. Als Parameter werden die ID des Benutzers und die konkrete betroffene Operation in Form einer NOP oder deren Identifikationsnummer allen drei Befehlen übergeben. Instanzen dieser Befehle werden vom Benutzer weder konkret ausgewählt noch abgeschickt, sondern automatisiert von der über die API integrierten Client-Bibliothek an den Server gesandt, sobald ein Benutzer innerhalb der graphischen Oberfläche des CAD-Werkzeugs eine entsprechende Aktion durchgeführt hat. Entsprechende Kommandos werden ebenfalls für die Bearbeitung von (globalen) Parametern bereitgestellt, wobei folgende Ausführungen für Operationen identisch auf Parameter zu übertragen sind; am Beispiel einer Extrusionsoperation soll nun das Hinzufügen einer Operation konkret veranschaulicht werden:

Sobald der Benutzer eine neue Extrusion erzeugt hat (Abbildung 16 – Schritt 1), wird vom zugrunde liegenden CAD-System über die API ein Ereignis ausgelöst, für welches sich die Benutzerbibliothek zuvor registriert hatte. Die Bibliothek fängt dieses Ereignis ab und liest über das von der API bereitgestellte Objektmodell die verschiedenen Parameter zur Beschreibung der neuen Extrusion wie die Höhe und das zugrunde liegende Profil aus. Mit Hilfe dieser Parameter wird eine Instanz einer entsprechenden neutralen Operation erzeugt, in das lokale prozedurale Modell integriert und schließlich in eine Zeichenkette serialisiert. Diese Zeichenkette wird als Parameter des *AddOperation*-Kommandos an die Plattform (Abbildung 16 – Schritt 2) gesendet. Auf der Plattform wiederum wird diese Zeichenkette in eine Objektinstanz deserialisiert, unverzüglich in das zentrale Modell integriert und die resultierende neutrale Operation an alle anderen angemeldeten Clients weitergeleitet (Abbildung 16 – Schritt 3).

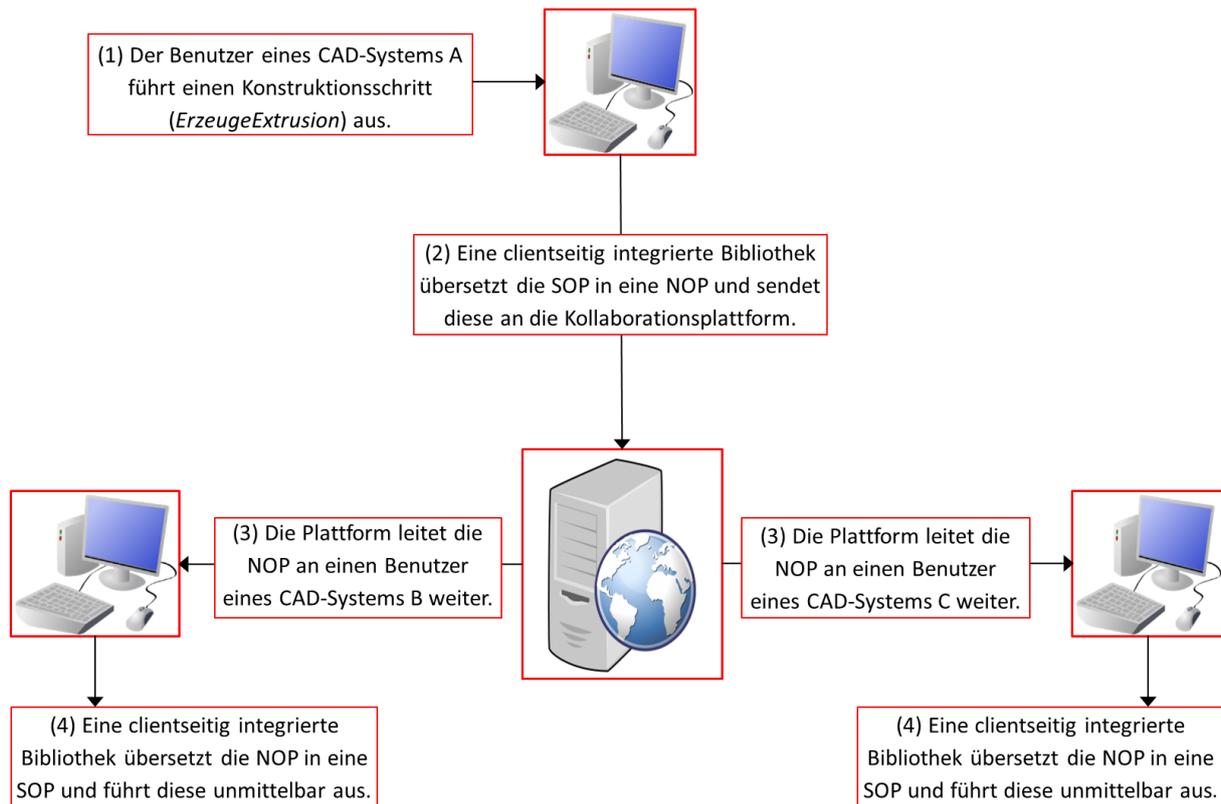


Abbildung 16: Der Prozess des Einfügens, Übertragens und der Übersetzung eines neuen Konstruktionsschritts

Die Clients sind nun gezwungen, diese Operationen ihrerseits unmittelbar in das lokale Replikat zu integrieren (Abbildung 16 – Schritt 4). Diese Strategie bildet die Grundlage für die Sicherung der durchgängigen Konsistenz der lokalen Replikate und des zentralen Modells, wobei wir auf zusätzlich nötige Maßnahmen zur Nebenläufigkeitskontrolle im folgenden Kapitel 3.7 eingehen werden.

Hiermit haben wir die Darstellung der grundlegenden Funktionsweise eines synchronen kollaborativen Modellierungsprozesses über die Kollaborationsplattform abgeschlossen. Es wurden das der Modellierung zugrunde liegende prozedurale Modell vorgestellt, die zentralen Serverkomponenten erläutert und der grundlegende Ablauf der gemeinsamen Modellierung über die Kollaborationsplattform erklärt. Nun werden wir uns eingehend um die Konsistenzsicherung während der synchronen Modellierung über die Kollaborationsplattform beschäftigen und die hier verwendeten und entwickelten Strategien und Konzepte präzisieren.

3.7. Nebenläufigkeitskontrolle durch die Kollaborationsplattform

Bisher wurden Konzepte erläutert, welche es prinzipiell ermöglichen, über die Plattform *Kollabserver* gemeinsam an einem Geometriemodell zu arbeiten. Der dabei zentrale Punkt der Konsistenzsicherung soll im Folgenden ausführlich besprochen werden. Nach einer allgemeinen Einführung in die Thematik der Nebenläufigkeitskontrolle wird ein Algorithmus präsentiert, der auf einem gerichteten und azyklischen Sperr-Abhängigkeitsgraphen basiert. Dieser wird direkt aus der im Modell enthaltenen Konstruktionshistorie abgeleitet und erlaubt einen pessimistischen Ansatz zur Konsistenzsicherung mit Hilfe von flexiblen Sperren.

3.7.1. Begriffsdefinitionen, Arten der Nebenläufigkeitskontrolle

Zunächst wird kurz der Begriff der Nebenläufigkeitskontrolle selbst diskutiert, bevor ihre verschiedenen Arten klassifiziert werden. Den Begriff Nebenläufigkeitskontrolle im Bereich von Datenbankmanagementsystemen definieren (Bernstein und Goodman 1980) folgendermaßen: „*Concurrency control is the activity of coordinating concurrent accesses to a database in a multiuser database management system (DBMS).*“ Nach Ellis et al. ist es in Groupwaresystemen das Ziel der Nebenläufigkeitskontrolle, Konflikte, die durch gleichzeitige Operationen verschiedener Benutzer entstehen, aufzulösen (Ellis und Gibbs 1989). In ihrem Standardwerk zur rechnergestützten Gruppenarbeit betonen (Borghoff und Schlichter 1995) die Bedeutung der Nebenläufigkeitskontrolle, wenn sie schreiben: „*Das gemeinsame Benutzen und Austausch von Informationen ist integraler Teil der Gruppenprozesse. Die Kontrolle der Nebenläufigkeit (concurrency control) ist daher essentiell, um die Informationen konsistent zu halten.*“ Bei der Begriffsdefinition fokussieren sich (Ressel et al. 1996) speziell auf die synchron kollaborative Arbeit: „*Concurrency control covers the management of parallel threads of human-computer interaction including potential conflicts. These typically arise in synchronous collaborative work when different users concurrently manipulate a shared object.*“ Im Bauwesen untersucht Borrmann die Aufgaben der Nebenläufigkeitskontrolle innerhalb eines synchronen Modellierungsprozesses und erklärt als ihr Ziel, gleichzeitige Zugriffe parallel laufender Programme auf ein und dieselbe Ressource zu koordinieren (Borrmann 2007). Im gegebenen Kontext ist die gemeinsame Ressource das durch die Kollaborationsplattform verwaltete prozedurale Geometriemodell. Die Koordination durch die Plattform besteht dann in der Sicherung der Konsistenz zwischen diesem zentral verwalteten Modell und den lokalen Replikaten des Modells auf den Client-Seiten.

Der entscheidende Punkt bei der Klassifizierung verschiedener Konzepte zur Nebenläufigkeitskontrolle ist, wie die Konsistenzsicherung konkret erreicht wird. Eine übliche Einordnung ist die grundlegende Unterscheidung in optimistische und pessimistische Ansätze (Borghoff und Schlichter 1995, Phillips W. 1999), wie sie auch in Abbildung 17 dargestellt ist. Prinzipiell ist eine pessimistische Nebenläufigkeitskontrolle dadurch gekennzeichnet, dass Konfliktsituationen von vornherein ausgeschlossen werden, während optimistische Ansätze

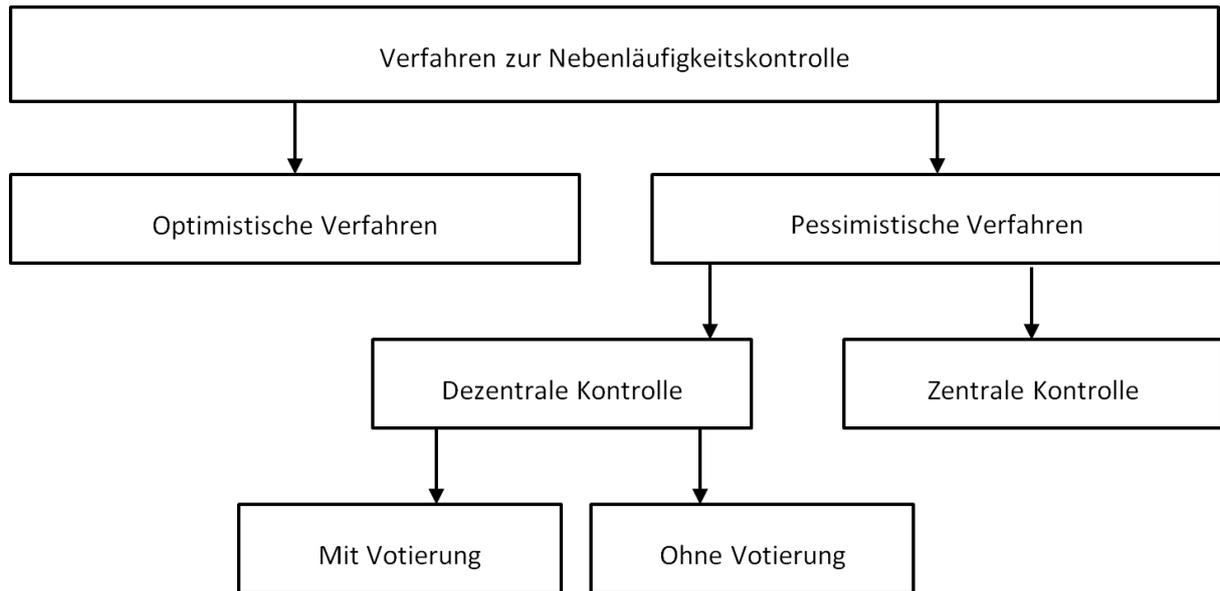


Abbildung 17: Eine Klassifikation für verschiedene Verfahren zur Nebenläufigkeitskontrolle nach (Borghoff und Schlichter 1995)

Konflikte zunächst zulassen und versuchen, diese Konflikte zu einem späteren Zeitpunkt wieder aufzulösen (Xue et al. 2000, Schwabe et al. 2001). Somit sind optimistische Verfahren in Anwendungsfällen sinnvoll, in denen nicht zu jeder Zeit eine (vollständige) Konsistenz der Daten gefordert werden muss oder Konflikte nur selten auftreten können. Pessimistische Verfahren hingegen werden verwendet, wenn jederzeit die Konsistenz des Datenbestands abgesichert werden muss. Diese Alternativen werden auch mit den Stichworten „prevent“ und „cure“ beschrieben, d.h. ob man Konfliktsituationen von vornherein vermeidet („prevent“) oder diese im Nachhinein wieder bereinigt („cure“) (Munson und Dewan 1996).

Die pessimistische Variante scheint zunächst als die sinnvollere, da zu jeder Zeit konsistente Daten garantiert werden. Natürlich hat diese höhere Sicherheit ihren Preis, nämlich einem niedrigeren Datendurchsatz, eine langsamere Antwortzeit der Anwendung oder im Extremfall einen völligen Stillstand der Anwendung (für andere Benutzer) (Ellis et al. 1991). Im Bereich der kollaborativen Arbeit werden bei optimistischen Ansätzen im Extremfall zunächst beliebige inkonsistente Versionen eines gemeinsamen Bearbeitungsgegenstandes zugelassen, welche man zu einem späteren Zeitpunkt wieder zu einer einheitlichen Version zusammenzuführen („Merging“) versucht. Im Bereich der Produktmodellierung sowie der geometrischen Modellierung findet man diesbezüglich ausführliche Diskussionen bei (Munson und Dewan 1994), (Firmenich 2002) und (Koch 2009).

Bei pessimistischen Verfahren unterscheiden Borghoff und Schlichter gemäß Abbildung 17 weiter zwischen zentraler und dezentraler Datenkontrolle (Borghoff und Schlichter 1995). Diese beiden Varianten sind durch die Systemarchitektur bedingt. So herrscht in Peer to Peer Systemen typischerweise eine dezentrale Datenkontrolle vor, während sich Client Server Ar-

chitekturen für eine zentrale Variante eignen, zumindest solange genau ein einzelner Rechner als ausgezeichnete Serverinstanz eingesetzt wird. In der vorliegenden Client Server Architektur bildet die Kollaborationsplattform den zentralen Zugriffspunkt, weshalb wir uns auf Techniken beschränken, die diese zentrale Architektur verwenden. Für Peer to Peer Architekturen verweisen wir den Leser insbesondere auf (Sun et al. 1998, Xue et al. 2000, Sun 2002) und (Li Min 2007). Sie verwenden in ihren Arbeiten derartige Systeme und stellen dabei interessante Aspekte der sogenannten *Operational Transformation*, insbesondere im Bereich der kollaborativen Textbearbeitung (Sun et al. 1998, Xue et al. 2000, Sun 2002) und kollaborativen CAD-Modellierung (Li Min et al. 2007) vor.

In einer zentralen pessimistischen Architektur bilden Sperren eine häufig verwendete Möglichkeit, die Datenkonsistenz zu wahren. Die Grundidee dabei ist es, dass ein Benutzer für eine Dateneinheit, die er modifizieren möchte, bei der zentralen Verwaltungsinstanz eine Sperre („*Lock*“) anfordern muss. Sobald er diese Sperre erhalten hat, kann er diese Einheit modifizieren, während den anderen Benutzern der Zugriff oder zumindest ein modifizierender Zugriff auf diese Einheit verwehrt wird. Hat der Benutzer seine Modifikation abgeschlossen, fordert er die zentrale Verwaltungseinheit auf, die Änderung zu übernehmen, an die anderen teilnehmenden Clients und damit Benutzer weiterzuleiten und abschließend die Sperre wieder aufzuheben. Die Granularität dieser gesperrten Einheiten ist für den Datendurchsatz von entscheidender Bedeutung: Denn je größer die zugrunde liegende zu sperrende Einheit ist, desto stärker werden die Möglichkeiten für die Benutzer eingeschränkt, den zugrunde liegenden Datenbestand simultan zu bearbeiten. Im Extremfall führt eine Sperre zur Blockade des gesamten Datenbestandes, so dass temporär nur ein Benutzer aktiv Modifikationen am Datenbestand vornehmen kann. Einen derartigen Zustand findet man auch bei sogenannten token-basierten Ansätzen. In einer derartigen Architektur besitzt ein ausgezeichnete Rechner das sogenannte Token und ist als einziger berechtigt, Modifikationen durchzuführen (Borghoff und Schlichter 1995). Hierbei gilt es natürlich zu überlegen, wie die Vergabe dieses Token zustande kommt. Auch im Bereich der kollaborativen synchronen Modellierung spielt dieser token-basierte Ansatz durchaus eine Rolle, wie etwa in der von Min Li beschriebenen Architektur (Min Li et al. 2007). Auch das in Kapitel 2.4 angesprochene *WEBSpiff* System fällt in diese Kategorie (Bidarra et al. 2002). Ein Beispiel für eine auf Sperren basierte pessimistische Architektur in der kollaborativen CAD-Modellierung, in der nur das direkt zu modifizierende Feature gesperrt wird, präsentiert (Hepworth 2014). Fan und Sun liefern einen flexiblen graph-basierten Ansatz zur Nebenläufigkeitskontrolle im Bereich der kollaborativen synchronen Quellcode-Erstellung (Fan und Sun 2012). Hier werden Abhängigkeiten zwischen einzelnen Quellcode-Segmenten wie etwa Funktionsrümpfen und Funktionsaufrufen identifiziert und daraus ein Abhängigkeitsgraph erstellt, der dann als Basis für einen hierarchischen Sperr-Algorithmus verwendet wird. Einen ebenfalls graph-basierten Ansatz entwickelten Cai et al. im Bereich der CAD-Modellierung (Cai et al. 2012). In diesem Ansatz werden die Abhängig-

keiten zwischen den einzelnen geometrischen Entitäten als Grundlage für den Abhängigkeitsgraphen verwendet, und auf dieser Basis wird wiederum ein hierarchisches Sperrkonzept implementiert. Dieser Ansatz wurde etwa zeitgleich mit dem hier vorzustellenden Ansatz entwickelt und ähnelt diesem insoweit, als er ebenfalls die geometrischen Abhängigkeiten zwischen den einzelnen Operationen als Basis verwendet. Darüber hinaus berücksichtigt der Ansatz, der im Folgenden präsentiert wird, Möglichkeiten für die Verwendung verschiedener Detaillierungsstufen (LoDs) und erlaubt im gegebenen Kontext der trassenbasierten Modellierung zusätzlich die Zerlegung des Gesamtmodells in einzelne unabhängige Teilmodelle, wodurch ein größerer Grad an möglicher simultaner Arbeit erreicht wird.

Nachdem einige Grundbegriffe der Nebenläufigkeitskontrolle diskutiert wurden, wird nun der Fokus auf die in dieser Arbeit entwickelten Ansätze gerichtet und dabei insbesondere ein neu entwickelter Algorithmus für flexible Sperren im vorliegenden Kontext darlegt. Für eine detaillierte Diskussion im Bereich der Nebenläufigkeitskontrolle seien dem Leser etwa (Borghoff und Schlichter 1995) oder (Tanenbaum und van Steen 2006) empfohlen.

3.7.2. Der prozedurale Modellierungsprozess als bipartiter Graph

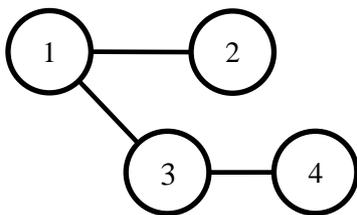
Zunächst wird der Prozess der prozeduralen Modellierung durch einen bipartiten Graphen beschrieben. Aus diesem bipartiten Graphen wird dann ein azyklischer und gerichteter Graph, der sogenannte Sperr-Abhängigkeitsgraph, abgeleitet. Mit seiner Hilfe beschreiben wir daraufhin den in der Kollaborationsplattform verwendeten Sperr-Algorithmus zur Nebenläufigkeitskontrolle. Wir beginnen mit einigen grundlegenden Definitionen und Begriffen der Graphentheorie, die wir im Folgenden wie auch in Kapitel 6.2 bei der Beschreibung der Integration von Personenstromsimulatoren verwenden werden.

Definition (Gerichteter und ungerichteter Graph)

Ein Graph ist ein Tupel (V, E) , das aus einer Menge V der Knoten (*Vertices*) und aus einer Menge $E \subseteq V \times V$ der Kanten (*Edges*) besteht. Dabei enthält die Menge der Kanten im Fall eines ungerichteten Graphen ungeordnete Paare von Knoten, im Fall eines gerichteten Graphen geordnete Paare. Anschaulich geben somit die Kanten an, ob zwei Knoten – möglicherweise nur in einer bestimmten Richtung – miteinander verbunden sind.

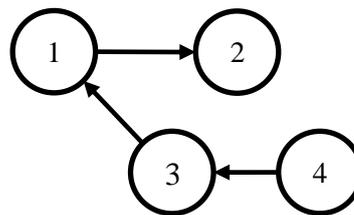
Beispiel (Ungerichteter Graph)

$$G = (\{1,2,3,4\}, \{(1,2), (1,3), (3,4)\})$$



Beispiel (Gerichteter Graph)

$$G = (\{1,2,3,4\}, \{(1,2), (3,1), (4,3)\})$$



Definition (Weg und Zyklus in einem Graphen)

Eine Folge von verschiedenen Knoten v_1, v_2, \dots, v_n ($n > 1$) mit $(v_i, v_{i+1}) \in E$ bezeichnet man als Weg von v_1 nach v_n . Sind die Knoten v_1, v_2, \dots, v_{n-1} paarweise verschieden und gilt $v_1 = v_n$, so spricht man von einem Zyklus.

Definition (Zyklischer und azyklischer Graph)

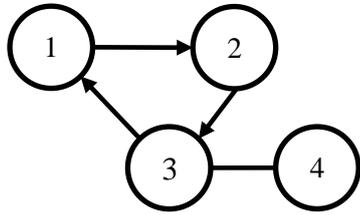
Enthält ein Graph zumindest einen Zyklus, so nennt man ihn zyklisch; enthält er keinen Zyklus, so nennt man ihn azyklisch.

Definition (Bipartiter Graph)

Einen Graphen nennen wir bipartit, wenn man seine Knotenmenge V in zwei disjunkte Teilmengen V_1 und V_2 aufteilen kann, so dass keine Kante zwei Elemente aus V_1 oder zwei Elemente aus V_2 verbindet, d.h. für eine Kante $e = (v_i, v_j)$ gilt entweder $v_i \in V_1 \wedge v_j \in V_2$ oder $v_i \in V_2 \wedge v_j \in V_1$.

Beispiel (Zyklischer Graph)

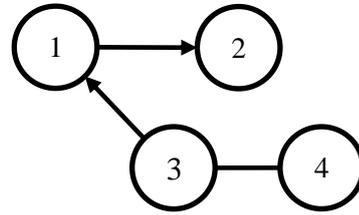
$$G = (\{1,2,3,4\}, \{(1,2), (2,3), (3,1), (3,4), (4,3)\})$$



Der Zyklus ist durch $\{1,2,3,1\}$ genauer durch $\{(1,2), (2,3), (3,1)\}$ gegeben.

Beispiel (Azyklischer Graph)

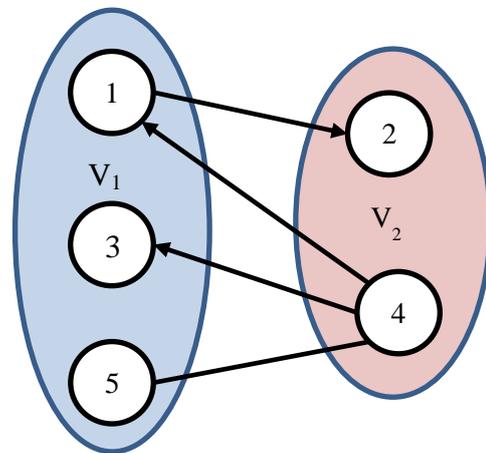
$$G = (\{1,2,3,4\}, \{(1,2), (3,1), (3,4), (4,3)\})$$



Beispiel (Bipartiter Graph)

$$G = (\{1,2,3,4,5\}, \{(1,2), (4,1), (4,3), (4,5), (5,4)\})$$

mit $V_1 = \{1,3,5\}$ und $V_2 = \{2,4\}$.



Definition (Konstruktionschritte und resultierende geometrische Objekte)

Während der Modellierung wählt der CAD-Experte konkrete Operationen aus der Menge der möglichen Konstruktionschritte (siehe Kapitel 3.3), etwa *ErzeugeSpline*, *ErzeugeSkizze*, *ErzeugeExtrusion* etc. Die Gesamtmenge der durchgeführten Konstruktionschritte bezeichnen wir im Folgenden mit C und die einzelnen dabei verwendeten Konstruktionschritte mit c_i , d.h. es gilt $C = \{c_1, c_2, \dots, c_n\}$.

Jeder Konstruktionschritt führt zu einem neuen geometrischen Objekt; beispielsweise erzeugt die *ErzeugeSpline*-Funktionalität ein neues Splineobjekt. Ein so (im i -ten Schritt) erzeugtes Objekt bezeichnen wir im Folgenden mit g_i , die Gesamtheit aller erzeugten geometrischen Objekte mit G , d.h. es gilt $G = \{g_1, g_2, \dots, g_n\}$. Die Vereinigung beider Mengen C und G beschreiben wir im Folgenden mit $V = C \cup G$.

Ein konkreter Konstruktionschritt c_i hat als Eingabeparameter kein, ein oder mehrere geometrische Objekte $g_{i_1}, g_{i_2}, \dots, g_{i_k}$ und erzeugt (nach Definition) ein geometrisches Objekt g_i .

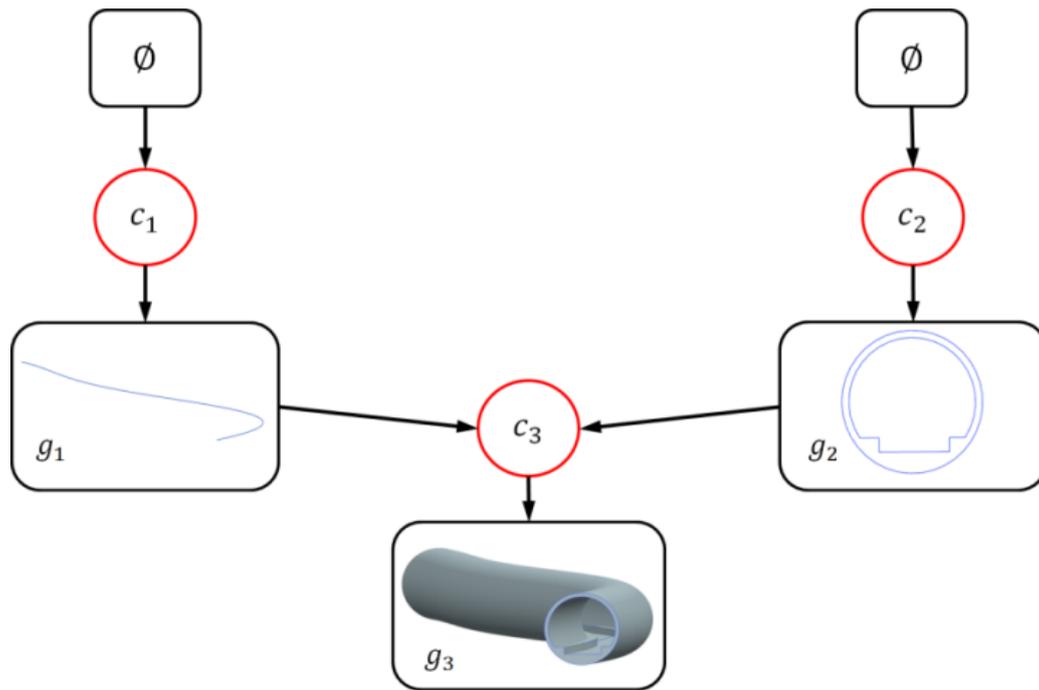


Abbildung 18: Ein einfaches Tunnelmodell, dem ein bipartiter Graph als Konstruktionshistorie zugrunde liegt

Diesen Sachverhalt bezeichnen wir im Folgenden mit $g_i = c_i(g_{i_1}, g_{i_2}, \dots, g_{i_k})$, im Falle einer leeren Parametermenge mit $g_i = c_i(\emptyset)$.

Nun konstruieren wir eine Menge von Kanten E , die aus Paaren von Konstruktionsschritten und zugehörigen Eingangsparametern oder Eingangsparametern und zugehörigen Konstruktionsschritten besteht, folgendermaßen: Führt ein Konstruktionsschritt c_i gemäß der Vorschrift $g_i = c_i(g_{i_1}, g_{i_2}, \dots, g_{i_k})$ zu einem geometrischen Objekt g_i , so fügen wir zur Menge E das Element $e = (c_i, g_i)$ sowie für jeden Eingangsparameter die Elemente $(g_{i_1}, c_i), (g_{i_2}, c_i), \dots, (g_{i_k}, c_i)$ hinzu. Dies führt zu folgender Definition.

Definition (Prozeduraler Modellierungsgraph)

Sei $G = (V, E)$ das Tupel aus den oben festgelegten Mengen V und E , dann nennen wir den so festgelegten Graphen den *prozeduralen Modellierungsgraphen*.

Folgerung

Die Definition der Menge V als disjunkte Vereinigung aus der Menge der Konstruktionsschritte C und der Menge der resultierenden geometrischen Objekte G sowie die Definition der Menge der Kanten, die entweder einen Konstruktionsschritt mit einem geometrischen Objekt oder umgekehrt ein geometrisches Objekt mit einem Konstruktionsschritt verbindet, ergeben unmittelbar, dass der prozedurale Modellierungsgraph ein bipartiter Graph ist.

Anhand eines einfachen CAD-Tunnelmodells werden wir dies verdeutlichen. Der Konstruktionsprozess für das in Abbildung 18 dargestellte Modell ergibt sich folgendermaßen:

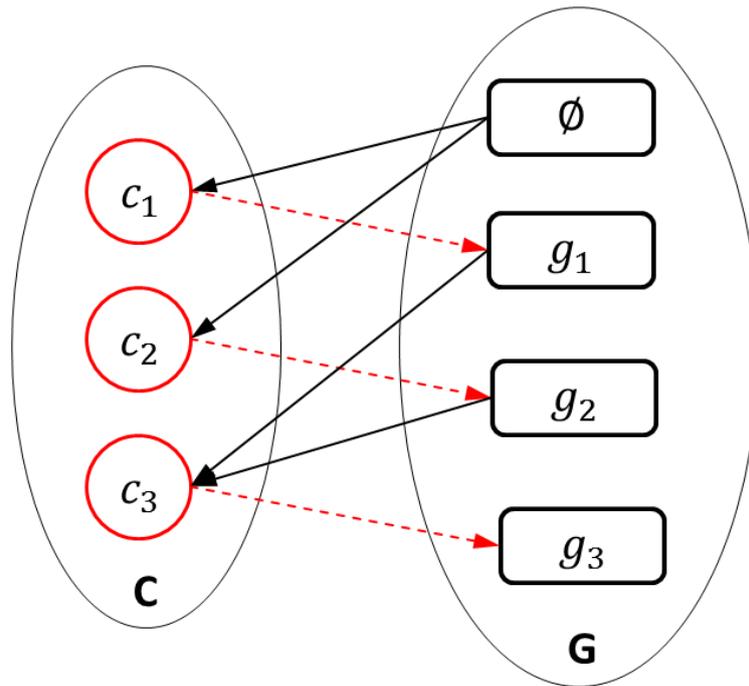


Abbildung 19: Resultierender bipartiter Modellierungsgraph

- 1) Zunächst wird mit Hilfe eines *ErzeugeSpline* Konstruktionsschritts c_1 ein neues Splineobjekt $g_1 = c_1(\emptyset)$ (mit leerer Parameterliste) erzeugt.
- 2) Anschließend wird mit Hilfe der *ErzeugeSkizze* Operation c_2 ein neues Skizzenobjekt $g_2 = c_2(\emptyset)$ (ebenfalls mit leerer Parameterliste) erstellt.
- 3) Die resultierenden geometrischen Objekte g_1 und g_2 werden dann als Eingangsparameter einer *ErzeugeSweep* Operation $g_3 = c_3(g_1, g_2) = c_3(c_1(\emptyset), c_2(\emptyset))$ verwendet.

So erhält man die drei Elemente umfassende Menge der Konstruktionsschritte $C = \{c_1, c_2, c_3\}$, die in Abbildung 19 durch (rote) Kreise symbolisiert wird, sowie die Menge der geometrischen Objekte $G = \{\emptyset, g_1, g_2, g_3\}$, die durch abgerundete (schwarze) Rechtecke dargestellt wird. Die Knotenmenge des prozeduralen Modellierungsgraphen ist durch $V = C \cup G = \{c_1, c_2, c_3, \emptyset, g_1, g_2, g_3\}$ gegeben, während sich die Kantenmenge zu $E = \{(\emptyset, c_1), (\emptyset, c_2), (c_1, g_1), (c_2, g_2), (g_1, c_3), (g_2, c_3), (c_3, g_3)\}$ ergibt. Hierbei beachte man die Möglichkeit, die Parameter eines Konstruktionsschritts rekursiv durch die Konstruktionsschritte zu ersetzen, die ihrerseits diese Parameter erzeugen. Für das Element g_3 bedeutet dies etwa $g_3 = c_3(g_1, g_2) = c_3(c_1(\emptyset), c_2(\emptyset))$.

Anmerkung

Da für Elemente der Kantenmenge der Form $e = (g_i, c_j)$ immer $i < j$ und für Konstruktionsschritte der Form $e = (c_i, g_j)$ stets $i = j$ gilt, ist der prozedurale Modellierungsgraph azyklisch, offensichtlich ist er auch gerichtet.

3.7.3. Ableitung des Sperr-Abhängigkeitsgraphen

Aus dem prozeduralen Modellierungsgraphen wird im Folgenden der sogenannte Sperr-Abhängigkeitsgraph abgeleitet, der dann die konkrete Grundlage für einen Sperr-Algorithmus bildet. Zunächst werden jedoch noch einige weitere benötigte Definitionen und Folgerungen angegeben.

Definition (Elternobjekte vom Grad 1)

Definitionsgemäß wird ein konkretes geometrisches Objekt g_i durch einen Konstruktions-schritt c_i erzeugt. Sind die direkten Eingangsparameter dieses Konstruktions-schritts die geo-metrischen Elemente $g_{i_1}, g_{i_2}, \dots, g_{i_m}$, dann nennen wir diese Objekte Elternobjekte (*Parents*) von g_i vom Grad 1.

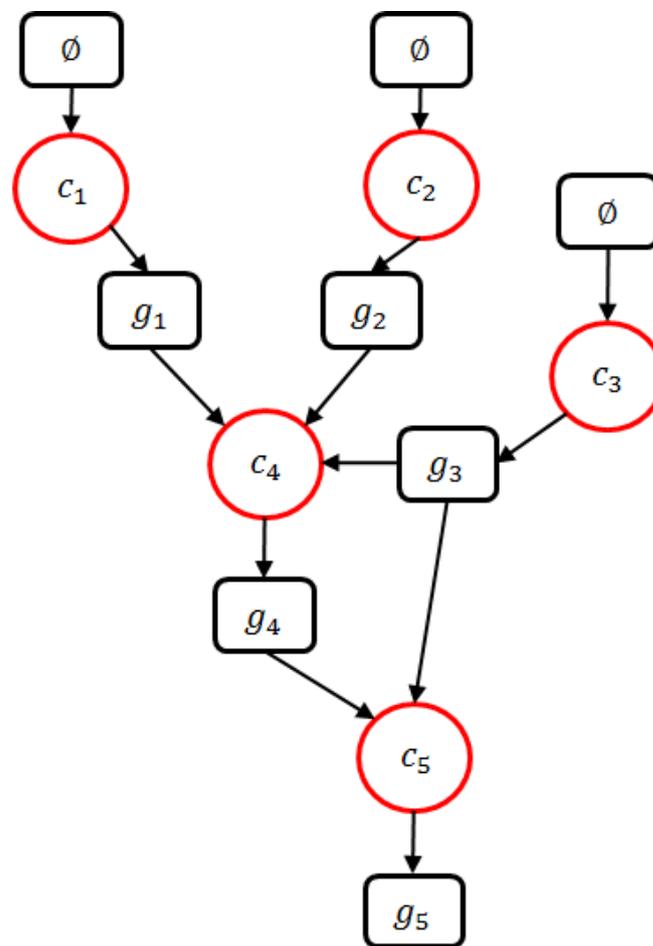


Abbildung 20: Elternelemente (vom Grad 1 und n) in einem prozeduralen Modellierungsgraphen

Definition (Elternobjekte vom Grad n)

Gegeben seien drei Elemente g_i, g_j und g_k (mit $i < j < k$). Es gelten folgende drei Aussagen:

- 1) g_i ist ein Elternelement des Elements g_j vom Grad 1,
- 2) g_i ist kein Elternelement von g_k vom Grad 1,

3) g_j ist ein Elternelement des Elements g_k vom Grad 1,

dann sagen wir g_i ist ein Elternelement von g_k vom Grad 2. Dies verallgemeinern wir auf natürliche Art und Weise, indem wir die Eltern vom Grad 1 eines Elternelements vom Grad $n - 1$ eines Element g_i als Eltern vom Grad n bezeichnen.

Beispiel (Elternelemente vom Grad n)

Im Beispiel aus Abbildung 20 ist g_2 ein Elternelement von g_4 mit Grad 1 und ein Elternobjekt von g_5 mit Grad 2. Die Bedeutung der zweiten Bedingung in der Definition der Elternobjekte wird in Abbildung 20 am Element g_3 deutlich. So hat dieses als Elternelement von g_5 den Grad 1, auch wenn es einen weiteren Weg von g_3 nach g_5 über g_4 gibt.

Mit Hilfe der Elternelemente definieren wir nun die Abhängigkeitsmenge eines Elements g_i .

Definition (Abhängige Elemente)

Die Menge $Abh(g_i)$ der abhängigen Elemente eines Elements g_i ist definiert als $Abh(g_i) := \{g \in G \mid g_i \text{ ist ein Elternobjekt des Elements } g \text{ vom Grad } \geq 1\}$.

Beispiel (Abhängige Elemente)

Im Beispiel aus Abbildung 20 gilt etwa $Abh(g_3) = \{g_4, g_5\}$ und $Abh(g_5) = \{\emptyset\}$.

Zu guter Letzt definieren wir die Menge P der Elternbeziehungen folgendermaßen: Besitzt ein Element g_i ein Elternobjekt g_j vom Grad 1, dann fügen wir der Menge P das Tupel (i, j) hinzu. Die Menge der geometrischen Elemente bezeichnen wir wie oben mit G und definieren dann:

Definition (Sperr-Abhängigkeitsgraph)

Den durch das Tupel (G, P) aus geometrischen Elementen G und Elternbeziehungen P definierten Graphen nennen wir den Sperr-Abhängigkeitsgraphen.

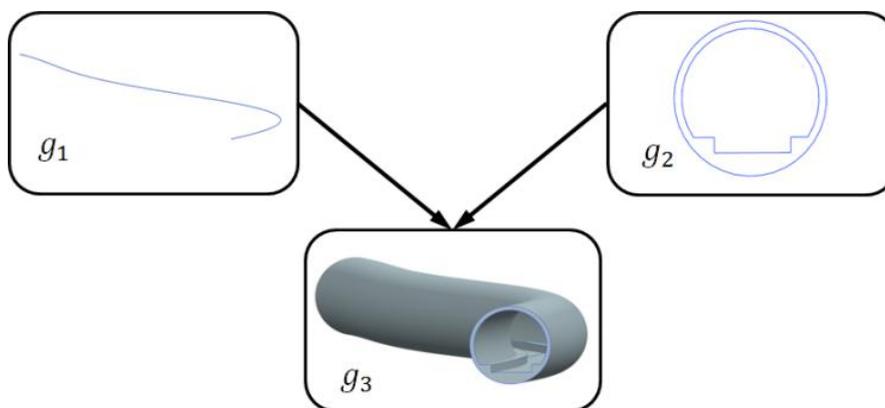


Abbildung 21: Sperr-Abhängigkeitsgraph zum Beispiel aus Abbildung 18 (ohne Berücksichtigung der leeren Parametermenge \emptyset)

Dieser Sperrabhängigkeitsgraph ist offensichtlich azyklisch und gerichtet, was sich unmittelbar aus den Eigenschaften des übergeordneten prozeduralen Modellierungsgraphen ergibt. Für unser ursprüngliches Beispiel ist der Sperrabhängigkeitsgraph in Abbildung 21 dargestellt.

Anmerkung:

Alternativ kann dieser Graph auch folgendermaßen hergeleitet werden: Die Menge der Kanten des bipartiten Modellierungsgraphen ist eine Teilmenge des kartesischen Produkts $E \subseteq V \times V$, so dass E in zwei Teilmengen $E_1 \subseteq G \times C$ und $E_2 \subseteq C \times G$ mit $E = E_1 \dot{\cup} E_2$ zerlegt werden kann. Dann ergeben sich die oben anschaulich definierten Elternbeziehungen formal auch aus der Hintereinanderausführung der beiden Relationen E_1 und E_2 zu $P = E_2 \circ E_1$. Diese alternative Herleitung ist sicherlich kürzer, die erste dafür näher am konkreten Modellierungsprozess orientiert.

3.7.4. Ein graph-basierter Sperr-Algorithmus

Wie bereits erläutert, beruht die Nebenläufigkeitskontrolle der Kollaborationsplattform auf einem pessimistischen Ansatz, der Sperren (*locks*) verwendet: Sobald ein Benutzer die Modifikation eines Konstruktionsschrittes startet, soll dieser Konstruktionsschritt für andere Benutzer gesperrt werden. Die gleichzeitige Bearbeitung durch andere Benutzer wird also unterbunden; somit wird eine durch gleichzeitige Bearbeitung möglicherweise entstehende Konfliktsituation per se vermieden. Da aber ein bestimmter Konstruktionsschritt, genauer das daraus resultierende geometrische Objekt, Eingangsparemeter für später folgende Konstruktionsschritte sein kann, hat seine Änderung implizit einen Einfluss auf in der Konstruktionshistorie folgende Schritte. Dieser Einfluss bezieht sich genau auf die Elemente, die von ihm nach obigen Erläuterungen abhängen, d.h. auf Elemente, die auf einem Pfad – ausgehend von besagtem Konstruktionsschritt – erreicht werden können. Deshalb muss zur Vermeidung von Konflikten ihre gleichzeitige Modifikation ebenfalls unterbunden werden. Im vorangegangenen Abschnitt haben wir die Menge der abhängigen Elemente $Abh(g_i)$ eines geometrischen Elements g_i bestimmt. Das Grundprinzip des im Folgenden beschriebenen Algorithmus ist es nun, genau diese Menge der abhängigen Elemente und natürlich das Element selbst, also $\{g_i \cup Abh(g_i)\}$, im Rahmen der Nebenläufigkeitskontrolle zu sperren. Denn nach unseren Überlegungen stellt die Menge $\{g_i \cup Abh(g_i)\}$ eine obere, aber nicht zwingend eine kleinste obere Schranke für die Menge der Elemente dar, auf welche die Modifikation des Elements g_i einen Einfluss haben kann. Ein Beispiel dafür, dass diese Menge nicht notwendigerweise eine kleinste obere Schranke für die zu sperrenden Elemente darstellt, zeigen wir im Anschluss. Konkret umfasst der Algorithmus nun die folgenden Schritte:

- 1) Sobald ein Benutzer die Modifikation eines Objektes g_i startet, wird automatisch – in der prototypischen Umsetzung ausgelöst durch ein Ereignis der Client API – eine Sperranforderung an die Plattform geschickt.

- 2) Auf der Serverseite wird die Menge abhängiger Elemente $Abh(g_i)$ durch eine Tiefensuche (*depth first search*, Cormen et al. 2010) bestimmt, d.h. im Sperrabhängigkeitsgraphen werden rekursiv die abhängigen Kindelemente durchlaufen.
- 3) Die Vereinigung $\{g_i\} \cup Abh(g_i)$ wird mit der Menge $Sperr$ der bereits gesperrten Elemente $Sperr = \{g_j | g_j \text{ ist gesperrt}\}$ verglichen.
- 4) Ist der Schnitt beider Mengen leer, gilt also $(\{g_i\} \cup Abh(g_i)) \cap Sperr = \emptyset$, so wird die Menge $\{g_i\} \cup Abh(g_i)$ der Menge $Sperr$ hinzugefügt und dem Benutzer wird die Sperrung gewährt.
- 5) Ist der Schnitt beider Mengen nicht leer, gilt also $(\{g_i\} \cup Abh(g_i)) \cap Sperr \neq \emptyset$, so wird dem Benutzer die Sperrung verweigert und er wird über die API gezwungen, seinen Modifikationsversuch (ergebnislos) zu beenden.
- 6) Wurde die Sperrung gewährt, führt der Benutzer nun seine Modifikationsschritte durch und sendet das Ergebnis an die Kollaborationsplattform.
- 7) Hier wird die Änderung in das zentrale Modell eingepflegt und alle anderen Benutzer werden über diese Änderungen benachrichtigt und dabei gezwungen, diese Änderungen unmittelbar in ihre lokale Kopie des zentralen Modells einzubauen. Dieser Einbau erfordert dabei implizit eine Neuberechnung aller abhängigen Elemente auf der Clientseite.
- 8) Schließlich wird die Menge $g_i \cup Abh(g_i)$ aus der Menge $Sperr$ gelöscht und der Benutzer über den erfolgreichen Abschluss seiner Änderung informiert.

Anmerkung: Nach obigen Ausführungen könnte man zunächst vermuten, dass man bei der Modifikation eines Konstruktionsschritts auch dessen Elternelemente sperren muss: Denn eine Modifikation eines Elternelement hätte ja dann wiederum Einfluss auf genau diesen Konstruktionsschritt, so dass eine angedachte Änderung des abhängigen Konstruktionsschritts möglicherweise gar nicht mehr zulässig wäre. Da jedoch vor der Änderung des Elternelements ebenfalls obiger Algorithmus durchlaufen werden würde, ergäbe sich beim Vergleich in Schritt 3 eine nicht leere Schnittmenge, womit gemäß Schritt 5 die Modifikation des Elternelements unterbunden werden würde.

Der präsentierte Algorithmus garantiert die Konsistenz des zentral verwalteten prozeduralen Modells und der lokalen Kopien auf den Client-Seiten, indem gemäß eines pessimistischen Ansatzes Inkonsistenzen von vorneherein ausgeschlossen werden. Da er aber im Gegensatz zu weit verbreiteten anderen Ansätzen nicht das komplette Modell sperrt, sondern nur explizit die gemäß obigem Algorithmus betroffenen Bereiche des Modells, ermöglicht er eine Zusammenarbeit bei der Modellierung derart, dass mehrere Experten synchron dasselbe Modell bearbeiten und tatsächlich auch synchron modifizieren können. Um unter Verwendung des präsentierten Ansatzes zur Nebenläufigkeitskontrolle möglichst große Bereiche des Modells für eine gleichzeitige Bearbeitung zuzulassen, werden wir in Kapitel 5.4 im Kontext der trasenbasierten Tunnelplanung ein Konzept vorstellen, welches eine Zerlegung des Gesamtmo-

dells in verschiedene „unabhängige“ Teilmodelle erlaubt. Aus mathematischer Sicht bedeutet dies gerade eine Zerlegung des Sperrabhängigkeitsgraphen in disjunkte Zusammenhangskomponenten.

Abschließend wollen wir exemplarisch zeigen, dass die oben bestimmte Menge der zu sperrenden Elemente nicht notwendigerweise optimal gewählt wurde, was einen Ausblick auf weitere mögliche Forschungsfragen ergibt. Dazu zeigt das folgende Beispiel, dass die oben bestimmte Menge der als zu sperrenden Elemente $\{g_i \cup Abh(g_i)\}$ für die Konsistenzsicherung hinreichend, aber nicht notwendig ist, oder anders formuliert, dass die Menge $\{g_i \cup Abh(g_i)\}$ eine obere Schranke für die Menge der zu sperrenden Elemente ist, aber keine kleinste obere Schranke darstellt:

An einem Spline wurde über eine Arbeitsebene eine Skizze zur Definition eines einfachen kreisförmigen Tunnelprofils angefügt, wie es in Abbildung 22 dargestellt ist. Die Existenz dieser Arbeitsebene, wie sie in konkreten CAD-Werkzeugen üblich ist, hat einen ganz entscheidenden Einfluss auf den konkreten Sperr-Abhängigkeitsgraphen.



Abbildung 22: Spline zur Repräsentation einer Trasse und eines kreisförmigen Tunnelprofils

Denn im zugehörigen Sperr-Abhängigkeitsgraphen hängt die Skizze nun vom übergeordneten Spline ab, da sich die Skizze gerade in der Arbeitsebene befindet, die mit dem Spline verknüpft ist. Somit wird im präsentierten Algorithmus bei der Modifikation des Splines automatisch auch die Skizze gesperrt, obwohl die Modifikation des Splines in diesem Fall keinerlei Einfluss auf die abhängige Skizze hat, so dass eine gleichzeitige Modifikation der Skizze möglich wäre. Extrudiert man allerdings im vorliegenden Beispiel die Skizze entlang des Splines, wie in Abbildung 23 dargestellt, so ändert sich die Situation grundlegend.



Abbildung 23: Das Beispiel aus Abbildung 22 um eine Extrusion erweitert

Eine gleichzeitige Modifikation von Spline und Skizze kann zu sogenannten sich selbstschneidenden Oberflächen führen, was zu einer Zerstörung des Volumenkörpers führt, ob-

wohl sowohl die Modifikation des Splines als auch der Skizze als Einzelmodifikationen erlaubt wären. Hier ist also die Menge der abhängigen Elemente, wie sie in unserem Algorithmus präsentiert wurde, tatsächlich eine kleinste obere Schranke. Die Präzisierung eines allgemeinen Algorithmus zur Bestimmung der kleinsten oberen Schranke im gegebenen Kontext wurde im Rahmen dieser Arbeit nicht näher verfolgt, kann aber einen interessanten zukünftigen Forschungsaspekt darstellen.

3.7.5. Sperren über verschiedene Levels von Detail hinweg

In Kapitel 3.4 hatten wir gesehen, dass in das im *3DTracks* Projekt entwickelte prozedurale Modell das Prinzip verschiedener Levels of Detail (LoD) fundamental integriert wurde. Hierdurch wird berücksichtigt, dass verschiedene Planer unterschiedlich abstrakte Sichtweisen auf den Planungsprozess und den zugehörigen Detaillierungsgrad des Planungsmodells haben. Der im letzten Kapitel präsentierte Sperr-Algorithmus kann auch in diesem Kontext vorteilhaft angewandt werden. So ist es möglich, dass ein spezieller Planer das Modell in einem für ihn geeigneten LoD lädt und bearbeitet, während die Plattform automatisch die Konsistenz über alle verschiedenen LoDs hinweg garantiert. Ein typisches Beispiel ist etwa folgende Situation: Ein Trassierungsexperte nimmt auf LoD 1 einige letzte Modifikationen am Trassenverlauf vor, während gleichzeitig der Tunnelexperte bereits innerhalb des Tunnelquerschnitts die einzelnen Räume (siehe Kapitel 3.4) auf LoD 4 definiert. Der Trassierungsexperte lädt und betrachtet dazu das Modell nur auf LoD 1, in dem ausschließlich die Trasse modelliert wurde; seine Änderungen werden jedoch konsistent in höherer LoDs propagiert. Die grundlegende Idee dafür ist, dass der LoD abhängiger Operationen stets größer oder gleich dem LoD der jeweiligen Elternoperation ist. Mit obigen Bezeichnungen bedeutet dies

$$\forall c_i \in \text{Abh}(c) \text{LoD}(c_i) \geq \text{LoD}(c).$$

Somit werden beim Laden des Modells bis zu einem bestimmten Detaillierungsgrad nur die Elemente bis zu einem bestimmten LoD berücksichtigt, Elemente höherer LoDs können in dem darzustellenden Konstruktionsbaum nicht auftreten. Im gegebenen Beispiel bedeutet dies, dass der Trassierungsexperte selbst keinerlei Kenntnis über die von der Trasse abhängigen Elemente verfügen muss, ja dass er diese nicht einmal geladen haben muss oder sie eventuell sogar gar nicht laden darf. Modifiziert nun ein Planer das Modell, welches er nicht bis zum maximalen LoD geladen hat, so erhält der präsentierte Algorithmus weiterhin die Konsistenz zwischen seinem lokalen Modell, dem zentralen Modell und den weiteren möglicherweise geladenen Replikaten: Denn vor der Modifikation durch den Benutzer wird seine Sperranfrage an die Plattform gesandt, die auch Elemente höherer LoDs kennt und generell im Sperr-Abhängigkeitsgraphen berücksichtigt. Selbstverständlich muss der Benutzer über die Ablehnung einer Sperranfrage informiert werden, denn seine Anfrage wird durch abhän-

gige Elemente in höheren LoDs verhindert, so dass er diese Elemente in seinem lokalen Modell selbst gar nicht identifizieren kann.

Gemäß diesen Ausführungen ist der präsentierte Sperr-Algorithmus hervorragend mit dem integrierten Prinzip verschiedener LoDs verträglich, insbesondere ist er dies, wenn ein Modell nur bis zu einem bestimmten LoD geladen wird, so dass tatsächlich eine aktive Modellierung über verschiedene LoDs hinweg möglich wird.

3.8. Praxistauglichkeit des prozeduralen Modells

Das beschriebene multi-skalige prozedurale Modell wurde zunächst für die Modellierung trassenbasierter Infrastrukturbauwerke und dabei mit Fokus auf der Modellierung von U-Bahn-Tunneln konzipiert. Die in das Modell integrierten prozeduralen Operationen, ihre Eigenschaften und Parameter haben sich in den Anwendungsbeispielen und Testszenarien als ausreichend und gut geeignet erwiesen; insbesondere konnte eine Reihe detaillierter Tunnel-Modelle erstellt werden (Borrmann et al. 2014). Es stellt sich damit unmittelbar die Frage, ob der beschriebene Ansatz auch für andere Modelltypen eingesetzt werden kann. Dazu wurden die vorgestellten Konzepte erfolgreich im Bereich der synchronen kollaborativen Modellierung von Bauwerken und dabei detailliert bei der Modellierung von Bahnhofsbauwerken getestet und somit auf Alltagstauglichkeit getestet.

Abbildung 24 zeigt exemplarisch das Modell des Bahnhofsbauwerks Ostbahnhof München,

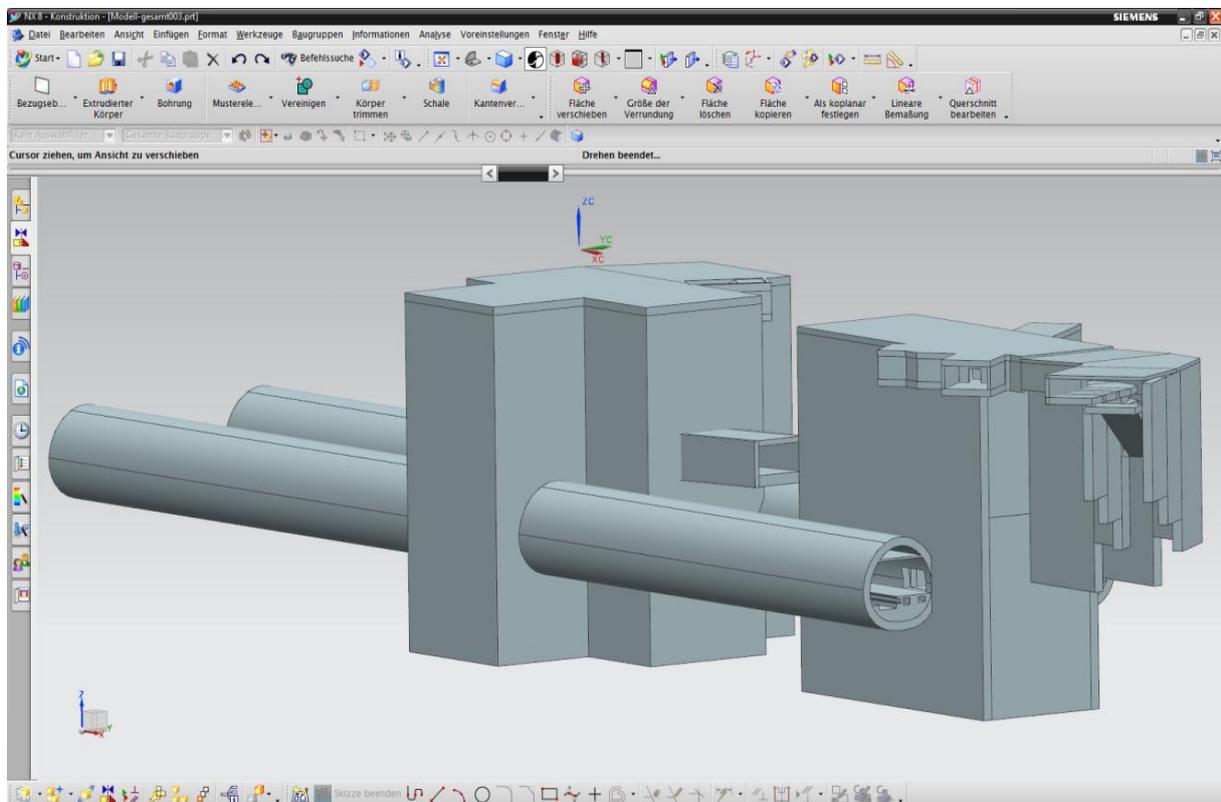


Abbildung 24: Modellierung eines Bahnhofsbauwerks auf der Basis des vorgestellten prozeduralen Modells

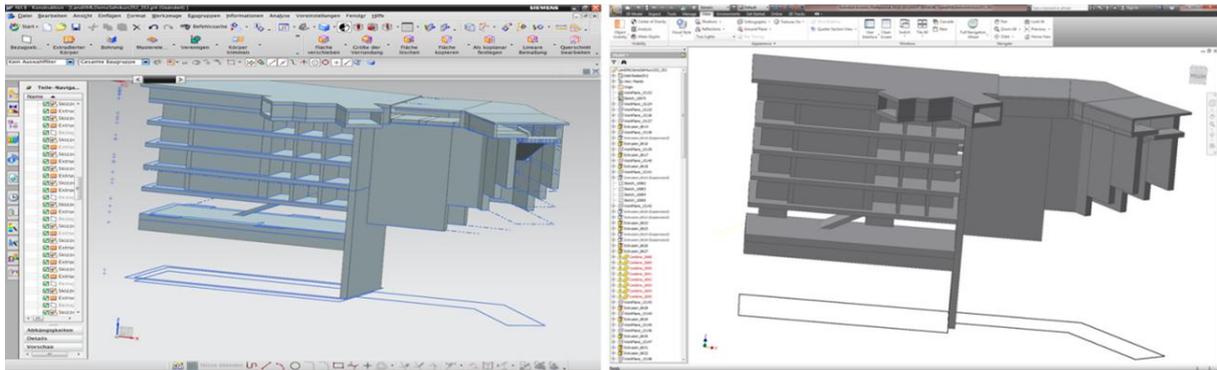


Abbildung 25: Synchrone Modellierung eines Bahnhofsbauwerks mit den beiden CAD-Werkzeugen Siemens NX und Autodesk Inventor

dem Ausgangsbahnhof der geplanten Zweiten Stammstrecke in München¹⁵, das im Rahmen einer Plausibilisierung der Plattform erstellt wurde. Dabei mussten einige konkrete Ergänzungen und Erweiterungen der Struktur des prozeduralen Modells, aber keine Änderung des ursprünglichen Konzepts selbst vorgenommen werden. Diese Notwendigkeit ergibt sich im Wesentlichen aus der Tatsache, dass die Modellierung der Tunnelbauwerke auf der Modellierung der Trasse basiert, während die Modellierung der Bahnhofsbauwerke durch das Referenzieren der sogenannten Leitskizze getrieben ist, die den prinzipiellen Grundriss des Bauwerks angibt (Schwietert 2014). Dazu mussten Beziehungen zwischen einzelnen Skizzenelementen hinzugefügt werden, die sich aber nahtlos in das Schema des gegebenen prozeduralen Modells integrieren ließen.

Weiterhin konnten bei der konkreten Modellierung des Ostbahnhofs der neuen geplanten Zweiten S-Bahn-Stammstrecke in München die Konzepte eines synchronen Modellierungsprozesses mit den beiden Systemen Siemens NX und Autodesk Inventor plausibilisiert werden. Dies ist für zwei Teilbauwerke des Ostbahnhofs in Abbildung 25 exemplarisch dargestellt. Im Besonderen konnte gezeigt werden, dass die Antwortzeiten der Kollaborationsplattform auch bei großen Anzahlen von Parametern (hier konkret einige 100) sehr gut im Bereich der Erwartungen lagen.

¹⁵ <http://www.2.stammstrecke-muenchen.de/>

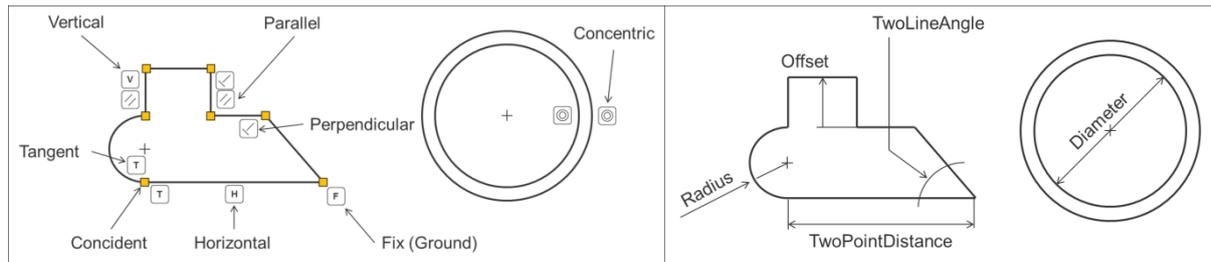


Abbildung 26: Constraints werden typischerweise in topologische und dimensionale Constraints unterteilt nach (Jubierre und Borrmann 2015).

3.9. Regelbasierte prozedurale Modellierung mit Unterstützung der Kollaborationsplattform

Prozedurale Modelle eignen sich zum Austausch von Geometriemodellen und können hier entscheidende Teile des *Design Intents* abbilden (Bianconi 2005, Bianconi et al. 2006, Kim et al. 2008). Dabei ist das Konzept der Constraints hilfreich, die es ermöglichen, Beziehungen zwischen geometrischen Elementen zu definieren; typisch sind hier etwa die Parallelität von Linien, die Konzentrität von Kreisen oder die Symmetrie von bestimmten Elementen (Pratt und Kim 2006, Ma et al. 2008). Abbildung 26 zeigt gängige Constraints, welche häufig in die zwei Gruppen der topologischen und der dimensional Constraints unterteilt werden. Constraints ermöglichen es zwar, wichtige Teile des *Design Intent* abzubilden, können aber nicht alle Aspekte abdecken (Flurl et al. 2015). Dazu werden wir im Folgenden anhand zweier Beispiele zeigen, dass sie Schwächen aufweisen, und werden als Ausblick zum Abschluss dieses Kapitels ein Konzept präsentieren, wie derartige Probleme – insbesondere innerhalb eines kollaborativen Modellierungsprozesses – in zahlreichen Fällen gelöst werden können. Ein weiteres interessantes Beispiel für diese Unzulänglichkeit von Constraints findet man in der Einführung der Dissertation von Bidarra (Bidarra 1999). Auch das dort geschilderte Problem lässt sich mit der hier entwickelten Strategie unschwer bewältigen. Das folgende Konzept wurde in gemeinsamer Forschung mit Dominic Singer, einem wissenschaftlichen Mitarbeiter des Lehrstuhls für Computergestützte Modellierung und Simulation an der technischen Universität München, entwickelt und ist – wie in den Überlegungen zuvor – speziell auf einen Modellierungsprozess zugeschnitten, in welchem die verschiedenen Benutzer die CAD-Werkzeuge ihrer Wahl verwenden können.

3.9.1. Problembeschreibung anhand zweier Beispiele

Dimensionale Constraints erlauben es, algebraische Beziehungen zwischen verschiedenen Elementen in Form von Gleichungen zu definieren. So könnte man etwa festlegen, dass die Länge l_1 einer ersten Linie immer das Doppelte der Länge l_2 einer zweiten Linie zuzüglich eines bestimmten festen Abstands d betragen muss, d.h. es besteht die Beziehung $l_1 = 2 \cdot l_2 + d$. Derartige Beziehungen lassen sich gut mit Constraints abbilden, und die Constraint-

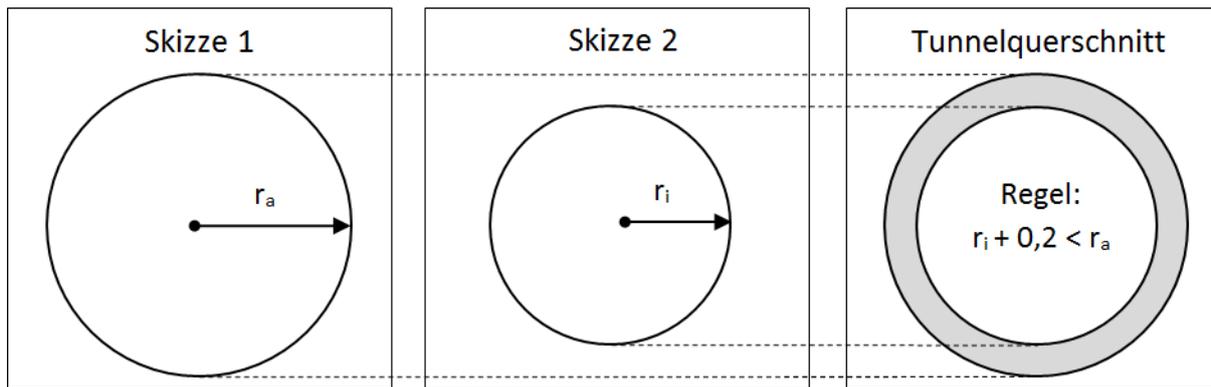


Abbildung 27: Eine Ungleichung definiert eine Beziehung zwischen zwei Profilen. Diese Ungleichung kann nicht durch gängige Constraints abgebildet werden.

solver (Bouma et al. 1995) moderner CAD-Werkzeuge bieten ausgereifte Funktionalitäten, um die Verträglichkeit zwischen verschiedenen Constraints sowohl zu prüfen als auch zu gewährleisten. Aber im Fall von Ungleichungen reichen diese Mechanismen nicht immer aus.

Beispiel 1:

Zur Erzeugung eines einfachen Tunnelmodells werden – ähnlich zu vorab gegebenen Beispielen, etwa dem in Kapitel 3.4 – zwei Skizzen entworfen, welche durch zwei Kreise den Querschnitt eines Tunnelbauwerks definieren. Beide Skizzen werden entlang eines Splines, welcher die Trasse repräsentiert, extrudiert und die resultierenden Volumina voneinander subtrahiert.

Offensichtlich führt diese Vorgehensweise nur dann zu einem sinnvollen Tunnelmodell, wenn der Radius r_a des äußeren Kreises echt größer als der Radius r_i des inneren Kreises ist, d.h. es gilt $r_a > r_i$ (Abbildung 27). Eine derartige Ungleichung lässt sich aber nicht durch Constraints abbilden, so dass diese offensichtlich fundamentale Beziehung auch nicht über Constraints in das prozedurale Modell integriert werden kann.

Beispiel 2:

Ein komplexeres Beispiel ist der Modellierung von Brücken entnommen. Die Breite des Überbaus einer Brücke sei durch einen Parameter l festgelegt (siehe Abbildung 28). Je nach der Größe dieses Parameters variiert die Anzahl der benötigten Stützpfiler. Im Idealfall passt sich diese Anzahl mit jeder Veränderung des Parameters l automatisch an. Mit den bisher besprochenen Möglichkeiten eines prozeduralen Modells, insbesondere mit Hilfe von Constraints, lässt sich diese Beziehung nicht abbilden.

Im Folgenden stellen wir ein Konzept vor, das genau dieses Problem im Rahmen einer kollaborativen Modellierung dadurch löst, dass es den Benutzern erlaubt, flexibel und dynamisch Regeln über die Kollaborationsplattform in den zuvor vorgestellten Modellierungsprozess zu integrieren. Dazu betrachten wir zunächst das Konzept von Regeln etwas allgemeiner, bevor wir unser eigenes Konzept konkret präsentieren.

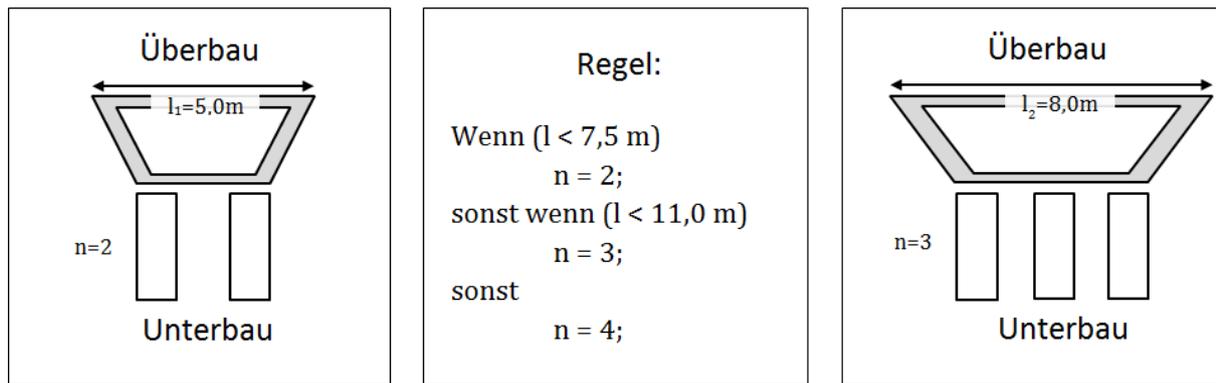


Abbildung 28: Ein einfaches Brückenmodell, in dem die Breite des Überbaus die Anzahl der nötigen Stützpfiler definiert wird

3.9.2. Regelbasierte Systeme

Der Ursprung regelbasierter Systeme, deren Vorteil (Puppe 1991) in „*ihrer Modularität und der Anpassungsfähigkeit des Regelformalismus an konkrete Anforderungen*“ sieht, stammt aus dem Forschungsgebiet der künstlichen Intelligenz und kann etwa auf die frühen Siebzigerjahre des letzten Jahrhunderts datiert werden (Dym 1985, Hayes-Roth und Jacobstein 1994). Eine ausführliche und klassische Diskussion der Vor- und Nachteile derartiger Systeme findet man etwa bei (Adeli 1988). Heute werden regelbasierte Systeme in vielen Gebieten angewandt und erfassen weite Bereiche des sogenannten Wissensmanagements: So finden wir sie etwa in Diagnosesystemen, bei der Verwaltung von Geschäftsregeln, bei Entscheidungsunterstützungssystemen und speziell im Bereich des sogenannten *Knowledge Based Engineering* (Dym und Levitt 1991). Die grundlegenden Komponenten eines solchen Systems sind die Wissens- oder Faktenbasis, die *Rule Engine* und eine Benutzerschnittstelle zur Definition der Regeln selbst, die im Bereich der CAD-Modellierung auch direkt in ein CAD-System integriert werden kann (Griffin und Lewis 1989).

Die *Rule Engine* testet die gespeicherten Regeln gegenüber der Faktenbasis, die in unserem Fall durch das prozedurale Modell gegeben ist. Diesen Prozess bezeichnet man als *Pattern Matching*, die Gesamtheit der auszuführenden Regeln als Konfliktmenge (Reijnders 2012), die vor der konkreten Ausführung in einer *Agenda* mit Hilfe einer Konfliktlösungsstrategie sortiert wird. Die grundlegenden Techniken sind hier die sogenannte Vorwärts- und Rückwärtsverkettung (Hartmann und Lehner 1990, Salvat und Mugnier 1996).

Die Hersteller moderner CAD-Werkzeuge haben die oben beschriebene Problematik erkannt und in ihre Produkte regelbasierte Systeme integriert, etwa *iLogic*¹⁶ für Autodesk Inventor oder *Knowledge Fusion*¹⁷ für Siemens NX. Diese regelbasierten Systeme sind aber genau auf

¹⁶ <https://knowledge.autodesk.com/de/support/inventor-products/learn-explore/caas/CloudHelp/cloudhelp/2014/DEU/Inventor/files/GUID-B98DF82D-E489-4C19-8351-11C0ED63C349-htm.html>

¹⁷ https://m.plm.automation.siemens.com/de_ch/Images/knowledge_fusion_tcm1425-62405.pdf

eine Produktlinie ausgerichtet, so dass die gespeicherten Regeln nicht zwischen CAD-Systemen verschiedener Hersteller ausgetauscht werden können und eine kooperative Zusammenarbeit unter Verwendung der Regelsysteme damit von vornherein – ähnlich dem diskutierten Szenario bei der CAD-Modellierung – generell ausgeschlossen ist.

3.9.3. Integration von Regeln in die Kollaborationsplattform

Um die in den Eingangsbeispielen beschriebene Problematik, speziell im Hinblick auf eine kooperative Modellierung zu mindern, soll dem Benutzer die Möglichkeit gegeben werden, komplexere Regeln zu definieren und über das prozedurale Modell in den Modellierungsprozess zu integrieren. Gleichzeitig müssen Mechanismen bereitgestellt werden, welche diese Regeln auf ihre Gültigkeit prüfen und angemessene, ebenfalls benutzerdefinierte Aktionen ausführen. Zunächst befassen wir uns jedoch mit der Form von Regeln im gegebenen Kontext.

Die Form kollaborativer Regeln

In der einfachsten Darstellung hat eine Regel die Form

WENN *Voraussetzung* DANN *Konsequenz*,

wobei *Voraussetzung* eine Bedingung repräsentiert, die im Fall, dass ihre Auswertung durch die *Rule Engine* den Wahrheitswert „wahr“ liefert, zur Durchführung von Schritten führt, die im Teil *Konsequenz* definiert sind (Reijnders 2012). Die einfachste Erweiterung ist es, eine Alternative in der *Konsequenz* zuzulassen

WENN *Voraussetzung* DANN *Konsequenz*₁ SONST *Konsequenz*₂,

wobei *Konsequenz*₂ die Schritte definiert, die auszuführen sind, falls die Evaluation der *Voraussetzung* den Wahrheitswert „falsch“ liefert. Die nächste Erweiterung besteht darin, in der *Konsequenz* selbst neue Aussagen der Form „WENN ... DANN ... SONST ...“ zuzulassen. Dies führt zu einer rekursiven Definition von Regeln und entspricht der Form, die im Rahmen dieser Arbeit unterstützt wird.

Die Voraussetzung im gegebenen Kontext der prozeduralen Modellierung

Wie oben gezeigt, bilden Parameter die Grundlage für eine feature-basierte Modellierung (Kim und Pratt 2008). In unserem Ansatz werden in der *Voraussetzung* algebraische Beziehungen zwischen den im Modell enthaltenen Parametern in Form von Gleichungen und Ungleichungen festgelegt. Ihre syntaktische Gültigkeit kann bereits bei ihrer Definition, also konkret bei der Benutzereingabe, durch einen Parser geprüft werden, während ihre semantische Sinnhaftigkeit natürlich in der Verantwortung des definierenden Ingenieurs liegt. Die spezifischen Parameter, die in einer Regel verwendet werden können, sind innerhalb des pro-

zeduralen Modells definiert und werden konkret durch ein sogenanntes „*RuleUsable*“ Attribut ausgezeichnet.

Bei der konkreten syntaktischen Formulierung der Regeln verwenden wir eine Darstellung ähnlich der Definition von Objekten in der Objektorientierten Programmierung, da eine derartige Darstellung den meisten in diesem Kontext arbeitenden Ingenieuren zumindest prinzipiell bekannt sein dürfte:

So wird etwa durch *Sketches*[„SketchID_02“].*Circles*[„CircleID_01“].Radius der Radius des Kreises mit der Identität „CircleID_01“ beschrieben, welcher Teil der Skizze mit der Identität „SketchID_02“ ist. Selbstverständlich sollte der Benutzer nicht gezwungen werden, die zur Benutzung verfügbaren Parameter mühsam manuell aus dem Schema des prozeduralen Modells zu entnehmen. So werden ihm automatisch bei der Regeldefinition nur Parameter bei der Auswahl einer prozeduralen Operation zur Verwendung angeboten, die zur Regeldefinition vorgesehen sind.

Die Konsequenz im gegebenen Kontext

In der *Konsequenz* werden die Schritte definiert, die auszuführen sind, nachdem die Voraussetzung mit „*wahr*“ bewertet wurde. Im gegebenen Kontext findet die Bewertung auf der Kollaborationsplattform innerhalb einer speziellen Komponente zur Regelinterpretation statt, die prinzipiell der oben genannten *Rule Engine* entspricht. Die vollständige Bewertung einer Regel führt zu Rückmeldungen an den Benutzer oder auch zu geometrischen Operationen, die automatisch am gemeinsamen Modell und somit auch implizit bei den an einer kollaborativen Sitzung teilnehmenden Clients ausgeführt werden müssen.

Prinzipiell wurden in der gegebenen Lösung drei Typen von Konsequenzen integriert. Eine „*Reject*“-Anweisung zeigt dem Benutzer an, dass die ausgeführte Aktion einer definierten Regel widerspricht, und zwingt diesen, die zuletzt durchgeführte Aktion zurückzunehmen. Selbstverständlich muss dem Benutzer dieser Konflikt und seine Ursache mitgeteilt und veranschaulicht werden.

Der zweite Fall „*Warning*“ tritt auf, wenn eine Aktion zwar einer definierten Regel zuwiderläuft, aber die Verantwortung für die Lösung des Konflikts dem Benutzer überlassen wird – eine Strategie, die in der kollaborativen Modellierung durchaus Anwendung findet (Munson und Dewan 1994, 1996). Diese zwei ersten Möglichkeiten sind weniger komplex als die dritte Möglichkeit „*Modify*“, bieten aber bereits beachtliches Potential, den Benutzer zu unterstützen, und rechtfertigen bereits den Einsatz von kollaborativen Regeln.

Die dritte Möglichkeit „*Modify*“ wurde in unseren Untersuchungen zumindest in Grundzügen geprüft; sie eröffnet ein umfangreiches und hochinteressantes zukünftiges Forschungsfeld. Diese Möglichkeit besteht in der Definition einer Menge von Operationen, welche automatisch ausgeführt werden sollen, sobald eine Voraussetzung mit „*wahr*“ evaluiert wurde.

Generell werden im Fall einer Modifikation eines geometrischen Elements durch den Benutzer das betroffene Element und die davon abhängigen Elemente automatisch durch die Kollaborationsplattform für andere Benutzer gesperrt. Nachdem der Benutzer seine Modifikation beendet hat, wird das resultierende modifizierte Element unverzüglich und automatisch an die Plattform gesandt. In unserem Ansatz kommt nun genau hier die am Server integrierte Regelinterpretationskomponente zum Einsatz, und zwar natürlich, bevor das modifizierte Element in das geteilte Modell integriert und an die anderen Teilnehmer weitergeleitet wird. Zunächst werden die Regeln, welche das zu modifizierende Element referenzieren, evaluiert und anschließend die in den *Konsequenzen* definierten Schritte ausgeführt. Die Konsequenzen sind in den ersten beiden Fällen Meldungen an den Benutzer, wobei im ersten Fall die Modifikation des Modells zugelassen und im zweiten verwehrt wird. Im dritten Fall werden Schritte definiert, welche die drei verschiedenen prinzipiellen Modellierungsoperationen umfassen: das Hinzufügen, das Modifizieren sowie das Löschen von vorhandenen Elementen. Um diese Schritte ausführen zu können, muss – wie im kollaborativen Modellierungsprozess zwischen den verschiedenen Teilnehmern einer kollaborativen Sitzung – die Konsistenz zwischen dem gemeinsamen Modell und den lokalen Replikaten gewahrt bleiben. Dieses Problem wird dadurch gelöst, dass die Regelinterpretationskomponente diese Schritte genau über die gleiche Schnittstelle ausführt wie alle anderen Teilnehmer einer kollaborativen Sitzung. Diese Strategie bietet den entscheidenden Vorteil, dass alle Mechanismen der Nebenläufigkeitskontrolle implizit verwendet werden und somit keine zusätzlichen Funktionalitäten integriert werden müssen.

Eine konkrete Regel, die aus Beispiel 1 in der Einführung resultiert, beschreiben beispielhaft die folgenden Anweisungen:

```
DEFINE param_ref1 := Sketches[SketchId_02].Circles[CircleId_01].Radius;  
DEFINE param_ref2 := Sketches[SketchId_03].Circles[CircleId_01].Radius;  
IF param_ref1 > param_ref2 THEN Warning(" ... .. ");
```

Offensichtlich werden hierin die Radien zweier Kreise verglichen und im Fall, dass der erste Radius größer als der zweite ist, wird eine Warnung an den betreffenden Benutzer zurückgegeben.

3.9.4. Die Kombination verschiedener Regeln

Ein grundlegendes Problem bei der Verwendung von Regeln ist die Kontrolle, inwieweit sich verschiedene Regeln untereinander zuwiderlaufen. Aus theoretischer Sicht wird dieses Problem etwa im sogenannten Resolutionskalkül untersucht (Hartmann und Lehner 1990, Schöning 2000). In der Praxis haben sich verschiedene Verfahren zur Konfliktlösung etabliert (Puppe 1991), wobei sich bei einer großen Anzahl von Regeln immer wieder erhebliche

Schwierigkeiten ergaben, bei komplexeren Strategien Konfliktlösungen in einer angemessenen Zeitspanne zu finden. Im Rahmen unserer Arbeit haben wir nur eine kleine Anzahl von Regeln zugelassen (ca. 10 – 50) und die einfachste Lösungsstrategie verwendet, nämlich die Regeln einfach in der Reihenfolge ihrer Definition sukzessive abzuarbeiten. Der im Rahmen dieser Arbeit interessante Punkt, ob es prinzipiell möglich ist, über die Plattform derartige Regeln in den Planungsprozess zu integrieren, kann insoweit positiv beantwortet werden; weitergehende Untersuchungen in diesem Kontext bieten sich als möglicher zukünftiger Forschungsgegenstand an.

3.10. Zusammenfassung

Im Rahmen dieser Forschung wurden prozedurale Geometriemodelle und ihre Eigenschaften ausführlich untersucht. Dabei wurde die Idee entwickelt und verfolgt, über ein neues prozedurales Geometriemodell einen Austausch von Modellen zwischen Anwendern zu ermöglichen, die CAD-Werkzeuge unterschiedlicher Hersteller verwenden. Dieses Modell integriert zudem fundamental das Konzept verschiedener Levels of Detail, welche die einzelnen Abstraktionsstufen der an der Planung beteiligten Experten abbilden sollen. Erfolgreich wurde es als Basis für einen synchronen Modellierungsprozess verwendet, in dessen Zentrum die neu konzipierte Plattform *Kollabsolver* stand, deren komponenten-basierte Architektur die Grundlage für eine flexible Entwicklung und Erweiterung ist; insbesondere wurden Komponenten fundamental integriert, die zur Lösung der im ersten Kapitel vorgestellten Problemstellungen beitragen. Dabei wurde vor allem gezeigt, dass es den verschiedenen Experten während des kollaborativen Modellierungsprozesses über die Plattform im Gegensatz zu anderen Ansätzen erlaubt ist, innerhalb einer kollaborativen Sitzung verschiedene CAD-Werkzeuge zu verwenden, etwa Siemens NX und Autodesk Inventor. Weiterhin unterstützt die Plattform die im präsentierten Modell integrierten verschiedenen LoDs, so dass einzelne Benutzer tatsächlich in der ihrer Aufgabe entsprechenden Abstraktionsstufe arbeiten und modellieren können.

Der zugrunde liegende prozedurale Modellierungsprozess wurde mathematisch durch einen bipartiten Graphen beschrieben, aus welchem der sogenannte Sperr-Abhängigkeitsgraph abgeleitet wurde, den wir dann als Basis für einen Algorithmus zur Nebenläufigkeitskontrolle eingesetzt haben. Hier ist es verschiedenen Planern erlaubt, synchron ein gemeinsames Modell zu bearbeiten und insbesondere zu modifizieren, solange sich ihre Modifikationen nur auf unabhängige Bereiche dieses Graphen beziehen. Im Gegensatz zu anderen Ansätzen erlaubt die präsentierte Vorgehensweise, dass die verschiedenen Modellierer aktiv in diesem synchronen Modellierungsprozess mitwirken und zwar gerade unter der Prämisse, die CAD-Werkzeuge ihrer Wahl zu verwenden. In Kapitel 5.4 werden wir sehen, wie wir die Einsatzmöglichkeit der vorgestellten Strategie zur Nebenläufigkeitskontrolle weiter verbessern können, indem wir das Modell im Fall von trassenbasierten Infrastrukturbauwerken entsprechend der zugrunde liegenden Trasse in unabhängige Teilmodelle zerlegen. Abschließend wurde als

Ausblick die Integration von kollaborativen Regeln in den Modellierungsprozess angesprochen. Auch bei der Integration dieser Regeln blieb im Gegensatz zu anderen Ansätzen die Prämisse gültig, dass verschiedene Teilnehmer einer Sitzung die CAD-Werkzeuge ihrer Wahl verwenden dürfen.

4. Integration von Geoinformationen – Geo Web Services

4.1. Geoinformationen im Planungsprozess von Infrastrukturbauwerken

Bei der Planung von Infrastrukturbauwerken spielt der geotechnische Kontext eine entscheidende Rolle: beim Entwurf eines U-Bahn-Tunnels etwa die unterirdischen Bauwerke und oberirdischen Gebäude, die Bodenschichtungen, das elektrische Leitungsnetz oder das Kanalnetz. Diesbezügliche Informationen werden heute häufig von sogenannten *Geo Web Services* zur Verfügung gestellt (Steuer et al. 2013, 2014). Einen *Geo Web Service* definiert Donaubauer dabei als eine Spezialisierung eines Geodienstes:

„Ein Geo Web Service ist ein Geodienst, der das World Wide Web als Distributed Computing Plattform verwendet.“

Dabei erklärt (Donaubauer 2004) einen Geodienst selbst folgendermaßen:

„Ein Geodienst stellt über Schnittstellen Funktionalität für die Nutzung von Geodaten bereit. Die Nutzung von Geodaten umfasst den Zugriff auf sowie die Erfassung, Manipulation, Transformation, Analyse und die Präsentation von Geodaten.“

Nicht zuletzt dank der 2007 ins Leben gerufenen europäischen Initiative INSPIRE¹⁸ (Infrastructure for SPatial InfoRmation in Europe) und des nationalen Geozugangsgesetzes¹⁹ wird die Menge der angebotenen Dienste und damit auch die Menge der verfügbaren Geodaten in den nächsten Jahren weiter steigen. Obwohl die den geotechnischen Kontext beschreibenden Informationen damit prinzipiell verfügbar sind, besteht das grundlegende Problem, dass deren Integration in die gängigen Planungs- und insbesondere CAD-Werkzeuge kaum nativ unterstützt wird (siehe auch Kapitel 4.4). Breunig et al. nennen drei weitere Probleme, die eine stringenteren Verwendung dieser Dienste erschweren:

- Unterschiedliche Informationen werden von verschiedenen und räumlich verteilten Servern zur Verfügung gestellt;
- verschiedene Dienste stellen zwar viele der gewünschten Daten zur Verfügung, doch differieren diese Daten bezüglich der verwendeten Skalen;
- die standardisierten Interfaces der Web Services sind im Umfeld eines kollaborativen Planungs- und Modellierungsprozesses nicht ausreichend (Breunig et al. 2011).

¹⁸ <http://inspire.ec.europa.eu/index.cfm> oder auch

<http://www.geoportal.de/DE/GDI-DE/INSPIRE/inspire.html?lang=de>

¹⁹ <http://www.bmub.bund.de/themen/umweltinformation-bildung/umweltinformation/umweltinformation-download/artikel/gesetz-ueber-den-zugang-zu-digitalen-geodaten-geodatenzugangsgesetz-geozg/>

Im Folgenden werden Ansätze präsentiert, die zur Lösung dieser Probleme beitragen. Dazu wird zum einen eine Containerstruktur präsentiert, welche die Integration der Dienste und somit der bereitgestellten Daten über die Plattform *Kollabserver* in die CAD-Werkzeuge unterstützt, zum anderen werden sogenannte *Web Service Context Documents* betrachtet. Diese wurden zur generischen Beschreibung dieser Dienste entworfen und bilden die Grundlage für den Entwurf der zu präsentierenden Containerstruktur. Im Speziellen wird gezeigt, dass diese Struktur eine einheitliche, dynamische und generische Integration dieser Dienste mit Hilfe der *Web Service Context Documents* zulässt. Die folgenden Abschnitte und die darin enthaltenen Erkenntnisse entstammen einer gemeinsamen Forschungstätigkeit zusammen mit einem zweiten Teilprojekt der DFG Forschergruppe *3DTracks* und hierbei vor allem einer engen Zusammenarbeit mit Herrn Dipl.-Inform. Horst Steuer, der sich dabei im Wesentlichen mit der Entwicklung der deskriptiven *Web Service Context Documents* befasst hat, während der Autor ihre Integration über die vorzustellende Containerstruktur in die Kollaborationsplattform untersuchte und dabei insbesondere die Integration der darin beschriebenen Web Services und der von diesen bereitgestellten Informationen in gängige CAD-Werkzeuge.

4.2. Geo Web Services – Web Map, Web Feature und Web Processing Service

Prinzipiell gibt es eine Reihe von verschiedenen Typen von *Geo Web Services*, von denen wir im Folgenden drei Typen herausgreifen werden, da sie im Rahmen dieser Forschung von besonderer Bedeutung waren: nämlich *Web Map Services*²⁰ (WMS), *Web Feature Services*²¹ (WFS) und *Web Processing Services*²² (WPS). Diese *Geo Web Services* wurden durch das OGC²³ (Open Geospatial Consortium) und/oder auch die ISO²⁴ (International Standards Organization) propagiert und standardisiert. *OGC Web Service Interfaces* bieten so Möglichkeiten eines standardisierten Zugriffs auf die bereitgestellten Daten, wobei zusätzlich für den Benutzer Möglichkeiten geschaffen wurden, durch räumliche, logische oder zeitliche Filter genau auf die von ihm gewünschten Features eines *Geo Web Services* zuzugreifen (Steuer et al. 2014).

Web Map Services (WMS)

Web Map Services, die sowohl von der OGC (erstmalig 1999) als auch durch die ISO (gemäß ISO 19128:2005) standardisiert wurden, bieten einen vereinheitlichten Zugriff auf Kartenmaterial, das in der Form von Bilddateien (etwa .png, .gif oder .jpeg) vorliegen muss. In einem solchen Dienst spezifiziert der Benutzer den Bereich, den er visualisieren möchte, und erhält

²⁰ <http://www.opengeospatial.org/standards/wms>

²¹ <http://www.opengeospatial.org/standards/wfs>

²² <http://www.opengeospatial.org/standards/wps>

²³ <http://www.opengeospatial.org/>

²⁴ <http://www.iso.org/iso/home.html>

einen entsprechenden Kartenausschnitt im gewünschten Format zurück. Unter Verwendung eines Transparenzgrades sind dabei auch Überlagerungen verschiedener Karten möglich (Reed 2006). Gemäß diesen Anforderungen ist eine Anfrage an einen *Web Map Service* immer rein lesender Natur.

Web Feature Services (WFS)

Nach der Norm ISO 19109:2006 wird im Kontext von Web Services unter einem Feature ein „*real world phenomenon*“²⁵ verstanden (Hampe und Intas 2006). Eine im Deutschen übliche Entsprechung für ein Feature in Kontext von *Geo Web Services* ist der Begriff Geobjekt, der konkret ein Sachobjekt beschreibt, das im Rahmen eines *Web Feature Services*²⁶ in Vektordaten gespeichert sein muss. So kann im gegebenen Kontext ein Feature beispielsweise ein See, ein Fluss, ein Gebäude oder ein ähnliches Objekt sein. Ein *Web Feature Service* (ISO 19142:2010) liefert nun über eine genau definierte Schnittstelle Zugriff auf die Beschreibungen derartiger Geobjekte. Das Rückgabeformat muss dabei immer zumindest die Sprache GML²⁷ (Geography Markup Language) unterstützen (Reed 2006, Pospesch 2013). GML ist selbst eine durch die OGC und ISO (ISO 1936:2007) standardisierte XML-basierte Beschreibungssprache zur Darstellung, zum Austausch und zur Speicherung von Geoinformationen. Neben den reinen geometrischen Informationen kann ein solcher Dienst eine beliebige Menge nicht räumlicher Informationen bereitstellen, etwa den Besitzer oder die Nutzungsart eines bestimmten Gebäudes. *Web Feature Services* sind zunächst nur lesender Natur, sogenannte *Transaktionale Web Feature Services* (T-WFS) erlauben es jedoch zusätzlich, Modifikationen an den zugrunde liegenden Daten vorzunehmen.

Web Processing Services (WPS)

Ein *Web Processing Service*²⁸ hingegen stellt primär nicht Geodaten zur Verfügung, sondern Schnittstellen zu deren Analyse. In der Definition durch die OGC liefert diese Analyse in vielen Fällen erst die Grundlage für eine effektive Nutzung der Daten. Prinzipiell kann dabei jede beliebige Datenquelle analysiert werden, insbesondere sowohl Vektor- als auch Rasterdaten (Steuer et al. 2014). Eine typische Operation ist die Verschneidung oder Überlagerung von Polygonen, allgemein von verschiedenen Geobjekten. Die Ergebnisdaten selbst sollen dabei wiederum im GML-Format an den Benutzer geliefert werden.

²⁵ Insbesondere ist der im Kontext der *Geo Web Services* verwendete Begriff eines Features nicht mit dem Begriff Feature im Bereich der prozeduralen Modellierung zu verwechseln (siehe Kapitel 3.3).

²⁶ <http://www.weichand.de/2011/11/30/grundlagen-web-feature-service-wfs-2-0/>

²⁷ <http://www.opengeospatial.org/standards/gml>

²⁸ OpenGIS Web Processing Service, Version 1.0.0 OGC Document 05-007r7. <http://www.opengeospatial.org/standards/wps>

4.3. Kollaborative Nutzung von Geo Web Services

Ein Ziel unserer Forschung soll es hier sein, *Geo Web Services* und die von diesen bereitgestellten Daten in den kollaborativen Planungsprozess zu integrieren. Dabei wurde der Fokus auf die Typen WMS, WFS und WPS gelegt, die nach Sikder die Schüsselemente einer kollaborativen Stadtplanung bilden (Sikder 2009). In diesem Abschnitt werden einige Beispiele für eine kollaborative Nutzung dieser Dienste gegeben, dabei aber nicht in die Tiefe vordringen. Die vorgestellten Arbeiten sollen lediglich einen ersten Anstoß für solche Problemstellungen geben und bieten einen interessanten Ausgangspunkt für eine intensivere Beschäftigung mit dieser Thematik. Insbesondere zeigen ausführliche Literaturstudien bezüglich der kollaborativen Nutzung von *Geo Web Services*, dass ihre Verwendung in einem kollaborativen Bauplanungsprozess noch kaum untersucht wurde, während es eine Reihe von Anwendungen in anderen Domänen gibt, in denen sie bereits heute intensiv und erfolgreich zur Zusammenarbeit eingesetzt werden.

Maiyo et al. betonen die Bedeutung der kollaborativen Nutzung von *Geo Web Services* bei einem modernen Management von Katastrophenszenarien (Maiyo et al. 2010). Dazu werden Möglichkeiten präsentiert, wie man mit ihrer Hilfe dynamisch Geoinformationen in zur Bewältigung dieser Szenarien benötigtes Kartenmaterial integrieren kann. Insbesondere können so verschiedene an der Aufgabe beteiligte Organisationen aktiv aktuelle Informationen einspeisen, mit anderen Helfern teilen und genauso von anderen Organisationen und Teams beziehen.

Ebenfalls mit der Bewältigung von Katastrophenszenarien unter dem kollaborativen Einsatz von *Geo Web Services* setzen sich Neis et al. auseinander (Neis et al. 2010). Mit Hilfe von *Transaktionalen Web Feature Services* sollen Möglichkeiten aufgezeigt werden, wie man Straßenkartenmaterial an die aktuellen Gegebenheiten des Szenarios anpassen kann. Somit können einzelne Helfer anderen die Zu- oder Unzugänglichkeit bestimmter Punkte dynamisch mitteilen.

Auch Cai befasst sich mit dem kollaborativen Einsatz von *Geo Web Services* beim Umgang mit Krisenszenarien (Cai 2005), während Klimke und Döllner einen kollaborativen Einsatz von 3D-Stadtmodellen im Bereich der (Stadt-)Planung präsentieren (Klimke und Döllner 2010). Zur Kommunikationsunterstützung, die eine klassische CSCW-Disziplin darstellt, präsentieren sie einen Ansatz zur dynamischen Integration von Anmerkungen (englisch *Annotations*) in diese Stadtmodelle. Aus technischer Sicht verwenden sie für diese Anmerkungen *Transaktionale Web Feature Services*. Diese Anmerkungen sind nicht nur rein textueller Natur, sondern dürfen auch ganz explizit (räumliche) Skizzen als präferierte Informationsdarstellung enthalten.

Eine kollaborative Anwendung von *Geo Web Services* im Bereich der Artenvielfaltsforschung wird von Fook et al. präsentiert (Fook et al. 2009). In ihrer Forschung entwerfen sie eine Ar-

chitektur zum Austausch von Ergebnissen der Modellierung von Artenverteilungsanalysen, die *Geo Web Services* als Grundlage des Datenaustausches verwenden.

Sikder liefert einen guten Überblick über kollaborative GIS-Projekte und einen Ansatz zur kollaborativen Modellierung im Bereich der Umweltplanung (Sikder 2009). Das in seiner Forschung entstandene Werkzeug GEO-ELCA (Exploratory Land Use Change Analysis) stellt eine kollaborative Plattform bereit, die verschiedene Experten dabei unterstützen soll, gemeinsam die umwelttechnischen Folgen der Änderungen in der Bodennutzung abzuschätzen. Im Jahr 2014 präsentierten Batita et al. das sogenannte WikiGIS (Batita et al. 2014). Dieses soll die typischen Content Managementsystem Eigenschaften eines Wikis auf den GIS-Bereich durch interaktiv und kollaborativ erstellte Karten übertragen, wobei auf die Nachvollziehbarkeit der Aktionen einzelner Benutzer im gemeinsamen Prozess besonderer Wert gelegt wird. Das Kartenmaterial wird dabei konkret durch *Web Map Services* bereitgestellt.

Im Jahr 2003 wurde von der OGC der sogenannte *Web Map Context Document*²⁹ als Standard eingeführt. Dieser wurde als Grundlage für die kollaborative Arbeit mit raumbezogenen Daten geschaffen, die von verschiedenen Web Services stammen (Steuer et al. 2014). Als Web Services sind hier zunächst ausschließlich *Web Map Services* zugelassen. Ein *Web Map Context Document* erlaubt es dann, in einem Dokument eindeutig den Kontext einer *Web Map Service Client Application* zu beschreiben, so dass ihr Zustand (möglicherweise von anderen Clients) wieder eindeutig hergestellt werden kann (Reed 2006). Nach zehnjähriger Entwicklungszeit veröffentlichte das OGC im Jahre 2014 die neueste Version eines *Context Documents* als Standard, der alle drei Typen WMS, WFS und WPS umfasst. Dieser Standard bildet auch die Grundlage für die von Steuer et al. entwickelten *Web Service Context Documents*, die aber über die Möglichkeiten der von der OGC standardisierten Dokumente hinausgehen (Steuer et al. 2014).

4.4. Integration von Geo Web Services in CAD-Werkzeuge

Im gegebenen Kontext ist hinsichtlich der Kooperation der verschiedenen Planer insbesondere die Möglichkeit der Integration von *Geo Web Services*, allgemeiner von GIS-Technologien und Werkzeugen in CAD-Werkzeuge von Bedeutung. Peachavanish et al. betonen, dass die beiden Disziplinen GIS und CAD zwar in einigen Bereichen gemeinsam genutzt und gemeinsame Daten sowie Operationen verwendet werden, ihre beiden Entwicklungen aber weitgehend voneinander unabhängig erfolgten und auch weiterhin erfolgen (Peachavanish et al. 2005). Wegen der Unabhängigkeit dieser beiden Bereiche erklären sie, dass ein Benutzer in beiden Gebieten gleichzeitig ein Experte sein muss, sobald es bei der Anwendung spezifischer Werkzeuge zu inhaltlichen Überschneidungen kommt. Einen Brückenschlag versuchen sie zu erreichen, indem sie eine gemeinsame Ontologie für beide Bereiche entwerfen und eine Ab-

²⁹ Web Map Context Documents, Version 1.1.0 OGC Document 05-005 <http://www.opengeospatial.org/standards/wmc>

fragen-basierte Methode entwickeln, die vor allem auf semantischer Ebene eine Interoperabilität zwischen beiden Gebieten in Grundzügen ermöglicht. Auch Karimi und Akinci erkennen die Notwendigkeit der Integration der beiden Gebiete GIS und CAD (Karimi und Akinci 2010). Um verschiedene Ansätze und Möglichkeiten, Entwicklungen und Fortschritte in dieser Integration detaillierter studieren zu können, wird hier explizit auf ihr grundlegendes Buch „CAD and GIS Integration“ verwiesen (Karimi und Akinci 2010). Insbesondere kann dieses Werk aufgrund der zahlreichen und interessanten Referenzen sehr gut als Ausgangspunkt für eine weitergehende Literaturstudie genutzt werden. Ebenso hilfreich ist das von Zlatanova und Prospero herausgegebene Werk „Large-Scale 3D Data Integration“ (Zlatanova und Prospero 2006), das sich ausführlich mit der gegebenen Thematik beschäftigt.

4.5. Web Service Context Documents – eine generische Beschreibung von Geo Web Services

Zur generischen Beschreibung von *Geo Web Services* greifen Steuer et al. sogenannte *Web Service Context Documents* des OGC auf (Steuer et al. 2013, 2014), die im gegebenen Kontext auf der Kollaborationsplattform hinterlegt werden. Den Benutzern stellt die Plattform dann eine Schnittstelle zur Verfügung, mit deren Hilfe sie einerseits vorhandene Dokumente abrufen können, um sich dann konkret mit Hilfe der darin enthaltenen Informationen mit einem *Geo Web Service* verbinden zu können; andererseits ermöglicht es diese Schnittstelle den Benutzern auch, neue Dokumente hinzuzufügen und diese somit anderen Sitzungsteilnehmern (dynamisch) zugänglich zu machen.

Um *Geo Web Services* zu nutzen, müssen durch die Sitzungsteilnehmer konkrete Anforderungen an die Server gestellt werden, die diese Dienste bereitstellen. Einzelne dieser Dienste stellen unter Umständen sehr große Datenmengen bereit: So bietet etwa ein WMS möglicherweise gleichzeitig Luftbilder, Entwicklungspläne oder auch hybride Bilddaten an, die typischerweise über sogenannte Layer zur Verfügung gestellt werden. Auch WFS kapseln verschiedene Daten durch Layer, etwa über Gebäude, Wände und Dächer, sowie über Geländemodelle der Umgebungen.

Neben diesen reinen Geodaten stellen Web Services zu den enthaltenen Layern und Geoobjekten Metadaten bereit. Diese Informationen werden dann bei konkreten Anfragen genutzt, um durch spezielle Filter nur die relevanten Daten, genauer nur entsprechende räumliche Ausschnitte (möglicherweise in einem bestimmten Detaillierungsgrad – Level of Detail) anzufordern und herunterzuladen. Das Ziel der genannten *Web Service Context Documents* ist es nun, eine Menge solcher Anfragen in einem speziell dafür zugeschnittenen Dokument zu kapseln und zu bündeln. Dies soll dabei so geschehen, dass alle für die Analyse eines bestimmten Problems relevanten Informationen mit Hilfe eines solchen Dokuments beschrieben werden. Spezifische Vorteile und Erweiterungen dieser *Web Service Context Documents* gegenüber den vom OGC eingeführten Web Map Context Documents findet man etwa wiederum bei

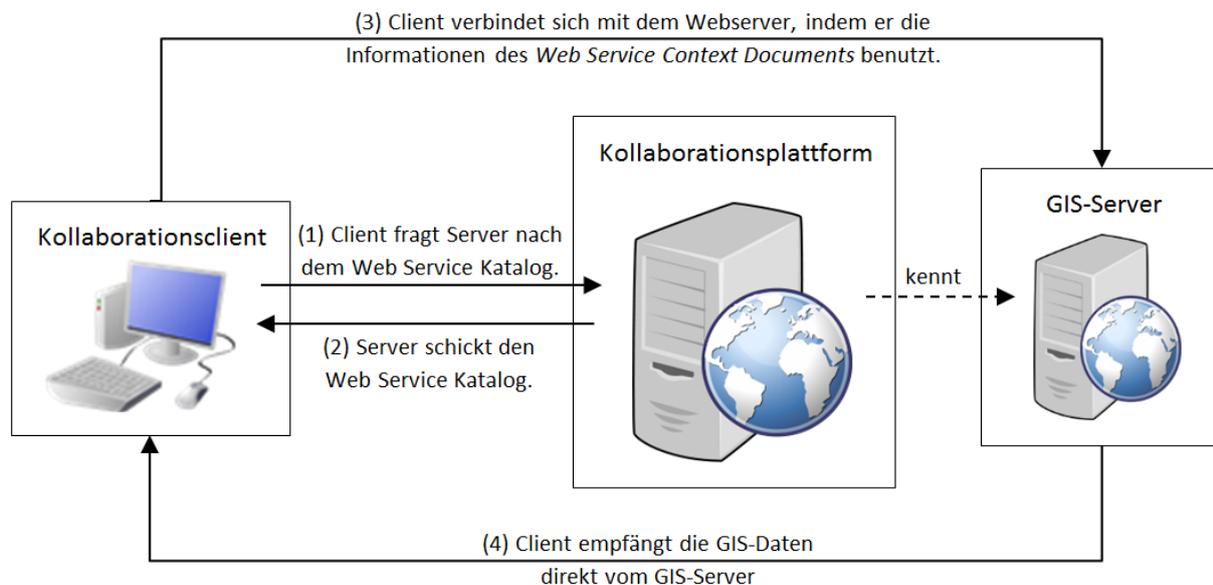


Abbildung 29: Prinzipieller Ablauf bei der Auswahl eines GIS Servers

(Steuer et al. 2013, 2014). Zumindest genannt werden sollen hier die Einführung und die Integration von benutzergruppen-spezifischen Regeln, Anmerkungen (Annotations), Kamera-Perspektiven und die intensive Integration von *Web Processing Services*.

4.6. Eine Containerstruktur zur Integration von Geo Web Services

Die Grundlage für die Integration von *Geo Web Services* in den im Rahmen dieser Arbeit vorgestellten kollaborativen Planungsprozess bildet ein Konzept für eine öffentliche Schnittstelle, welche die Kollaborationsplattform den verschiedenen Teilnehmern einer kollaborativen Sitzung zur Verfügung stellt. Zunächst ermöglicht es diese Schnittstelle den Benutzern, die am Server hinterlegten *Web Service Context Documents* abzufragen (Abbildung 29 – Schritt 1) und herunterzuladen (Abbildung 29 – Schritt 2). So gelangen sie insbesondere an die Informationen, die es ihnen dann erlauben, sich mit einem konkreten GIS-Server zu verbinden (Abbildung 29 – Schritt 3) und dann in einem weiteren Schritt die konkret gewünschten Geodaten abzurufen (Abbildung 29 – Schritt 4). Entscheidend in diesem Ansatz ist, dass der Benutzer von der Kollaborationsplattform nur Metadaten anfordert, welche meist nur einen Bruchteil der Größe der wirklichen Daten ausmachen. Die realen Daten selbst werden dann mit Hilfe dieser Informationen direkt vom GIS-Server ohne eine weitere Beteiligung der Kollaborationsplattform angefragt. So wird insbesondere vermieden, dass die Plattform innerhalb der gegebenen Client Server Architektur zu einem Flaschenhals wird, wie wir es bereits in Kapitel 3.2 angedeutet hatten.

Grundlegend beim Entwurf dieser Schnittstelle waren drei Gesichtspunkte mit dem Ziel, die Integration der Geodaten zu erleichtern. Die Schnittstelle sollte einen einheitlichen, dynamischen und einen generischen Zugriff ermöglichen:

- Unter einheitlich verstehen wir hier die Eigenschaft, dass das Laden eines Dienstes, genauer seiner Daten, für alle Typen von Diensten auf die gleiche Art und Weise durchgeführt werden kann. Insbesondere soll dies für die Skalierung und Visualisierung der bereitgestellten Daten gelten.
- Das Laden der Daten sollte zu jeder Zeit einer kollaborativen Sitzung, also sowohl zu Beginn als auch während einer Sitzung möglich sein, so dass man von einer dynamischen Integration sprechen kann.
- Da noch nicht alle Typen von Diensten zu Beginn einer kollaborativen Sitzung bekannt sind, sollte es auch prinzipiell möglich sein, zur Laufzeit neue Typen von *Geo Web Services* zu integrieren, solange diese nur bestimmte Kriterien gegenüber ihrem Inhalt erfüllen, weshalb wir von einer generischen Integration sprechen können.

Diese Anforderungen erfüllt die neu entwickelte Containerstruktur, die wir jetzt vorstellen. Grundlegend für diese Struktur sind zwei Datencontainertypen: Sogenannte *Parts* kapseln konkrete (zu visualisierende) Daten, sogenannte *Assemblies* hingegen erlauben es, *Parts* oder *Assemblies* zu aggregieren. Mit Hilfe dieser Container wird dann die Integration der Daten in

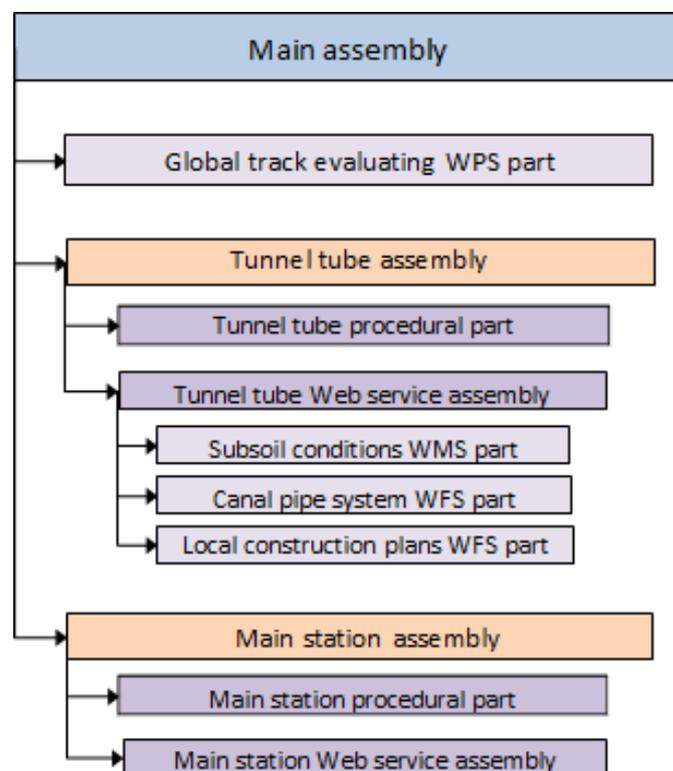


Abbildung 30: Die Komposition von verschiedenen *Parts* und *Assemblies* führt zu einer baumartigen Struktur

die einzelnen Planungswerkzeuge erreicht. Die Komposition verschiedener *Parts* in verschiedene *Assemblies* und ihre Komposition wiederum in andere *Assemblies* ergeben eine Baumstruktur, wie sie in Abbildung 30 dargestellt ist. Aus Sicht eines Softwarearchitekten ergibt sich eine solche Baumstruktur ähnlich der im sogenannten Composite-Patterns beschriebenen Struktur, die nach (Gamma et al. 1995) gerade das Ziel verfolgt, „to ,compose‘ objects into tree structures to represent part-whole hierarchies. Implementing the composite pattern lets clients treat individual objects and compositions uniformly.“ Für uns spielt dabei gerade die einheitliche Behandlung von Objekten und Komposition, also von *Parts* bzw. *Assemblies* die entscheidende Rolle.

So greifen wir die Grundidee dieses Entwurfsmusters allgemein auf und verstehen unter *Parts* grundlegende Einheiten zum Kapseln von zu integrierenden Datenquellen, die dabei Funktionalitäten zum Laden, Visualisieren und Skalieren von Geodaten bereitstellen müssen. Unter *Assemblies* verstehen wir Container, die ebenfalls Funktionalitäten zum Laden, Visualisieren und Skalieren anbieten müssen, aber zusätzlich auch Möglichkeiten zum Aggregieren von Datencontainern bereitstellen, also zum Aggregieren von *Parts* und auch von *Assemblies* selbst. *Parts* stellen in dieser Konstruktion die Möglichkeiten zum Laden, Visualisieren und Skalieren tatsächlich zur Verfügung, während *Assemblies* die konkreten Funktionalitäten ihrer enthaltenen Kindelemente nützen, d.h. *Assemblies* delegieren diese Aufgaben und zwar gerade solange, bis sie auf *Part*-Ebene konkret ausgeführt werden können.

Bei der konkreten Anwendung in den einzelnen Modellierungswerkzeugen ordnen wir dann jedem Container vom Typ *Part* eine sogenannte Teile-Umgebung zu, während ein Container vom Typ *Assembly* durch eine Baugruppen-Umgebung repräsentiert wird. Das Teile- und

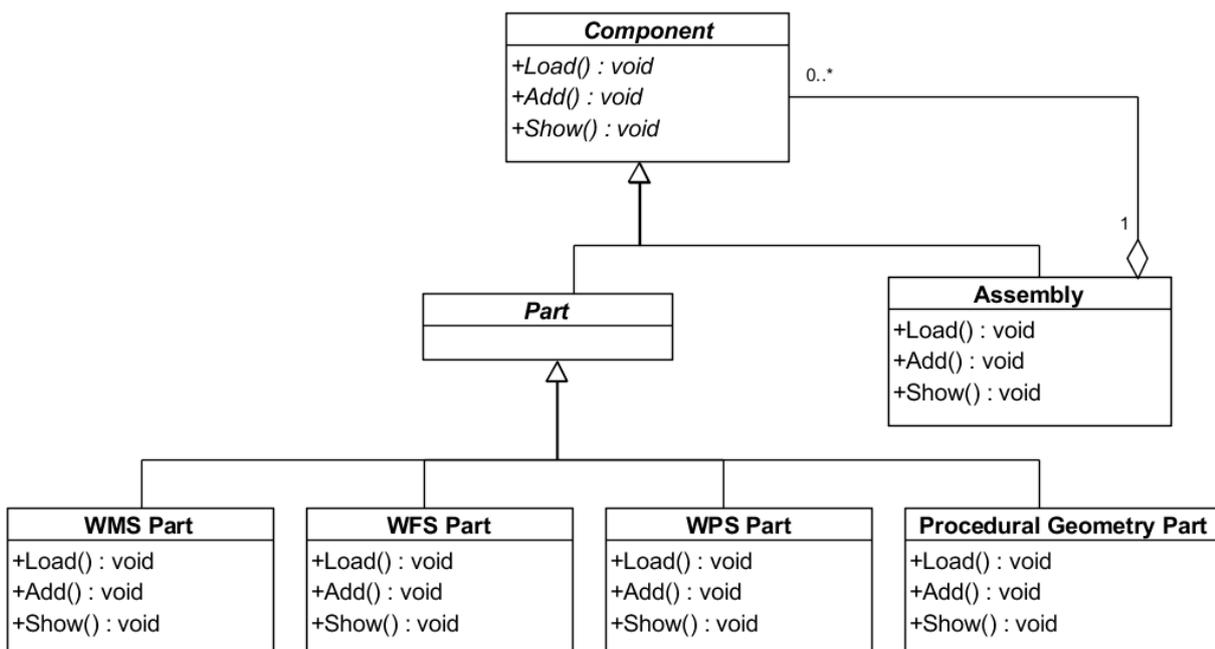


Abbildung 31: Klassendiagramm zur Umsetzung der vorgeschlagenen Containerstruktur

Baugruppenkonzept wird in heutigen CAD-Systemen allgemein unterstützt und bildet so eine Grundlage für unser Integrationskonzept. In der konkreten Umsetzung dieses Konzepts ergibt sich dann auf natürliche Art und Weise, dass selbstverständlich für jedes einzelne Modellierungswerkzeug eine entsprechende softwaretechnische Implementierung zumindest für das Visualisieren der Daten vorliegen muss.

Wir dehnen nun die Datenkapselung auf die zuvor besprochenen prozeduralen Modelle aus, um das Konzept noch weiter zu generalisieren. So kapseln wir die bereits oben besprochenen prozeduralen Modelle (siehe Kapitel 3.3) in sogenannte *Procedural Geometry Part Container*, während wir die Daten von WMS, WFS und WPS Services dementsprechend in WMS, WFS und WPS *Parts* kapseln. Dieser Schritt ermöglicht nun eine einheitliche Behandlung von prozeduralen Modellen und *Geo Web Services* innerhalb der CAD-Werkzeuge hinsichtlich der grundlegenden Funktionalitäten des Ladens, Visualisierens und Skalierens. So wird eine Integration und Zusammenführung dieser verschiedenen Informationsquellen auf theoretischer und auch konkreter Ebene ermöglicht.

Wie oben erklärt, beschreibt ein *Web Service Context Document* einen oder auch mehrere Web Services, weshalb wir ein solches Dokument als *Assembly* kapseln. Neben den Web Services darf ein solches *Assembly* nun auch eine beliebige Anzahl von prozeduralen Geometrie *Parts* enthalten; dies ist gerade die Grundlage für eine gemeinsame Visualisierung der prozeduralen Geometriedaten und der durch die Web Services bereitgestellten Geodaten. Die Abbildung einer entsprechenden Klassenstruktur als Grundlage für die prototypische Implementierung des vorgestellten Konzepts ist in Abbildung 31 gegeben. Die präsentierte Strategie hat zumindest zwei besondere Stärken:

- Die Kollaborationsplattform bildet einen ausgezeichneten und zentralen Zugriffspunkt für alle Teilnehmer einer kollaborativen Sitzung und stellt dabei alle Informationen sowohl für die prozeduralen Geometriemodelle als auch für Web Services zur Verfügung. Somit nimmt er dem Benutzer insbesondere die Verwaltung einer Vielzahl von Adressen und Orten verschiedener Datenquellen ab, wobei die Aktualität und Gültigkeit dieser Daten zentral verwaltet und gewartet wird. So wird eine Integration von Daten und Arbeitsprozessen mit Hilfe einer zentralen Konfigurationskomponente erreicht, während gleichzeitig die Vorzüge einer verteilten Architektur genutzt werden können.

Mit dieser Strategie ist insbesondere das erste in Kapitel 4.1 angesprochene Problem gelöst, nämlich dass unterschiedliche Informationen von verschiedenen und räumlich verteilten Servern zur Verfügung gestellt werden.

- Durch die eingeführte Indirektion zwischen Datenquellen und Kollaborationsplattform ist diese nicht dafür verantwortlich, große Datenmengen an die einzelnen Sitzungssteilnehmer zu versenden. So wird vermieden, dass die Plattform durch das Laden von großen Datenmengen zu einem Flaschenhals innerhalb der kollaborativen Sitzung wird. Somit ist sie in

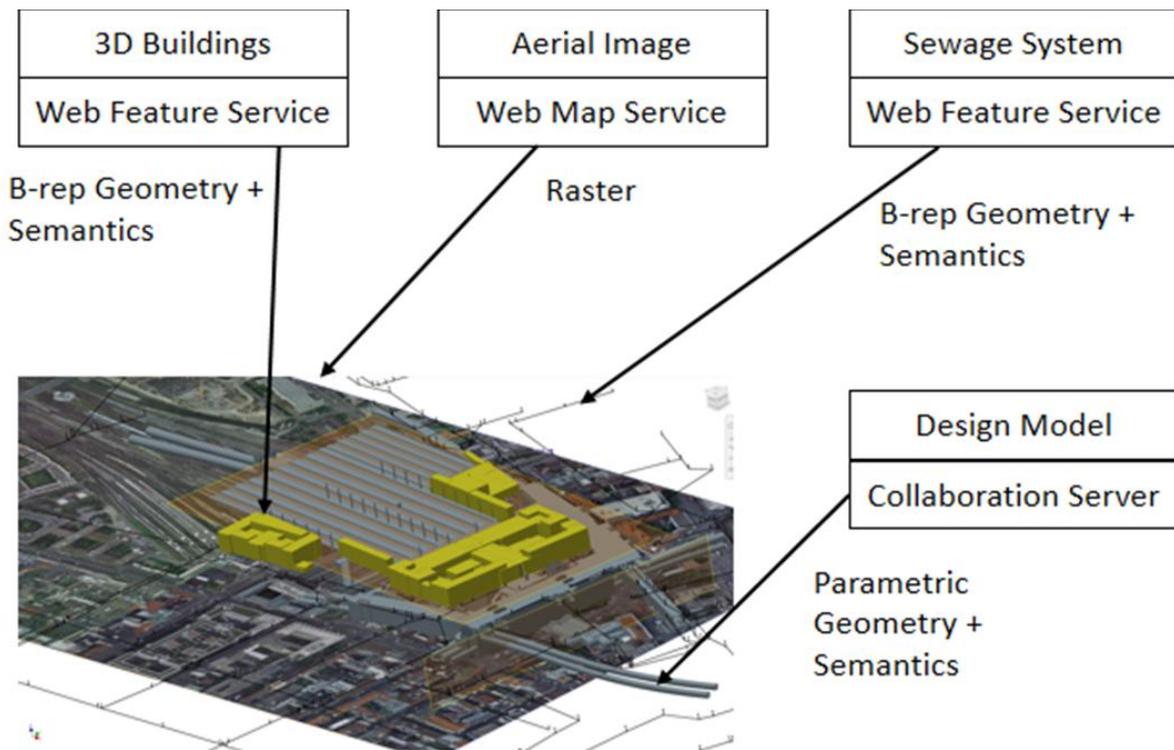


Abbildung 32: Überlagerung eines prozeduralen Trassenmodells und mehrerer Web Services nach (Steuer et al. 2015)

erster Linie für die Administration des kollaborativen Arbeitsprozesses verantwortlich und wird nicht zum Verschicken großer Datenmengen missbraucht.

Die vorgestellte Containerstruktur erlaubt es, verschiedene in einem *Assembly* enthaltene *Parts* prinzipiell in einem Schritt zu laden und als verschiedene Teile in einer gemeinsamen Baugruppen-Umgebung simultan zu visualisieren. Exemplarisch ist dies in Abbildung 32 dargestellt für ein *Assembly*, das ein den Münchner Hauptbahnhof repräsentierendes prozedurales Modell und zusätzlich ein Web Service *Assembly* enthält (Steuer et al. 2014). Das Web Service *Assembly* selbst umfasst hier drei Web Services: einen WFS zur Darstellung von Gebäuden in City-GML, einen WMS Service, der ein Luftbild der Umgebung bereitstellt, und als drittes einen WFS, der die Informationen zum Kanalrohrsystem bereitstellt.

Das hier dargestellte Beispiel liefert auch direkt einen weiteren besonders wichtigen Anwendungsfall bei der Planung von U-Bahn-Tunneln: Die Daten der äußeren Tunnelhülle (LoD 2 im prozeduralen Modell) und die Daten etwa bezüglich des Kanalrohrsystems (gemäß dem vorigen Beispiel aus dem *WFS*) oder der unterirdischen Bauwerke werden als Eingangsdaten für einen *Web Processing Service* verwendet. Dieser prüft diese Daten dann auf (mögliche) Kollisionen zwischen geplantem Tunnel und existierendem Rohrnetz oder unterirdischen Bauwerken. Als Beispiel ist dies in Abbildung 33 dargestellt: Hier wird eine Kollision, die in

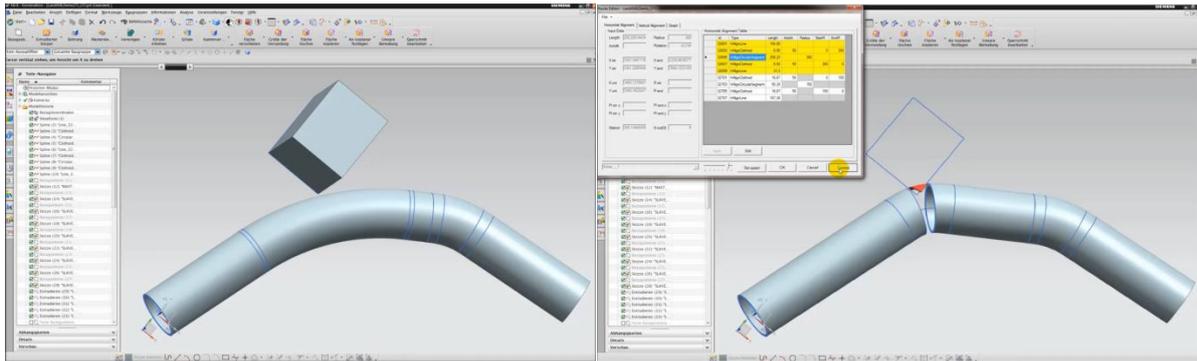


Abbildung 33: Eine Veränderung der Trasse führt zu einer Kollision mit einem unterirdischen Bauwerk und wird durch einen WPS entdeckt.

Folge einer Trassenveränderung zwischen dem Tunnel und einem unterirdischen Bauwerk entsteht, automatisch durch einen WPS entdeckt.

Bei der simultanen Visualisierung von verschiedenen *Parts* in einer einzelnen Baugruppen-Umgebung gibt es zumindest zwei grundlegende Probleme:

- Verschiedene *Parts* kapseln verschiedene Daten und Datenformate,
- die gekapselten Daten beziehen sich auf verschiedene Koordinatensysteme und unterschiedliche Skalen (siehe auch Breunig et al. in der Einführung dieses Kapitels).

Das erste Problem wird durch den Aufbau der oben beschriebenen Containerstruktur gelöst: Denn jeder einzelne *Part* kennt natürlich seine Daten und ihr Format, so dass er entsprechende Funktionalitäten zu ihrer Visualisierung bereitstellen kann, für die er in der gegebenen Konstruktion auch gerade selbst verantwortlich ist.

Zur Lösung des zweiten Problems aus Kapitel 4.1, nämlich dass unterschiedliche Services auch verschiedene Skalen zugrunde legen, wird folgendermaßen vorgegangen: Ein *Hauptassembly*, das alle weiteren Datencontainer enthält, definiert ein im gegebenen Kontext anwendungsweit gültiges Koordinaten- und Einheitensystem. Um die enthaltenen Container, also sowohl *Assemblies* wie auch *Parts* korrekt zu skalieren, werden ihre Funktionalitäten zur Skalierung vor der Visualisierung aufgerufen, wobei das allgemein gültige Koordinaten- und Einheitensystem des *Hauptassemblys* als Referenz verwendet wird. Während der darauf folgenden Visualisierung können dann die untergeordneten Container die Koordinaten ihrer Datenobjekte dementsprechend anpassen, so dass insgesamt insbesondere das zweite in Kapitel 4.1 angesprochene Problem gelöst ist.

Im Folgenden werden wir das Vorgehen noch einmal anhand eines konkreten Beispiels verdeutlichen. Wir betrachten den Fall eines einfachen *Web Service Assemblys*, das ausschließlich einen WMS Service bereitstellt.

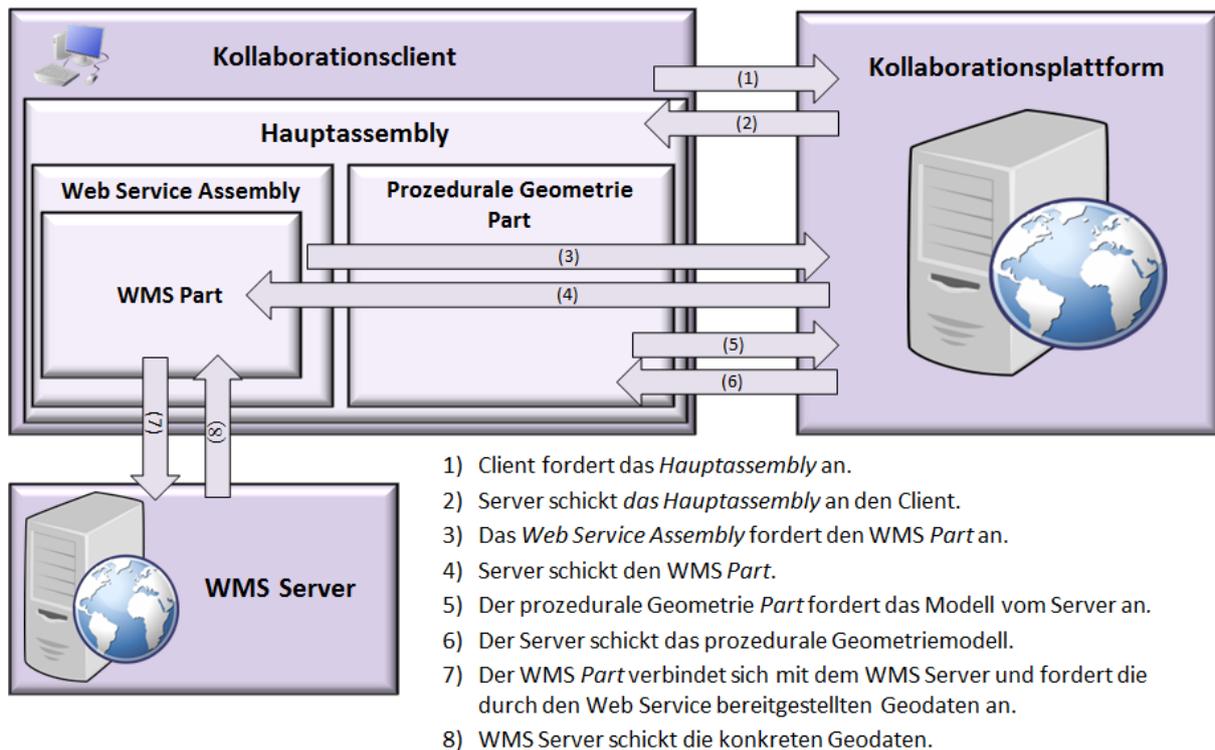


Abbildung 34: Beispiel für die Überlagerung eines prozeduralen Geometriemodells mit einem WMS Service

4.7. Ein Beispiel für die Integration eines Web Map Service

Im Folgenden wollen wir abschließend ein konkretes Beispiel durchspielen; die dabei angegebenen Schritte beziehen sich auf Abbildung 34; das konkrete Ergebnis ist in Abbildung 35 dargestellt. Zunächst hat der Benutzer aus der Liste der auf der Kollaborationsplattform hinterlegten *Web Service Context Documents* ein bestimmtes *Hauptassembly* ausgewählt, das sowohl einen prozeduralen Geometrie *Part* als auch ein weiteres *Assembly*, im Folgenden als *Web Service Assembly* bezeichnet, enthält (1).

Nach dessen Erhalt (2) interpretiert er dieses und fordert das *Web Service Assembly* und alle darin enthaltenen Elemente vom Server an. In unserem Fall ist das gerade genau ein *WMS Part* (3), welcher konkret eine hybride Darstellung des Entwicklungsplans in der Nähe des Münchner Hauptbahnhofs enthält (Abbildung 35). Anschließend ruft es rekursiv die Load-Funktionalität all seiner enthaltenden Elemente auf, hier genau die des enthaltenen *WMS Parts* (4).

Im nächsten Schritt (5) fordert der prozedurale Geometrie *Part* das prozedurale Geometriemodell vom Server an, der diese unverzüglich zurückschickt (6). Dieser Part enthält das Trassenmodell der dortigen ersten S-Bahn Stammstrecke und der U-Bahnlinien U1/2 und U4/5 (Abbildung 35). Die zur Anforderung des WMS Service benötigten Metainformationen werden gerade dem *Web Service Context Document* entnommen, so dass der *WMS Part* sich mit dem entsprechenden GIS-Server verbinden und die echten Geodaten herunterladen kann (7). Zu guter Letzt benachrichtigt der *WMS Part* das übergeordnete *Assembly*, dass der Ladevor-

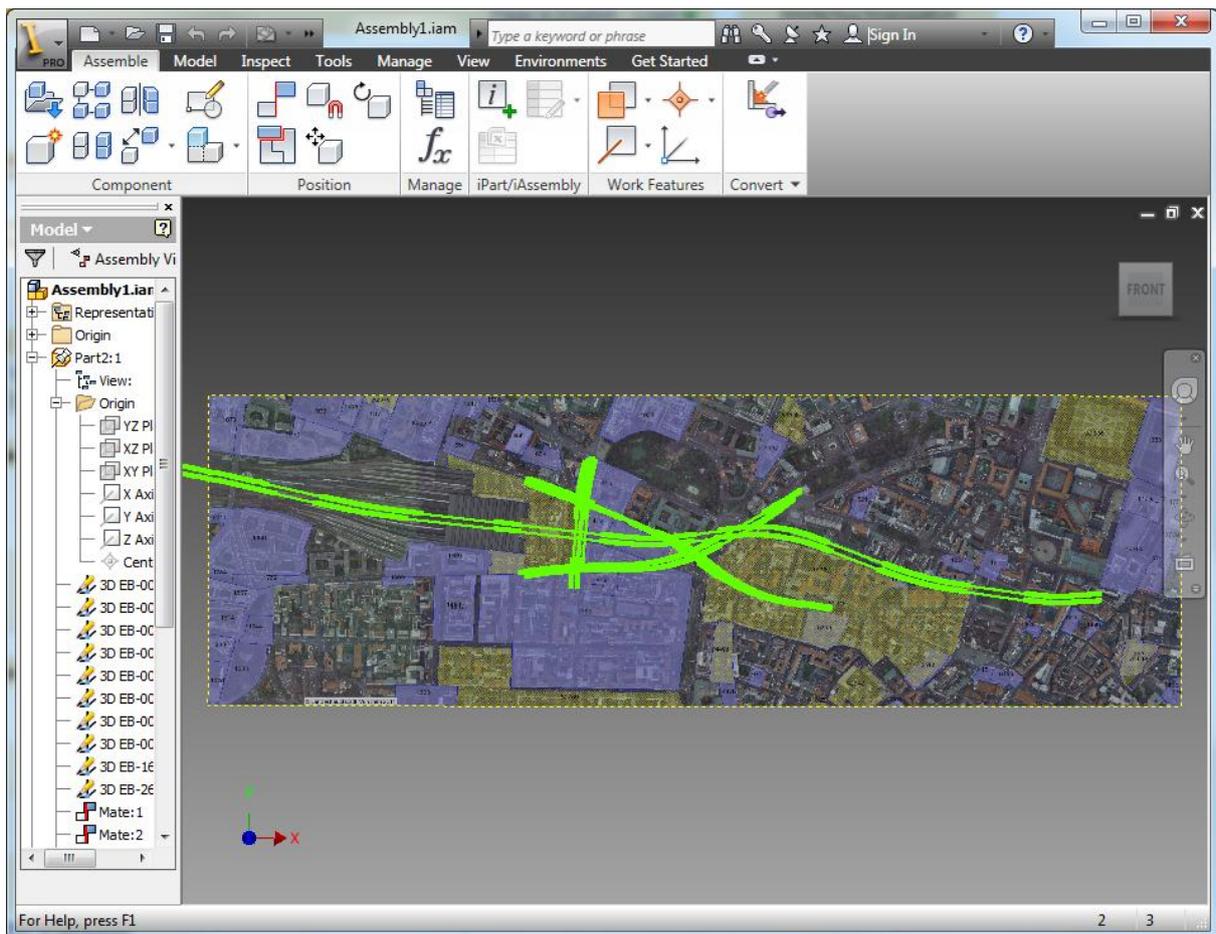


Abbildung 35: Überlagerung eines prozeduralen Trassenmodells von U- und S-Bahn in der Umgebung des Münchner Hauptbahnhofs und ein WMS, der einen hybriden Entwicklungsplan der dortigen Umgebung zeigt. Verwendet wurde hier das CAD-System Autodesk Inventor.

gang der Geodaten abgeschlossen ist, so dass das Web Service *Assembly* selbst die Visualisierungsfunktionalität des WMS *Parts* aufrufen kann (8), wodurch sich der WMS *Part* anschließend selbstständig in einer neuen Teile-Umgebung darstellen kann.

4.8. Zusammenfassung

Für die Planung von Infrastrukturbauwerken liefern *Geo Web Services* eine Vielzahl planungsrelevanter Geoinformationen. Die historisch bedingt getrennten Entwicklungen von GIS- und CAD-Werkzeugen erfordern neue Konzepte, um diese Informationen bei der Modellierung nutzen zu können. *Web Service Context Documents* können vorteilhaft zur generischen Beschreibung von *Geo Web Services* verwendet werden. Diese Dokumente können wir insbesondere im Rahmen des von der Kollaborationsplattform unterstützten Planungsprozesses nutzen.

Eine speziell auf diese zugeschnittene Containerstruktur wurde vorgestellt, mit deren Hilfe im Gegensatz zu bisherigen Ansätzen eine einheitliche, dynamische und generische Integration

von Web Services in den Planungsprozess ermöglicht werden können. Diese Strategie zur Integration von Web Services ermöglicht es verschiedenen Planern unterschiedlicher Fachdisziplinen, insbesondere den GIS-Experten und Modellierungsexperten, in einer neuartigen Weise aktiv zusammenzuarbeiten, so dass insbesondere die zwei Fachdisziplinen GIS und CAD näher zueinander geführt werden.

5. Das Alignmentmodell – Grundlage der Infrastrukturplanung

Bei der Planung von Infrastrukturbauwerken wie Tunneln, Brücken und Fernstraßen bildet das Trassen- oder auch Alignmentmodell die Planungsgrundlage. Dieses beschreibt zunächst den prinzipiellen Verlauf dieser Bauwerke, genauer den Verlauf der Achse dieser Bauwerke und stellt den höchst möglichen Abstraktionsgrad dieser Bauwerke dar (Amann et al. 2014, AREMA 2015). Im Rahmen dieser Arbeit ist ausschließlich die Beschreibung dieser Achse als Grundlage für die geometrische Modellierung von trassenbasierten Infrastrukturbauwerken von Interesse. Andere ebenfalls in Alignmentmodellen enthaltene Informationen, etwa von Querschnitts-, Gelände- oder Oberflächenprofilen (Ji 2014), sollen deshalb in diesem Kontext nicht näher untersucht werden.

Für die konkrete Erstellung eines Trassenmodells verwenden Fachexperten in einer frühen Planungsphase hochspezielle Softwarewerkzeuge. Diese erleichtern über eine graphische Oberfläche die komplexe Modellierung gegenüber der klassischen Planung am Zeichenbrett und verbergen dabei weitgehend das zugrunde liegende mathematische Modell zur Beschreibung des Trassenverlaufs, so dass sich die Planer auf die Einhaltung komplexer (geo- und ingenieurtechnischer) Randbedingungen fokussieren können. Bekannte Beispiele für derartige Softwareprodukte zur Trassenmodellierung sind AutoCAD Civil 3D³⁰, RIB Stratis³¹ oder Obermeyer ProVI³². Diesen Werkzeugen und der heutigen Planungspraxis ist das Grundproblem gemeinsam (Borrmann et al. 2014), dass

die Modellierung der Trasse in der heutigen Planungspraxis weitgehend von der Modellierung der darauf basierenden Infrastrukturbauwerke entkoppelt ist. So sind Änderungen der zugrunde liegenden Trasse ohne eine Neumodellierung der darauf basierenden Modelle weitgehend unmöglich.

Der Grund dafür liegt wohl in der klassischen Trennung der beiden Fachdisziplinen zur Modellierung von Trassen und der von Infrastrukturbauwerken, aus der wiederum fachspezifische Softwarelösungen resultieren, so dass eine native Integration trassierungs-spezifischer Objekte in die typischen dreidimensionalen CAD-Modellierungswerkzeuge wie Siemens NX oder Autodesk Inventor bis heute nicht realisiert wurde.

Um dieses Problem zu lösen, wird eine spezielle Strategie zur Integration des Alignmentmodells in die Architektur der Kollaborationsplattform und damit in den von ihr unterstützten Planungsprozess präsentiert. So wird es dem Trassierungsspezialisten selbst dann möglich, Modifikationen an der Trasse vorzunehmen, wenn die CAD-Experten bereits mit der Modellierung der von Bauwerken begonnen haben, die von der Trasse abhängen. Dazu wird zunächst eine sehr kurze Einführung in die *Alignmentmodellierung* und die hierfür

³⁰ <http://www.autodesk.de/products/autocad-civil-3d/overview>

³¹ <http://www.rib-software.com/de/loesungen/strassenbau-tiefbau-infrastrukturmanagement/stratis.html>

³² <http://www.provi-cad.de/>

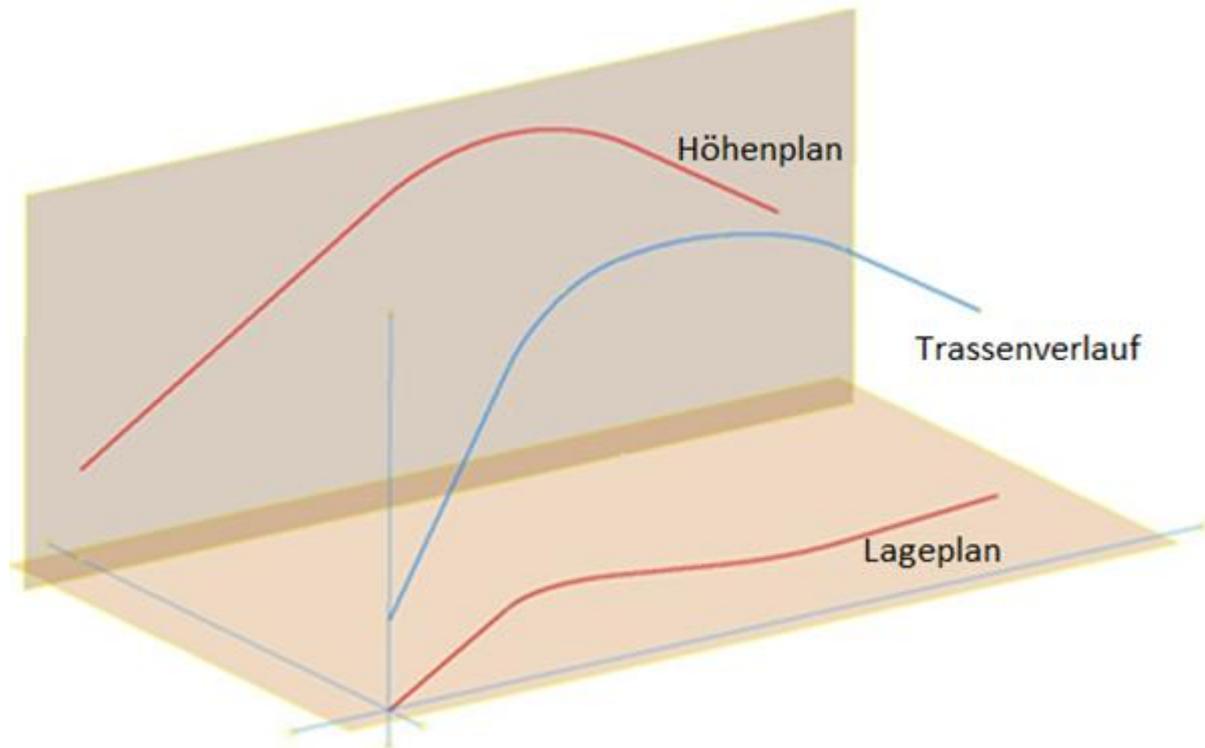


Abbildung 36: Die Überlagerung von Lageplan und Höhenplan beschreibt den dreidimensionalen Trassenverlauf.

grundlegenden Begriffe gegeben, bevor wir einen Lösungsansatz für das beschriebene Problem konkret aufzeigen. Anschließend wird untersucht, wie durch die vorgestellte Architektur und die natürliche Segmentierung des Trassenmodells eine Zerlegung des Gesamtmodells in weitgehend unabhängige kleinere Teilmodelle erreicht werden kann, so dass verschiedene Planer synchron an diesen einzelnen Teilmodellen arbeiten können. Darüber hinaus wird in Kapitel 6.1 gezeigt, wie das Alignmentmodell als Basis einer Simulation zur Berechnung des Energieverbrauchs für einen späteren Zugbetrieb eingesetzt werden kann.

5.1. Mathematische Grundlagen

Die Grundidee bei der Darstellung der Achse einer Trasse ist die Überlagerung zweier ebener Kurven zu einer dreidimensionalen Kurve. Diese beiden Kurven werden im sogenannten Lage- und Höhenplan beschrieben, wie es schematisch in Abbildung 36 gezeigt wird. Die mathematische Verknüpfung der Kurven aus Lage- und Höhenplan wird über die Stationierung erreicht, die als zurückgelegte Streckenlänge auf der Kurve des Lageplanes definiert ist.

5.1.1. Der Lageplan

Im Lageplan wird die Projektion der Trassenachse in die x-y-Ebene als zweidimensionale Kurve beschrieben. Diese setzt sich aus einzelnen Kurvenstücken zusammen, die selbst aus Geraden, Übergangs- und Kreisbögen zwischen Elementen unterschiedlicher Krümmung bestehen. Abbildung 37 zeigt einen solchen Lageplan, der mit Hilfe von AutoCAD Civil 3D erstellt wurde: Grünen Kurvenstücken in dieser Darstellung entsprechen Geradenstücke, blau-



Abbildung 37: Lageplan eines U-Bahnlinienabschnitts in AutoCAD Civil 3D

en Abschnitten Kreisbögen und roten Abschnitten Klothoiden als Übergangskurven zwischen Geradenstücken und Kreisbögen.

Wie in Abbildung 37 zu sehen, sind dabei Abfolgen von ungekrümmten Geradenstücken, Übergangskurven mit sich ändernder Krümmung und Kreisbögen mit konstant positiver Krümmung und umgekehrt typisch. Aber auch direkte Abfolgen von Übergangsbögen, etwa zweier Klothoiden, werden in der Praxis regelmäßig verwendet, während direkte Übergänge zwischen Elementen unterschiedlicher Krümmung auszuschließen sind. Mathematisch gesehen, bedeutet der Einsatz von Übergangsbögen das Vermeiden von Unstetigkeits- und damit anschaulich von Sprungstellen im Krümmungsband, wie in Abbildung 38 dargestellt, während Übergangsbögen im Praxisalltag etwa bei Autofahrten ruckartige Lenkbewegungen reduzieren oder sogar völlig vermeiden (Weise und Durth 2005, Amann et al. 2014). Formal

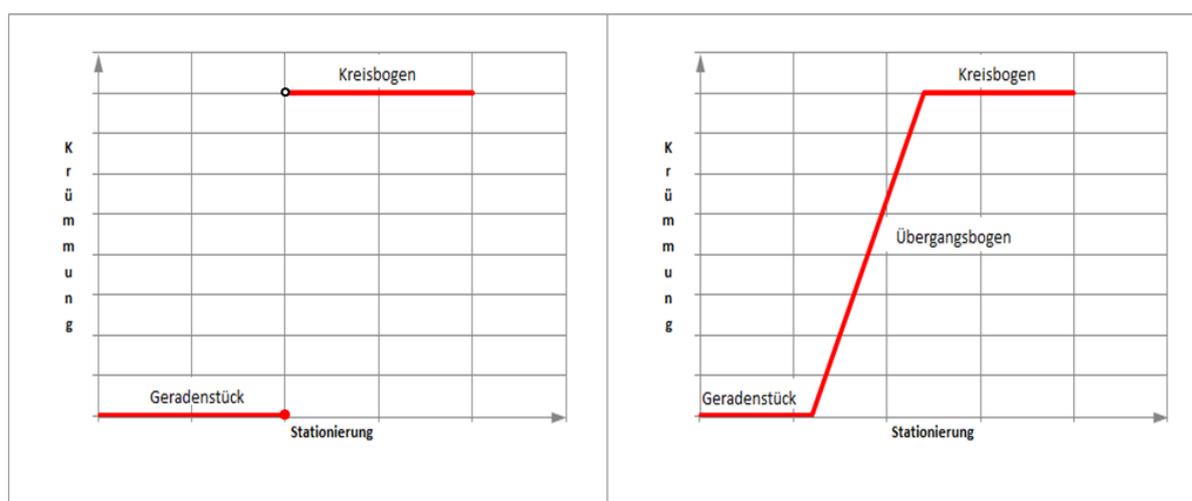


Abbildung 38: Krümmungsband links ohne Übergangsbogen zwischen Gerade und Kreisbogen, rechts mit Übergangsbogen

sind alle diese (gesetzlich zulässigen) Möglichkeiten im Regelwerk für die Trassierung festgelegt; einen umfassenden Überblick über geltende Vorschriften und Regeln findet man etwa bei (Freystein et al. 2015).

5.1.2. Die Klothoide als Übergangskurve im Lageplan

Als Übergangsbögen zwischen Geraden und Kreisbögen werden eine Reihe verschiedener mathematischer Kurven verwendet; sehr häufig ist dabei die Verwendung der Klothoide. Deshalb haben wir in unserem Forschungsvorhaben Klothoiden als Übergangskurven untersucht und im Rahmen der prototypischen Implementierung in unsere konkreten Modelle integriert. Eine Klothoide, auch Euler Spirale genannt, ist durch die Forderung charakterisiert, dass ihre Krümmung K , die als das Inverse des Krümmungsradius R definiert ist, linear mit der Kurvenlänge s ansteigt, d.h. dass $K = \frac{1}{R} \sim s$ gelten soll. Diese Forderung führt im Alltag anschaulich bei einem auf einer Klothoide basierenden Übergangsbogen dazu, dass der Fahrer das Lenkrad nicht ruckartig, sondern mit konstanter Geschwindigkeit einschlagen muss (Menniger 1954). Bereits aus der Proportionalitätsforderung

$$K = \frac{1}{R} \sim s \quad (1)$$

lässt sich die parametrische Darstellung $(x(s), y(s))$ einer Klothoide vollständig zu

$$\begin{pmatrix} x(s) \\ y(s) \end{pmatrix} = A \cdot \pi \cdot \int_0^s \begin{pmatrix} \cos\left(\frac{\pi \cdot t^2}{2}\right) \\ \sin\left(\frac{\pi \cdot t^2}{2}\right) \end{pmatrix} dt \quad (2)$$

herleiten (Brown 2008).

Diese Herleitung gelang zum ersten Mal Leonhard Euler im Jahre 1743 (Brown 2008). Die in der Beschreibung (2) auftretenden Euler- oder auch Fresnel-Integrale sind nicht elementar integrierbar und nur für den Spezialfall $s \rightarrow \infty$ exakt lösbar. Deshalb müssen zur Bestimmung von Punkten auf Klothoiden immer geeignete Näherungsverfahren, etwa numerische Integrationsverfahren, verwendet werden (Raymond 1918). Der in (2) auftretende sogenannte Klothoiden-Parameter A , genauer dessen Quadrat A^2 stellt einen Zusammenhang zwischen dem Krümmungsradius und der Länge der Klothoide her: Denn A^2 ist gerade der Festwert in der definierenden Proportionalitätsbeziehung, d.h. es gilt $R = \frac{A^2}{s}$. Im Bereich der Trassierung

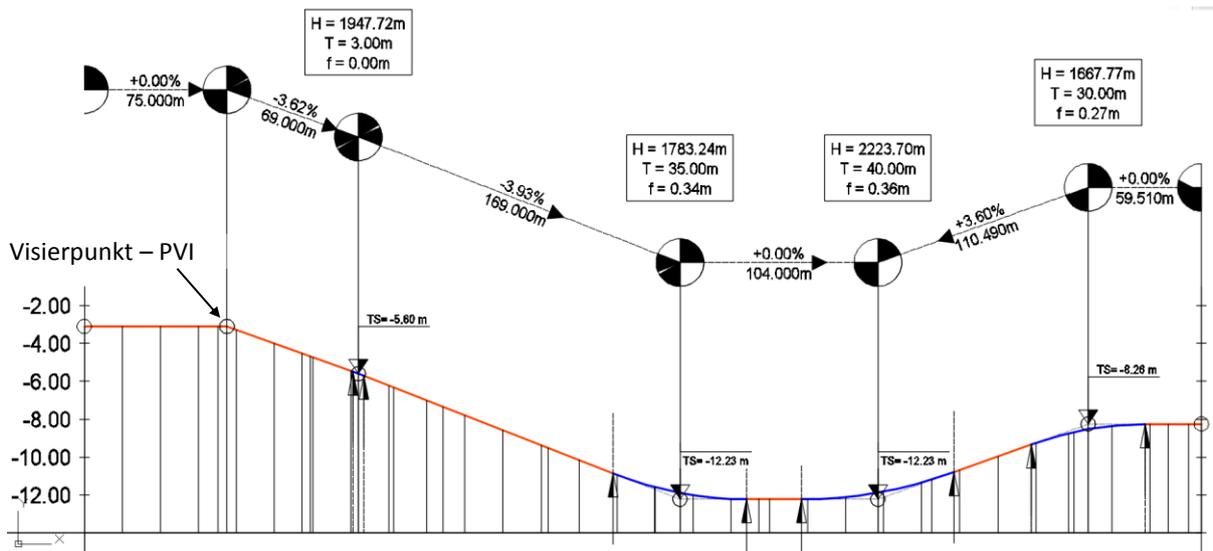


Abbildung 39: Höhenplan - Ausschnitt aus der Trassierung einer U-Bahnlinie in Auto CAD Civil 3D

von Landstraßen sind Werte des Parameters A im Bereich $\frac{R}{3} < A < R$ üblich, wie den Richtlinien für die Anlage von Landstraßen (RAL) zu entnehmen ist.

Andere Beispiele für Übergangsbögen sind etwa der Blossbogen oder der Wiener Bogen. Einen Überblick über verschiedene mögliche Übergangsbögen findet man etwa in (Amann et al. 2014).

5.1.3. Der Höhenplan

Im Höhenplan wird der Verlauf der Trasse in der s - z -Ebene beschrieben. Dabei ist z im dreidimensionalen Raum als Höhe (etwa gegenüber Normalnull) zu interpretieren, während die sogenannte Stationierung s , wie oben gesehen, die Länge der zurückgelegten Strecke auf der Kurve im Lageplan angibt. Die Stationierung ermöglicht so gerade die Verknüpfung von Höhe- und Lageplan: Formal wird jedem Punkt P des Lageplans mit den Koordinaten (x, y) über die streng monotone Abbildung f_1 , welche gerade die zurückgelegte Streckenlänge beschreibt, ein Stationierungswert s zugeordnet $f_1: (x, y) \mapsto s$ und jedem Stationierungswert s über eine Abbildung f_2 wiederum eine Höhe z im Raum: $f_2: s \mapsto z$. Über die Hintereinanderausführung

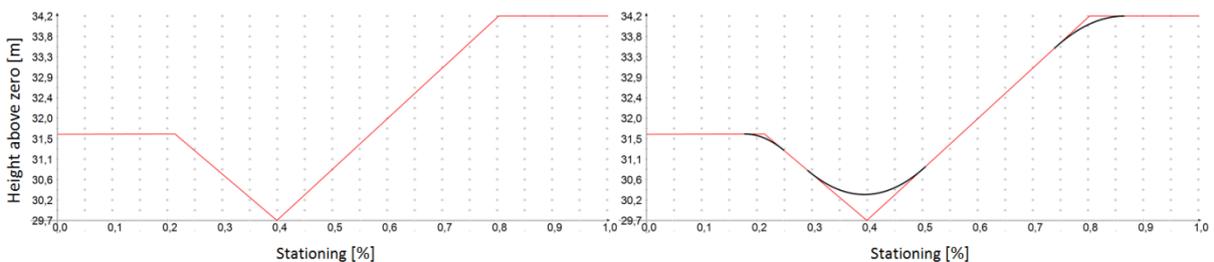


Abbildung 40: Ausschnitt aus einem Höhenplan: Links sind nur die Geradenstücke zwischen den PVI (PVI – Points of Vertical Intersection), rechts zusätzlich die parabelförmigen Ausrundungen in Form von Kuppen und Wannen dargestellt.

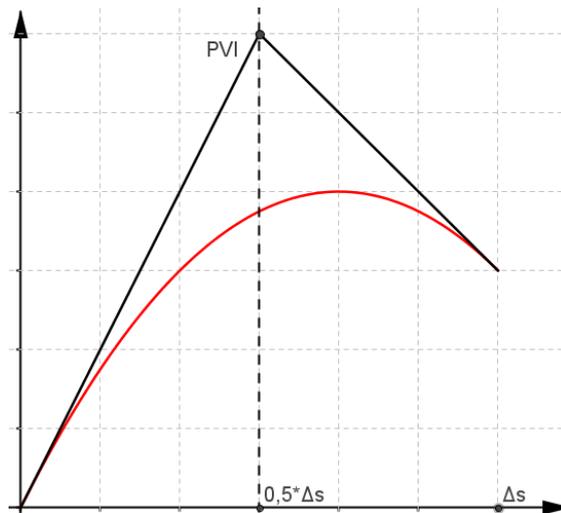


Abbildung 41: Bei der Ausrundung durch eine Parabel in Form einer Kuppe halbiert der PVI gerade den zugehörigen Stationierungsbereich.

der beiden Abbildung $f_2 \circ f_1$ ergibt sich also eine Zuordnung, die jedem Punkt des Lageplans die entsprechende Höhe im Höhenplan zuordnet, also $f_2(f_1(x, y)) = f_2(s) = z$.

Bei der konkreten Trassenmodellierung definiert der Experte zunächst die sogenannten Visierpunkte (*PVI – Points of Vertical Intersection*), die in Abbildung 39 durch Kreise angedeutet sind. Die Visierpunkte werden dann durch Geradenstücke verbunden, bevor deren Schnittstellen abschließend durch Parabelbögen (Abbildung 40 rechts) oder seltener Kreisbögen – in Form von Wannen und Kuppen – ausgerundet werden. Im Straßenbau sind diese Möglichkeiten in den „Richtlinien für die Anlage von Landstraßen (RAL)“ beschrieben, deren aktuelle Version im Jahr 2013 eingeführt wurde. Für detaillierte Informationen sei der Leser auf Informationen bei der Bundesanstalt für Straßenwesen BASt verwiesen³³. Für den Bau von Bahnstrecken findet man entsprechende Informationen in den Regelwerken der DB Netz AG³⁴.

5.1.4. Die Parabel als Übergangselement im Höhenplan

Die im Tunnelbau am häufigsten – in Deutschland ausschließlich – verwendete Methode für Übergänge im Höhenplan sind Ausrundungen in Form von Parabelbögen. Diese Methode soll hier näher erläutert werden, da sie im Rahmen des Forschungsprojekts dieser Arbeit untersucht und auch prototypisch implementiert wurde, um Änderungen am Höhenplan zuzulassen und die automatische Anpassung des abhängigen Modells an diese Änderungen zu demonstrieren.

³³ http://www.bast.de/DE/Home/home_node.html

³⁴ <http://fahrweg.dbnetze.com/fahrweg-de/nutzungsbedingungen/regelwerke/>

Prinzipiell genügt es in der gegebenen Situation, ausschließlich die Schnittpunkte der Geradenstücke – also die Visierpunkte – und die Stationierungsdifferenz Δs zwischen dem Start- und Endpunkt eines Parabelstücks zu kennen, um den Verlauf einer Parabelausrundung vollständig zu berechnen. Um dies einzusehen, macht man sich zunächst klar, dass der Tangentschnittpunkt den vorgegebenen Stationierungsbereich der Ausrundungsparabel genau halbiert (Abbildung 41).

Ohne Beschränkung der Allgemeinheit ist es hier hilfreich, den Koordinatenursprung in den Berührungspunkt von linker Tangente und Ausrundungsparabel zu legen. Falls der betrachtete *PVI* durch die Koordinaten (\tilde{s}, \tilde{z}) gegeben ist, haben die beiden Tangenten t_l und t_r ($l := \text{links}; r := \text{rechts}$) mit den Steigungen m_l und m_r die Gleichungen $t_l: y = m_l \cdot s$ und $t_r: y = m_r(s - \tilde{s}) + \tilde{z} = m_r(s - \tilde{s}) + m_l \tilde{s}$.

Die Funktionsvorschrift der Ausrundungsparabel hat allgemein die Form $p(s) = a \cdot s^2 + b \cdot s + c$. Die direkt aus der Wahl des Koordinatensystems resultierende Bedingung $p(0) = 0$ liefert unmittelbar $c = 0$, während die Bedingung $p'(0) = m_l$ den Parameterwert $b = m_l$ ergibt. Schließlich kann der Parameter a aus $p'(\Delta s) = m_r$ zu $a = \frac{m_r - m_l}{2\Delta s}$ bestimmt werden. So lässt sich zu guter Letzt der Stationierungswert \tilde{s} des *PVI* angeben, wenn man beachtet, dass die Parabel und die rechte Tangente bei der Stationierung Δs den gleichen Funktionswert $t_r(\Delta s) = p(\Delta s)$ besitzen:

$$\begin{aligned} m_r(\Delta s - \tilde{s}) + m_l \tilde{s} &= \frac{m_r - m_l}{2\Delta s} \Delta s^2 + m_l \Delta s \\ \Leftrightarrow \tilde{s}(m_l - m_r) &= \frac{m_r - m_l}{2} \Delta s + m_l \Delta s - m_r \Delta s \\ \Leftrightarrow \tilde{s}(m_l - m_r) &= \frac{m_l - m_r}{2} \Delta s \Leftrightarrow \tilde{s} = \frac{1}{2} \Delta s. \end{aligned}$$

Diese Erkenntnis stärkt auch die beim Entwurf eines Höhenplans gängige Vorgehensweise, zunächst die einzelnen Visierpunkte aufgrund gegebener (äußerer) Bedingungen zu definieren, diese durch einen Linienzug zu verbinden und dann unter Vorgabe der Längen der Übergangsstücke Ausrundungsparabeln einzufügen.

5.2. Formate zum Austausch von Alignmentmodellen

Zur Beschreibung und zum Austausch von Alignmentmodellen wurde eine Reihe von verschiedenen Dateiformaten entwickelt, die sich im Laufe der Jahre vor allem regional etablieren konnten, wobei sich kein Format weltweit als anerkannter Standard durchgesetzt hat. Aktuell versucht die buildingSMART³⁵ Initiative, mit dem Format IFCAAlignment³⁶ einen solchen internationalen Standard zu schaffen. Die verschiedenen, momentan etablierten Dateiformate unterscheiden sich hinsichtlich der Beschreibung der Achse im Wesentlichen in den unterstützten Typen von Übergangskurven und in den für die Modellierung der Kurven verwendeten Parametern.

Häufig angewendet wird das im Jahr 2002 in der Version 1.0 erschienene LandXML-Format³⁷ (Rebolj et al. 2008), das sich sehr flexibel hinsichtlich der unterstützten Typen von Übergangskurven zeigt; dabei werden konkret 16 verschiedene Typen integriert, insbesondere die oben beschriebene Klothoide. Auch bezüglich der verwendeten Parameter gewährt LandXML dem Benutzer große Freiheiten, schließt dabei aber die Verwendung redundanter Parameter nicht aus. Zwischen 2009 und 2014 wurde der LandXML-Standard nicht weiterentwickelt, bevor im Jahr 2014 – möglicherweise etwas überraschend – eine neue Version 2.0 im Internet präsentiert wurde.

RoadXML³⁸ ist ein häufig im Straßenbau und im Bereich der Fahrsimulation verwendetes Format, das sich durch seine klare, komprimierte und einfache Struktur deutlich vom oben beschriebenen LandXML-Format unterscheidet (Chaplier et al. 2010). So wird in RoadXML ein einziger Übergangskurventyp, nämlich die Klothoide, unterstützt. Gleichzeitig wird nur ein minimales Parameterset zur Beschreibung der einzelnen Kurven zugelassen, bei einer Klothoide gerade sechs Stück. Damit wird das oben beschriebene Redundanzproblem per Definition vermieden.

Ein ausschließlich im deutschen Sprachraum verwendeter Standard für die Modellierung von Straßentrassierungsmodellen ist OKSTRA³⁹ (Schultze und Buhmann 2008). Der eingeschränkte Verbreitungsraum ist wohl in den konkreten Definitionen des Standards in deutscher Sprache zu sehen. Wie in RoadXML ist nur die Klothoide als Übergangskurventyp integriert. So wird die im Rahmen dieser Arbeit gemachte Einschränkung gestützt, nur diesen Typ von Übergangskurve im Lageplan detaillierter zu untersuchen und zu modellieren.

Einen ausführlichen Vergleich verschiedener Formate und ein Konzept, durch die Einführung der domänenspezifischen Sprache IFCPL den Austausch von Alignmentmodellen auf eine neue Basis zu stellen, findet man bei (Amann et al. 2014). Das Problem redundanter Parame-

³⁵ <http://www.buildingsmart.org/>

³⁶ <http://www.buildingsmart-tech.org/infrastructure/projects/alignment>

³⁷ <http://www.landxml.org/>

³⁸ <http://www.road-xml.org/>

³⁹ <http://www.okstra.de/>

ter wird hier durch die Verwendung eines minimalen Parametersets von vorneherein ausgeschlossen. Zusätzlich wird vorgeschlagen, neben den Parametern standardisierte (IFCPL) Programme zu ihrer Interpretation auszutauschen, so dass eine Fehlinterpretation der Parameterwerte von vornherein ausgeschlossen ist.

5.3. Integration von Alignmentmodellen in die Kollaborationsplattform

Wir demonstrieren nun einen Lösungsansatz für die in der Einführung dieses Kapitels beschriebene Grundproblematik der momentanen Trennung bei der Modellierung von Trassen und den darauf basierenden Bauwerken. Hier ist es insbesondere das Ziel, die Modellierung von Trassen als Grundlage eines prozeduralen Infrastrukturmodells innerhalb des durch die Plattform *Kollabserver* unterstützten Arbeitsprozesses zu ermöglichen und dabei den Anforderungen eines Trassierungsspezialisten zu entsprechen. Die Trasse wird hierzu zunächst mit Hilfe spezieller Trassierungssoftware (etwa Autodesk Civil 3D) oder mittels der im Rahmen dieser Arbeit entworfenen *Alignmenteditor*-Komponente modelliert. Dabei werden die klassischen Trassierungselemente verwendet, bevor die Trasse bei der späteren Modellierung des Tunnelbauwerks in den benutzten CAD-Systemen durch Splines visualisiert wird. Die Verwendung von Splines bietet sich an, da CAD-Systeme wie etwa Siemens NX oder Autodesk Inventor Splines zur Darstellung von (glatten) Kurven zur Verfügung stellen, während typische Trassierungselemente wie etwa Klothoiden nicht nativ unterstützt werden. Im gegebenen Kontext wäre auch der Einsatz von Polygonzügen denkbar. Ihre Verwendung würde aber bei der Darstellung stets zu zahlreichen (leichten) Knicken sowohl im Trassen- als auch im Tunnelverlauf führen, selbst wenn man die Anzahl der vorgegebenen Interpolationspunkte beliebig erhöht.

Die vorgeschlagene Trennung bei der Modellierung der Trasse und ihrer Visualisierung erfolgt in Anlehnung an das klassische Paradigma, das eine Trennung von zugrunde liegenden Daten und ihrer Darstellung propagiert (siehe etwa MVC-Pattern bei Gamma et al. 1995). Diese Trennung von Daten und Visualisierung wird konkret durch eine bidirektionale Verknüpfung zwischen einzelnen nativen Kurvenelementen (des Lageplans) und repräsentierenden Splines ermöglicht (Abbildung 42).

In ihrer Gesamtheit wird die Trasse nicht durch einen einzelnen Spline dargestellt, sondern auf Basis des Lageplans in verschiedene Splines unterteilt; so wird jeder Teil der dreidimensionalen Trasse, der durch die Superposition genau eines Elements im Lageplan und des zugehörigen Bereichs im Höhenplan entsteht, durch genau einen Spline dargestellt. Alle Berechnungen hinsichtlich von Änderungen an der Trasse werden weiterhin auf Basis der originären Trassierungselemente durchgeführt, während die darstellenden Splines gemäß den Berechnungsergebnissen angepasst werden.

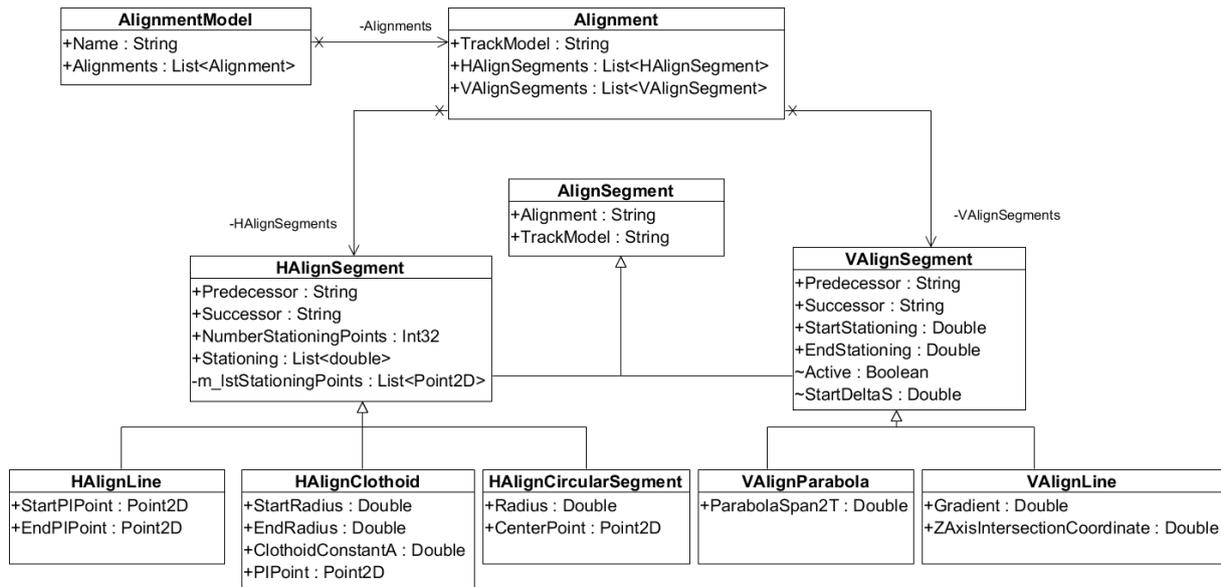


Abbildung 43: Objektstruktur für eine Implementierung des Alignmentmodells

Im folgenden Kapitel werden wir sehen, dass diese Unterteilung den Vorteil bietet, das gesamte prozedurale Modell des Infrastrukturbauwerks in weitgehend unabhängige Teilmodelle zerlegen zu können. Diese Zerlegung erweitert die Möglichkeiten einer synchronen Modellierung signifikant, denn sie erlaubt es verschiedenen Planer, simultan an diesen Teilmodellen zu arbeiten.

Für eine dieser Problemstellung angemessene Beschreibung des Alignmentmodells wurde eine Objektstruktur entworfen, die schematisch in Abbildung 43 dargestellt ist. In diese Struktur wurden neben den beschreibenden Eigenschaften im Gegensatz zu den gängigen rein deskriptiven Dateistandards auch Methoden integriert, die insbesondere Möglichkeiten zur Berechnung von Stationierungswerten auf den Elementen des Lageplans zur Bestimmung der Länge beliebiger Kurvenabschnitte umfassen. So kann dann die Superposition von Lageplan- und Höhenplan-Elementen zur Berechnung dreidimensionaler Kurvenpunkte automatisiert (und bei Modifikationen in Echtzeit) durchgeführt werden. Für jeden einzelnen Kurventyp sowohl des Lage- als auch des Höhenplans wurde dabei eine eigene Klasse entworfen: So repräsentieren die Klassen vom Typ „*HAlignSegment*“ die Elemente des Lageplans und ermöglichen die angesprochene Berechnung der Stationierungswerte, während die Klassen vom Typ „*VAlignmentSegment*“ den Elementen des Höhenplans entsprechen und eine Berechnung der z-Koordinate in Abhängigkeit der Stationierung erlauben.

Zusätzlich wurde eine Softwarekomponente *Alignmenteditor* entworfen und prototypisch implementiert: Sie erlaubt zunächst einen Import von in LandXML gespeicherten Trassierungsdaten aus gängigen Trassierungswerkzeugen in die in Abbildung 43 dargestellte Objektstruktur. Zudem wurden in diese Funktionalitäten sowohl zur Veränderung des Höhenplans durch eine Variation der Visierpunkte als auch zur Veränderung des Lageplans integriert. Damit ist

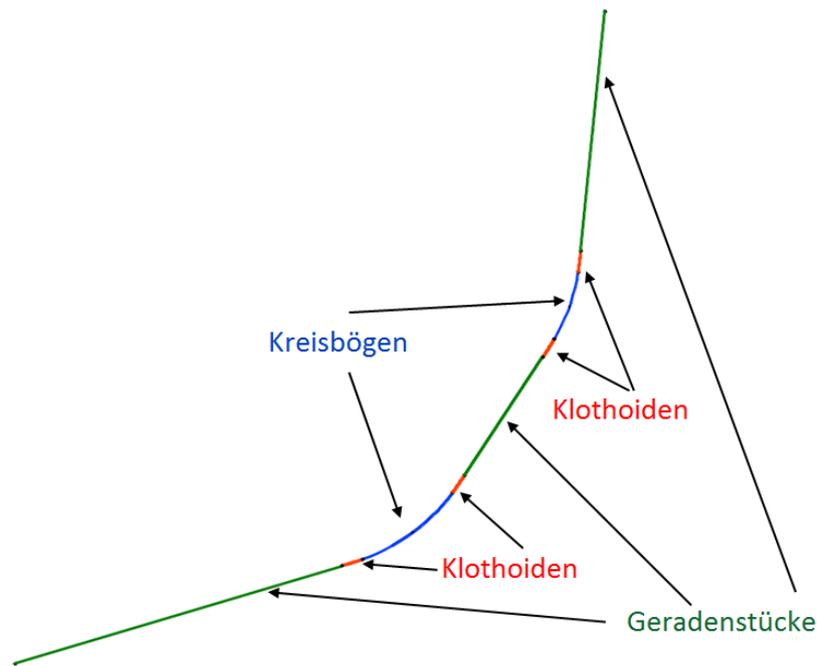


Abbildung 44: Ausschnitt aus dem Lageplan einer Trasse bestehend aus neun verschiedenen Elementen. Dabei sind in grüner Farbe drei Geradenstücke, in roter Farbe vier Klothoiden und in blauer Farbe zwei Kreisbögen abgebildet.

dem Trassierungsexperten im Modellierungsprozess über die Plattform *Kollabserver* die Möglichkeit gegeben, (auch in einer fortgeschrittenen Planungsphase) Änderungen an der Trasse vorzunehmen. Denn das von der Trasse abhängige Geometriemodell wird automatisch durch die in das Modell integrierten geometrischen Abhängigkeiten von den Splines angepasst, die selbst auf der Plattform gemäß den Anforderungen des Experten modifiziert und Neuberechnet werden. Dabei gehen wir davon aus, dass in dieser Modellierungsphase die Planung der Trasse weitgehend abgeschlossen ist, der Planer aber noch einzelne Parameter der verschiedenen Trassenelemente verändern darf, ohne dabei die Gesamtabfolge in ihrem Kern zu ändern, d.h. die Abfolge der einzelnen Elemente bleibt unverändert, während die konkrete Form einzelner Elemente nach wie vor angepasst werden kann. Exemplarisch ist in Abbildung 44 der Lageplan eines Trassenabschnitts bestehend aus neun einzelnen Elementen abgebildet, auch der in Abbildung 37 dargestellte Ausschnitt liefert eine gute Grundlage für die folgenden Überlegungen. In solchen Fällen erlaubt der vorgestellte Ansatz in der angesprochenen Phase der Modellierung noch etwa die Änderung der Radien der Kreisbögen oder der Klothoidenparameter. Im Anhang zeigen wir dazu insbesondere, dass eine Modifikation der Kreisradien nur einen Einfluss auf die benachbarten Klothoiden und Geradenstücke hat, während die weiteren Elemente unbeeinflusst bleiben. Dies bedeutet dann insbesondere, dass eine Modifikation eines Kreisradius im hier vorgestellten Prozess der synchronen prozeduralen Modellierung nur eine Sperrung der benachbarten Klothoiden und Geradenstücke sowie die der davon abhängigen prozeduralen Elemente erfordert, ohne dass andere Teile des Modells durch Sperrungen blockiert werden.

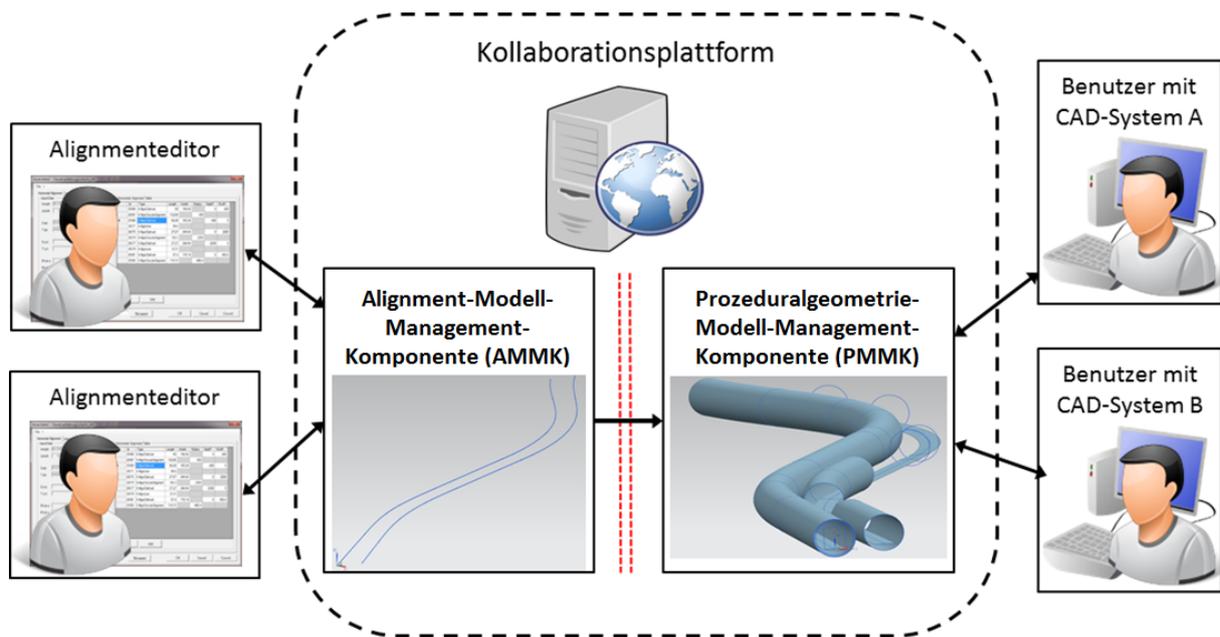


Abbildung 45: Eine Komponentenstruktur erlaubt die Trennung von Trassierungsdaten und den darauf basierenden prozeduralen Geometriedaten.

Die Grundlage für diesen Mechanismus in der Architektur der Plattform *Kollabserver* bildet die sogenannte *Alignment-Modell-Management-Komponente* (AMMK), die konkret die Verwaltung trassierungsspezifischer Daten übernimmt und über einen dedizierten Kommunikationskanal mit der in Kapitel 3.6 besprochenen *Prozeduralgeometrie-Modell-Management-Komponente* (PMMK) interagiert (Abbildung 45). Einzelnen Trassierungselementen werden dabei, wie oben erläutert, prozedurale Geometrieelemente zur Visualisierung zugeordnet, wobei die Verwaltung dieser Zuordnungen der AMMK unterliegt.

In der konkreten Umsetzung benutzt eine Instanz einer solchen AMMK eine Instanz der angegebenen Objektstruktur zur Beschreibung der Trasse. Wie in Abbildung 45 dargestellt, arbeitet der Trassierungsexperte zur Modifikation der Trasse mit einem speziellem Trassierungstool, dem *Alignmenteditor*, der ihm gerade die Möglichkeiten zur Modifikation auf Basis der Elemente des Lage- und Höhenplans ermöglicht, ohne ihm dabei einen direkten Zugriff auf die im CAD-System verwendeten Splines zu gewähren, die (nur) zu ihrer Darstellung benutzt werden.

Sobald ein solches Trassierungselement modifiziert wurde, werden automatisch von der AMMK die relevanten Informationen an die PMMK geschickt. Diese wiederum passt automatisch die zugehörigen Splines an und schickt die entsprechenden Aktualisierungen an die anderen Clients und fordert von diesen ein, ihre Umgebungen zu aktualisieren. Dieser Prozess passiert völlig analog zur Modifikation eines prozeduralen Elements, wie er in Kapitel 3.6 beschrieben wurde. Da bei dieser Vorgehensweise die durch die AMMK angestoßenen Modifikationen genauso wie die Modifikationen eines beliebigen CAD-Clients behandelt werden, werden die oben angeführten Strategien zur Nebenläufigkeitskontrolle automatisch implizit

genutzt: So werden automatisch alle betroffenen Splines gesperrt, die mit der zuvor beschriebenen eindeutigen Verknüpfung bestimmt werden können. Dies führt wiederum gemäß den Ausführungen in Kapitel 3.7 zur Sperrung aller abhängigen Elemente. Gesperrte Elemente werden dabei gemäß den Prinzipien über ein gemeinsames Bewusstsein in kollaborativen Softwarewerkzeugen für die Aktionen anderer Benutzer sowohl im Aligneditor als auch in den CAD-Werkzeugen (farblich) besonders visualisiert (siehe auch Abbildung 8). Die Strategie, die AMMK wie einen eigenen CAD-Client in den Modellierungsprozess einzubeziehen, sichert gemäß den Ausführungen in Kapitel 3.7 die Konsistenz zwischen dem zentralen Modell und den lokalen Replikaten auf Seiten der Benutzer, und zwar sowohl der CAD-Modellierer als auch der Trassierungsexperten. Von entscheidendem Vorteil in dieser Strategie ist, dass diese Mechanismen zur Konsistenzwahrung keinerlei explizites Eingreifen der Benutzer erfordern, sondern völlig transparent arbeiten, so dass sich die Benutzer voll und ganz auf ihre inhaltlichen Fragestellungen konzentrieren können.

Exemplarisch konnten mit Hilfe der besprochenen Komponenten mehrere Trassen, etwa die Trasse der neuen Zweiten Stammstrecke München und – darauf basierend – eine Menge (detaillierter) Tunnelmodelle erstellt werden (Borrmann et al. 2014). Insbesondere konnte so in der Praxis gezeigt werden, dass die vorgeschlagene Strategie die Grundforderung einer nahtlosen Modellierung der Trasse und des davon abhängigen Infrastrukturbauwerks löst.

5.4. Teilmodellerzeugung basierend auf dem Alignmentmodell

Im folgenden Abschnitt wird gezeigt, wie das oben erläuterte Konzept zur Integration der Alignmentdaten als Grundlage für die Zerlegung des darauf basierenden Gesamtmodells in mehrere kleinere Teilmodelle genutzt werden kann. Dadurch wird es möglich, dass sich ein einzelner Experte auf einen kleinen Ausschnitt des Gesamtmodells fokussieren und genau diesen bearbeiten kann, wobei die aus Modifikationen resultierenden Sperrungen nur gerade diese Ausschnitte betreffen und die Konsistenz des Gesamtmodells – automatisiert durch die Kollaborationsplattform – aufrechterhalten wird. Zunächst ist hier festzustellen, dass die Trasse (im Wesentlichen) eine C^2 -stetige Kurve, also eine Kurve ohne Sprungstellen im Krümmungsband ist und im vorgeschlagenen prozeduralen Modellierungsprozess in ihrer Gesamtheit – wie zuvor beschrieben – durch verschiedene aneinandergefügte Splines repräsentiert wird. Jeder einzelne Spline selbst ist eine C^2 -stetige Kurve; kritisch sind jedoch die Übergänge zwischen den einzelnen Splines. Eine stetige Visualisierung, also eine Darstellung ohne Lücken zwischen den einzelnen Splines, lässt sich im prozeduralen Modell durch eine Zwangsbedingung zwischen zwei Splines abbilden, bei der das Zusammenfallen des Endpunkts des einen Splines mit dem Anfangspunkt des zweiten festgelegt wird. Ähnliches gilt für eine gemeinsame Tangente und Krümmung an eben besagtem Übergang. Dieses Vorgehen führt dann aber zu Abhängigkeiten zwischen den einzelnen Splines im prozeduralen Modellierungsgraphen, die sich über die komplette Repräsentation der Trasse hinweg fortsetzen. Alternativen, bei denen Abhängigkeiten im prozeduralen Modell vermieden werden und gleichzeitig die geforderte Kontinuität aufrechterhalten wird, gibt es nicht. Bei der Verwendung des oben beschriebenen Sperrmechanismus ergibt dies einen entscheidenden Nachteil: Die Sperrung eines einzelnen Splines würde zur Sperrung des von der entsprechenden Trasse abhängigen Teilmodells, also im Fall einer einzelnen Trasse sogar zur Sperrung des gesamten Modells führen. Denn die in diesem Fall untereinander abhängigen Splines bilden im beschriebenen Anwendungsfall aus Sicht der Prozeduralgeometrie die Basis für das komplette weitere Modell. Die Grundidee zur Zerlegung des Gesamtmodells ist es nun, auf den einzelnen Splines basierende unabhängige Teilmodelle zu generieren und den stetigen Übergang zwischen den einzelnen Splines von außerhalb des prozeduralen Modells, insbesondere ohne die Verwendung von Abhängigkeiten erzeugenden Zwangsbedingungen zu steuern, so dass insbesondere eine Unabhängigkeit im Sinne des oben beschriebenen Algorithmus zur Nebenläufigkeit entsteht. Diese Idee kann schematisch in Abbildung 46 beobachtet werden: Das gesamte Modell zerfällt in zwei Teilmodelle, die zwei voneinander völlig unabhängige Modellierungsgraphen aufweisen. Im Folgenden wird beschrieben, wie diese Idee konkret mit Unterstützung durch die AMMK umgesetzt werden kann.

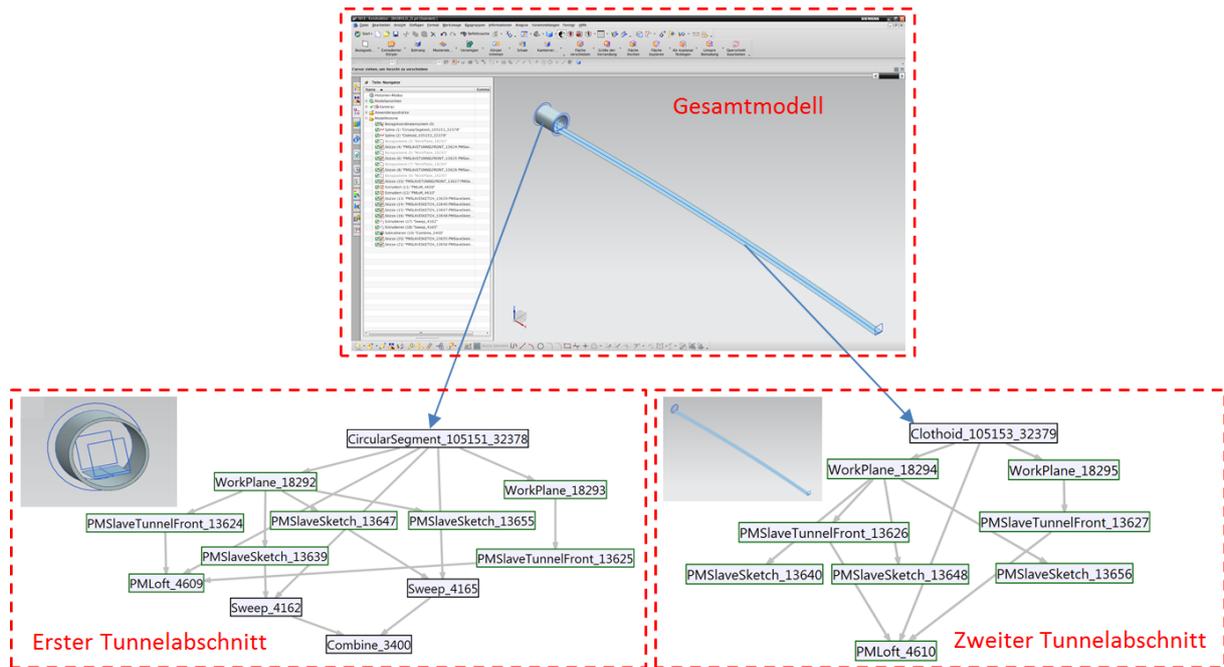


Abbildung 46: Zwei aufeinanderfolgende Teilmodelle mit voneinander unabhängigen Abhängigkeitsgraphen

Bei der geometrischen Modellierung von trassenbasierten Infrastrukturbauwerken, insbesondere bei der Modellierung von Tunneln, ist nach der Modellierung der Splines, welche die Trasse repräsentieren, der typische Arbeitsschritt die Abbildung der Tunnelquerschnitte durch Skizzen, die senkrecht zu den Splines liegen (vergleiche dazu auch das Beispiel am Ende von Kapitel 3.5). Splines und Skizzen werden dann als Basis für darauf folgende Sweep-Operationen verwendet. Die Verwendung einer einzigen Skizze und einer alle verschiedenen Splines umfassenden einzelnen Sweep-Operation wäre theoretisch möglich, würde aber völlig dem Ziel widersprechen, das Gesamtmodell in kleinere Einheiten zu zerlegen. Im Fall der Modellierung eines Tunnels ist eine Abfolge der zugehörigen prozeduralen Operationen als Beispiel in Abbildung 13 in Kapitel 3.5 dargestellt, insbesondere sind in dieser Abbildung die verschiedenen Levels of Detail gemäß der multi-skaligen Definition hervorgehoben.

Diese Folge verschiedener Querschnitte, die zu Beginn eines jeden Splines angefügt werden müssen, kann man durch eine Serie von Master-Replikatskizzen geschickt abbilden. Zunächst wird eine sogenannte Masterskizze als Querschnitt senkrecht an einen Spline angefügt. Diese Skizze wird dann automatisiert an die anderen Spline-Enden repliziert. Dabei definiert die Masterskizze die konkrete Querschnittsgeometrie, die von den Replikaten zunächst nur kopiert, aber entsprechend der aktuellen Querneigung angepasst wird, die aus der aktuellen Krümmung der Trasse resultiert. Diese Vorgehensweise erlaubt es, dass die Querschnittsgeometrie nur ein einziges Mal definiert werden muss, während sich die Replikatskizzen automatisch gemäß der durch die Trassenkrümmung bedingten Querneigung anpassen (Borrmann et al. 2014).

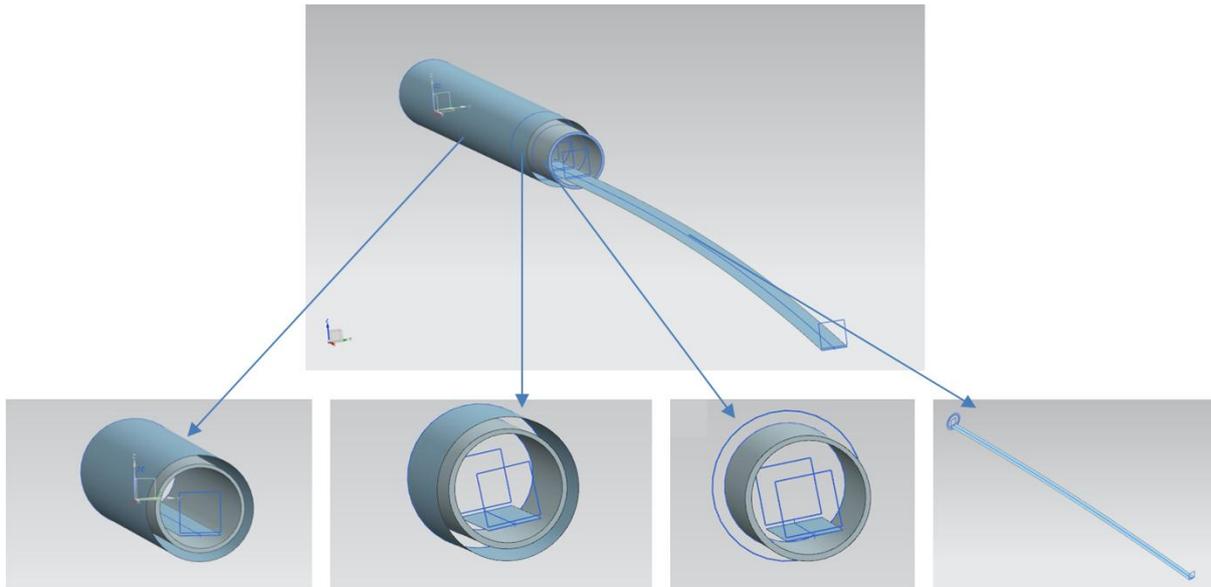


Abbildung 47: Das oben dargestellte Modell kann gemäß der zugrunde liegenden Trassierung in vier einzelne Teilmodelle zerlegt werden.

Diese Master- und Replikatskizzen werden nun als Ausgangsbasis für verschiedene Sweep-Operationen verwendet, die etwa im Fall eines Tunnels den Tunnelquerschnitt, im Fall einer Brücke den Fahrbahnquerschnitt repräsentieren. So definiert die Masterskizze und jede einzelne Replikatskizze mit dem zugehörigen Spline jeweils genau eine Sweep-Operation. Diese Vorgehensweise liefert unmittelbar kleinere Teilmodelle, die nur über die Master-Replikat-Beziehungen voneinander abhängen. Ein Beispiel für die Zerlegung eines Tunnelmodells in kleinere Teilmodelle ist in Abbildung 47 zu sehen.

Diese so definierten Teilmodelle lassen sich einfach aus dem Gesamtmodell extrahieren, um ihre spezifische Bearbeitung zuzulassen. Der dazu entwickelte Algorithmus „*Extrahiere Teilmodell*“ ist in Abbildung 48 angegeben und soll hier kurz erläutert werden: Zunächst müssen der Spline oder die Splines angegeben werden, die das Teilmodell definieren. Anschließend werden die Objekte, die von diesen abhängen, rekursiv durchlaufen und dem Teilmodell hinzugefügt. Dies geschieht konkret in der Routine „*FügeObjekUndAbhängigeHinzu*“, der das zugrunde liegende Element als Parameter („*Basiselement*“) übergeben wird.

Um die Konsistenz des Teilmodells und des restlichen Modells zu wahren, muss im Falle, dass es sich beim übergebenen Basiselement um eine Replikatskizze handelt, eine Verbindung zur Masterskizze, ihrem direkten Elternobjekt und deren Elternobjekten gewahrt bleiben. Dazu wird dies innerhalb der Routine zunächst überprüft und gegebenenfalls die Routine „*FügeObjektUndElternHinzu*“ aufgerufen. Dies sichert insbesondere die Aktualität des Teilmodells im Fall der Modifikation einer Masterskizze durch andere an der kollaborativen Modellierung teilnehmende Experten. Die Masterskizze (und ihre Elternobjekte) werden dabei nicht visualisiert, sondern für den Modellierer transparent im Hintergrund verwaltet, um ihre Modifikationen an die Replikatskizzen und die darauf basierenden Elemente weiterzugeben.

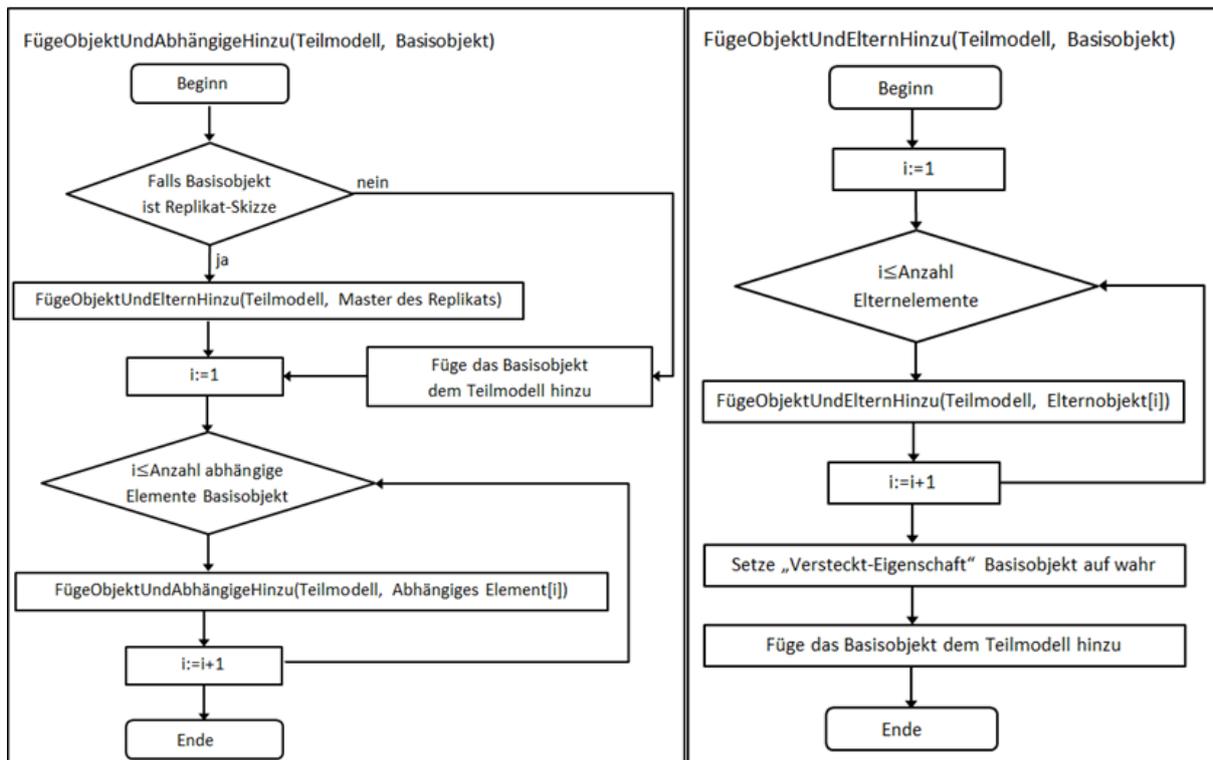
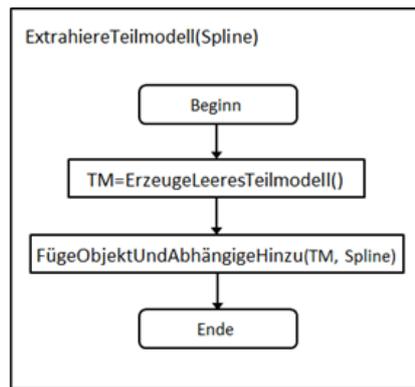


Abbildung 48: Struktur des Algorithmus zur Extraktion von Teilmodellen

Im Sinne des prozeduralen Modellierungsgraphen sind die einzelnen Tunnelsegmente so völlig voneinander unabhängige Zusammenhangskomponenten. Natürlich muss aber zwischen den einzelnen Segmenten ein nahtloser und glatter Übergang bestehen. In unserem Ansatz wird jedoch diese Stetigkeit durch die von der PMMK getrennte AMMK gewährleistet, d.h. insbesondere kennt nur diese die Verbindungen und Abhängigkeiten der einzelnen Splines, während diese für das prozedurale Geometriemodell völlig transparent sind.

Deshalb müssen die Abhängigkeiten zwischen den Splines auch nicht in ihre Abhängigkeitsmengen aufgenommen werden, wie in obigem Beispiel beim Kreissegment `CircularSegment_105151_32378` und der Klothoide `Clothoid_105153_32379` (Abbildung 47) konkret im Abhängigkeitsgraphen abgelesen werden kann. Bei der Modifikation eines einzelnen Splines, also eines einzelnen Trassenabschnitts, berechnet die AMMK automatisch alle notwendigen

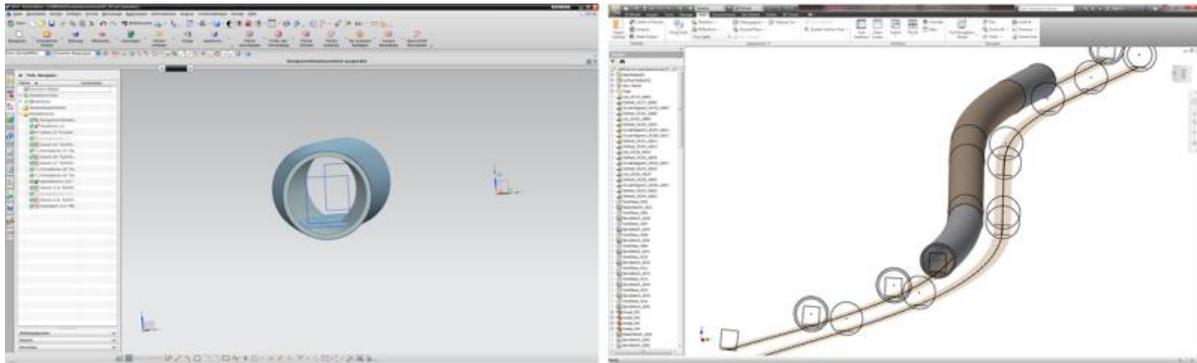


Abbildung 49: Ein Benutzer extrahiert eine Tunnelteilmodell (Siemens NX), während ein anderer Benutzer gleichzeitig das Gesamtmodell betrachtet (Autodesk Inventor).

Modifikationen für die einzelnen Nachbarelemente der Trasse und leitet diese an die PMMK weiter. Insbesondere sperrt die AMMK alle betroffenen Elemente, so dass die Konsistenz des Gesamtmodells erhalten bleibt.

Der dargestellte Ansatz hat zumindest zwei grundlegende Stärken: Zum einen erlaubt er dem Experten, Teile des Bauwerks zu extrahieren, zu visualisieren und zu modifizieren, ohne den Rest des Modells überhaupt kennen zu müssen oder im Falle von benutzerrechtlichen Einschränkungen kennen zu dürfen (Abbildung 49). Zum anderen ermöglicht das Fehlen der Abhängigkeiten im geometrischen Modell und deren Auslagerung in die externe AMMK gemäß dem oben präsentierten Sperr-Algorithmus (siehe auch Kapitel 3.7) eine Modifikation von Teilmodellen, ohne andere Teilmodelle zu beeinflussen und daher ohne diese von der Bearbeitung durch andere Benutzer ausschließen zu müssen.

5.5. Zusammenfassung

In diesem Kapitel haben wir zunächst einige Grundlagen der Alignmentmodellierung behandelt, da das Alignment die Basis für die Modellierung von Infrastrukturbauwerken wie Tunnel, Fernstraßen oder Brücken bildet.

Dann haben wir ein Konzept präsentiert, wie die typischen Trassierungselemente in den prozeduralen Modellierungsprozess integriert werden können. Dabei löst die Strategie zur separierten Verwaltung der Alignmentmodelle durch die AMMK und der Verwaltung des prozeduralen Geometriemodells durch die PMMK das grundlegende Problem anderer Ansätze, dass das Trassierungsmodell nicht mehr modifiziert werden kann, sobald mit der Modellierung der darauf basierenden Geometrie begonnen wurde. Diese Strategie ermöglicht insbesondere eine synchrone Zusammenarbeit zwischen den Spezialisten der Trassierung und denen der Bauwerksmodellierung. Insbesondere konnten wir dabei zeigen, dass sich hier die in Kapitel 3.7 vorgestellten Konzepte zur Nebenläufigkeitskontrolle nahtlos integrieren und sehr wirkungsvoll einsetzen lassen.

Weiter konnten wir darlegen, dass man die natürliche Segmentierung der Trasse durch die Lageplan-Elemente wirkungsvoll nutzen kann, um das Gesamtmodell des Bauwerks in kleinere Einheiten zu zerlegen. Dazu haben wir ein Konzept zur Replikation von Masterskizzen in einem Master-Replikat-Konzept aufgezeigt. Dieses Konzept und die Segmentierung durch das Alignmentmodell ermöglichen im Gegensatz zu vorhandenen Ansätzen sowohl eine simultane Visualisierung als auch eine gleichzeitige Bearbeitung von Teilmodellen durch verschiedene Spezialisten innerhalb des kollaborativen Planungsprozesses. Dabei ist es sogar möglich, dass einzelne Experten nur spezielle Teilmodelle, nicht aber das Gesamtmodell kennen.

Insgesamt wurde so ein Konzept zur synchronen Bearbeitung und Modifikation einzelner Teilmodelle bei gleichzeitiger automatisierter Aufrechterhaltung der Konsistenz zwischen zentralem Modell und den lokalen Replikaten auf der Benutzeroberfläche geschaffen.

6. Einbindung von Simulationssoftware

Im Bauwesen finden Simulationen in immer zahlreicheren Szenarien Eingang (siehe auch Kapitel 1). Auch hinsichtlich der Bewertung der Modelle, die über die Kollaborationsplattform entworfen werden, ergeben sich verschiedenste Einsatzmöglichkeiten. Allen ist das Ziel gemeinsam, die unterschiedlichen Experten bei dieser Bewertung zu unterstützen, und zwar im besten Fall so, dass die Ergebnisse der Simulationsprozesse direkt in den Modellierungsprozess zurückgeführt werden können.

Ein generell immer wiederkehrendes Problem beim Einsatz von Simulationswerkzeugen besteht in der Aufbereitung der (geometrischen) Modelldaten in Formate, die als Eingangsdaten für spezifische Simulationswerkzeuge verwendet werden können. Exemplarisch werden wir für zwei Simulationstypen aufzeigen, wie wir diese in den Modellierungsprozess über die Kollaborationsplattform integrieren. Insbesondere können wir ihre Ergebnisse dann direkt gewinnbringend in den weiteren Planungsprozess einbringen.

Zunächst werden wir ein Konzept für eine Energiesimulation präsentieren. Diese unterstützt die Planer in einer frühen Phase dabei, die Trassierung aus energetischen Gesichtspunkten zu beurteilen und zu optimieren. Dazu stellen wir insbesondere eine Möglichkeit vor, mit Hilfe eines Optimierungsverfahrens Höhenprofile automatisiert so zu verbessern, dass der Energieverbrauch reduziert oder sogar minimiert wird. Als zweites mögliches Szenario werden wir darstellen, wie wir Bahnhofsmodele automatisiert so transformieren, dass die resultierenden Daten direkt als Eingangsquelle für Personenstromsimulatoren verwendet werden können. Diese helfen den Experten bei der Beurteilung der Bauwerke hinsichtlich des Alltagsbetriebs oder auch der Beurteilung von Evakuierungsszenarien.

6.1. Energiesimulation auf Basis des Alignmentmodells

Während der Trassenfindung unterliegt der Experte einer Vielzahl von komplexen Bedingungen, die unter anderem durch geografische und geotechnische Gegebenheiten vorgegeben sind. Der Aspekt der Energieeffizienz einer Trasse in Bezug auf den späteren U-Bahnbetrieb findet dabei momentan kaum Eingang in diesen Prozess. Diese fehlende Unterstützung konnte auch im Rahmen einer intensiven Literaturstudie belegt werden, in welcher deutlich wurde, dass Untersuchungen der Energieeffizienz im Wesentlichen a posteriori durchgeführt und so Möglichkeiten der Optimierung im operativen Betrieb gesucht werden, etwa verbesserte Geschwindigkeitsprofile zu berechnen (siehe auch Kapitel 6.1.1). Dagegen werden (a priori) Untersuchungen hinsichtlich eines optimierten Trassenverlaufs kaum durchgeführt. Prinzipiell ist hinsichtlich der Energieeffizienz zumindest folgende Strategie weit verbreitet: Direkt nach dem Verlassen eines Bahnhofes wird die Trasse abwärtsgelegt, so dass die Gewichtskraft die Beschleunigung des Zuges unterstützt. Vor dem nächsten Zielbahnhof kann dann die (möglicherweise erforderliche) potentielle Energie für eine Fahrt aufwärts beim Bremsen aus der kinetischen Energie gewonnen werden. Detaillierte quantitative Informationen bezüglich des

Energieverbrauchs sind jedoch für den Experten in der momentanen Planungspraxis nicht zugänglich. Deshalb wurde das Konzept einer autonomen Energiesimulation entworfen. Diese ist dadurch charakterisiert, dass sie bei Änderungen des Trassenverlaufs automatisch im Hintergrund die sich ergebenden Änderungen im Energieverbrauch berechnet. Im Folgenden soll zunächst dieses Konzept dargestellt und daran anschließend gezeigt werden, dass bereits kleine Änderungen im Trassenverlauf zu merklichen Energieeinsparungen führen können. Über eine Betriebsdauer von üblicherweise vielen Jahrzehnten können diese Einsparungen im Energieverbrauch dann zu erheblichen Kostenreduzierungen führen. Dieses Einsparpotential werden wir durch konkrete Beispiele belegen, die sich im Rahmen der Validierung des Modells bei der Verwendung von realen Daten der Metro Bilbao ergaben. Abschließend werden wir ein Konzept vorstellen, wie man mit Hilfe eines Optimierungsverfahrens Höhenprofile so modifizieren kann, dass sie hinsichtlich des Energieverbrauchs deutlich verbessert oder sogar optimal sind.

6.1.1. Modelle für die Berechnung der Energieeffizienz von Schienenfahrzeugen

In diesem Abschnitt werden wir einige (grundlegende) Arbeiten im Gebiet der Energieeffizienz von Trassen aufführen und ihre zentralen Ergebnisse kurz darlegen. Zunächst gibt es zahlreiche, grundlegende Erkenntnisse hinsichtlich der Schienenfahrzeuge selbst. Insbesondere gibt es eine Vielzahl an Untersuchungen zu Kräften und ihren Wirkungen im Fahrbetrieb genauso wie zu Reibungseinflüssen und deren Beschreibung. Für detaillierte Informationen in dieser Richtung verweisen wir auf (Wende 2003), (Knothe und Stichel 2003), (Iwinicki 2006) und (Janicki et al. 2013), für eine detaillierte Untersuchung von Reibungseinflüssen insbesondere auf (Lukaszevicz 2001).

Fundamentale Untersuchungen zur konkreten Energieeffizienz von Trassen hingegen lieferte Howlett. Im Mittelpunkt seiner Forschungen stand insbesondere die Frage, ob es bei einem gegebenen Trassenprofil ein Geschwindigkeitsprofil gibt, das zu einem minimalen Energieverbrauch führt. Dabei standen lange Überlandstrecken von Frachtzügen in seinem Heimatland Australien im Zentrum seiner Untersuchungen. Diese Fragestellung konnte positiv, wenn auch in einer eher theoretischen Form beantwortet werden. So konnte mathematisch nachgewiesen werden, dass ein derartiges Profil existiert und auch eindeutig ist, zumindest wenn das Streckenprofil als einfach genug angenommen wird. Da seine Beweise meist nicht konstruktiv durchgeführt wurden, werden diese Profile aber nicht konkret angegeben. Wir verweisen hier auf (Howlett 1990, Howlett et al. 1994, Howlett und Cheng 1997, Howlett 2000, Howlett et al. 2009). Ebenfalls mit der Optimierung von Geschwindigkeitsprofilen befassen sich Jong sowie Jong und Chang (Jong 2003, Jong und Chang 2005). Sie präsentieren eine Computersimulation, in der zwei auf Kräftenmodellen basierende Algorithmen zur Generierung von Geschwindigkeitsprofilen implementiert wurden. Der eine dieser Algorithmen hat dabei das Ziel,

bei gegebenen Randbedingungen eine möglichst kurze Fahrzeit zu erreichen, während der zweite bei gegebener Fahrzeit den Energieverbrauch zu minimieren versucht. Reibungseinflüsse werden dabei nur rudimentär behandelt. Auch die Möglichkeit der Energierückgewinnung bei Bremsvorgängen findet hier keinen Eingang in die Simulation. Großer Wert wird hingegen auf die Integration und vor allem auf die automatische Generierung von gegebenen Geschwindigkeitsbegrenzungen infolge des Trassenverlaufs gelegt.

Kim und Chien entwickelten eine „Train Performance Simulation (TPS)“ (Kim und Chien 2010). Diese besitzt drei Grundkomponenten: ein Kräfte-Modul zur Beschreibung der physikalischen Rahmenbedingungen hinsichtlich des Zugbetriebs, ein Alignment-Modul und eine Zugsteuerungskomponente. Ziel dieser Simulation ist es, Unterschiede im Energieverbrauch bei verschiedenen Zugsteuerungsmustern zu bestimmen. Prinzipiell sind Änderungen im Höhenprofil, nicht aber im Lageplan möglich, auch die Möglichkeit der Energierückgewinnung findet keinen Eingang. Die Ergebnisse liefern Steuerungsmuster für die Züge und richten sich damit primär an die Zugbetreiber und nicht – wie in unserem Fall – an die in der Planung der Trasse involvierten Experten.

Auch Su et al. untersuchen den Energieverbrauch von Zügen, insbesondere den von U-Bahnzügen (Su et al. 2013). Hierbei ist es ihr Ziel, energieeffiziente Geschwindigkeitsprofile zu bestimmen, die dann direkt in sogenannten „Automatic Train Operation (ATO)“-Systemen verwendet werden können. Einerseits untersuchen sie Abschnitte zwischen einzelnen U-Bahnstationen, andererseits nehmen sie auch die Gesamtstrecke in ihre Untersuchungen auf. So konnten sie konkret Geschwindigkeitsprofile für die U-Bahnlinie „Beijing Yizhuang“ in China bestimmen, die zwischen maximal 10% und 15% energieeffizienter als die momentan verwendeten Profile sind. Hull et al. versuchen ebenfalls, durch bessere Strategien bei der Zugsteuerung den Energieverbrauch zu reduzieren, und berücksichtigen im Gegensatz zu den zuvor beschriebenen Ansätzen sehr wohl die Möglichkeit, bestimmte Energiemengen beim Bremsen wieder zurückzugewinnen (Hull et al. 2010).

Mit der Optimierung von Geschwindigkeitsprofilen setzen sich auch Yang et al. auseinander, insbesondere im Rahmen des durch die Bauindustrie Bayern geförderten Projekts „ProZeit“ (Yang und Meng 2011, Yang et al. 2015). Dabei ist die Kernfrage, wie sich eine kürzere Fahrzeit zwischen zwei gegebenen Punkten durch eine Veränderung des Geschwindigkeitsprofils unter Beachtung der gegebenen Randbedingungen erreichen lässt. Dabei berücksichtigen sie als Randbedingungen insbesondere die maximal zulässigen Geschwindigkeiten in Kurvenstücken. Ein zentraler Punkt ist es dann, die Überhöhung in diesen Kurvenstücken so anzupassen, dass sich eine entsprechend höhere und trotzdem zulässige Geschwindigkeit ergibt und diese Änderungen der Überhöhungen zugleich im Rahmen von kostengünstigen Wartungsarbeiten durchgeführt werden können, ohne dabei die Trasse komplett neu aufschütten zu müssen. Zusätzlich werden dabei automatisch die bei der Optimierung der Überhöhung entstehenden Kosten abgeschätzt.

Wie wir gesehen haben, ist es typisch, die Energieeffizienz von Trassen erst in der Betriebsphase zu untersuchen. Das Ziel ist es deshalb häufig, effizientere Geschwindigkeitsprofile zu finden. Die Energieeffizienz einer Trasse bereits in der frühen Planungsphase zu untersuchen, ist momentan nach bestem Wissen des Autors kaum Gegenstand der aktuellen Forschungen. Allen obigen Modellen ist weiterhin gemeinsam, dass die unterlegten Kräfteemodelle eine Analyse hinsichtlich der verschiedenen Einflussfaktoren (Höhenprofil, Krümmungsprofil, etc.) nicht separat zulassen. Gerade diese verschiedenen Faktoren können aber für den planenden Ingenieur in der frühen Planungsphase von enormer Wichtigkeit bei der Beurteilung der Energieeffizienz und ganz besonders beim Verständnis für die Ursachen des (unterschiedlichen) Energieverbrauchs sein. Generell gehen wir hinsichtlich der Anwendung unseres Modells von folgendem Planungsfortschritt aus: Der Trassierungsexperte hat den grundlegenden Trassenverlauf etwa aufgrund von Geoinformationen abgeschlossen und ist nun mit der Feinabstimmung der Trasse beschäftigt. In dieser Phase sind nur noch kleinere Änderungen möglich; so kann etwa das Höhenprofil noch etwas angepasst, Kurvenradien leicht verändert oder das Geschwindigkeitsprofil modifiziert werden. In Kapitel 6.1.4 werden wir zeigen, wie wir mit unserem Modell genau diese Problemstellungen lösen können, werden aber zuvor die theoretischen physikalischen Grundlagen für unser Modell darlegen.

6.1.2. Ein physikalisches Modell zur Energiesimulation eines U-Bahnzuges

Zunächst wird jetzt das verwendete physikalische Modell zur Berechnung des Energieverbrauchs eines U-Bahnzuges präsentiert. In diesem Modell unterscheiden wir als vier Grundgrößen die aufzuwendende kinetische und potentielle Energie, den Energieaufwand infolge von Reibungseinflüssen und – davon getrennt – den Einfluss der Reibung bei Kurvenfahrten, den sogenannten Kurvenwiderstand. Die Eingangsgrößen zur Berechnung dieser vier Größen sind das Geschwindigkeits-, das Höhen- und das Krümmungsprofil:

- Geschwindigkeitsprofil $v := v(s)$,
 - Höhenprofil $h := h(s)$,
 - Krümmungsprofil $c := c(s)$,
- (3)

wobei alle Profile von der Stationierung s abhängen. Dabei können entweder durch den Benutzer manuell festgelegte Geschwindigkeitsprofile oder durch das System generierte einfache Profile (Wende 2003) verwendet werden. Höhen- und Krümmungsprofile werden automatisiert aus den Trassendaten gewonnen. Bei geringfügigen Änderungen des Trassenverlauf, insbesondere des Höhenprofils (etwa zur Optimierung des Energieverbrauchs) kann das Geschwindigkeitsprofil dann (in einer ersten Näherung) durch eine lineare Transformation T auf die neuen Geometriedaten angepasst werden und muss so nicht bei jeder Änderung vom Benutzer komplett neu definiert werden. Wenn s_{A0}, s_{B0} und s_{A1}, s_{B1} die Start- und Endpositionen

der ursprünglichen bzw. modifizierten Stationierung beschreiben, ist diese Transformation folgendermaßen definiert:

$$T: [s_{A1}; s_{B1}] \rightarrow [s_{A0}; s_{B0}]$$

$$s \mapsto s_{A0} + \frac{s_{B0} - s_{A0}}{s_{B1} - s_{A1}} \cdot (s - s_{A1}) \quad (4)$$

Dann ergibt sich das neue Geschwindigkeitsprofil \tilde{v} aus dem ursprünglichen Profil v mit Hilfe dieser Transformation durch

$$\tilde{v}: [s_{A1}; s_{B1}] \rightarrow \mathbb{R}$$

$$s \mapsto \tilde{v}(s) := (v \circ T)(s) = v(T(s)). \quad (5)$$

Da die im Rahmen dieser Forschung durchgeführten Variationen der Höhenprofile typischerweise zu Änderungen der Gesamtstreckenlänge im Bereich von wenigen Promille geführt haben, scheint dieser Ansatz in einer ersten Näherung gerechtfertigt. In Abbildung 50 ist links ein Geschwindigkeitsprofil abgebildet, dem Messdaten der Metro Bilbao auf dem Streckenabschnitt von Casco Viejo nach Abando zugrunde liegen, während in Abbildung 50 rechts das Höhenprofil desselben Streckenabschnitts dargestellt ist. In beiden Fällen ist die Stationierung in Prozent der Gesamtstreckenlänge angegeben, die Geschwindigkeit in $\frac{km}{h}$ und die Höhe in Metern (über Normalnull).

6.1.2.1. Die vier verschiedenen energetischen Terme

Zunächst sollen nun die verschiedenen energetischen Terme, die im Simulationsmodell berücksichtigt wurden, einzeln dargestellt werden. Generell behandeln wir dabei vereinfachend einen U-Bahnzug wie einen Massepunkt, und zwar so, als ob die Masse des Zuges in seinem Schwerpunkt vereinigt wäre. Dies mag zunächst als eine starke Vereinfachung erscheinen, die Vergleiche zwischen den in unserem Modell berechneten Werten und Messdaten der Metro

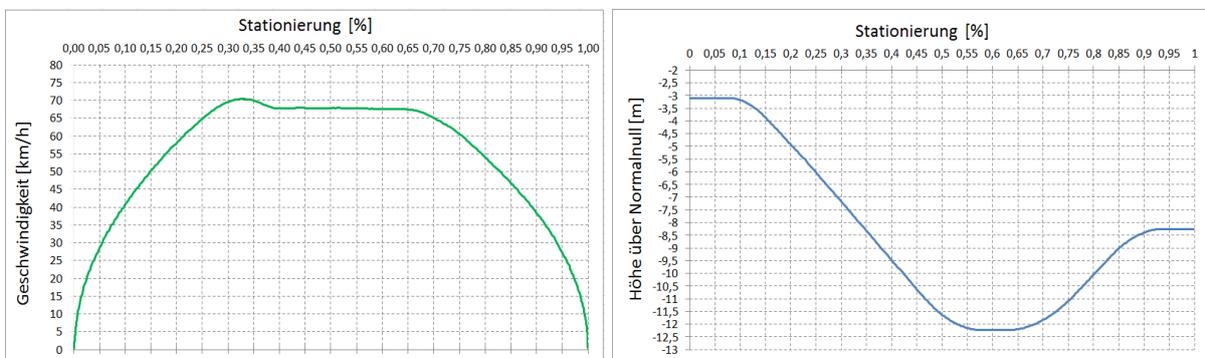


Abbildung 50: Beispiele für ein Geschwindigkeits- und ein Höhenprofil – Originalhöhenprofile eines Streckenabschnitts der Metro Bilbao

Bilbao rechtfertigen dies aber im Nachhinein sehr gut. Die separate Berechnung der einzelnen Einflussfaktoren ist enorm hilfreich für das Verständnis des Gesamtenergieverbrauchs, wie wir in Abschnitt 6.1.4 sehen werden.

Kinetische Energie

Die kinetische Energie – auch Bewegungsenergie genannt – gibt den Energiebetrag an, der aufzuwenden ist, um einen Zug während seiner Fahrt (auf die durch das Geschwindigkeitsprofil vorgegebenen Geschwindigkeiten) zu beschleunigen. Wie allgemein üblich, wird ein Bremsvorgang durch eine negative Beschleunigung erfasst. Prinzipiell verbirgt sich dahinter das Arbeitsintegral $W = \Delta E_{Kin} = \int_{s_1}^{s_2} F_B(s) ds$, in welchem $F_B(s)$ die beschleunigende Kraft in Abhängigkeit des Orts s angibt und über einen Weg von s_1 nach s_2 integriert wird (Demtröder 2015). Der Wert dieses Integrals ergibt sich zu $\Delta E_{Kin} = \frac{1}{2} \cdot m \cdot (v_2^2 - v_1^2)$, hängt also nicht vom konkret gewählten Weg, sondern nur von den Geschwindigkeiten v_1 am Ausgangspunkt s_1 und v_2 am Zielort s_2 ab. Somit spielt der Verlauf der Kraft in Abhängigkeit des Orts keine Rolle. Diesen Zusammenhang werden wir nun kurz herleiten. Es gilt zunächst

$$\Delta E_{Kin} = \int_{s_1}^{s_2} F_B ds = \int_{s_1}^{s_2} F_B \cdot \frac{ds}{dt} dt = \int_{s_1}^{s_2} F_B \cdot v dt. \quad (6)$$

Nach dem zweiten Newtonschen Gesetz gilt $F_B = \frac{d}{dt}(m \cdot v) = \underbrace{\frac{dm}{dt}}_{=0} \cdot v + m \cdot \frac{dv}{dt} = m \cdot a$, wenn

man die Masse als konstant ansieht, was im betrachteten Fall der Fahrt eines U-Bahnzuges zwischen zwei Stationen sicherlich zutrifft. Beachtet man nun, dass $\frac{d}{dt}(v^2) = \frac{d}{dt}(v \cdot v) = \frac{dv}{dt} \cdot v + v \cdot \frac{dv}{dt} = 2 \cdot v \cdot \frac{dv}{dt}$ gilt, so ergibt sich

$$F_B \cdot v = \frac{d}{dt}(m \cdot v) \cdot v = \frac{1}{2} \cdot \frac{d}{dt}(m \cdot v^2),$$

und man erhält

$$\Delta E_{Kin} = \int_{s_1}^{s_2} F_B \cdot v dt = \frac{1}{2} \cdot \int_{s_1}^{s_2} \frac{d}{dt}(m \cdot v^2) dt = \frac{1}{2} \cdot m \cdot (v_2^2 - v_1^2) \quad (7)$$

mit $v_2 = v(s_2)$ und $v_1 = v(s_1)$. Der Übergang beim letzten Gleichheitszeichen in dieser Gleichung ist dabei gerade eine direkte Anwendung des Hauptsatzes der Differential- und Integralrechnung (HDI).

Abbildung 51 verdeutlicht nochmals diese Wegunabhängigkeit bei der Berechnung der kinetischen Energie. Bei der Berechnung der kinetischen Energie ist hinsichtlich unserer Problemstellung insbesondere zu beachten, ob diese Energie durch den Motor zu liefern ist oder ob die

aufzuwendende kinetische Energie oder zumindest ein Teil von ihr durch einen Unterschied der Höhenenergie infolge eines Gefälles aufgebracht werden kann. Diese Fragestellung werden wir detailliert betrachten, wenn wir später die Bilanz über die einzelnen Energierme aufstellen.

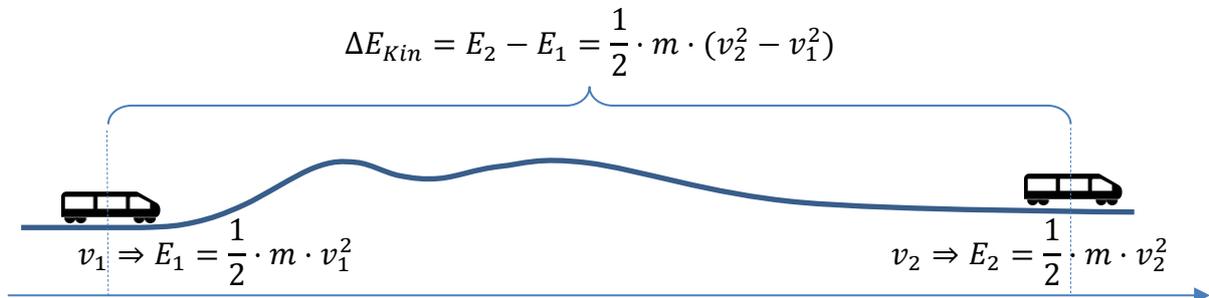


Abbildung 51: Die Änderung der kinetischen Energie hängt nur von der Anfangs- und Endgeschwindigkeit ab und ist damit unabhängig vom konkreten Weg, auf dem eine Strecke durchlaufen wurde.

Potentielle Energie

Um einen Körper von einer Höhe h_1 auf eine zweite Höhe h_2 mit $h_1 < h_2$ zu bringen, muss Arbeit gegen die wirkende Gewichtskraft verrichtet werden. Umgekehrt verrichtet im Fall $h_1 > h_2$ die Gewichtskraft selbst Arbeit am Körper. Dieser Arbeitsbetrag W definiert die Änderung seiner potentiellen Energie ΔE_{Pot} , die auch als Lageenergie bezeichnet wird. Formal ist die potentielle Energie durch das Arbeitsintegral $W = \Delta E_{Pot} = \int_{s_1}^{s_2} G(s) ds$ definiert, wenn ein Körper gegen die ortsabhängige Gewichtskraft $G(s)$ von einem Ort s_1 an einen anderen Ort s_2 gebracht wird. Ähnlich wie zuvor für die kinetische Energie kann für die Gewichtskraft $G(s)$ gezeigt werden, dass die potentielle Energie nur von Anfangs- und Zielhöhe (der Orte s_1 und s_2) abhängt und durch den Ausdruck $\Delta E_{Pot} = m \cdot g \cdot (h_2 - h_1)$ gegeben ist (Abbildung 52). Allgemein verbirgt sich dahinter die Aussage, dass in einem konservativen Kraftfeld – wie hier – der Arbeitsbetrag entlang eines Weges in diesem Feld nur vom Anfangsort und Zielort abhängt (Demtröder 2015). In unserem Fall geben dabei h_1 die Höhe des Zugschwerpunktes zu Beginn und h_2 die Höhe am Ende des Vorgangs gegenüber einem frei

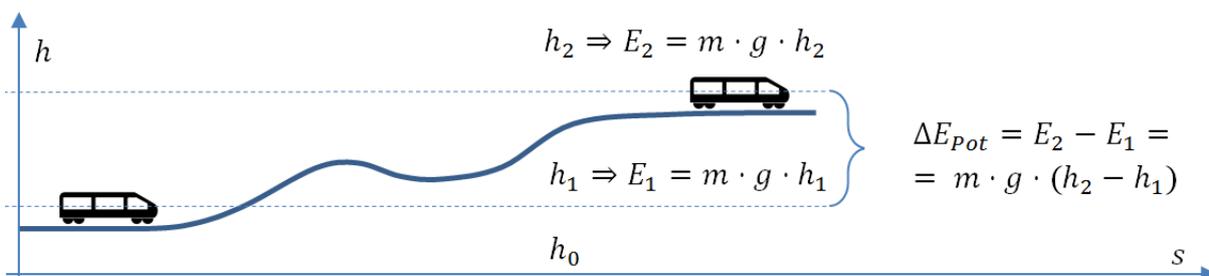


Abbildung 52: Der Unterschied in der potentiellen Energie zwischen zwei Orten hängt nur von der Anfangs- und Endhöhe ab.

wählbaren Nullniveau h_0 an. Dieses sogenannte Nullniveau definiert die Höhe, auf welcher dem Körper eine potentielle Energie mit dem Wert null zugeordnet wird (Abbildung 52).

Einfluss von Reibung

Während der Fahrt eines Zuges treten verschiedenste Reibungseinflüsse auf wie etwa die Reibung zwischen den Rädern und der Gleisoberseite oder die Reibung zwischen den Flanschen der Räder und den Gleisen. Hinzu kommen ein Reibungsverlust in den Lagern der Achsen und der Einfluss des Luftwiderstandes. Um diese verschiedenen Einflussfaktoren zu berücksichtigen, wurden sehr komplexe Modelle erstellt (Wende 2003). Eine einfache, aber wirkungsvolle Möglichkeit, diese Einflüsse zu modellieren, bietet die sogenannte *Davis Equation*⁴⁰, welche auch heute noch verbreitet Anwendung findet (Hay 1982, Rochard und Schmid 2000, James et al. 2005, Denver et al. 2013, Allen und Chien 2016). In ihr wird eine Reibungskraft F_R angenommen, die sich aus drei einzelnen Summanden zusammensetzt. Diese Summanden werden durch die Parameter A , B und C gewichtet, wodurch spezielle Zugtypen (zumindest grob) berücksichtigt werden können (siehe Tabelle 3). Die konkreten Werte für diese Parameter erhält man aus Abschätzungen und Messungen. In ihrer Originalformulierung nach *Davis* lautet diese Gleichung

$$F_R = A + B \cdot v + C \cdot v^2. \quad (8)$$

Der erste Summand soll dabei von der Geschwindigkeit unabhängige Einflüsse wie etwa die Reibung zwischen Rollen und Gleisen beschreiben. Der zweite Term wächst linear mit der Geschwindigkeit und beschreibt Einflüsse, wie etwa das „Schwanken“ des Zuges, während der dritte Summand Faktoren berücksichtigt, die mit dem Quadrat der Geschwindigkeit wachsen, wie etwa den Luftwiderstand. Nach Davis kann in (4) der Parameter A durch $A = 1.3 + 29/w$ ersetzt werden, wenn w das Gewicht pro Achse beschreibt. Weiterhin kann C durch $C = \frac{K \cdot Q}{W}$ ersetzt werden, wenn K den Luftwiderstandsbeiwert, Q die Zugquerschnittsfläche und W das Gesamtgewicht des Zuges angibt. So erhält man die *Davis Equation* in der modifizierten Form zu

$$F_R = 1.3 + \frac{29}{w} + B \cdot v + \frac{K \cdot Q}{W} \cdot v^2. \quad (9)$$

Als Beispiel sind Werte für die verschiedenen Parameter nach Iwnicki in folgender Tabelle 3 dargestellt (Iwnicki 2006), eine detaillierte Erläuterung der Bedeutung der Parameter A , B und C findet man auch bei (Boschetti und Mariscotti 2012) und (Baker 2014).

⁴⁰ Der Name dieser Gleichung geht auf ihren Urheber W. J. Davis Jr. zurück, welcher die prinzipielle Form dieser Gleichung bereits 1926 einführte.

Zugtyp/Parameter	A $\left[\frac{N}{\text{ton}}\right]$	B $\left[\frac{N \cdot h}{\text{ton} \cdot \text{km}}\right]$	C $\left[\frac{N \cdot h^2}{\text{ton} \cdot \text{km}^2}\right]$
French Bogie	14.7	0	0.00218
ICE	9.81	0.024525	0.053955

Tabelle 3: Beispielwerte für die Parameter A , B und C nach (Iwnicki 2006)

Der gesamte Energieaufwand infolge der durch (8) und (9) beschriebenen Gleichungen und damit der ihr zugrunde liegenden Reibungseinflüsse ergibt sich dann als Integral über $F_R = A + B \cdot v + C \cdot v^2$ entlang des Weges, d.h. es gilt

$$\begin{aligned}
 E_R &= \int_{s_a}^{s_e} F_R ds = \int_{s_a}^{s_e} (A + B \cdot v + C \cdot v^2) ds \\
 &= A \cdot (s_e - s_a) + \int_{s_a}^{s_e} (B \cdot v + C \cdot v^2) ds.
 \end{aligned} \tag{10}$$

Dabei ist natürlich zu beachten, dass sowohl die Geschwindigkeit als auch deren Quadrat vom Ort abhängen.

Kurvenwiderstand

Der sogenannte Kurvenwiderstand ergibt sich als weiterer Einflussfaktor aus der Reibung zwischen Gleisen und Flanschen der Räder während einer Kurvenfahrt. Dabei hängt die wirkende Reibungskraft vom Krümmungsradius der zu durchfahrenden Kurvenstrecke ab und kann als proportional zum Inversen des Krümmungsradius angenommen werden, d.h. es gilt $F_{KR} = D \cdot \frac{1}{r}$ mit einer Konstanten D , welche der einschlägigen Literatur entnommen werden kann (Iwnicki 2006). Der hierdurch verursachte Energieaufwand berechnet sich dann durch das Integral

$$E_{KR} = \int_{s_1}^{s_2} \frac{D}{r} ds. \tag{11}$$

Wie oben ist hier zu berücksichtigen, dass der Krümmungsradius r vom Ort s abhängt. Eine gerade Strecke wird durch einen unendlichen Krümmungsradius beschrieben, so dass der entsprechende Integralanteil gerade zu null verschwindet. Auch für die zwei weiteren innerhalb unserer Untersuchungen verwendeten Kurven – Kreisbögen und Klothoiden – ist dieses elementar zu integrieren. So ist auf einem Kreissektor der Radius konstant, so dass der Wert sich

$$E_{KR;Kreis} = \int_{s_1}^{s_2} \frac{D}{r} ds = \frac{D}{r} \cdot \int_{s_1}^{s_2} 1 ds = \frac{D}{r} \cdot (s_2 - s_1) \tag{12}$$

ergibt. Auf der Klothoide schließlich steigt der Radius linear vom Anfangswert r_1 an der Position s_1 zum Endwert r_2 bei s_2 an. Somit gilt

$$r = r_2 + (s - s_1) \cdot \frac{r_2 - r_1}{s_2 - s_1} = r_1 + c \cdot (s - s_1) \quad (13)$$

mit der Konstanten $c := \frac{r_2 - r_1}{s_2 - s_1}$. Dieser Ausdruck kann in das Integral eingesetzt werden

$$\begin{aligned} E_{KR;Klothoide} &= \int_{s_1}^{s_2} \frac{D}{r_1 + c \cdot (s - s_1)} ds = \left[\frac{D}{c} \cdot \ln(r_1 + c \cdot (s - s_1)) \right]_{s_1}^{s_2} \\ &= \frac{D}{c} \cdot \left(\ln(r_1 + c \cdot (s_2 - s_1)) - \ln(r_1) \right) = \frac{D}{c} \cdot \ln \left(1 + \frac{c}{r_1} \cdot (s_2 - s_1) \right). \end{aligned} \quad (14)$$

Somit kann der Kurvenwiderstand in allen hier verwendeten Kurventypen durch elementare Integration berechnet werden. Die auftretenden Radien können dem zuvor beschriebenen Krümmungsprofil entnommen werden.

6.1.2.2. Energiebilanz und Gesamtenergie

Wir haben nun die grundlegenden Einflussfaktoren unseres Energiemodells erfasst: kinetische und potentielle Energie, Reibungseinflüsse generell und im Speziellen den Kurvenwiderstand. Gemäß dem zweiten Newtonschen Gesetz (in vektorieller Form) gilt für einen Körper der Masse m die Grundgleichung $\mathbf{F}_{Ges} = m \cdot \mathbf{a}$, wenn \mathbf{a} die Beschleunigung des Körpers und \mathbf{F}_{Ges} die auf ihn wirkende (beschleunigende) Gesamtkraft angibt. In unserem Modell nehmen wir die Gesamtkraft \mathbf{F}_{Ges} als Summe von vier Einzelkräften an

$$\mathbf{F}_{Ges} = m \cdot \mathbf{a} = \mathbf{F}_B + \mathbf{F}_G + \mathbf{F}_R + \mathbf{F}_{KR}. \quad (15)$$

Hier steht \mathbf{F}_G für die Gewichtskraft, \mathbf{F}_R für die Reibungskraft gemäß der *Davis Equation* und \mathbf{F}_{KR} für den Kurvenwiderstand. Die Kraft \mathbf{F}_B muss von außen (durch den Motor) aufgebracht werden, um den Zug so zu beschleunigen oder zu bremsen, dass er sich aufgrund der resultierenden Gesamtkraft \mathbf{F}_{Ges} gemäß dem vorgegebenen Geschwindigkeitsprofil bewegt. Da die Bewegung eines Zuges durch die Gleise geführt wird, kann man von obiger vektorieller Form zu einer eindimensionalen vereinfachten Bewegungsgleichung übergehen. Dabei legt man fest, dass eine Kraft in Bewegungsrichtung durch ein positives, eine entgegengesetzt gerichtete Kraft durch ein negatives Vorzeichen gekennzeichnet wird. Wenn man zusätzlich beachtet, dass für die Bewegung des Zuges nur der Anteil F_H ⁴¹ der Gewichtskraft in Bewegungsrichtung

⁴¹ Den Anteil F_H der Gewichtskraft in oder entgegen der Bewegungsrichtung bezeichnet man üblicherweise als Hangabtriebskraft.

tung von Bedeutung ist, erhält man.

$$F_{Ges} = m \cdot a = F_B + F_H + F_R + F_{KR}. \quad (16)$$

Gemäß der Festlegung der Vorzeichen bedeutet eine positive Kraft F_B anschaulich, dass der Motor eine beschleunigende Kraft ausüben muss, während eine negative Kraft F_B für eine Bremskraft steht. Generell ergibt sich die Änderung der Energie ΔE eines Körpers durch das Arbeitsintegral $\Delta E = W = \int_{s_1}^{s_2} F ds$. Somit ist die gesamte Energieänderung eines Zugs auf einer Fahrt von s_1 nach s_2 durch das folgende Integral gegeben

$$\Delta E = W = \int_{s_1}^{s_2} F_{Ges} ds = \int_{s_1}^{s_2} (F_B + F_H + F_R + F_{KR}) ds. \quad (17)$$

Für den Zugbetreiber ist dabei der Energieanteil E_{Auf} von Bedeutung, der durch den Motor zur Beschleunigung des Zugs von außen aufgebracht werden muss, denn genau dieser Anteil verursacht für ihn die primären (Energie-)Kosten. Dieser Anteil ist durch das Integral

$$E_{Auf} = \int_{s_1}^{s_2} F_B^+ ds \quad (18)$$

gegeben, wenn F_B^+ den positiven Anteil von F_B angibt, also den Anteil, den der Motor zur Beschleunigung des Zuges aufzubringen hat. Dieser Anteil ist formal folgendermaßen festgelegt

$$F_B^+(s) := \begin{cases} F_B(s), & \text{falls } F_B(s) \geq 0 \\ 0 & \text{falls } F_B(s) < 0. \end{cases} \quad (19)$$

Mit Beziehung (15) ergibt dies

$$F_B^+(s) = \begin{cases} F_{Ges} - F_H - F_R - F_{KR}, & \text{falls } F_{Ges} \geq F_H + F_R + F_{KR} \\ 0, & \text{falls } F_{Ges} < F_H + F_R + F_{KR} \end{cases}$$

bzw.

$$F_B^+(s) = \begin{cases} m \cdot a - F_H - F_R - F_{KR}, & \text{falls } m \cdot a \geq F_H + F_R + F_{KR} \\ 0, & \text{falls } m \cdot a < F_H + F_R + F_{KR}. \end{cases} \quad (20)$$

Bereiche mit einem negativen Wert von F_B müssen bei der Berechnung von E_{Auf} explizit unberücksichtigt bleiben, da ein Bremsvorgang für den Zugbetreiber keinen Energieaufwand bedeutet, sondern etwa bei der Verwendung mechanischer Bremsen ausschließlich in deren Erwärmung resultiert, d.h. die Bewegungsenergie wird als Wärme an die Bremsen abgeführt. Um E_{Auf} konkret zu bestimmen, müssen somit die Bereiche entlang des Wegs bestimmt wer-

den, in denen $F_B^+(s)$ einen von Null verschiedenen Wert aufweist, und dann entsprechend durch das Integral aufsummiert werden.

In der prototypischen Softwarekomponente zur Umsetzung des Modells berechnen wir den Energieanteil E_{Auf} (alternativ) durch eine diskrete Energiebilanzierung, welche eine ausreichende Näherung an das Integral (18) darstellt. Diese Bilanzierung basiert auf den Gleichungen, die in Abschnitt 6.1.2.1 hergeleitet wurden, und ist aufgrund der vorgegebenen Werte und Profile einfach und insbesondere schnell durchzuführen. Außerdem erhält man so zugleich die Werte der einzelnen in 6.1.2.1 definierten Energiebeiträge, die dem Ingenieur die Interpretation des Gesamtergebnisses erleichtern (siehe auch 6.1.4).

Zunächst werden die verschiedenen Eingangsprofile (3) auf Basis einer gemeinsamen Stationierung diskretisiert, d.h. gemäß einer gemeinsamen Stationierung werden die zugehörigen Geschwindigkeits-, Höhen- und Krümmungswerte errechnet:

- Geschwindigkeitsprofil $v := v(s) \rightarrow \{(s_1, v_1), (s_2, v_2), \dots, (s_n, v_n)\}$,
- Höhenprofil $h := h(s) \rightarrow \{(s_1, h_1), (s_2, h_2), \dots, (s_n, h_n)\}$,
- Krümmungsprofil $c := c(s) \rightarrow \{(s_1, c_1), (s_2, c_2), \dots, (s_n, c_n)\}$.

Für alle Teilintervalle $[s_i; s_{i+1}]$ mit $i \in \{1, 2, \dots, n - 1\}$ wird dann mit Hilfe der in Kapitel 6.1.2.1 präsentierten energetischen Größen die Energiebilanz berechnet, d.h. die einzelnen Energieanteile werden bestimmt und aufsummiert, insbesondere wird so das Vorzeichen der resultierenden Summe deutlich. Ein positives Ergebnis der Summe trägt dann gerade zu E_{Auf} bei, ein negatives Ergebnis liefert keinen Beitrag, könnte aber zumindest teilweise durch entsprechende Vorrichtungen zur Energierückgewinnung genutzt werden. Eine geeignete Komponente zur Energierückgewinnung kann in die prototypische Software-Architektur ohne größeren Aufwand integriert werden. Grundlage hierfür ist ihre modulare Architektur, die im folgenden Kapitel kurz beschrieben wird. Die konkrete Summation erfolgt unter Berücksichtigung der Vorzeichen, die sich bei korrekter Anwendung der Energiedefinitionen in Kapitel 6.1.2.1 ergeben und in Tabelle 4 detailliert dargestellt sind.

Sind etwa durch h_i und v_i die Höhe bzw. die Geschwindigkeit an der i -ten Stationierungsstelle gegeben, so wird der Verbrauch pro Masseneinheit zwischen der i -ten und $i + 1$ -ten Stationierungsstelle durch das Energiemodul folgendermaßen berechnet:

$$\Delta E_{i,i+1} = \text{Maximum} \left(0 \left| g \cdot (h_{i+1} - h_i) + \frac{1}{2} \cdot (v_{i+1}^2 - v_i^2) + R_{i,i+1} + KR_{i,i+1} \right. \right) \quad (21)$$

Dabei gibt $R_{i,i+1}$ den Reibungsanteil und $KR_{i,i+1}$ den Kurvenwiderstandsanteil im entsprechenden Abschnitt an. Die angegebene Bildung des Maximums entspricht gerade der Bestimmung des Kraftanteils $F_B^+(s)$. Die Gesamtenergie (pro Masseneinheit) ergibt sich dann bei $n + 1$ Stationierungswerten, welche von 0 bis n nummeriert sind und daher $n - 1$ verschiedene Abschnitte definieren, zu

Einflussart	Vorzeichen	Anschauliche Interpretation
Potentielle Energie	+	Fahrt aufwärts: Dem Zug muss Energie zugeführt werden. Bremsst der Zug gleichzeitig, so kann kinetische Energie in potentielle Energie umgewandelt werden.
Potentielle Energie	-	Fahrt abwärts: Es wird Energie frei, die etwa zum Beschleunigen oder auch zum Überwinden von Reibungswiderständen genutzt werden kann. Ist eine konstante Geschwindigkeit durch das Geschwindigkeitsprofil vorgegeben und ist die benötigte Reibungsenergie bereits aufgebracht, so geht überschüssige Energie verloren, wenn keine weiteren Mechanismen zur Energierückgewinnung verwendet werden.
Kinetische Energie	+	Der Zug beschleunigt: Es muss Energie zugeführt werden. Möglichweise kann die benötigte Energie zumindest teilweise aus der potentiellen Energie bei einer Abwärtsfahrt gewonnen werden.
Kinetische Energie	-	Der Zug wird gebremst: Prinzipiell wird Energie frei (siehe auch Punkt 2).
Reibungseinflüsse	+	Das Vorzeichen ist stets positiv, denn es ist stets Energie zuzuführen.
Kurvenwiderstand	+	Das Vorzeichen ist stets positiv, denn es ist stets Energie zuzuführen.

Tabelle 4: Vorzeichen der einzelnen energetischen Einflussfaktoren

$$E_{ges} = \sum_{i=1}^{n-1} \Delta E_{i,i+1}. \quad (22)$$

Es bleibt anzumerken, dass die Genauigkeit der gewählten Berechnungsmethode natürlich von der Feinheit der Diskretisierung abhängt. Praxisbeispiele zeigen, dass ein Abstand $\Delta s_{i,i+1} = s_{i+1} - s_i$ von etwa einem halben Meter in weiten Bereichen ausreicht. Für diesen Fall wollen wir noch den maximalen Fehler für die Höhen- und die Geschwindigkeitsänderung abschätzen: Bei einer maximal zulässigen Steigung von etwa 4,5% bedeutet dies eine maximale Höhenänderung von 0,0225m zwischen zwei Stationierungspunkten und bei einer

maximal zulässigen Beschleunigung von etwa $0,15g$ und bei einer Ausgangsgeschwindigkeit von minimal $v_0 = 1,0 \frac{m}{s}$ eine maximale Geschwindigkeitsänderung von $\Delta v \approx 1,5 \frac{m}{s}$; denn es gilt folgende Abschätzung

$$v^2 - v_0^2 = 2 \cdot (0,15 \cdot g) \cdot \Delta s \Leftrightarrow (v - v_0)(v + v_0) = 0,3 \cdot g \cdot \Delta s$$

$$\Leftrightarrow \Delta v = \frac{0,3 \cdot g \cdot \Delta s}{v + v_0} \leq \frac{0,3 \cdot g \cdot \Delta s}{1,0 \frac{m}{s}} \approx 1,5 \frac{m}{s}.$$

Je größer die Ausgangsgeschwindigkeit, desto geringer ist hier natürlich der gemachte Fehler, so dass sich bei einer Geschwindigkeit v_0 von $v_0 = 10 \frac{m}{s}$ und einer maximal zulässigen Beschleunigung von etwa $0,15g$ ein maximaler Fehler von $\approx 0,15 \frac{m}{s}$ ergibt.

6.1.3. Eine Architektur zur Integration des physikalischen Modells

Bei der prototypischen Umsetzung des physikalischen Modells wurde eine modulare Architektur konzipiert: Den verschiedenen energetischen Eingangsgrößen wird, wie in Abbildung 53 dargestellt, jeweils eine eigene Klasse zur Berechnung (*Calculator*) zugeordnet, so dass später eine separate sowie vergleichende Analyse der einzelnen Einflussfaktoren möglich wird. Diese separate Berechnung ist die Grundlage für eine Interpretation durch den Ingenieur, die ein Verständnis für den Einfluss des Trassenverlaufs, insbesondere des Höhenprofils überhaupt erst ermöglicht.

Alle Klassen implementieren das *IEnergyCalculator*-Interface und verfügen über die für eine spezielle Energiekomponente benötigten Kenntnisse. Instanzen dieser Klassen werden innerhalb einer *TotalEnergyCalculator*-Klasse aggregiert, welche die konkrete Aufgabe der korrekten Summation übernimmt, wie sie gerade zuvor beschrieben wurde.

Die modulare Architektur der Berechnungsklassen hat zudem den entscheidenden Vorteil, einzelne Berechnungskomponenten zu einem späteren Zeitpunkt durch neuere Komponenten ersetzen zu können, ohne dabei andere Komponenten in ihrer Funktionalität zu beeinflussen. So kann beispielsweise im Fall besserer Kenntnisse über Reibungseinflüsse oder für die Berücksichtigung der Energierückgewinnung einfach eine neue Klasse vom Typ *ICalculator* hinzugefügt werden.

Die resultierende Bibliothek wurde als eigene Komponente in die Plattform integriert und wird hier als *Energie-Modell-Management-Komponente* (EMMK) bezeichnet. In Abbildung 54 sind die drei zentralen Komponenten PMMK, AMMK und EMMK sowie ihre Interaktion schematisch dargestellt. Die PMMK bezieht die Trasseninformationen als Grundlage der darauf basierenden Infrastrukturbauwerke von der AMMK. Änderungen an der zugrunde lie-

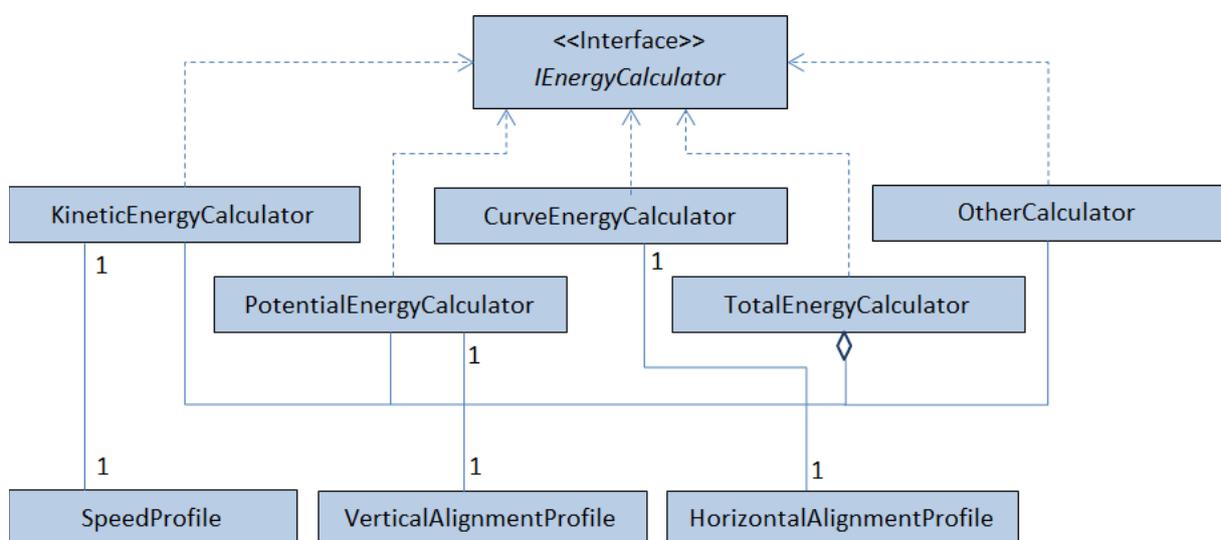


Abbildung 53: Modulare Architektur für die Berechnung der einzelnen energetischen Komponenten

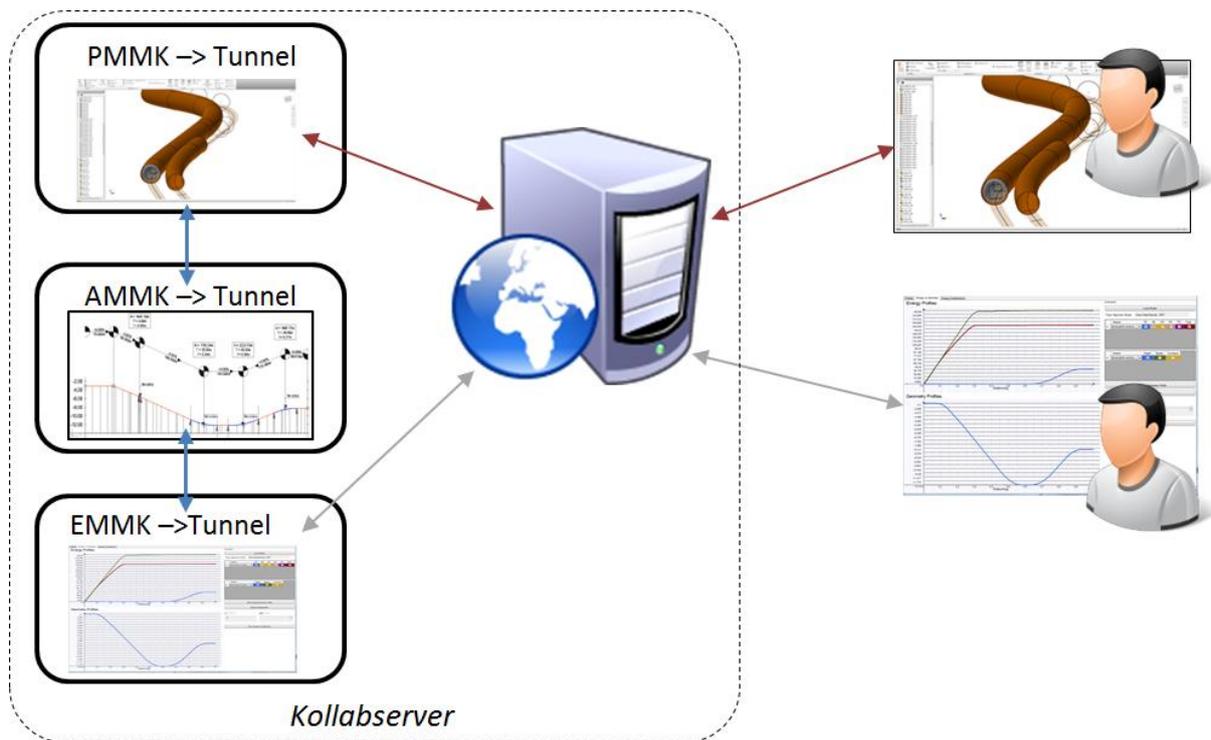


Abbildung 54: Die Komponenten für die Verwaltung von prozeduralen Modellen (PMMK) und Trassenmodellen (AMMK) sowie zur Energieberechnung (EMMK) bilden zentrale Module der Kollaborationsplattform.

genden Trasse werden der PMMK von der AMMK unmittelbar mitgeteilt, so dass die PMMK automatisch die abhängigen Modelle anpassen kann. Die EMMK bezieht die zur Energieberechnung benötigten Trassendaten, also Höhen- und Krümmungsprofil, direkt von der AMMK. Diese wiederum informiert die EMMK automatisch über Änderungen in der Trassenführung, so dass die EMMK eine Neuberechnung des resultierenden Energieverbrauchs starten kann.

6.1.4. Darstellung der Simulationsresultate

Nachdem wir die Grundlagen des physikalischen Modells zur Energiesimulation und die Integration in die Kollaborationsplattform besprochen haben, werden wir uns nun kurz der Darstellung der Simulationsresultate widmen.

Zunächst ist natürlich der Gesamtenergieverbrauch (pro Kilogramm bewegter Masse) eine entscheidende Größe. Von ebenso großer Bedeutung ist aber sicherlich die Frage, wie sich dieser Gesamtenergieverbrauch ergibt. Um den Ingenieur bei der Beantwortung dieser oder ähnlicher Fragen zu unterstützen, werden verschiedene Möglichkeiten für die Visualisierung der Ergebnisse angeboten.

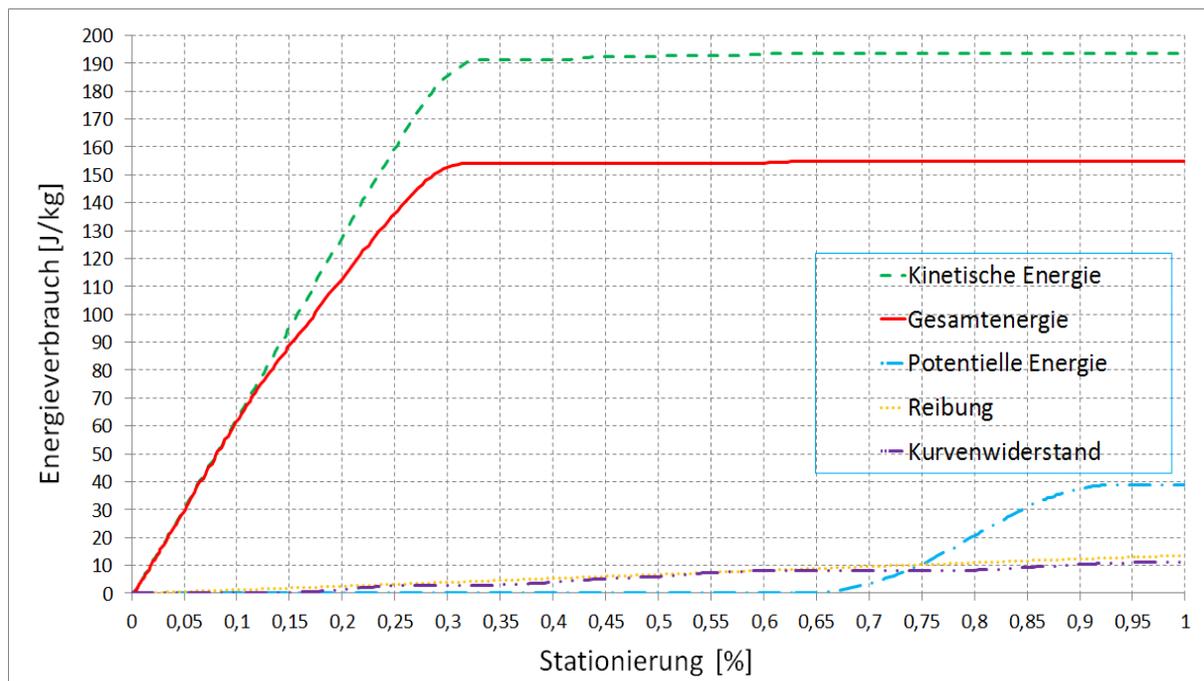


Abbildung 55: Verschiedene Visualisierungsmöglichkeiten erlauben dem Benutzer eine Interpretation der Ergebnisse. Dazu sind hier zunächst die einzelnen Anteile der verschiedenen Energien als Kurven dargestellt.

Eine erste Visualisierungsmöglichkeit erlaubt es, den Aufwand für die einzelnen Energieanteile kumuliert in Abhängigkeit der Stationierung zu betrachten. Die Grundlage dafür bildet die zuvor besprochene modulare Architektur, die diese separate Berechnung zulässt. Diese einzelnen Anteile sind in Abbildung 55 dargestellt: Die fünf hier dargestellten Kurven entsprechen den positiven Anteilen der einzelnen Energieterme, also den Anteilen, die (theoretisch) von außen aufzubringen sind. In Abbildung 55 sind dies die kinetische Energie (grün – gestrichelt), die Gesamtenergie (rot – durchgezogen), die potentielle Energie (blau – strich/punkt) und die zwei separat berechneten Reibungseinflüsse gemäß der *Davis Equation* (gelb – gepunktet) und dem Kurvenwiderstand (violett – punkt/punkt/strich).

Um das mögliche Vorgehen bei der Interpretation durch einen planenden Ingenieur zu zeigen, wollen wir dieses Beispiel ein wenig näher untersuchen. In diesem mag es zunächst verwunderlich erscheinen, dass der vom Zugmotor zu leistende Gesamtenergieaufwand niedriger ist als der Aufwand für die kinetische Energie (Abbildung 55). Dies ergibt sich aus der Tatsache, dass ein Teil der zuzuführenden kinetischen Energie aus der Abnahme der potentiellen Energie des Zuges gewonnen werden kann. Dazu beobachtet man in Abbildung 55 im Detail die rote, durchgezogene Kurve und die grüne, gestrichelte Kurve. Zunächst steigen beide Kurven bis etwa zehn Prozent der Gesamtstreckenlänge nahezu identisch an, während ab etwa zehn Prozent die grüne Kurve steiler ansteigt.

Dies folgt aus der Tatsache, dass zunächst die komplette kinetische Energie von außen zuzuführen ist, während ab etwa zehn Prozent der Gesamtstreckenlänge die Beschleunigung durch

die Fahrt abwärts unterstützt wird, also die Abnahme der Höhenenergie des Zuges zu seiner Beschleunigung genutzt werden kann.

Um diese Fragestellung besser interpretieren zu können, wurde eine zweite Visualisierungsmöglichkeit integriert, die zusätzlich die Geometrie- und Geschwindigkeitsprofile darstellt. So sieht man in Abbildung 56 und Abbildung 57 die Energieprofile zusammen mit zugehörigem Höhen- bzw. Geschwindigkeitsprofil. In unserem Beispiel lässt sich konkret erkennen, dass die im letzten Drittel für die Fahrt bergauf bereitzustellende potentielle Energie vollständig aus der kinetischen Energie gewonnen werden kann, wenn man dazu insbesondere das Geschwindigkeitsprofil berücksichtigt.

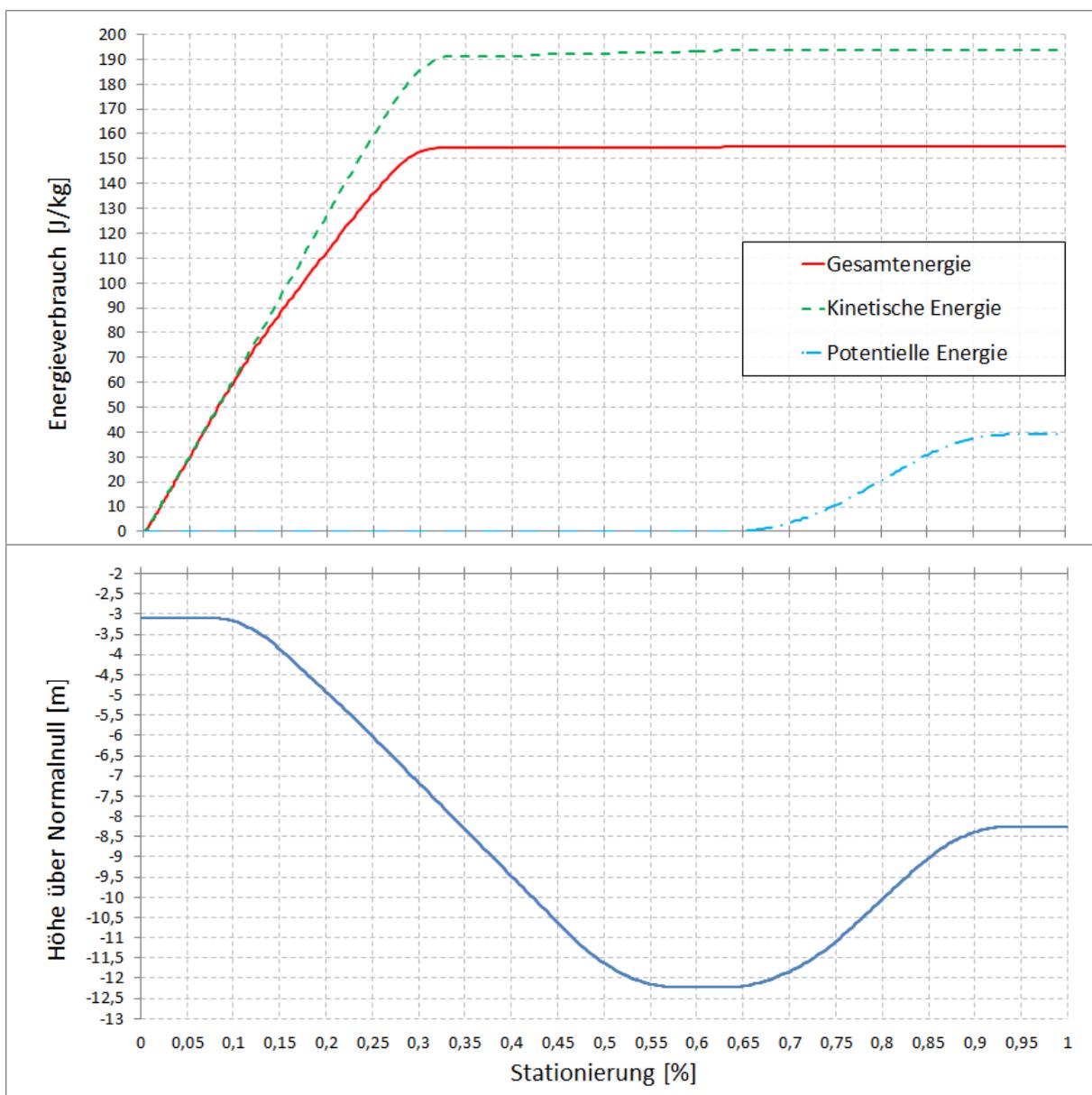


Abbildung 56: Gleichzeitige Visualisierung von Energie- und Höhenprofil

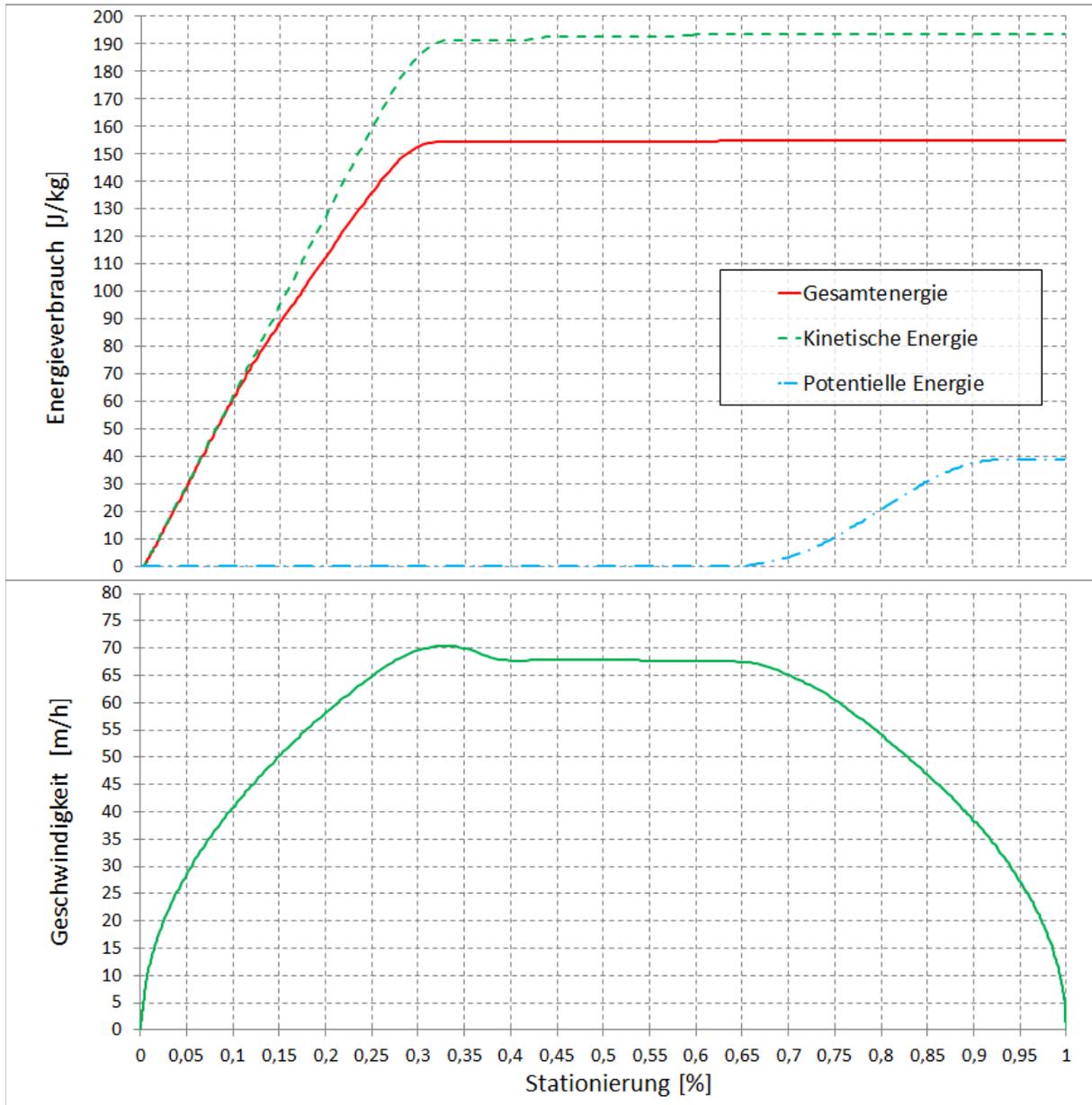


Abbildung 57: Gleichzeitige Visualisierung von Energie- und Geschwindigkeitsprofil

Insgesamt werden so den planenden Ingenieuren nicht nur finale Zahlenwerte hinsichtlich des Energieverbrauchs geboten, sondern zusätzlich Möglichkeiten zur Visualisierung, welche die Ursachen für das Zustandekommen des Energieverbrauchs verständlicher machen. Dies ist dann die Grundlage für eine interaktive Verbesserung der Energieeffizienz: So kann der Ingenieur in dieser Oberfläche manuell etwa das Höhenprofil (a priori) variieren und erhält dabei in Echtzeit eine direkte Rückmeldung und Darstellung der resultierenden Änderungen beim Energieverbrauch.

6.1.5. Validierung des Modells anhand von Messdaten der Metro Bilbao

Um den Einsatz des beschriebenen Energiemodells zu rechtfertigen, muss dessen Validität geprüft werden. Dazu ist der Vergleich von mit diesem Modell berechneten und gemessenen Werten die beste Möglichkeit. Im Rahmen dieser Forschungsarbeit konnte ein Kontakt mit den Betreibern der Metro Bilbao geknüpft werden, welche uns freundlicherweise sehr detaillierte Messdaten für Teilstrecken des dortigen U-Bahnnetzes zur Verfügung stellten. Diese Daten der Metro Bilbao wurden zur Validierung herangezogen; sie umfassen:

- die Trassierungsdaten über sämtliche Abschnitte des dortigen U-Bahnnetzes, also Lage- und Höhenpläne, so dass Höhen- und Krümmungsprofile extrahiert werden konnten,
- detaillierte Angaben zu den verwendeten U-Bahnzügen, insbesondere die Massen der verschiedenen Zugtypen und ihre Zusammenstellung auf verschiedenen Streckenabschnitten,

Abschnitt	Gemessener Wert [J/kg]	Berechneter Wert [J/kg]	Verhältnis bei Wirkungsgrad 93%	Verhältnis bei Wirkungsgrad 85%
Casco Viejo → San Tutxu	421,2	401,4	95,30%	104,3%
San Tutxu → Casco Viejo	133,7	120,0	89,75%	98,20%
Casco Viejo → Abando	177,2	158,1	89,22%	97,62%
Abando → Casco Viejo	178,4	158,8	89,01%	97,39%
Etxebarri → Bolueta	189,4	170,0	89,75%	97,14%
Bolueta → Etxebarri	242,7	223,6	92,13%	100,8%
Indautxu → San Mames	166,0	157,9	95,12%	104,1%
San Mames → Indautxu	161,0	153,4	95,28%	104,3%
Bolueta → Basarrate	209,1	195,7	93,36%	102,4%
Basarrate → Bolueta	181,2	164,9	91,00%	99,57%
San Inazio → Lutzana	343,5	326,0	94,90%	103,8%
Lutzana → San Inazio	227,4	215,6	94,81%	103,7%

Tabelle 5: Vergleich von Messdaten der Metro Bilbao und berechneten Werten

- Messdaten bezüglich der anliegenden Spannung an der Oberleitung und der Eingangsstromstärke,
- Messdaten der tatsächlich gefahrenen Geschwindigkeit in Abhängigkeit der Stationierung, also gerade das Geschwindigkeitsprofil,
- die Fahrgastzahlen auf bestimmten Streckenabschnitten während der Energiemessungen.

Aus der anliegenden Spannung und Stromstärke kann der Energieverbrauch einfach als Produkt $E = U \cdot I$ bestimmt werden (Demtröder 2013). Da laut den Angaben der Betreiber etwa sieben Prozent des Energieverbrauch aus der Klimatisierung und Beleuchtung der Züge resultieren, müssen die entsprechenden Produktwerte gerade um diese sieben Prozent reduziert werden, um einen sinnvollen Vergleich mit den berechneten Werten zu ermöglichen. Hinzu kommt, dass Motor und Getriebe die Eingangsenergie nicht vollständig an den Antrieb weitergeben. Die entsprechenden Wirkungsgrade waren nicht bekannt und mussten deshalb abgeschätzt werden. Ein Wert zwischen 85% und 93% scheint gemäß der entsprechenden Fachliteratur eine geeignete Näherung zu sein (Klimentew et al. 2012). In Tabelle 5 sind die Vergleichsdaten aufgeführt. Die erste Spalte enthält die Namen von Start- und Endbahnhof eines Abschnitts, Spalte zwei den gemessenen Verbrauchswert in $\left[\frac{J}{kg}\right]$, wobei die oben erwähnten sieben Prozent infolge von Klimatisierung und Beleuchtung bereits abgezogen wurden. Spalte drei zeigt die berechneten Werte wiederum in $\left[\frac{J}{kg}\right]$, Spalte vier und fünf geben das Verhältnis von berechnetem zu gemessenem Wert bei einem Wirkungsgrad von $\epsilon = 93\%$ bzw. $\epsilon = 85\%$ an. Insgesamt ergaben sich bei einem angenommenen Wirkungsgrad von $\epsilon = 93\%$ ein Genauigkeitsbereich von $[89\%; 95\%]$ und bei einem geschätzten Wirkungsgrad von $\epsilon = 85\%$ ein Bereich von $[97\%; 104\%]$. Unter den oben angegebenen Erwartungen an die Genauigkeiten des Modells fügen sich diese Werte in diesen Erwartungsbereich ein.

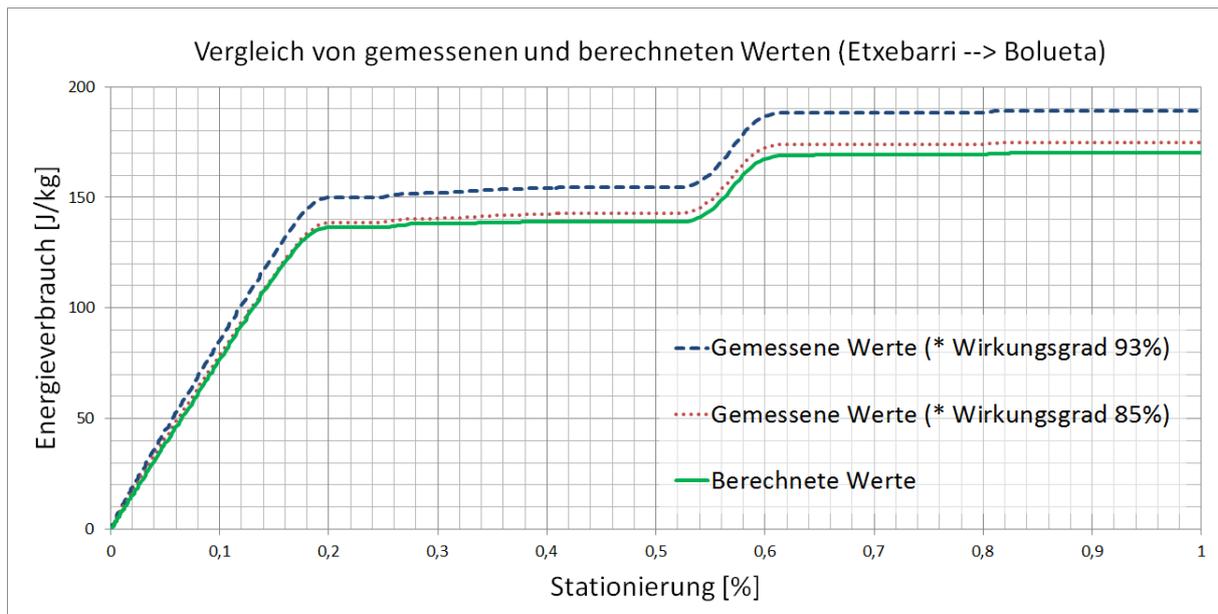


Abbildung 58: Gesamtenergieverbrauch in Abhängigkeit der Stationierung auf dem Abschnitt Etxebarri nach Bolueta

Neben dem Vergleich der berechneten Gesamtverbrauchswerte wurden auch die Kurvenverläufe hinsichtlich der Entwicklung des kumulativen Energieverbrauchs in Abhängigkeit der Stationierung gegenübergestellt. Dies kann aus Abbildung 58 sowie aus Abbildung 59 entnommen werden.

Hier sind der kumulative Energieverbrauch auf dem Streckenabschnitt zwischen Etxebarri und Bolueta sowie auf dem Streckenabschnitt zwischen Lutxana und San Inazio gegenüber

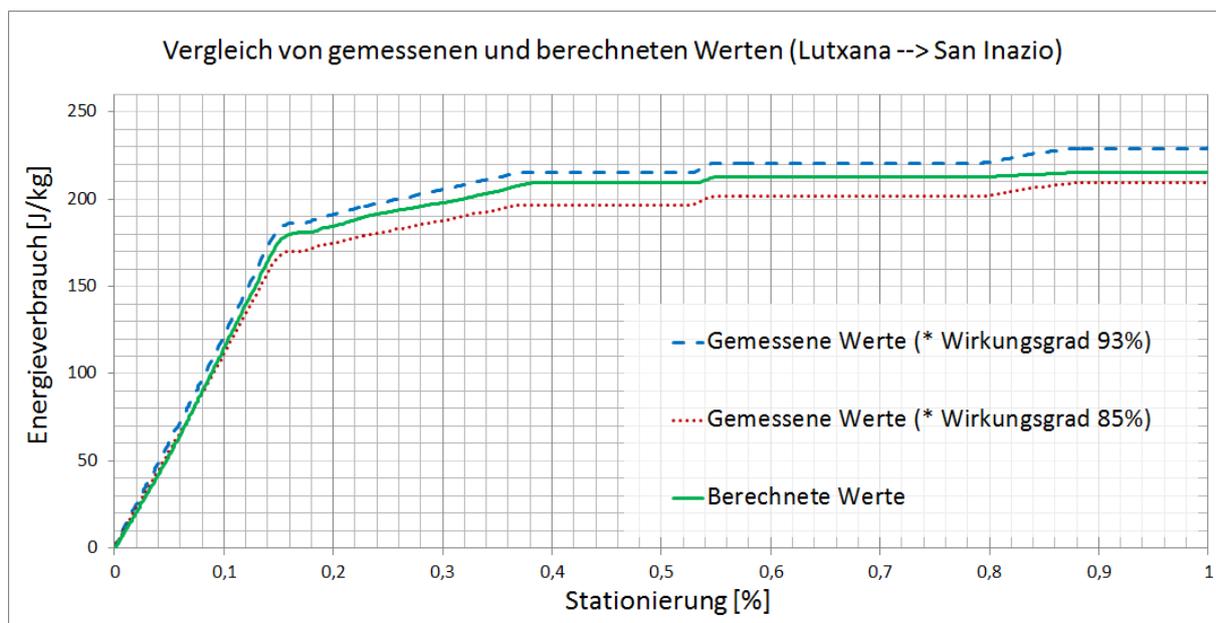


Abbildung 59: Gesamtenergieverbrauch in Abhängigkeit der Stationierung auf dem Teilabschnitt Lutxana nach San Inazio.

der Stationierung aufgetragen. Die blaue gestrichelte und rote gepunktete Kurve stellen die gemessenen Werte dar, multipliziert mit einem angenommenen Wirkungsgrad von $\epsilon = 93\%$ bzw. $\epsilon = 85\%$, die grüne durchgezogene Kurve die durch die Simulation berechneten Werte. Hier kann man beobachten, dass bei einem Absenken des geschätzten Wirkungsgrades von $\epsilon = 93\%$ in Richtung $\epsilon = 85\%$ die Kurven von gemessenen und berechneten Werten etwas besser übereinstimmen. Somit scheint ein Wirkungsgrad von knapp $\epsilon = 90\%$ tatsächlich angemessen. Betrachtet man den Verlauf in Abbildung 59 genauer, so fällt auf, dass die berechneten Werte zum Ende hin die Messwerte etwas stärker als erwartet unterschätzen. Dies mag sich daraus erklären, dass der Zug auf dieser Strecke zunächst den Tunnel verlässt und etwa bei 85% der Strecke wieder in einen Tunnel hineinfährt. Eine derartige Situation, die zu wechselnden Luftwiderstandsbeiwerten führt, wurde allerdings bisher nicht im Modell berücksichtigt und bietet sicherlich Verbesserungspotential, um die erzielte Genauigkeit zu erhöhen.

6.1.6. Manuelle Optimierung der Energieeffizienz einer Trasse

Das vorgestellte Modell dient in erster Linie dazu, dem Planer Änderungen des Energieverbrauchs infolge von Änderungen im Trassenverlauf zu vergegenwärtigen. Es beinhaltet zunächst keine Funktionalitäten für eine automatisierte Optimierung der Trasse, auch wenn das zugrunde liegende Modell und insbesondere die prototypisch entwickelte Software dazu eine gute Ausgangsbasis bilden. Da die entwickelte Simulationskomponente über ihre graphische Oberfläche aber zumindest die Möglichkeit bietet, durch den konkreten Vergleich mehrerer unterschiedlicher Varianten einer Trasse einen reduzierten Energieverbrauch zu finden, sollen hier zwei so entstandene Beispiele für eine manuelle Optimierung präsentiert werden. Diese belegen, mit wie geringem Aufwand bezüglich der Veränderung des Trassenverlaufs eine beachtliche Reduzierung des Energieverbrauchs zu erreichen ist. Diese Beispiele basieren wiederum auf Daten der Metro Bilbao. Im folgenden Kapitel werden wir diese Ideen aufgreifen, um abschließend zu zeigen, wie hieraus Möglichkeiten für eine automatisierte Optimierung entwickelt wurden.

Beispiel 1: Horizontale Verschiebung (des Zuges im Bahnhof)

Das erste Beispiel hat als Grundlage den Streckenabschnitt San Tutxu nach Casco Viejo. Das zugehörige Höhenprofil ist in Abbildung 60 dargestellt und weist einen beachtlichen Höhenunterschied von etwa $30,5m$ auf. Der Bahnhof in San Tutxu besitzt eine Bahnsteiglänge von etwa $90m$, während ein verwendeter Metro-Zug mit vier Wagons knapp $80m$ lang ist. Somit ist es möglich, den Zug um eine Strecke von bis zu gut zehn Metern im Bahnhof zu verschieben, ohne das Ein- und Aussteigen der Fahrgäste zu behindern. So ergeben sich zwei extreme Startpositionen, die in Abbildung 60 mit x_a und x_b markiert sind.

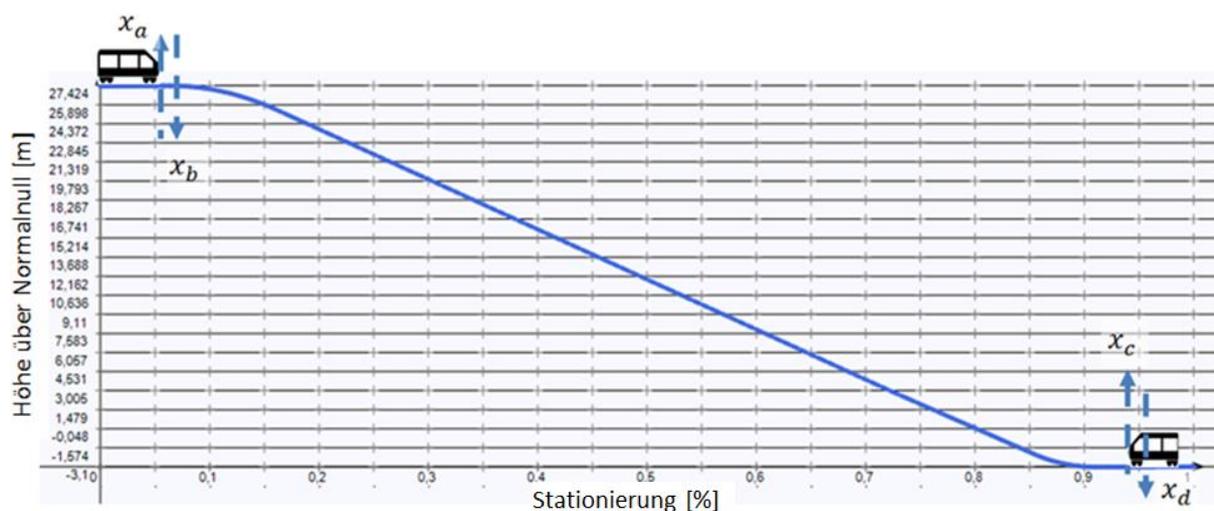


Abbildung 60: Höhenprofil der Strecke von San Tutxu nach Casco Viejo in Bilbao

Die berechneten Werte für den Energieverbrauch entsprechend den beiden Startpositionen x_a und x_b sind der Tabelle 6 zu entnehmen.

Startposition	Energieverbrauch [J/kg]
x_a	111,6
x_b	106,5

Tabelle 6: Energieverbrauch bei verschiedenen Startpositionen (San Tutxu nach Casco Viejo)

Das heißt: Ein bloßes Verschieben des Zuges an den rechten Rand des Bahnsteigs reduziert den Energieverbrauch um etwa 4,8%. Die Ursache für den hier niedrigeren Verbrauch ist einfach zu erkennen: Je weiter rechts der Zug startet, desto früher fährt er in der Beschleunigungsphase abwärts, so dass die Schwerkraft einen Teil der beschleunigenden Kraft und somit der benötigten kinetischen Energie liefern kann. Betrachten wir nun umgekehrt (Abbildung 60) die Fahrt von Casco Viejo nach San Tutxu, so erhalten wir für die Startpositionen x_c und x_d die Werte, die in Tabelle 7 dargestellt sind.

Wiederum ergibt ein einfaches Verschieben des Zuges im Bahnhof eine Energieersparnis von gut einem Prozent. Die Interpretation ist in diesem Fall nicht ganz so einfach und ergibt sich wie folgt: Startet der Zug weiter rechts, so findet ein größerer Teil der Beschleunigungsphase in einem flachen Bereich statt, der Zug erreicht dann eher seine Höchstgeschwindigkeit, und ein prozentual größerer Anteil der zuzuführenden potentiellen Energie kann aus der Energie beim Abbremsen des Zuges gewonnen werden.

Startposition	Energieverbrauch [J/kg]
x_c	390,8
x_d	386,0

Tabelle 7: Energieverbrauch bei verschiedenen Startpositionen (Casco Viejo nach San Tutxu)

Beispiel 2: Vertikale Verschiebung (Tiefergraben des Tunnelstollens)

Grundlage des zweiten Beispiels ist der Streckenabschnitt Casco Viejo nach Abando, wiederum entnommen der Metro Bilbao. Hier startet der Zug in Casco Viejo auf einer Höhe von $-3,1m$ und fährt dann abwärts bis auf eine Höhe von $-12,2m$. Schließlich befindet sich der Bahnhof in Abando auf einer Höhe von $-8,7m$ (Abbildung 61).

In einem zweiten möglichen Profil gräbt man den Stollen etwas tiefer bis auf eine Höhe von $-13,5m$, also etwa $1,3m$ tiefer. Bei Verwendung einer Tunnelbohrmaschine sollte das

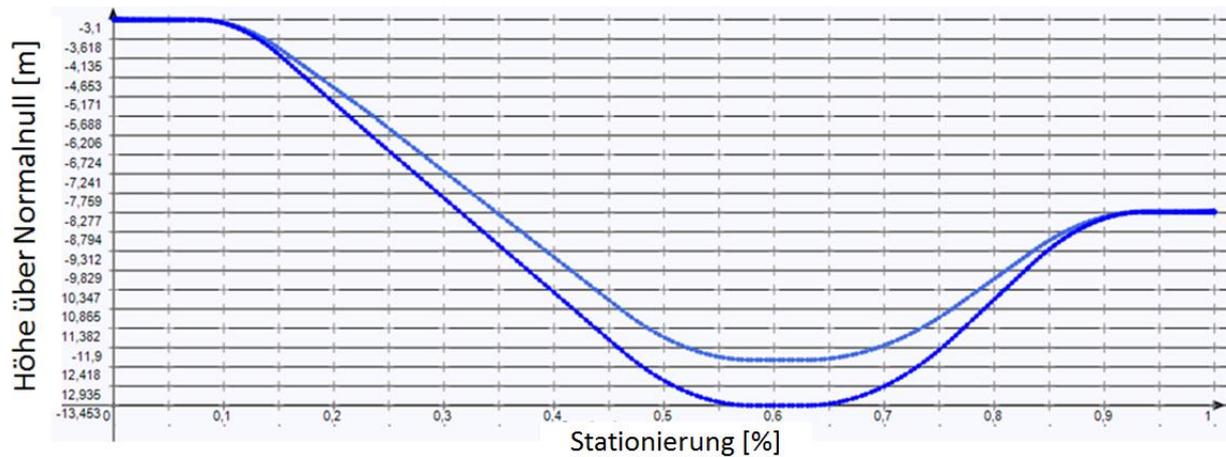


Abbildung 61: „Tiefergraben“ des Tunnels – zwei mögliche Höhenprofile

keine nennenswerten Mehrkosten verursachen. Beide Höhenprofile sind in Abbildung 61 dargestellt. Berechnet man nun für beide Profile den Energieverbrauch, so ergeben sich folgende Verbrauchswerte, die der Tabelle 8 zu entnehmen sind. Somit ergibt ein einfaches „Tiefergraben“ eine Energieersparnis von etwa 4,8%. Die Ursache ist einfach zu finden. Aufgrund des steileren Beginns ist der beschleunigende Anteil der Gewichtskraft größer, so dass ein größerer Anteil der kinetischen Energie mit ihrer Hilfe erreicht werden kann.

Maximale Tiefe [m]	Energieverbrauch [J/kg]
-12,2	155,0
-13,5	147,9

Tabelle 8: Energieverbrauch bei verschiedenen Höhenprofilen Casco Viejo → Abando

Prinzipiell kann man nun diesen Vorgang iterieren und mehrfach tiefergraben. Interessant ist dabei, dass der Energieverbrauch zunächst immer weiter sinkt, bis er ab einem bestimmten Wert wieder zu steigen beginnt. Das ist dann der Fall, wenn die kinetische Energie nicht mehr ausreicht, um bei gegebenem Geschwindigkeitsprofil die benötigte Energie für den Anstieg in Abando aus der kinetischen Energie zu gewinnen. Es bleibt hier allerdings anzumerken, dass der Prozess des Tiefergrabens durch die maximal zulässigen Steigungen von etwa 4,5% begrenzt ist, so dass in der Realität nur ein- oder maximal zweimaliges „Tiefergraben“ möglich gewesen wäre. Wie wir aber zuvor berechnet haben, bedeutet bereits ein ein- oder zweimaliges Tiefergraben eine beachtliche Reduktion des Energieverbrauchs um 5% bzw. 10%. Der Vorgang des mehrmaligen Tiefergrabens ist in Abbildung 62 dargestellt. Hierbei sind in beiden Diagrammen je sechs Kurven (gekennzeichnet mit 1 – 6) dargestellt, welche dem ein-, zwei-, dreimaligen etc. Tiefergraben entsprechen. Die Kurven im oberen Teil der Abbildung geben dabei den Gesamtenergieverbrauch in $\left[\frac{J}{kg}\right]$ an, die im unteren Teil die zugehörigen Höhenprofile.

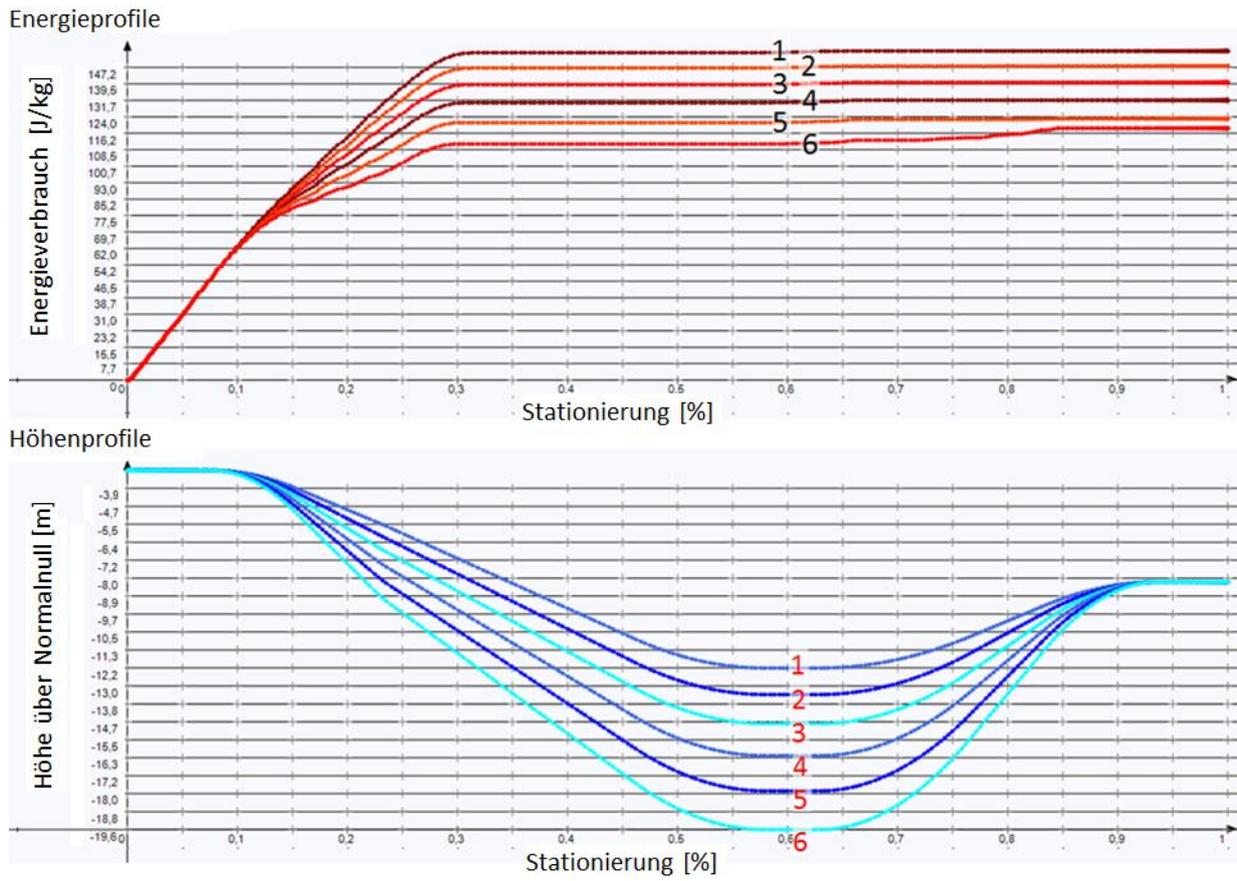


Abbildung 62: Mehrfaches Tiefergraben und (zunächst) stetig sinkender Energieverbrauch; oben der Energieverbrauch in [J/kg], unten die Höhe in [m] über Normalnull.

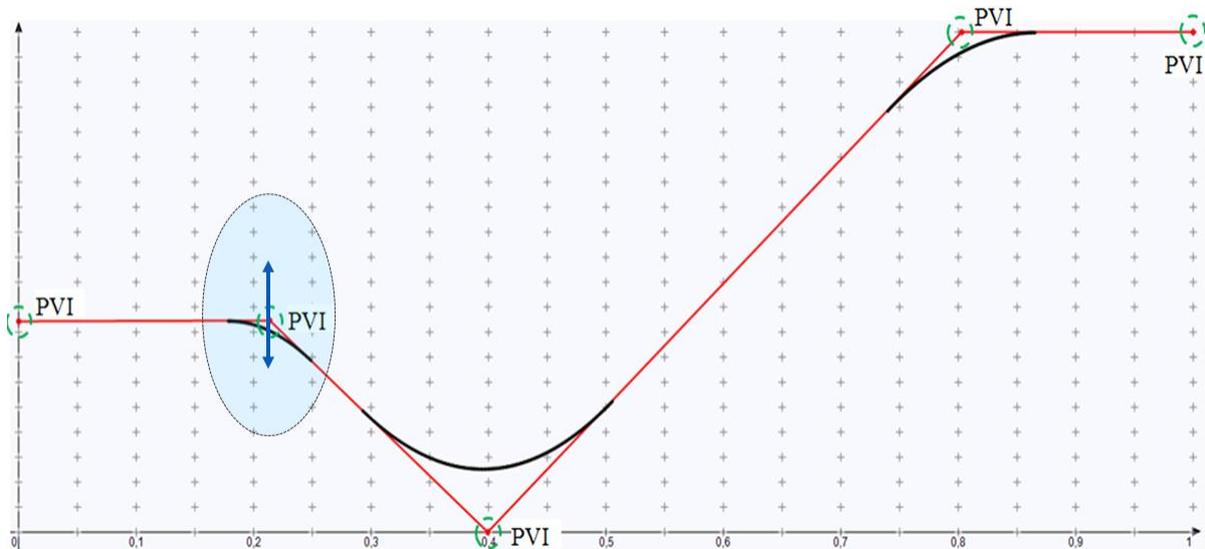


Abbildung 63: Grundlegend für den vorgestellten Optimierungsansatz ist die vertikale Verschiebung einzelner PVI.

6.1.7. Automatisierte Optimierung der Energieeffizienz einer Trasse

6.1.7.1. Automatische Variation des Höhenprofils

Im vorangehenden Abschnitt haben wir gesehen, wie man manuell mit ein wenig Experimentieren unter Verwendung der präsentierten Werkzeuge eine Reduktion des Energieverbrauchs erreichen kann. Die Vorgehensweise des zweiten Beispiels wollen wir nun verallgemeinern und einen automatisierten Ansatz zur Optimierung eines Höhenprofils zwischen zwei Bahnhöfen vorstellen. Prinzipiell haben wir in diesem Beispiel gesehen, wie wir durch eine Streckung des Höhenprofils nach unten den Energieverbrauch reduzieren konnten. Von der etwas allgemeineren Idee einer vertikalen Variation der Höhenprofile lassen wir uns im Folgenden leiten.

Die zentrale Größe zur Erzeugung eines Höhenprofils sind die bereits besprochenen *Points of Vertical Intersection* (PVI, siehe Kapitel 5.1.3), die bei der Erstellung eines Höhenplans zu Beginn vom Ingenieur unter Berücksichtigung gegebener Randbedingungen festgelegt werden. In unserem Ansatz gehen wir entsprechend typischen realen Planungsbedingungen davon aus, dass ihre vertikalen Positionen nicht völlig fixiert sind, sondern einen gewissen Spielraum zulassen (Abbildung 63).

So legt der Ingenieur zum Zweck der Optimierung gemäß unserem Konzept einen konkreten Bereich zur vertikalen Variation der einzelnen PVI fest. Innerhalb dieses Bereichs definiert er zudem eine diskrete Schrittweite, die zu neuen (diskreten) Positionen für die einzelnen (verschobenen) PVI führt. Dabei ist es dem Ingenieur natürlich durch Angabe eines Bereichs der Länge Null möglich, einzelne PVI gemäß äußeren Gegebenheiten zu fixieren. Insbesondere fixieren wir in unserem Ansatz immer den ersten und letzten PVI, um ein bestimmtes Optimierungsproblem zu spezifizieren. Dieses Vorgehen kann man folgendermaßen auf einen

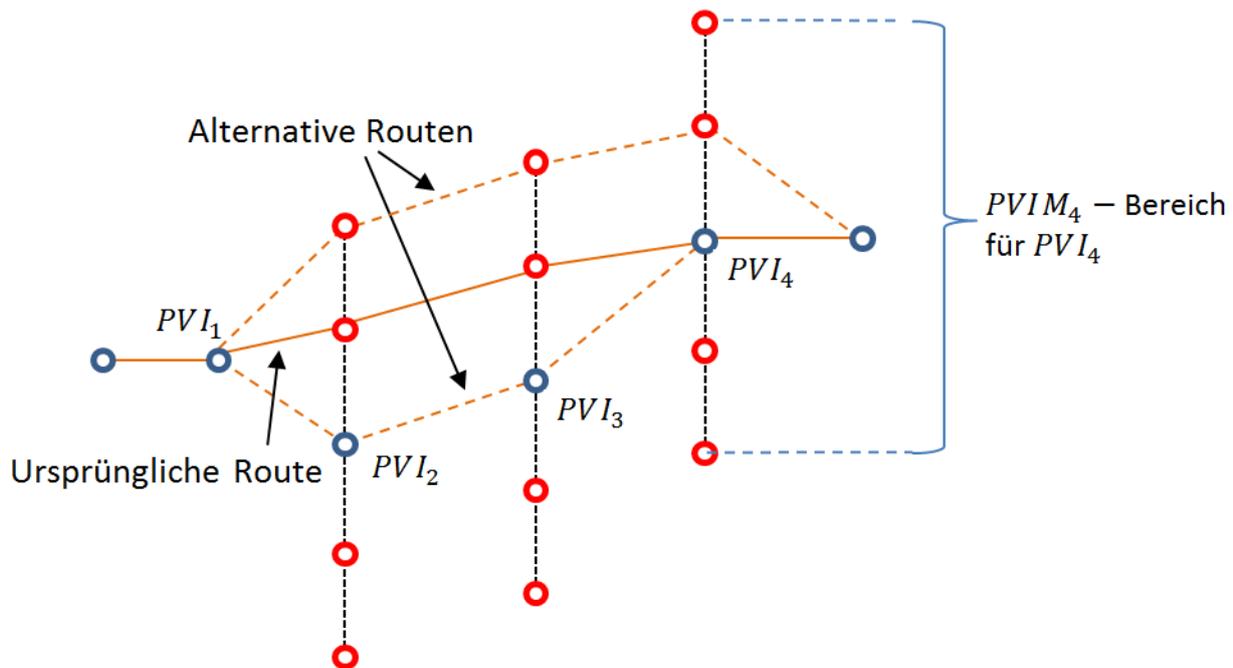


Abbildung 64: Die vertikale Verschiebung von PVI's führt zu alternativen Routen.

ganzen zu optimierenden Streckenabschnitt erweitern: Mit PVI_k bezeichnen wir für $k \in \{1, 2, \dots, n\}$ die einzelnen PVI's eines konkreten Streckenabschnitts, mit $PVIM_k$ die Menge der Punkte, die durch die Verschiebung eines Punkts PVI_k entsteht. Ein neues mögliches Profil erhalten wir nun, indem wir aus jeder Menge $PVIM_k$ für $k \in \{1, 2, \dots, n\}$ genau ein Element P_k wählen, so dass wir insgesamt eine Menge von Punkten $\{P_1, P_2, \dots, P_n\}$ als Grundlage für ein neues Profil erhalten. Die Menge aller so erzeugten Punkte bezeichnen wir im Folgenden mit $PVIM = \bigcup_{k=1}^n PVIM_k$. Die beschriebene Methode ist schematisch in Abbildung 64 dargestellt. Dabei entstehen aus dem ursprünglichen Profil (durchgezogener Linienzug) durch die Verschiebung der PVI-Positionen neue Profile (gestrichelte Linienzüge).

Dabei muss man beachten, dass nicht jedes so erzeugte Profil P_k aus bautechnischen Voraussetzungen als sinnvoll erscheint: So sind im U-Bahnbau etwa eine maximale Steigung sowie ein maximales Gefälle von etwa 4,5% vorgeschrieben. Deshalb werden bei der Generierung neuer Profile automatisch nur Möglichkeiten in Betracht gezogen, die dieser Einschränkung genügen.

Durch die Vorgabe einer diskreten Schrittweite enthält jede Menge $PVIM_k$ endliche viele Elemente, deren Anzahl wir mit m_k bezeichnen. Somit ist die Gesamtzahl N aller (theoretisch) möglichen neuen Profile gerade durch $N = m_1 \cdot m_2 \cdot \dots \cdot m_n$ gegeben, wobei jedes Profil zu einem bestimmten Energieprofil E_k führt. Die Menge aller Energieprofile $\{E_1, E_2, \dots, E_N\}$ enthält zwar möglicherweise sehr viele, aber sicherlich nur endlich viele Elemente. Diese Endlichkeit sichert dann gerade die Existenz eines Minimums, welches wir im Folgenden (näherungsweise) bestimmen wollen. Allerdings erfordert die häufig hohe Anzahl

an Elementen einen erheblichen Rechenaufwand zur Bestimmung des Minimums, zumindest wenn man gemäß einem *Brute Force* Ansatz versucht, das Minimum durch Probieren zu bestimmen, also durch die Berechnung aller Möglichkeiten. Rechenzeiten von mehreren Stunden erwiesen sich in unseren Experimenten als typisch, so dass eine sinnvolle Integration dieses Ansatzes in den Arbeitsprozess des Planers nicht möglich ist. Deshalb beschreiben wir im Folgenden, wie wir mit Hilfe eines Optimierungsverfahrens Näherungen dieses Minimums bestimmen und dabei die Rechenzeit auf einige Minuten oder sogar nur wenige Sekunden reduzieren können.

6.1.7.2. Höhenprofiloptimierung mit Schwarmintelligenz

Im Bereich der Kombinatorischen Optimierung, einem Teilgebiet der Diskreten Mathematik, gilt es, aus einer diskreten, aber möglicherweise sehr großen Menge eine Teilmenge zu bestimmen, die bezüglich einer bestimmten Kostenfunktion optimal ist. Hier sind diese diskrete Menge gerade durch die Menge der möglichen Profile und die Kostenfunktion durch die Berechnung des Energieverbrauchs für die unterschiedlichen Profile gegeben. Für eine detaillierte Einführung in das Feld der Kombinatorischen Optimierung sei der Leser etwa auf (Korte und Vygen 2012) verwiesen. Eine sehr bekannte Fragestellung in diesem Umfeld ist das sogenannte *Traveling Salesman Problem* (TSP): Ein Handlungsreisender muss eine Menge von Städten besuchen, wobei die Entfernungen dieser Städte untereinander vorgegeben sind. Das Problem besteht nun darin, die kürzeste aller möglichen Routen (in einem realistischen Zeitraum) zu bestimmen. Da die Anzahl der möglichen Routen mit der Fakultät der Anzahl der Städte wächst, ist die Berechnung der optimalen Route bereits bei relativ kleinen Anzahlen mit einem dementsprechend hohen Rechenaufwand verbunden (Lawler et al. 1985). Aus Sicht der Informatik wird das TSP in die Klasse der NP-vollständigen Probleme eingeordnet. Die Klasse der NP-vollständigen Probleme ist gerade dadurch definiert, dass für ihre Lösung (zumindest bisher) keine Algorithmen mit polynomialer Komplexität gefunden wurden (Schöning 2008). Zur Berechnung einer Näherungslösung schlugen Dorigo et al. einen sogenannten Ameisenalgorithmus vor. Diesen Typ von Algorithmus benutzen wir zur Optimierung unserer Höhenprofile und wollen seine Grundzüge kurz beschreiben. Eine detaillierte Beschreibung dieses Algorithmus und einiger seiner verschiedenen Varianten findet man etwa bei Dorigo, insbesondere bei (Dorigo et al. 1991, 1992).

Ameisenalgorithmen sind dem Verhalten von Ameisen bei ihrer Futtersuche nachempfunden. Als einzelne Individuen sind Ameisen bei der Futtersuche mehr oder minder hilflos, in der Gegenwart ihrer Artgenossen aber sehr erfolgreich; eine Fähigkeit, die man üblicherweise mit dem Begriff kollektive Intelligenz oder Schwarmintelligenz bezeichnet (Bonabeau et al. 1999, Miller 2007). So wählen Ameisen die Route zur Futterquelle zunächst mehr oder minder willkürlich, hinterlassen aber bei ihrem Weg zur Futterquelle und zurück entlang ihrer Route eine Duftspur (*Pheromone*). Andere Ameisen tendieren dazu, diesen Duftspuren zu folgen. Je

schneller eine Menge von Ameisen von der Futterquelle zurückkehrt, desto intensiver bleibt die von ihnen hinterlassene Duftspur, so dass andere Ameisen dieser bevorzugt folgen und ihrerseits Duftspuren hinterlassen. So verstärkt sich dieser Effekt im Laufe der Zeit immer weiter selbst, bis die Ameisen allmählich einem Weg folgen, der zumindest eine Näherung an die optimale Route darstellt. Dieser Prozess wurde von Dorigo im sogenannten *Ant System* algorithmisch nachempfunden, daher auch der Name Ameisenalgorithmus. Er schlug vor, diesen zum Auffinden von Näherungslösungen für das TSP zu verwenden, und konnte seine Eignung dafür nachweisen (Dorigo et al. 1991, Dorigo 1992). Diese Art von Algorithmus setzen wir zur Höhenprofiloptimierung ein, wobei die Wahl dieses Algorithmus Typs durch die augenscheinliche Ähnlichkeit des TSPs mit dem gegebenen Optimierungsproblem bedingt wurde. Dorigo und andere Forscher schlugen verschiedene Varianten von Ameisenalgorithmen vor. So diskutieren Dorigo et al. etwa die Varianten ANT-Density, ANT-Quantity und ANT-Cycle (Dorigo et al. 1991). Dabei werden insbesondere unterschiedliche Möglichkeiten zur Verteilung des Duftstoffs auf den durch die Ameisen besuchten Wegen diskutiert. So wird etwa vorgeschlagen, auf alle Pfade die gleich große Duftstoffmenge zu verteilen oder eine variable Duftstoffmenge zu verwenden, die von der Länge des Pfades abhängt, etwa indem diese Menge durch die Länge des Pfades dividiert wird. In unserer Implementierung benutzen wir eine ANT-Ellitist-Variante (Negulescu et al. 2008), in der die beste momentan gefundene Route im Laufe der einzelnen Runden zusätzlich betont wird.

In der Adaption des Algorithmus auf das Energieoptimierungsproblem lassen wir virtuelle Ameisen Routen ablaufen, die bis auf die Wahl des Ausgangspunkts nach dem zuvor besprochenen Verfahren erstellt wurden: Wir setzen zunächst eine bestimmte Anzahl m von virtuellen Ameisen auf m verschiedene Punkte aus der Menge PVIM und wählen dann heuristisch für diese Ameisen Routen. Wurde dabei für eine spezielle Ameise etwa ein Ausgangspunkt aus der Menge PVIM _{j} gewählt, so läuft ihre Route auf Punkten aus den Mengen PVIM _{j} , PVIM _{$j+1$} , ..., PVIM _{n} , PVIM _{1} , ..., PVIM _{$j-1$} , PVIM _{j} . Bei der Wahl der Folgepunkte ist zunächst die sogenannte Sichtbarkeit (*visibility*) von Bedeutung. Im ursprünglichen TSP wurden erfolgreich Strategien vorgeschlagen, nähere (sozusagen sichtbare) Städte bei der Wahl der Route im Stil einer Greedy-Heuristik zu bevorzugen (Gerdes et al. 2004). In unserem Ansatz interpretieren wir unter dem Parameter *visibility* den Energieverbrauch zu einem darauffolgenden PVI und bevorzugen so zunächst Wege, die lokal energieeffizient sind. Nachdem eine Route durchlaufen wurde, wird der für sie benötigte Energieverbrauch berechnet und eine Effizienz-Bewertung vorgenommen. Dies entspricht dem *Pheromone*-Parameter oder in der Realität dem Hinterlassen einer Duftspur. Die konkrete Berechnung und den konkreten Einfluss dieser beiden Parameter entnimmt man der Masterarbeit von Ricardo Morelos, der im Rahmen dieser gemeinsamen Forschung im Wesentlichen die konkrete Implementierung des Algorithmus übernommen hat (Morelos 2016). Nachdem alle Ameisen ihre Route beendet haben, startet ein Folgedurchlauf, in welchem die Ameisen nicht mehr ausschließlich heuris-

tisch unter Verwendung des *Visibility*-Parameters, sondern auch unter Berücksichtigung der Bewertung des Vorlaufs durch den *Pheromone*-Parameter ihre Route wählen. Das Verhältnis zwischen dem Einfluss des *Pheromone-Parameters* und des *Visibility-Parameters* wurde wählbar gestaltet. Hierdurch und durch die zufällige Auswahl der Startpunkte der einzelnen Ameisen wird gesichert, dass die Routenwahl noch nicht aufgrund der während des ersten Durchlaufs entstandenen Duftspur völlig festgelegt ist, sondern weiterhin neue Wege exploriert werden. Jede solche Runde wird in diesem Zusammenhang als Zyklus (*Cycle*) bezeichnet; die Anzahl der Zyklen wird vom Ingenieur mit dem Ziel bestimmt, ein ausgewogenes Verhältnis zwischen Gesamtlaufzeit und der Genauigkeit der Näherung für das gegebene Optimierungsproblem zu erreichen. Wir wollen nun einige konkrete Rechenergebnisse an einem Beispiel betrachten.

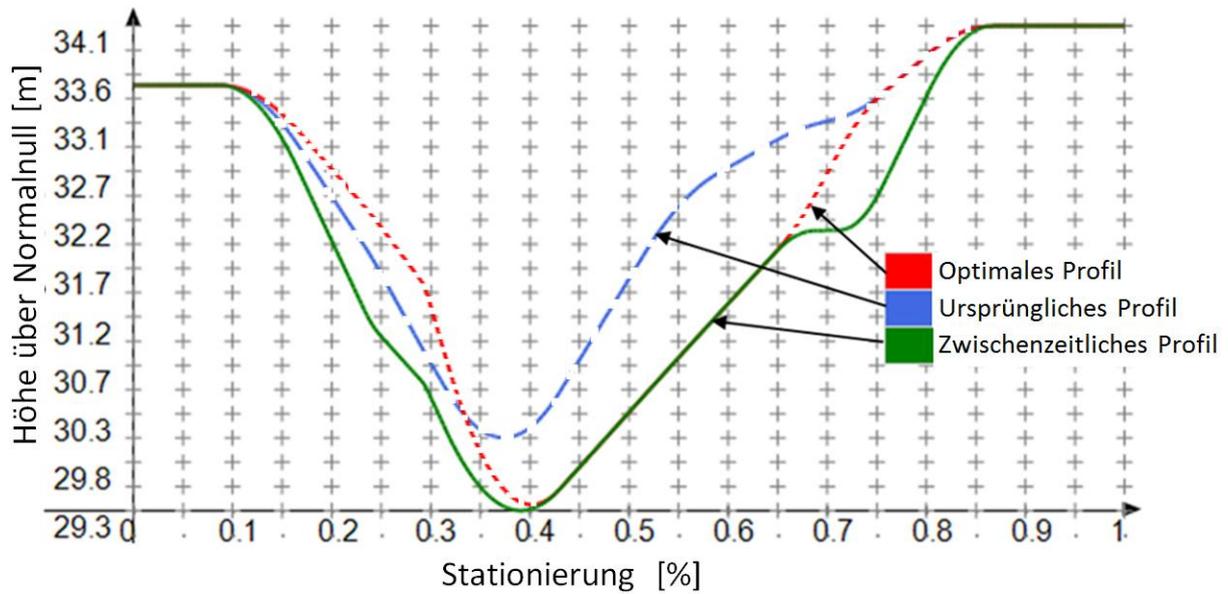


Abbildung 65: Drei Höhenprofile für die Strecke von Bolueta nach Etxebarri: in blauer Farbe (gestrichelt) das ursprüngliche Profil, das optimale in roter Farbe (gepunktet) und ein vom Ameisenalgorithmus erzeugtes (nicht optimales) in grüner Farbe (durchgezogen)

6.1.7.3. Beispiele zur automatischen Höhenprofiloptimierung

Als Ausgangsdaten für konkrete Berechnungen zur automatisierten Optimierung wurden wiederum Daten der Metro Bilbao verwendet. Innerhalb dieses Abschnitts beziehen wir uns der Anschaulichkeit halber auf die Strecke Bolueta (B) nach Etxebarri (E), wobei die dabei angesprochenen konkreten Strategien selbstverständlich ohne prinzipielle Änderung auf allgemeine Problemstellungen ausgedehnt werden können.

Das Originalprofil dieser Strecke entspricht der blauen, gestrichelten Kurve in Abbildung 65 und ergibt in unserem Modell einen Verbrauch von etwa $208,5 \frac{J}{kg}$. Insgesamt umfasst dieses

PVI	Stationierung [%]	Höhe über Normalnull [m]	Oberer Variationsbereich des PVI [m]	Unterer Variationsbereich des PVI [m]	Anzahl diskreter Schritte (oben/unten)
1	0,126	33,5	0	0	0
2	0,241	31,8	1,35	0,415	3/3
3	0,294	30,8	0,415	0,659	3/3
4	0,372	29,5	0,659	1,93	3/3
5	0,552	32,4	1,57	1,57	3/3
6	0,668	33,06	0,994	1,15	3/3
7	0,734	33,2	1,15	0,994	3/3
8	0,835	34,08	0	0	0

Tabelle 9: Variationsmöglichkeiten für die acht PVIs auf der Strecke Bolueta nach Etxebarri

Methode	Ergebnis [J/Kg]	Reduktion [J/Kg]	Zyklen/Ameisen	Dauer [hh:mm:ss]
Ameisenalgorithmus	196.8	11.66	5 x 4	00:00:12
Ameisenalgorithmus	193.4	15.04	15 x 4	00:00:21
Ameisenalgorithmus	193.2	15.34	30 x 4	00:00:34
Alle Permutationen	193.2	15.34	117649	08:49:43

Tabelle 10: Ergebnisse der Optimierung Bolueta nach Etxebarri

Profil acht PVI's. Die grundlegenden Parameterwerte zu ihrer Modifikation sind Tabelle 9 dargestellt: zunächst die nicht veränderlichen Werte der prozentualen Stationierung entlang der Strecke sowie die Originalhöhen, dann die Größe des oberen und des unteren Bereichs, in dem die einzelnen PVI's verschoben werden können, und zuletzt die Anzahl der diskreten Schritte, die innerhalb dieser Bereiche zugelassen waren.

Die Ergebnisse dreier Optimierungsläufe enthält Tabelle 10. Die Berechnung des Energieverbrauchs für alle möglichen Profile benötigte fast neun Stunden und lieferte ein optimales Profil mit einem Verbrauch von $193,2 \frac{J}{Kg}$. Dazu musste für 117 649 verschiedene Profile der Energieverbrauch berechnet werden. Zur Bestimmung einer Näherung wurden mehrere Varia-

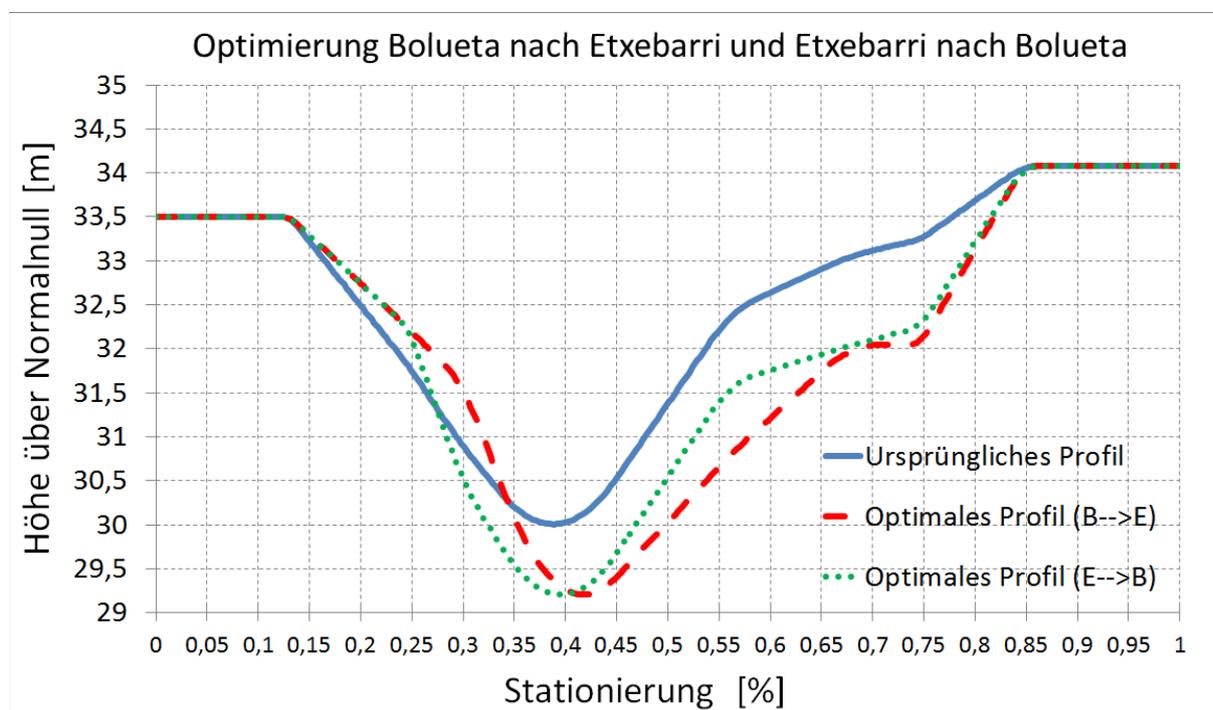


Abbildung 66: Die blaue, durchgezogene Kurve zeigt das ursprüngliche Höhenprofil der Strecke von Bolueta nach Etxebarri. Die rot gestrichelte Kurve ergibt sich aus einer Optimierung dieser Strecke ($B \rightarrow E$) ohne Berücksichtigung der Rückrichtung, die grün gepunktete Kurve entsteht aus einer Optimierung, die von Etxebarri her begonnen wurde ($B \leftarrow E$) und die Rückrichtung dabei ebenfalls vernachlässigte.

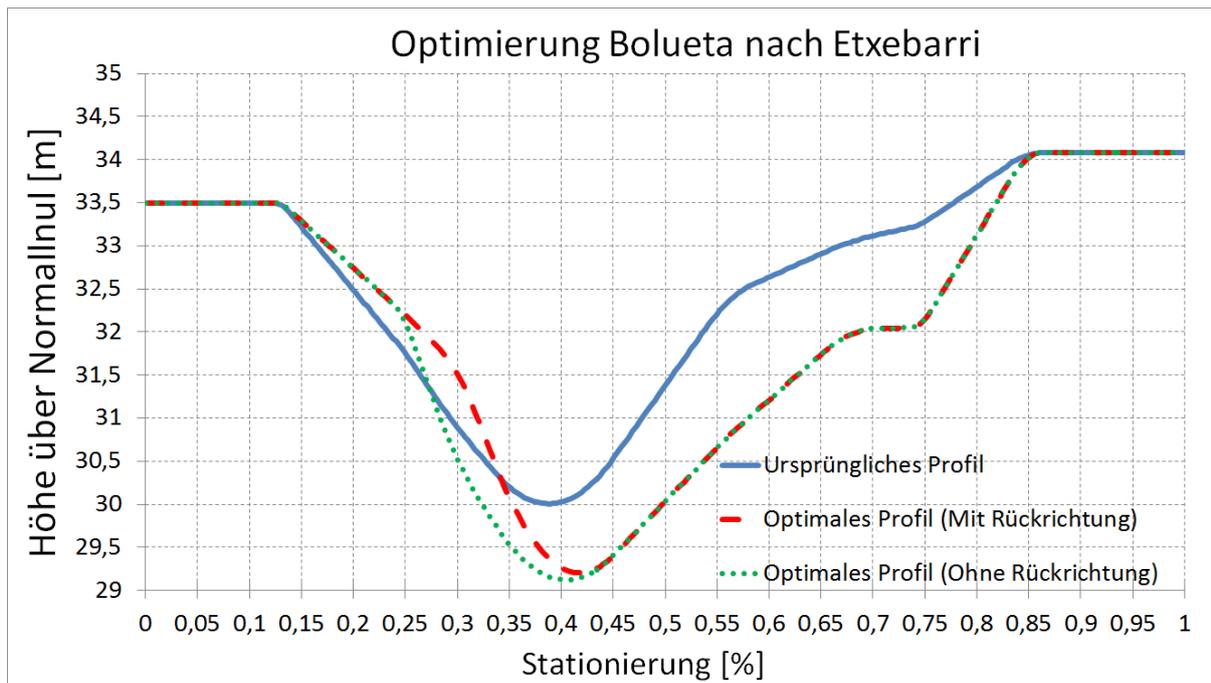


Abbildung 67: Die blau durchgezogene Kurve stellt das ursprüngliche Höhenprofil der Strecke von Bolueta nach Etxebarri dar. Die rot gestrichelte Kurve zeigt eine Optimierung dieser Strecke (B→E) ohne, die grün gepunktete Kurve eine Optimierung mit Berücksichtigung der Rückrichtung.

tionsläufe durchgeführt. Dabei wurde sowohl die Anzahl der beteiligten Ameisen als auch die Anzahl der Zyklen variiert.

Bereits bei vier (virtuellen) Ameisen konnten sehr gute Ergebnisse erzielt werden, während zugleich die Anzahl der Zyklen variiert wurde. Die Ergebnisse für vier Ameisen sowie 5, 15 und 30 Zyklen sind in Tabelle 10 abgebildet. Bereits bei 30 Zyklen und 4 Ameisen ist die Näherung auf vier gültige Ziffern mit dem optimalen Profil identisch und liefert eine Reduktion um etwa 15%. Es wurden dafür nur etwa 34 Sekunden benötigt, eine extrem kurze Dauer im Vergleich zu knapp neun Stunden bei der Berechnung aller Möglichkeiten. In Abbildung 65 sind das ursprüngliche (blau, gestrichelt) und das optimale (rot, gepunktet) Höhenprofil sowie ein weiteres nicht optimales, durch den Ameisenalgorithmus gefundenes Höhenprofil (grün, durchgezogen) dargestellt. Dabei ist zu beachten, dass der maximale Unterschied in der Höhe zwischen optimalem und ursprünglichem Höhenprofil nur etwa einen Meter beträgt, auch wenn das aufgrund der (verzerrten) Darstellung in Abbildung 65 auf den ersten Blick als mehr erscheinen mag; natürlich müssen dann in dieser Situation weitere Untersuchungen durch den Ingenieur unternommen werden, welche eine Realisierbarkeit dieser optimierten Route bestätigen.

Bei den bisherigen Untersuchungen wurde ein Abschnitt zwischen zwei Stationen zunächst in einer bestimmten Richtung untersucht. Diese Fragestellung ist natürlich hinsichtlich der Alltagspraxis nur interessant, wenn für die Rückrichtung ein anderer Tunnel verwendet wird. Denn ein energieoptimiertes Profil in der einen Richtung kann ein weniger effizientes in der anderen nach sich ziehen. Dies hat insbesondere auch zur Folge, dass die Bestimmung eines

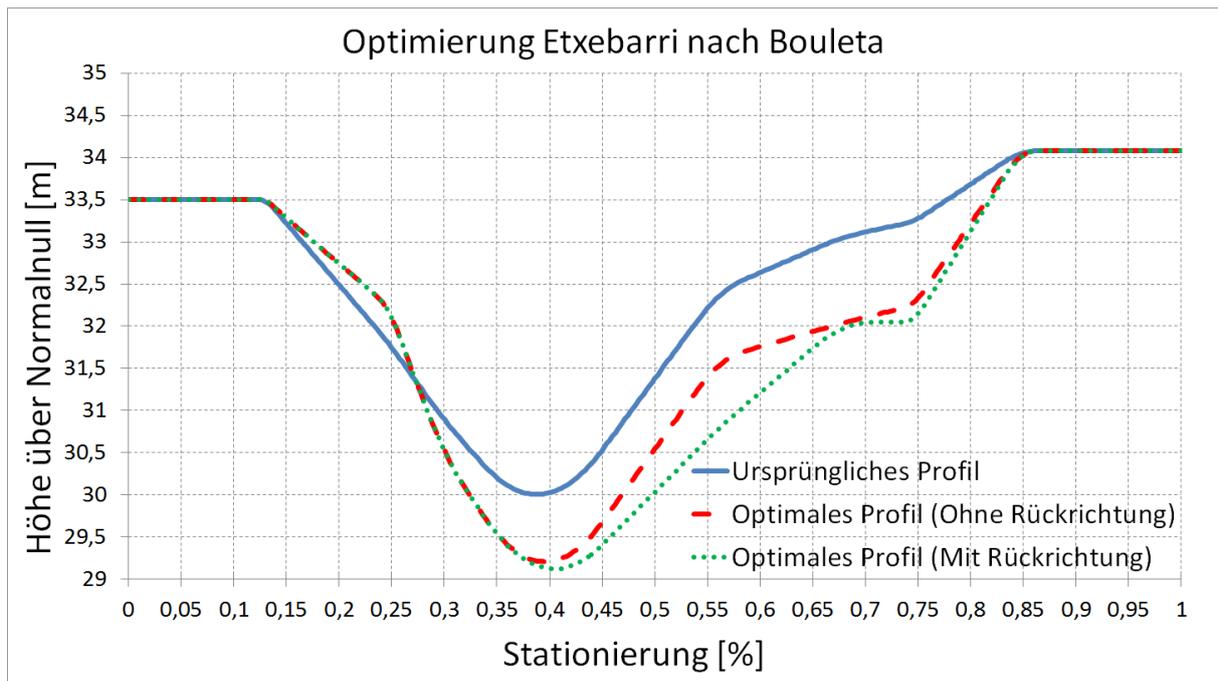


Abbildung 68: Die blau durchgezogene Kurve stellt das ursprüngliche Höhenprofil der Strecke von Bolueta nach Etxebarri dar. Die rot gestrichelte Kurve zeigt eine Optimierung dieser Strecke ($B \leftarrow E$) ohne Berücksichtigung der Rückrichtung, während die grün gepunktete Kurve eine Optimierung angibt, welche die Rückrichtung beachtet und mit der Kurve der Optimierung übereinstimmt, die sich bei einem Beginn von Bolueta ($B \rightarrow E$) her ergab.

optimalen Höhenprofils, das nach obigem Verfahren (6.1.7.1) festgesetzt wurde, genau für eine Richtung $B \rightarrow E$ gilt, die Rückrichtung $B \leftarrow E$ im Allgemeinen aber ein völlig anderes optimales Profil besitzt. Dieser Sachverhalt ist in Abbildung 66 dargestellt: Hier zeigt die blaue Kurve das ursprüngliche Profil der Strecke Etxebarri nach Bolueta ($B \rightarrow E$), die rot gestrichelte Kurve ein optimiertes Höhenprofil, bei dem die Berechnung von Bolueta aus begonnen wurde, ohne die Rückrichtung zu berücksichtigen. Die grün gepunktete Kurve weicht erheblich von der rot gestrichelten ab und gibt gerade das Höhenprofil an, das bei einer Optimierung ohne Beachtung der Rückrichtung entsteht, wenn die Optimierung aber von Etxebarri aus begonnen wurde ($B \leftarrow E$).

Um zu einem allgemeingültigen Optimum zu gelangen, haben wir unseren Ansatz nochmals erweitert: Wie in 6.1.7.1 beschrieben, erzeugen wir neue Profile und berechnen für diese das entsprechende Energieprofil. Ist somit ein Höhenprofil für die eine Richtung ($B \rightarrow E$) festgelegt, so addieren wir den Energieverbrauch auf dieser Route (in umgekehrter Richtung) für den Rückweg ($B \leftarrow E$) hinzu. Schließlich bestimmen wir das Minimum nicht für den Verbrauch für die Route von ($B \rightarrow E$), sondern für die so bestimmte Hin- und Rückroute ($B \rightarrow E \rightarrow B$), so dass wir ein (Gesamt-)Optimum für beide Richtungen erhalten. Abschließend sollte man noch untersuchen, ob man bei einer Wahl ($B \rightarrow E \rightarrow B$) zu einem anderen optimalen Profil gelangt als bei einer Wahl ($E \leftarrow B \leftarrow E$). Dies wurde im Rahmen dieser Forschung

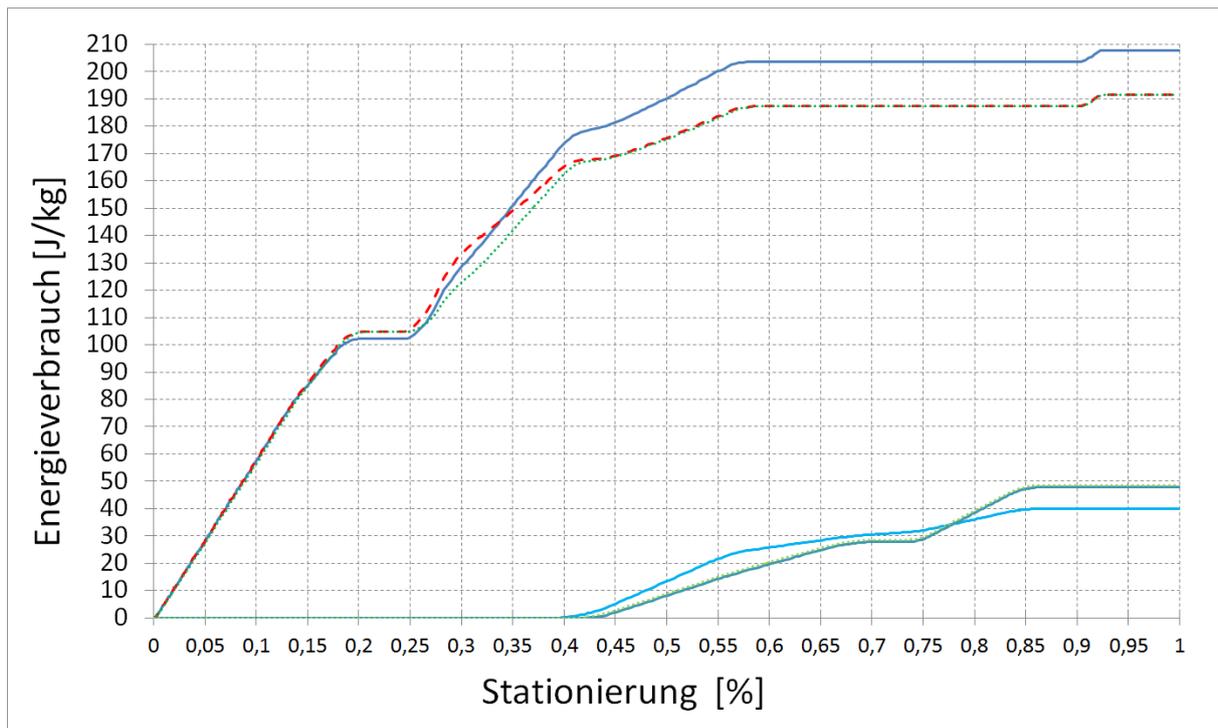


Abbildung 69: Energieverbrauch für die drei in Abbildung 67 dargestellten Höhenprofile, oben der Gesamtenergieverbrauch und unten die zugehörigen potentiellen Energien

nur exemplarisch nachgewiesen, ein mathematisch vollständiger Beweis soll hier nicht erbracht werden (siehe auch Abbildung 68).

Diese Überlegungen und die sich so ergebenden Resultate wollen wir noch kurz veranschaulichen: In Abbildung 67 und Abbildung 68 zeigt die blaue Kurve jeweils das ursprüngliche Profil; die rot gestrichelte Kurve in Abbildung 67 stellt ein optimiertes Höhenprofil von Bolueta nach Etxebarri ($B \rightarrow E$) ohne, die grün gepunktete Kurve unter Beachtung der Rückrichtung dar.

In Abbildung 68 gehen wir von Etxebarri aus. So stellt hier die rot gestrichelte Kurve ein optimiertes Höhenprofil von Etxebarri nach Bolueta ($B \leftarrow E$) ohne Berücksichtigung der Rückrichtung dar, die grün gepunktete Kurve mit Berücksichtigung der Rückrichtung. Insbesondere sieht man hier, dass die grün gepunkteten Kurven aus Abbildung 67 und Abbildung 68 übereinstimmen, obwohl sie prinzipiell unterschiedlichen Prozessen entspringen. Somit ergibt sich für die beiden Optimierungen ($B \rightarrow E \rightarrow B$) und ($E \leftarrow B \leftarrow E$) gerade das gleiche optimale Profil. In Abbildung 69 ist der Energieverbrauch für die Abbildung 67 dargestellten Profile angegeben. Hier zeigen die oberen drei Kurven den zugehörigen Gesamtenergieverbrauch, die unteren drei den Bedarf an potentieller Energie. Auffällig ist, dass das ursprüngliche Profil zwar den niedrigsten Bedarf an potentieller Energie ergibt, man aber trotzdem zu einem höheren Gesamtenergieverbrauch gelangt. Dies resultiert aus der „schlechteren“ Nut-

Strecke	Optimierung +: mit Optimierung -: ohne Optimierung	Rückrichtung +: berücksichtigt -: nicht berücksichtigt	Energieverbrauch $\left[\frac{J}{kg}\right]$
$B \rightarrow E$	-	-	207,9
$B \rightarrow E$	+	-	191,7
$B \rightarrow E \rightarrow B$	-	+	369,4
$B \rightarrow E \rightarrow B$	+	+	352,3
$B \leftarrow E$	-	-	161,5
$B \leftarrow E$	+	-	157,9
$E \leftarrow B \leftarrow E$	-	+	369,4
$E \leftarrow B \leftarrow E$	+	+	352,3

Tabelle 11: Verschiedene Optimierungsvarianten für die Strecke Bolueta (B) nach Etxebarri (E) liefern komplett unterschiedliche Energieverbrauchswerte.

zung der kinetischen Energie, d.h. nur ein kleinerer Anteil an potentieller Energie kann aus der kinetischen Energie gewonnen werden.

In Tabelle 11 sind abschließend die Ergebnisse für Berechnungen gemäß den obigen Überlegungen dargestellt. Insgesamt sieht man dabei insbesondere ein Gesamteinsparpotential von knapp fünf Prozent, wobei für das optimale Profil nur etwa einen Meter tiefer gegraben werden muss. Diese Reduzierung lässt sich den blau und rot markierten Feldern entnehmen: Blau markiert sind die nicht optimierten Fahrten hin und zurück, während rote Felder die optimierten Fahrten hin und zurück darstellen. Dabei wurde einmal (gemäß Spalte eins) die Route von Bolueta nach Etxebarri und einmal von Etxebarri nach Bolueta gewählt. Dies ist ein für unsere Berechnungen nicht untypischer Wert, wenn das zugehörige Einsparpotential auch eher niedrig erscheint: So ergab sich für zwölf bisher berechnete Profile insgesamt ein Bereich für das Optimierungspotential von ca. 2% bis ca. 16%.

6.1.8. Zusammenfassung Energiesimulation

In diesem Kapitel haben wir ein Konzept zur (autonomen) Energiesimulation eines U-Bahnzuges vorgestellt. Diese Energiesimulation unterstützt den Experten bei der Abschätzung der Änderungen des Energieverbrauchs infolge von Modifikationen des Trassenverlaufs bereits in der frühen Planungsphase. Zunächst haben wir dazu das zugrunde liegende physikalische Modell präsentiert, bevor die Integration des Modells und der resultierenden Software-

komponente in die Kollaborationsplattform erläutert wurde. Im Gegensatz zu anderen Ansätzen erlaubt der vorgestellte die Beurteilung der Trasse hinsichtlich der Energieeffizienz bereits a priori, d.h. bereits in der Planungsphase, und hier in Echtzeit. So erhält der Ingenieur unmittelbar Rückmeldung über jede Änderung, die er an einer Trasse vornimmt. Um den Einsatz unseres Modells zu rechtfertigen, konnten wir seine Validität im Rahmen der gewünschten Genauigkeit durch den Vergleich von realen Messdaten der Metro Bilbao mit den durch die Simulation berechneten Werten belegen. Weiterhin haben wir gezeigt, wie bereits kleine Modifikationen im Trassenverlauf zu verschiedenen Varianten führen, die sich signifikant im Energieverbrauch unterscheiden. Diese Erkenntnisse können dann zu nicht unerheblichen Kostenersparnissen führen und natürlich auch zu einem nachhaltigeren Umgang mit den natürlichen Ressourcen. Von diesen zunächst intuitiv gewonnenen Ergebnissen haben wir uns leiten lassen und abschließend Möglichkeiten gezeigt, mit Hilfe eines Ameisenalgorithmus Optimierungen an existierenden Höhenprofilen automatisch vorzunehmen, die in den Modellrechnungen zu einer beachtlichen Reduktion des Energieverbrauchs geführt haben.

6.2. Integration von Personenstromsimulatoren

Als zweites Beispiel für die Integration von Simulationen in den von der Plattform unterstützten kollaborativen Arbeitsprozess betrachten wir nun Simulationen von Personenströmen. Im Bereich der Tunnel- und vor allem der Bahnhofsplanung spielen Simulationen des Alltagsbetriebs sowie von Evakuierungsszenarien, die durch äußere Einflüsse wie Rauch und Feuer bedingt sind, eine wichtige und immer bedeutendere Rolle (Kneidl 2013). Die hierfür verwendeten Werkzeuge werden zwar in Bezug auf die Realitätstreue ihrer Resultate immer effektiver und ausgereifter, haben aber doch nach wie vor alle das Problem gemeinsam, dass die Transformation und Integration der geometrischen Modelldaten in diese Werkzeuge aufwendige und fehleranfällige Prozesse sind. Einen halbautomatisierten Ansatz für genau diesen Prozess der Datentransformation – hier konkret bei Modelldaten von U-Bahnhöfen – beschreiben wir in diesem Kapitel. Dieses Vorgehen unterstützt den planenden Ingenieur bei der Transformation insofern, als er sich auf das eigentliche Problem fokussieren kann, nämlich die Bewertung des Bauwerks aus sicherheitstechnischen Aspekten. Durch die direkte Integration der Simulation wird es zudem möglich, ihre Ergebnisse unmittelbar in den Modellierungsprozess zurückzuführen. Somit ergibt sich die im Folgenden zu lösende Grundproblematik:

Wie lassen sich die dreidimensionalen Geometriedaten prozedural modellierter Bauwerke (halb-)automatisiert in die meist zweidimensionalen Eingangsformate von typischen Personenstromsimulatoren umwandeln?

Das zugehörige Forschungsgebiet der Personenstromsimulation reicht zurück bis in die Mitte des letzten Jahrhunderts. Im Rahmen dieser Arbeit können wir nur einen ganz knappen Einblick in die Grundlagen dieses Gebiets geben. Für einen ausführlichen Überblick sowie für grundlegende Ansätze und Theorien sei auf (Schadschneider et al. 2009) und (Kneidl 2013) verwiesen. Einen hilfreichen Überblick über die momentan verwendeten Modelle (etwa Social Force, Zelluläre Automaten oder Agentensysteme) findet man auch bei (Duives et al. 2013). Dabei betonen wir auch von Anfang an, dass es nicht Ziel unserer Forschungen war, in diesem Bereich neue Erkenntnisse hinsichtlich der Simulationstheorien zu gewinnen, sondern Konzepte zu entwickeln, um vorhandene Ergebnisse dieses Forschungsbereichs gewinnbringend in den Planungsprozess zu integrieren.

Nachdem zu Beginn der Forschungen die rein deskriptive Untersuchung von Personenströmen im Vordergrund stand, lieferte Henderson einen ersten physikalisch orientierten Ansatz zur Beschreibung von Fußgängerströmen (Henderson 1971). In einem makroskopischen Ansatz überträgt er Grundlagen der kinetischen Gastheorie auf dieses Problemfeld. Nach Kneidl umfassen makroskopische Modelle allgemein solche Modelle, die eine ganzheitliche Sichtweise auf das zu simulierende Szenario haben und somit in gewisser Weise „allwissend“ sind

(Kneidl 2013). In diesen Modellen werden somit alle Personen – analog zur Behandlung verschiedener Teilchen in einem kinetischen Modell für ideale Gase – gleichbehandelt, während individuelle Verhaltensweisen einzelner Personen unberücksichtigt bleiben. In den Bereich der makroskopischen Modelle fallen auch sogenannte Netzwerkflussmodelle, die auch zur Simulation von Evakuierungsszenarien entwickelt wurden. Einen Überblick über diese Art von Modellen und die dahinter stehenden Theorien findet man etwa bei (Hamacher und Tjandra 2001). Wie Hamacher und Tjandra zeigen, eignen sich diese Modelle speziell zur Bestimmung einer unteren Schranke für die Evakuierungszeiten, insbesondere in Bahnhöfen und Flughäfen. Einen fundamentalen Nachteil von makroskopischen Modellen beschreiben (Duvies et al. 2013) so: „*Macroscopic models can correctly predict the aggregate trends, while not indicating which pedestrian is where at what moment in time.*“ Nach (Kneidl 2013) werden diese Aspekte gerade in mikroskopischen Modellen untersucht: „*Mikroskopische Modelle dagegen modellieren Fußgänger als Individuen mit dem Ziel, Interaktionen zwischen Fußgängern und das Verhalten in Menschenmengen abzubilden.*“

Mikroskopische Ansätze gehen zurück auf (Helbing und Molnár 1995), die in diesem Bereich zunächst sogenannte *Social Force* Modelle schufen. In diesen führen sie die Interaktionen verschiedener Fußgänger und die hieraus resultierenden Bewegungsmuster auf den Einfluss von sogenannten *Social Forces* zurück und verwenden diese als ein Maß für die individuelle Motivation des Individuums, bestimmte Bewegungen auszuführen.

Durch mikroskopische Modelle können klassische Probleme wie die Bahnenbildung, Stoßwellen und der Stau vor Engstellen, die bei Duvies et al. als *self-organization phenomena* bezeichnet werden (Duvies et al. 2013), sehr gut simuliert werden. Diese Bewegungsmuster können häufig in Bahnhofsbauwerken am Bahnsteig oder vor Treppen und Aufzügen beobachtet werden und werden von mikroskopischen Modellen hervorragend abgebildet (Schadschneider et al. 2009).

Eine alternative Möglichkeit zur Erzeugung mikroskopischer Modelle bieten sogenannte zelluläre Automaten. Hier werden die Bewegungsmöglichkeiten der Fußgänger durch Gitter vorgegeben, wobei vier- und sechseckige Gitter am weitesten verbreitet sind. Kinkeldey und Rose liefern einen Überblick über verschiedene Typen von zellulären Automaten (Kinkeldey und Rose 2003). Ein bekanntes, auf einem zellulären Automaten beruhendes Modell ist das von Nagel und Schreckenberg, das ursprünglich zur Simulation von Verkehrsstaus entwickelt wurde, aber sehr wohl auch bei der Simulation von Fußgängern eingesetzt wird (Nagel und Schreckenberg 1992). Kneidl versucht, mit Hilfe eines Navigationsgraphen eine Brücke zwischen makroskopischen Modellen und auf zellulären Automaten basierenden mikroskopischen Modellen zu schlagen (Kneidl 2013). Insbesondere werden bei der Erzeugung des Navigationsgraphen kognitive Prozesse bei der Wegfindung mit dem Ziel berücksichtigt, möglichst realitätsnahe Bewegungsmuster zu generieren. Im Rahmen der Plausibilisierung unserer eigenen Ansätze zur Integration von Personenstromsimulatoren greifen wir auf die Arbeiten

von (Biedermann et al. 2014) und (Kielar et al. 2014) zurück, die mit dem Werkzeug *MomentumTUMV2*⁴² einen leistungsstarken Personenstromsimulator entwickelt haben.

Wir werden hier der Vollständigkeit halber noch einige spezielle Simulationswerkzeuge im Bereich der Evakuierungsszenarien nennen. Basierend auf der Simulationssoftware *CFD_Phoenics 3.5*, untersuchten Yang et al. Evakuierungsszenarien im Brandfall und lieferten dabei insbesondere Abschätzungen für absolute Evakuierungszeiten (Yang et al. 2006), während für Park et al. die Simulation der Rauchentwicklung in U-Bahnstationen unter Verwendung der Software *FDS (Fire Dynamics Simulator)* von Interesse war (Park et al. 2006). Mit *FDS* experimentierten auch Chen et al., wobei sie die Personenstromsimulationen mit dem Multiagentensystem *Legion* durchführen (Chen et al. 2015). Unter Verwendung der weit verbreiteten Softwarelösung *Smartfire* simulierten Hao et al. Fußgängerströme in einem U-Bahnhof bei Rauchentwicklung (Hao et al. 2009 I), während sie ebenfalls 2009 Evakuierungsszenarien in U-Bahn-Tunnels infolge von Bränden mit dem Ziel simulierten, die Abstände zwischen verschiedenen Tunnels zur Entfluchtung zu optimieren (Hao et al. 2009 II). Erwähnt werden sollen auch (Kucera und Bradacova 2012) sowie (Li et al. 2013), die interessante Beiträge in diesem Bereich lieferten. Dabei stehen hier die Entwicklung hoher Temperaturen, die Rauchentwicklung und die kritische Ventilationsgeschwindigkeit im Zentrum der Untersuchungen des Evakuierungsszenarios. In seiner Dissertation untersuchte Schatz einen *Serious Gaming* Ansatz, der insbesondere den Einfluss von Gebäudeeigenschaften auf das menschliche Verhalten im Evakuierungsfall abbilden soll (Schatz 2014). Im Mittelpunkt seiner Untersuchungen stehen auch die Kombinationen von BIM-Technologien und Simulationen zur Rauchentwicklung, um zu verifizieren, dass sich Menschen in einem derartigen *Serious Gaming* Ansatz ähnlich wie in realen Extremsituationen verhalten.

6.2.1. Prozedurale Modellierung von Bahnhofsbauwerken

In Kapitel 3 haben wir ausführlich prozedurale Geometriemodelle besprochen, insbesondere haben wir dabei das multi-skalige prozedurale Modell beschrieben, das die Grundlage des von der Plattform *Kollabserver* orchestrierten, kollaborativen Modellierungsprozesses bildet. Dieses Modell wurde ursprünglich für die Modellierung von trassenbasierten Bauwerken, insbesondere der Modellierung von U-Bahn-Tunnels entworfen. Es konnte aber gezeigt werden, dass sich das Modell auch sehr gut zur Abbildung von Bahnhofsbauwerken eignet (siehe Kapitel 3.8). Zur Modellierung der Bahnhofsmodelle werden die üblichen prozeduralen Operationen verwendet; die hier entscheidenden sind Arbeitsebenen, Skizzen, Extrusionen und die Booleschen Operationen Schnitt, Differenz und Vereinigung. Arbeitsebenen werden zur Definition der räumlichen Lage von Skizzen verwendet, die dann selbst als Grundlage für Extrusionen zur Erzeugung von Volumenkörpern benutzt werden; diese können dann durch Boole-

⁴² <https://www.cms.bgu.tum.de/de/team/kielar/31-forschung/projekte/456-momentum>

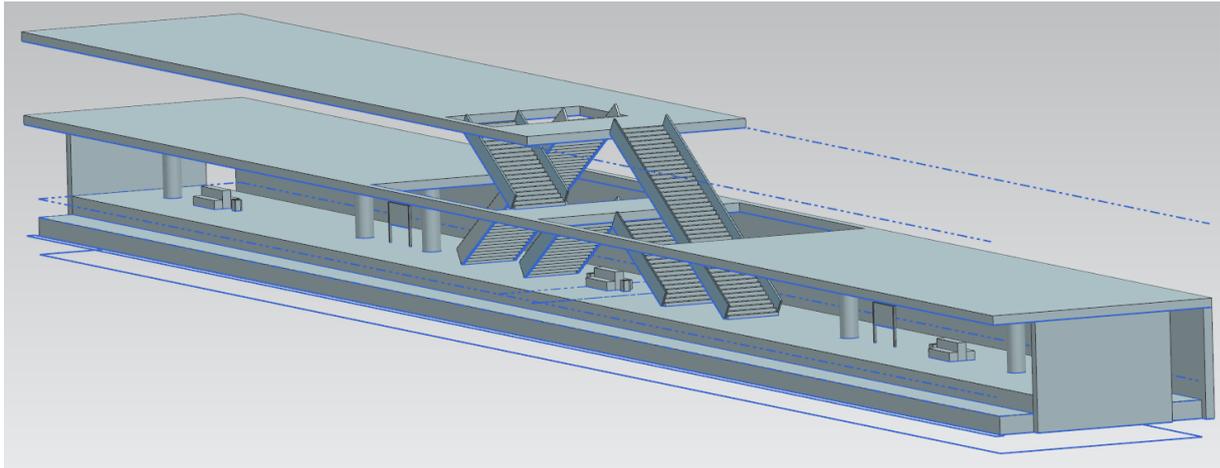


Abbildung 70: Ein über die Kollaborationsplattform erstelltes, vereinfachtes Modell des Münchner U-Bahnhofs Theresienstraße

sche Operationen kombiniert werden. Zusätzlich wurden sogenannte Musterfunktionalitäten (*Pattern*) in das Modell integriert. Diese eignen sich sehr gut zur Abbildung von Objekten und Objektgruppen, die im Kontext der Bahnhofmodellierung immer wiederkehren, etwa von Treppen, Sitzbänken, Fahrstandanzeigern oder Schildern (siehe das Modell in Abbildung 70). Muster fassen eine beliebige Menge von prozeduralen Operationen als eine Einheit auf und lassen sich dann an vorgegebener Position integrieren und nach bestimmten Regeln – etwa in bestimmten Abständen entlang einer Linie – replizieren.

Als grundlegend für die Modellierung eines Bahnsteiges sowie der eines Bahnhofsbauwerks hat sich die Verwendung einer sogenannten Leitskizze etabliert. Alle weiteren Skizzen referenzieren dann die Entitäten dieser Leitskizze, um die weiteren Elemente an der richtigen räumlichen Position zu platzieren. Um sich eine gewisse Flexibilität in der Modellierung zu erhalten, ist gerade hier der intensive Gebrauch von Parametern und Constraints einerseits bei der Definition der Leitskizze, andererseits bei den referenzierenden Skizzen von entscheidender Bedeutung; er konnte hier nochmals intensiv bei der prototypischen Modellierung plausibilisiert werden.

In Abbildung 71 ist der Beginn der sukzessiven Modellierung eines Bahnsteiges dargestellt: Links ist die in diesem Fall sehr einfache Leitskizze, in der Mitte eine erste Extrusion zu se-

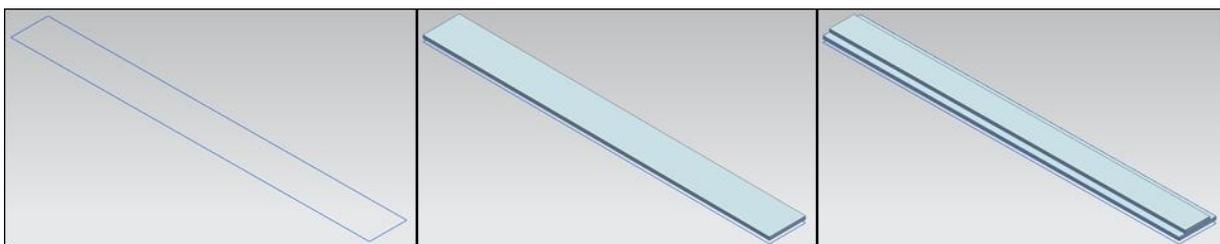


Abbildung 71: Sukzessive Modellierung eines Bahnsteiges, ausgehend von einer rechteckigen Leitskizze

hen, welche die Bodenplatte repräsentiert, während rechts bereits der (prinzipielle) Bahnsteig zu erkennen ist. In Abbildung 70 ist das finale prototypische Modell des Münchner U-Bahnhofs Theresienstraße dargestellt, welches sich durch sukzessive Anwendung der oben beschriebenen Konstruktionsschritte ergibt. Basierend auf dieser Grundidee, wurde prototypisch eine Reihe von Bahnsteigen modelliert, welche die konkrete Grundlage bei der Plausibilisierung des zu präsentierenden Ansatzes zur Datentransformation der prozeduralen Geometrien in die Eingangsformate typischer Personenstromsimulationen bildeten.

6.2.2. Eingangsformate von Personenstromsimulatoren

Im Bereich der Personenstromsimulation innerhalb von Bauwerken hat sich die Strategie einer etagenweisen Simulation etabliert (Duvies et al. 2013). So werden die einzelnen Etagen eines Bauwerks als separate Einheiten (häufig als *Floors* bezeichnet) aufgefasst, modelliert und schließlich simuliert. Übergänge zwischen verschiedenen Etagen sind dabei möglich, werden aber gesondert betrachtet und abgebildet. Die Beschreibung einer konkreten Ebene erfolgt dabei zweidimensional: Zunächst wird die Grundfläche der Etage als ein prinzipiell für die Fußgänger möglicher Bereich in Form eines Polygons festgelegt (Szenario). Aus dieser Polygonfläche werden dann Bereiche eliminiert, die nicht begehbar sind. Diese Bereiche werden typischerweise als Wände (*Walls*) modelliert, die den Bewegungsbereich durch einzelne nicht notwendigerweise geschlossene Linienzüge einschränken, während Hindernisse (*Obstacles*) durch Polygone abgebildet werden und auf natürliche Art und Weise alle Gegenstände repräsentieren, die sich einem Fußgänger in den Weg stellen. Dies sind gerade Elemente des Bauwerks selbst, aber auch alle Arten von Einrichtungs- oder Alltagsgegenständen. Im gegebenen Modellierungsparadigma steht die Verwendung von Hindernissen im Vordergrund, während Wandelemente (momentan) nur zur Beschreibung des gesamten Szenarios benötigt werden. In Abbildung 72 ist ein derartiges zweidimensionales Modell der Bahnsteigebene dargestellt, welche aus dem vorangehenden Beispiel des U-Bahnhofs Theresienstraße in München (Abbildung 70) gewonnen wurde. Als rechteckige Grundfläche erkennt man den prinzipiellen Bereich des Bahnsteigs, in welchem als Hindernisse Schilder, Bänke, Säulen, Treppen etc. integriert wurden.



Abbildung 72: Modell der Bahnsteigebene des Münchner U-Bahnhofs Theresienstraße aus Abbildung 70

Um eine Verknüpfung zwischen diesen verschiedenen Ebenen eines Bauwerks zuzulassen, werden Verbindungen zwischen diesen modelliert, indem die in der Realität auftretenden Treppen und Aufzüge einen Übergang von einer Simulationsebene in eine andere durch sogenannte Quellen und Ziele zulassen. Quellen beschreiben allgemein die Startpunkte der einzelnen Fußgänger, während Ziele die Orte festlegen, zu denen sich die Fußgänger bewegen sol-

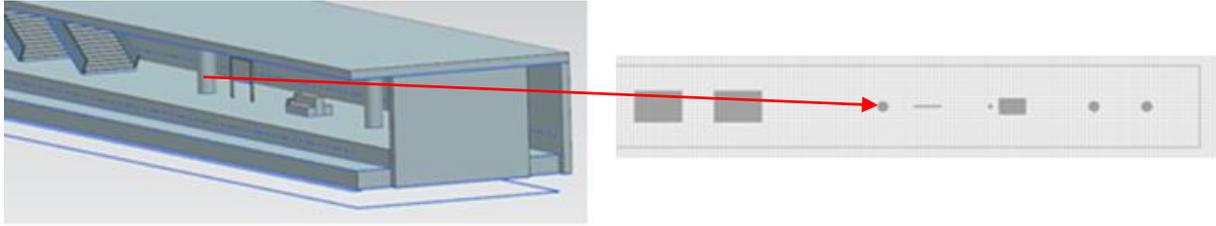


Abbildung 73: Projektion einer dreidimensionalen Säule in die zweidimensionale Bahnsteigebene

len. Welche Strategien Fußgänger bei der Wahl der Ziele verwenden, ist bei der Konfiguration der Simulation konkret anzugeben. Auch eine automatische Zielwahl, algorithmisch etwa durch eine *Shortest Distance Strategy* repräsentiert, ist möglich und im Bereich der Personenstromsimulation an einem Bahnsteig in vielen Fällen durchaus sinnvoll. Bei der Simulation von mehrstöckigen Bauwerken können die Fußgänger so eine konkrete Ebene an Zielen verlassen, die in der Konfiguration definiert wurden, und eine andere Ebene an Quellen wieder betreten. Die beschriebene Verwendung einzelner Ebenen und ihre Verknüpfung führen dann zur Verbindung von mehreren Ebenen (*Floors*). Die konkreten Wege, auf denen sich die Fußgänger innerhalb einer Ebene bewegen können, werden meist automatisiert durch die Simulation bestimmt und in einem sogenannten Navigationsgraphen abgebildet (Kneidl 2013).

Die konkrete physikalische Repräsentation der Eingangsdaten wird häufig über proprietäre XML-Formate erreicht, die von Werkzeug zu Werkzeug zwar leichte Unterschiede aufweisen, aber doch in weiten Teilen einfach ineinander umzuwandeln sind. Typischerweise befindet sich auf oberster Ebene ein sogenanntes *Szenario*-Element, welches den gesamten begehbaren Bereich in Form eines Polygons beschreibt. Außerdem werden Listen von Quellen und Zielen sowie Listen von Wänden und Hindernissen integriert, die geometrisch als Polygone repräsentiert werden. Zusätzlich werden Informationen zur Konfiguration übernommen, welche die zu verwendenden Algorithmen und Simulationsmodelle definieren und sich zwischen den einzelnen Werkzeugen am stärksten unterscheiden.

6.2.3. Transformation von prozeduralen Bahnhofsmodellen

Wir werden uns nun der Lösung des zu Beginn dieses Kapitels 6.2 beschriebenen Grundproblems widmen, die dreidimensionalen Ausgangsdaten prozeduraler Geometriemodelle in die Eingangsformate typischer Personenstromsimulatoren zu transformieren. Im vorangehenden Abschnitt wurde gezeigt, dass die entscheidenden Elemente für die Beschreibung der Eingangsdaten der Personenstromsimulatoren die zweidimensionalen Hindernisse darstellen. Diese können als Projektionen einzelner Volumenkörper in die Bahnsteigebene aufgefasst werden. Exemplarisch ist dies für eine Säule des Bahnhofsmodells Theresienstraße in Abbildung 73 dargestellt. Das detaillierte Vorgehen zur Bestimmung dieser Volumenkörper und der sich daraus im Simulationsszenario ergebenden Hindernisse beschreiben wir im Folgenden.

6.2.3.1. Identifikation von Volumenkörpern

Prinzipiell gilt es somit, alle Volumenkörper zu identifizieren, welche den (zunächst) dreidimensionalen Bewegungsfreiraum der Fußgänger einschränken. Die so definierten Einschränkungen sind dann aus dem auf oberster Ebene festgelegten Gesamtbereich des ebenen Simulationsszenarios auszuschließen, ergeben also in diesem gerade zweidimensionale Hindernisse.

Um den Bereich zu bestimmen, in dem sich die Fußgänger bewegen können, wird die Bahnsteigfläche zunächst manuell identifiziert. Dazu wählt der Experte die entsprechende Skizze aus der Menge der prozeduralen Operationen – in der prototypischen Umsetzung direkt über den Konstruktionsbaum des CAD-Werkzeugs. Da im prozeduralen Modell weitergehende semantische Informationen fehlen, muss dieser Schritt manuell ausgeführt werden. Die gewählte Skizze gibt dann den gesamten ebenen Bereich vor, der maximal als Lauffläche für die Fußgänger zur Verfügung steht, und wird in Form von Wänden in das Simulationsszenario integriert. Diese ebene Fläche definiert aber auch gerade den maximalen räumlichen Bereich, in dem sich die Fußgänger bewegen können. So wird generell eine bestimmte Höhe für den Raum des Bahnsteigs angenommen, die flexibel festgelegt werden kann; ein Wert von 3.0 Metern entspricht einer typischen Wahl. Dieser Wert kommt in der Realität zwar nicht für die Deckenhöhe in Frage, aber Fahrstandanzeiger befinden sich etwa in dieser Höhe und sollten nicht als Hindernisse in das Szenario aufgenommen werden. Somit werden nun insgesamt alle Volumenkörper des Modells identifiziert, die mit dem so beschriebenen Bewegungsraum einen nicht-leeren Schnitt haben.

Volumenkörper entstehen im prozeduralen Modell aus Extrusionen, Kombinationen von Extrusionen in Form von Booleschen Operationen sowie aus Musterelementen. Eine Extrusion selbst wird durch eine Skizze, genauer durch ein in der Skizze enthaltenes Profil, eine Richtung und Höhe festgelegt. Die Richtung ist im einfachsten, aber hier auftretenden Fall durch einen dreidimensionalen Vektor gegeben. Komplexere Möglichkeiten, wie etwa Extrusionen entlang von Splines sind möglich, aber bei der Modellierung von Bahnhofsbauwerken im Gegensatz zu der von Tunnelbauwerken eher selten anzutreffen. Die Höhen von Extrusionen werden typischerweise parametrisch festgelegt, so dass ein gewisses Maß an Flexibilität hinsichtlich späterer Änderungen fundamental integriert wird. Der konkrete Teil einer Skizze, der als Basis für eine spätere Extrusion verwendet werden kann, wird durch einen geschlossenen Linienzug definiert und als Profil bezeichnet. Ähnlich wie bei der Modellierung eines Tunnels (3.4), wird zur Modellierung von Bauwerken in unserem Ansatz davon ausgegangen, dass eine Skizze jeweils höchstens ein Profil enthalten darf. Dies erleichtert die Interpretation der verwendeten Skizzen, stellt aber keinen prinzipiellen Nachteil hinsichtlich der Modellierungsmöglichkeiten dar (siehe auch Erläuterungen zur Granularität in Kapitel 3.3).

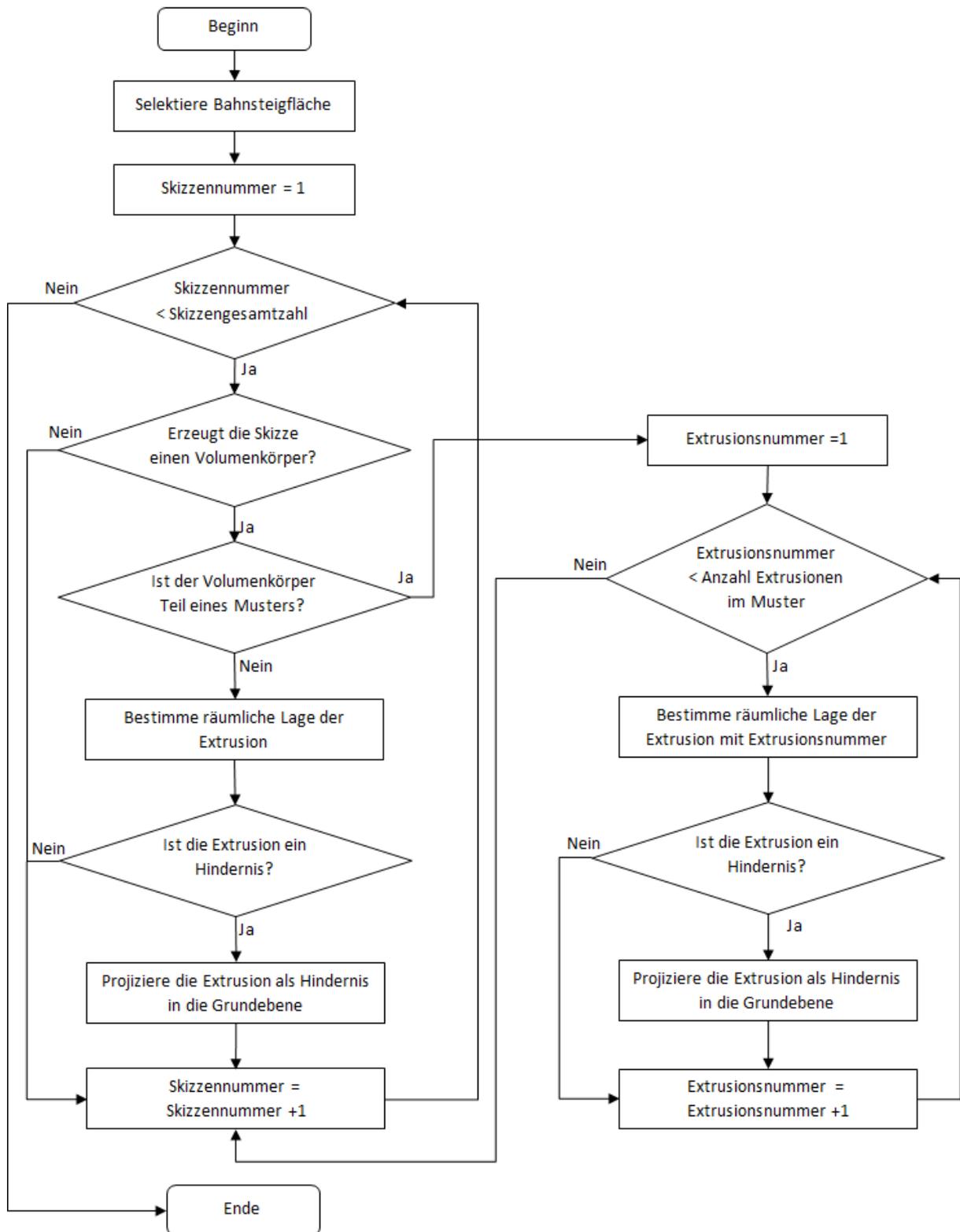


Abbildung 74: Basisalgorithmus zur Transformation der dreidimensionalen prozeduralen Geometrie in die Personenstromsimulatoren

Dazu werden alle Skizzen des prozeduralen Modells (als mögliche Basis von Volumenkörpern) hinsichtlich ihrer Lage in Relation zu dieser ausgewählten Skizze untersucht. Dazu werden die abhängigen Elemente der Skizze in der Konstruktionshistorie gemäß einem *Depth*

First Algorithmus (Tiefensuche) durchlaufen (Cormen et al. 2010). Gemäß dem vorgegebenen Modellierungsparadigma sind Skizzen zweidimensionale Elemente. Ihre räumliche Lage und damit auch die räumliche Lage von Volumenkörpern, die auf diesen Skizzen basieren, werden durch die Arbeitsebenen festgelegt, die bei der Definition der Skizze als Basis ausgewählt wurden. Somit müssen diese Arbeitsebene und ihre Lage stets in alle Überlegungen mit einbezogen werden.

Ist eine untersuchte Skizze Basis für eine einzelne Extrusion, so wird konkret überprüft, ob diese den Bewegungsraum der Fußgänger einschränkt. Natürlich ist dabei zunächst die Richtung der Extrusionen zu beachten, tatsächlich kann eine Extrusion nämlich auch von oben in Richtung des Bahnsteigs durchgeführt worden sein, so dass der senkrechte Abstand der definierenden Skizze (bei der Projektion) um diese Höhe zu mindern ist.

Basiert hingegen auf der Skizze ein Muster, welches sich selbst wieder aus einzelnen Extrusionen zusammensetzt, so müssen alle im Muster enthaltenen (replizierten) Extrusionen hinsichtlich ihrer Position im Raum untersucht werden. Diese Extrusionen müssen allesamt untersucht werden, auch wenn sie nicht explizit im Modell enthalten sind, sondern gemäß der im Muster hinterlegten Regel dynamisch zur Laufzeit erzeugt werden. Muster definieren so zum einen die enthaltenen Basisextrusionen, andererseits einen Algorithmus, gemäß dem diese repliziert werden. In unserem Modellierungsparadigma haben wir uns auf lineare Muster beschränkt. Diese definieren eine Verschiebungsrichtung und einen (parametrisierbaren) Verschiebungsabstand, so dass die einzelnen enthaltenen Extrusionen gemäß diesen Festlegungen geprüft werden können. Ein typisches Beispiel für ein Muster ist etwa eine Treppe: Hier wird die erste Stufe als Basisextrusion definiert und entlang eines Steigungsvektors Stufe für Stufe bis zum Treppende repliziert.

Um den Ablauf des gesamten Basisalgorithmus noch einmal zu verdeutlichen, sind seine hier besprochenen Schritte nochmals in Abbildung 74 schematisch mit Hilfe eines Flussdiagramms dargestellt.

6.2.3.2. Identifikation von Profilen

Sind alle relevanten Volumenkörper bestimmt worden, müssen die dadurch betroffenen (ebenen) Bereiche aus dem Simulationsszenario ausgeschlossen werden. Die Basis dafür bilden die Profile der Skizzen, die als Grundlage für die Extrusionen entworfen wurden. Die einzelnen Skizzenelemente zur Beschreibung eines Profils wie Linien und Kreisbögen verfügen im prozeduralen Modell zwar stets sowohl über einen Anfangs- als auch über einen Endpunkt, es fehlt aber die Information, wie sich aus diesen Kurven ein geschlossener Linienzug ergibt, der dann später als Hindernis in Form eines Polygons in das Simulationsszenario integriert werden kann. Im einfachsten Fall enthält eine Skizze nur einen Kreis als Profil, der dann durch ein regelmäßiges Vieleck angenähert in das Simulationsszenario aufgenommen wird. In allen

anderen Fällen müssen geschlossene Linienzüge in einer Skizze (als Grundlage der die Extrusionen definierenden Profile) algorithmisch identifiziert werden.

Dazu werden die Elemente einer Skizze auf natürliche Art und Weise in eine Graphstruktur eingebettet: Start- und Endpunkte der einzelnen Elemente der Skizze werden als Knoten, Linien und Kreisbögen als Kanten aufgefasst. Dabei müssen wir beachten, dass es im prozeduralen Modellierungsparadigma üblich ist, im Raum übereinstimmende Start- oder Endpunkte verschiedener Elemente getrennt zu speichern und sie beim Aufbau der Skizze durch Constraints zur Übereinstimmung zu bringen (*Coincident Constraint*). Derartige Punkte werden beim Aufbau des Graphen zu einem einzelnen Punkt verschmolzen. Der entstandene Graph ist zunächst ungerichtet, so dass zusätzlich gefordert wird, dass bei der späteren Identifikation eines geschlossenen Linienzugs jede Kante nur einmal und damit in genau einer Richtung durchlaufen werden kann.

Zusätzlich gilt es zu beachten, dass ein Teil der Elemente einer Skizze durch Projektionen aus anderen Skizzen entstanden sein kann. In diesem Fall enthalten die Skizzen nicht die Koordinaten entsprechender Elemente. Um diese Koordinaten aus dem Modell zu extrahieren, werden die übergeordneten prozeduralen Elemente (*Parent-Elemente*, siehe auch Kapitel 3.7.2) rekursiv durchlaufen, bis die sie definierende Skizze identifiziert wurde. Dieser können dann die konkreten Koordinaten der Elemente entnommen werden, um sie dann (senkrecht) in die aktuelle Skizzenebene zu projizieren. Hier müssen wir wieder beachten, dass die räumliche Lage der Skizzen durch die sie definierenden Arbeitsebenen festgelegt ist. Da definierende Skizzen einerseits schräg zur Skizze, welche die Projektionen enthält, verlaufen können, andererseits der Skizzenursprung verschoben sein kann, stimmen die zweidimensionalen Koordinaten von referenzierendem und referenziertem Element im Allgemeinen tatsächlich nicht überein.

6.2.3.3. Identifikation von Zyklen

Im so gebildeten Graphen lassen sich dann geschlossene Linienzüge als Zyklen (siehe Kapitel 3.7.2) identifizieren. Die Grundlage zu ihrer Bestimmung ist wiederum eine *Tiefensuche*, deren Grundidee wir aufgreifen und auf unseren Fall anpassen (Turau 2004). Durch eine dynamische Identifikation der Nachbarknoten lässt sich dieser Algorithmus auch auf den zunächst ungerichteten Skizzengraphen anwenden. Die Grundidee des Algorithmus stellt sich folgendermaßen dar: Um einen Zyklus zu erkennen, werden alle Nachbarknoten eines vorgegebenen Startknotens und dann wiederum rekursiv ihre Nachbarknoten untersucht. Gelangt man in dieser rekursiven Suche zu einem bereits untersuchten Knoten, so wurde ein Zyklus gefunden. Dabei wird nicht nur die Existenz eines Zyklus nachgewiesen, sondern der Zyklus kann sogar konkret angegeben werden.

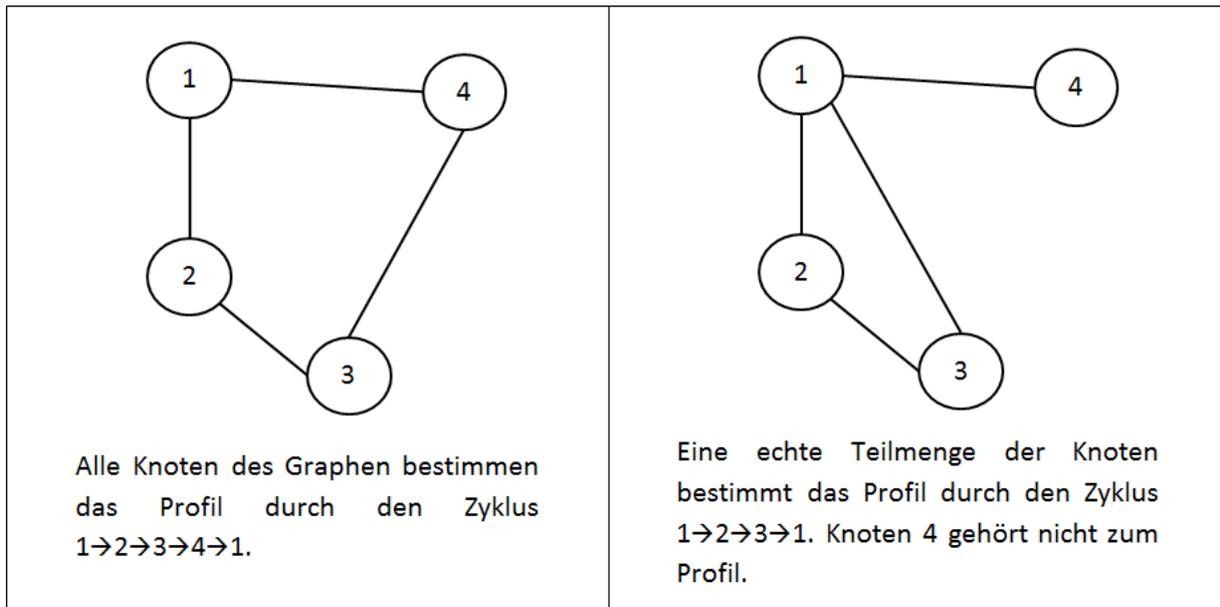


Abbildung 75: Profile in einem Skizzengraphen; rechts erkennt man: Zyklen werden auch entdeckt, wenn der Ausgangsknoten des Algorithmus nicht zum Zyklus gehört.

Im Kern dieses Algorithmus, der im Anhang konkret in Pseudocode angegeben ist, steht eine Methode *FindeZyklus*. Diese wird aus einer äußeren Iteration über alle Knoten des Graphen für jeden dieser Knoten einzeln aufgerufen und untersucht, dann vom jeweiligen Knoten aus, ob ein Zyklus gefunden werden kann. Dieser Eingabeknoten muss dabei selbst nicht zum Zyklus gehören. Es genügt, wenn man von ihm aus zu einem Zyklus gelangen kann. Wird die Funktion *FindeZyklus* etwa im rechten Beispiel der mit dem Knoten 4 als Startknoten aufgerufen, so wird ausgehend von Knoten 4 der Zyklus $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$ gefunden, ohne dass der Knoten 4 selbst in diesem Zyklus enthalten ist. Nachdem ein Zyklus gefunden wurde, kann die äußere Iteration sofort verlassen werden, da es nur das Ziel war, einen Zyklus zu finden und nicht etwa alle möglichen Zyklen. Insbesondere kann es in unserem Anwendungsfall beim Auffinden eines Profils in einer Skizze – bis auf das Verschieben der Indizes – nur genau einen Zyklus geben, da eine Skizze definitionsgemäß nur ein Profil enthalten darf. Enden alle Aufrufe aus der Hauptschleife heraus ohne Erfolg, so enthält der Graph gerade keinen Zyklus.

6.2.3.4. Integration von Quellen und Zielen

Um nun die Fußgängerströme tatsächlich zu simulieren, müssen noch die Quellen und Ziele konkret definiert werden. Dabei bezeichnen Quellen die Startpunkte der Fußgänger und Ziele die Orte, zu denen sich die Fußgänger bewegen sollen. Bei der Simulation etwa des Ausstiegsvorgangs entsprechen die Quellen zunächst (im Wesentlichen) den Türen des einfahrenden U-Bahnzugs, die Ziele dagegen den Treppen und Ausgängen. Die Position der U-Bahntüren kann dem prozeduralen Bahnhofsmo-
dell nicht entnommen werden, da es diese

Information offensichtlich nicht enthält. Um den Anwender zusätzlich zu unterstützen, wurde ein einfaches Softwarewerkzeug entwickelt, das es erlaubt, Konfigurationen für U-Bahn-Waggons vorzugeben und einzelne Waggons zu Zügen zusammzusetzen. In dieser können beispielsweise die Positionen der Türen eingefügt und Informationen über die Position des Zugs am Bahnsteig hinterlegt werden. Diese Informationen können dann automatisch in das Simulationsszenario integriert werden. Eine automatisierte Erkennung von Treppen, Rolltreppen oder Aufzügen ist momentan nicht realisiert und ohne weitere semantische Informationen im Modell schwierig zu realisieren. Konkret kann der Benutzer in einer Konfigurationskomponente des Szenarios weitere Quellen und Ziele einfach (als Rechtecke) markieren, um diese ebenfalls in das Simulationsszenario zu integrieren.

Ein so entstandenes Szenario wird in der prototypischen Implementierung in einem neutralen Format (XML-basiert) gespeichert. Aus diesem kann dann auf natürliche Art und Weise das proprietäre Format einzelner Hersteller mit üblichen Transformationsstrategien erzeugt werden (siehe etwa Kay 2008). So ist insgesamt eine zumindest (halb-)automatisierte Integration gängiger Personenstromsimulatoren möglich, womit das zu Beginn des Kapitels 6.2 formulierte Grundproblem als gelöst betrachtet werden.

6.2.4. Exemplarische Plausibilisierung des Algorithmus

Zur Plausibilisierung des Algorithmus wurden mehrere prozedurale Bahnhofsmodele erstellt und eine halb-automatisierte Umwandlung der jeweiligen Bahnsteigbereiche in das beschriebene neutrale Format konkret durchgeführt. Dabei wurden insbesondere die fünf Münchener U-Bahnhöfe Hauptbahnhof, Karlsplatz, Odeonsplatz, Marienplatz und Theresienstraße in ausreichendem Detail und der notwendigen Genauigkeit modelliert. In Abbildung 76 ist exemplarisch ein Modell des Münchner U-Bahnhofs Odeonsplatz mit zwei Bahnsteigen dargestellt,

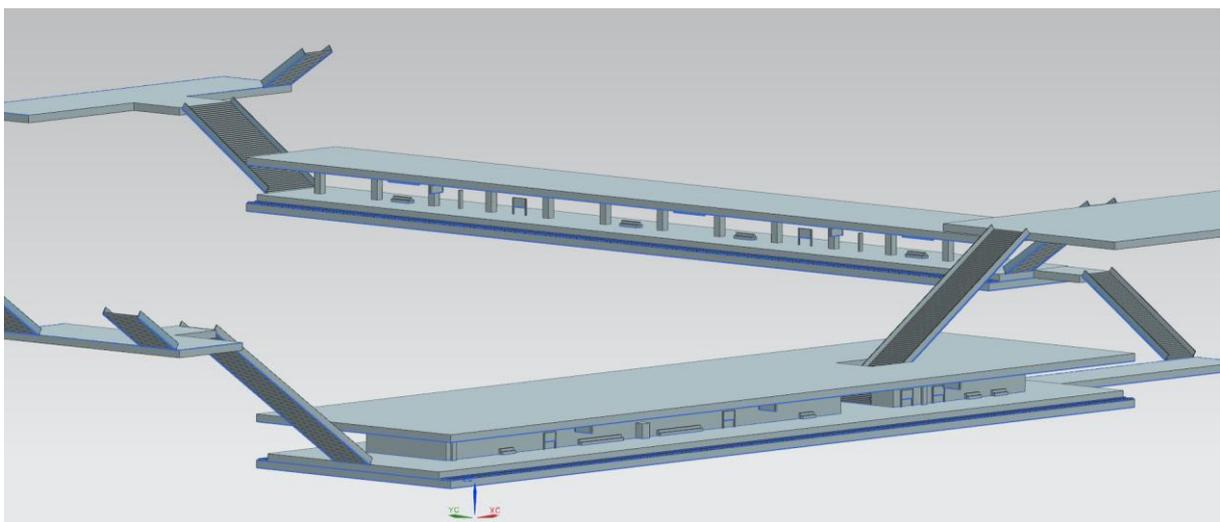


Abbildung 76: Prozedurales Modell des Münchner U-Bahnhofs Odeonsplatz mit den zwei Bahnsteigen der Linien U4/U5 (in der Abbildung unten) und der Linien U3/U6 (in der Abbildung oben)

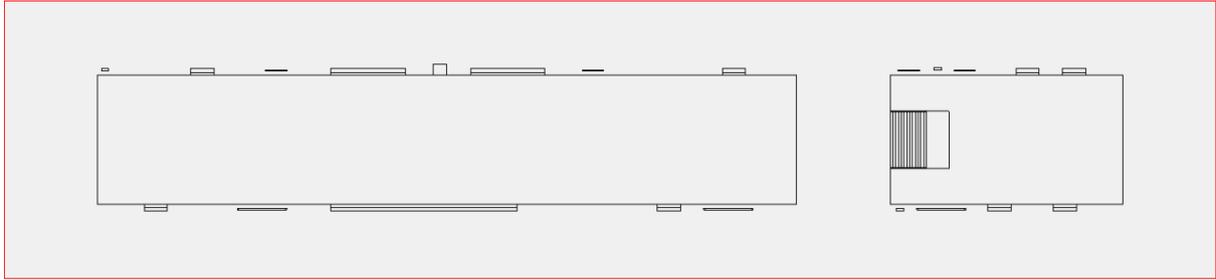


Abbildung 77: Aus dem Modell des U-Bahnhofs Odeonsplatz im Bereich der Linien U4/U5 extrahiertes Fußgängerszenario

die Zugang zu den Linien U4/U5 bzw. U3/U6 gewähren.

Das aus diesem Modell nach der Wahl der unteren Bahnsteigfläche (der Linien U4/U5) automatisch erzeugte zweidimensionale Szenario ist in Abbildung 77 dargestellt. In ihr lassen sich der Quergang zwischen den Gleisen und die von dort nach oben führende Treppe erkennen. Weiterhin sieht man alle Sitzbänke, Fahrplanschaukästen und auch Mülleimer am Rande des Bahnsteigs.

Hingegen zeigt Abbildung 78 das dreidimensionale Modell des S- und U-Bahnhofs Marienplatz in München und Abbildung 79 ein daraus generiertes Fußgängerszenario, welches den darin enthaltenen U-Bahnbahnsteig darstellt. Deutlich kann man darin die zurückversetzten Gänge erkennen, die den Fußgänger zusätzlichen Bewegungsraum bieten. Diese Gänge wurden erst gut 30 Jahre nach der Eröffnung des Bahnhofs im Vorfeld der Fußballweltmeisterschaft 2006 nachträglich in einem sich über Jahre erstreckenden Prozess integriert, da es im Alltagsbetrieb zu permanenten Stauungen beim Ein- und Ausstieg gekommen war. Genau derartige Situationen sollte man mit der vorgestellten Strategie bei einer Simulation in einer frühen Modellierungs- und Planungsphase bemerken und somit vermeiden können.

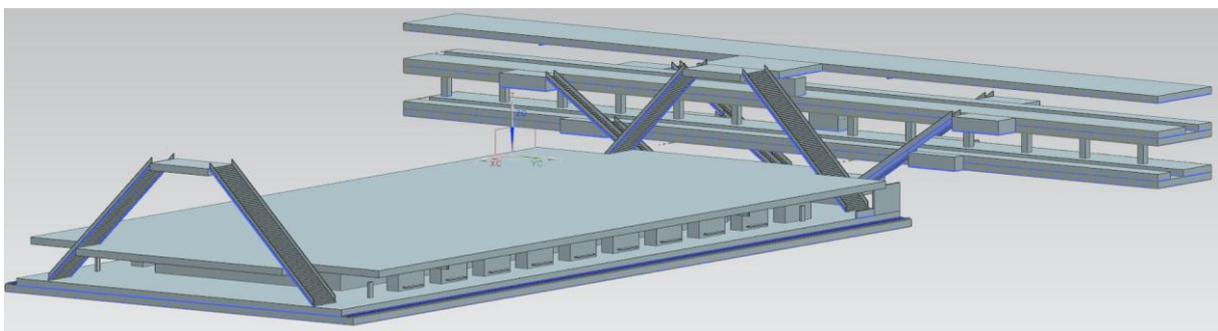


Abbildung 78: S- und U-Bahnhof München Marienplatz. Vorne sieht man die U-Bahngleise der Linien U3 und U6; dahinter sind in zwei Etagen die Bahnsteige der S-Bahn-Stammstrecke München in Richtung Hauptbahnhof und in Richtung Ostbahnhof zu sehen.

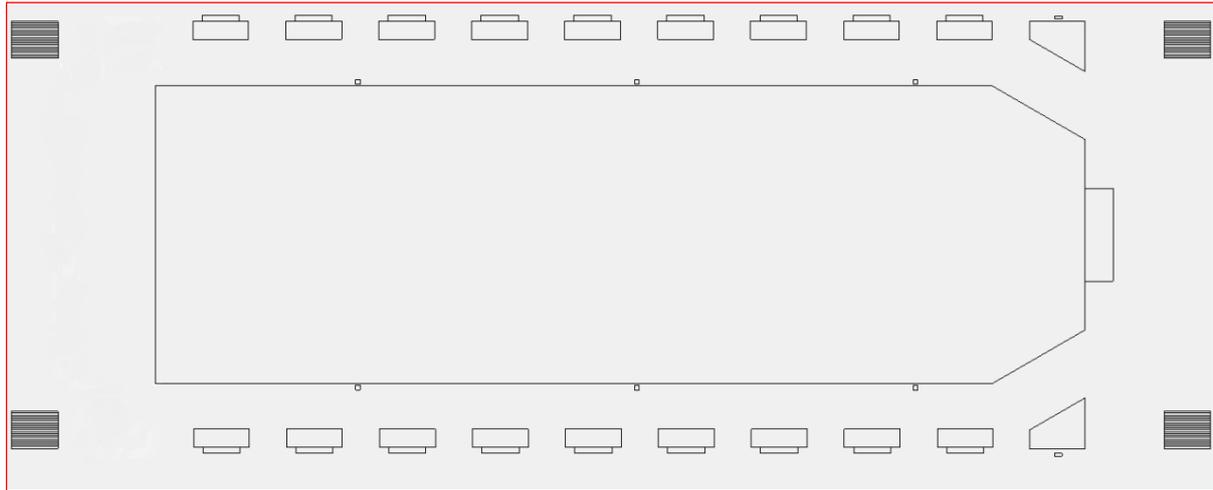


Abbildung 79: Zweidimensionales Szenario des U-Bahnhofs München Marienplatz, generiert aus dem in Abbildung 78 dargestellten Modell

Exemplarisch wurde ein leicht vereinfachtes Modell der U-Bahnstation Theresienstraße in München (Abbildung 72) genutzt, um konkrete Simulationen durchzuführen. Die U-Bahnstation Theresienstraße befindet sich im Münchner Zentrum nahe dem Stammgelände der Technischen Universität München TUM und auch nahe einiger Institute der Ludwig-Maximilians-Universität LMU. Ein typisches Szenario am Bahnhof ist die morgendliche Ankunft eines hauptsächlich mit Studentinnen und Studenten gefüllten Zugs, die als Ziel das nahe gelegene TUM-Gelände und die Institute der LMU haben. Um diesen Ausstiegsvorgang zu simulieren, wurde das oben dargestellte dreidimensionale Modell (Abbildung 72) automatisiert in ein zweidimensionales Eingangsszenario des Simulators *MomentUMV2* transformiert und nach geringen Anpassungen die Simulation gestartet. Dabei sollen die durchgeführten Simulationen einen ersten groben Eindruck für das vorliegende Szenario vermitteln, für detaillierte Untersuchungen müssen entsprechende (Simulations-)Spezialisten herangezogen und insbesondere Normen und Richtlinien zur Simulation derartiger Szenarien beachtet wer-

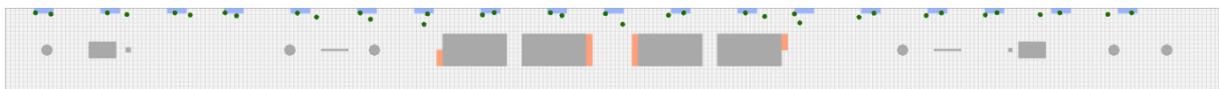


Abbildung 80: Simulation des Ausstiegsvorgangs am U-Bahnhof Theresienstraße; zu Beginn befinden sich nur einzelne Aussteigende am Bahnsteig.

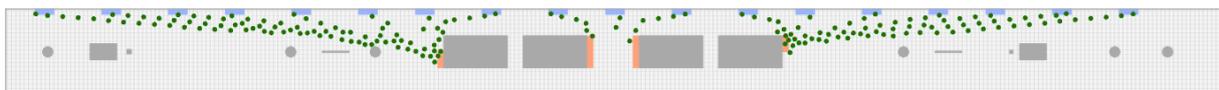


Abbildung 81: Simulation des Ausstiegsvorgangs am U-Bahnhof Theresienstraße; im Laufe der Zeit bilden sich deutliche erkennbare Bahnen in Richtung Treppen- und Rolltreppen.

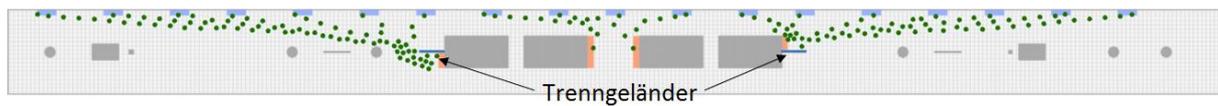


Abbildung 83: Eine leichte Veränderung des Modells aus Abbildung 81 durch ein Trenngeländer zwischen aufwärts- und abwärtsführender Rolltreppe.

den (siehe auch RiMEA – Richtlinien für Mikroskopische Entfluchtungs-Analysen⁴³).

Hier zeigen Abbildung 80 und Abbildung 81 zunächst die Fußgängerströme zu Beginn des Ausstiegsvorgangs bzw. nachdem sich deutlich sichtbare Bewegungsbahnen der Fußgänger herausgebildet haben. In blauer Farbe sind hier die Positionen der U-Bahntüren, also die Ausstiege, in oranger Farbe hingegen die Anfänge der Treppen und Rolltreppen dargestellt. Deutlich erkennt man hier auch die erhöhte Fußgängerdichte direkt vor den Rolltreppen. In Abbildung 83 ist in das Modell zusätzlich ein Trenngeländer an das Ende zwischen aufwärts- und abwärtsführender Rolltreppe eingefügt worden, so dass Kollisionen zwischen Fußgängern, die sich zum Bahnsteig begeben, und denen vermieden werden, die den Bahnhof verlassen. Dieser Effekt wird durch die Simulation deutlich bestätigt. Die Bereiche um den Rolltreppeneinstieg mit und ohne Geländer aus den Beispielen in Abbildung 81 und Abbildung 83 sind nochmals als Ausschnitt vergrößert in Abbildung 82 dargestellt.

Die Ergebnisse der Simulationsläufe sind in Tabelle 12 aufgeführt. Dabei wurde die Anzahl der aussteigenden Personen variiert. So wurden während des Ausstiegsvorgangs 90, 180, 270, 540 oder 810 Personen angenommen. 810 Personen entsprechen dabei gerade einem komplett besetzten U-Bahnzug, also der maximal möglichen Anzahl an aussteigenden Personen. Ein Ausstiegsvorgang wird dabei als abgeschlossen angesehen, wenn 95% der Personen den Bahnsteig verlassen haben, da man immer einzelne Nachzügler annimmt. Für jede Personenanzahl wurden zehn Simulationsläufe ausgeführt und dann der Mittelwert gebildet. Die Ausstiegszeiten in Abhängigkeit der Anzahl der aussteigenden Personen sind in Tabelle 12 darge-

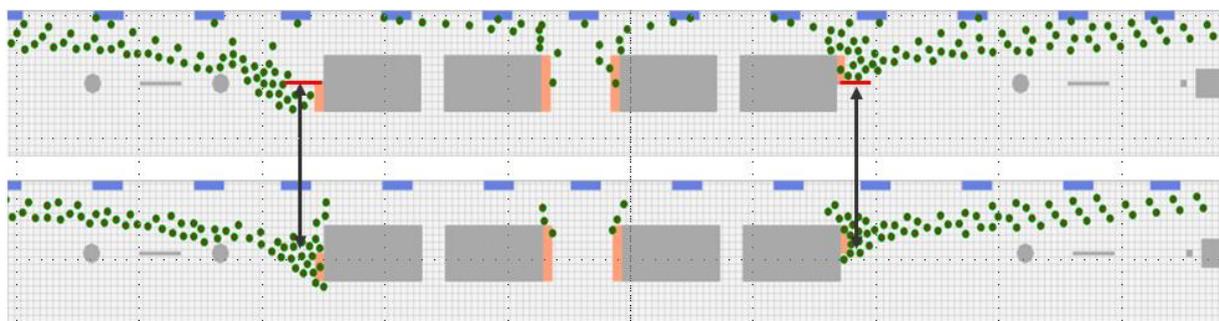


Abbildung 82: Der Bereich nahe der Rolltreppen gemäß den vorherigen Beispielen mit und ohne zusätzliches Trenngeländer

⁴³ <http://www.rimea.de/>

stellt.

Anzahl der Personen	Dauer des Ausstiegsvorgangs [s] bei zehn Simulationsläufen	Mittelwert [s]
$0,95 \cdot 90 \approx 86$	42,3/49,0/42,6/46,0/42,7/41,4/42,2/40,9/42,6/46,8	43,7
$0,95 \cdot 180 \approx 171$	50,5/51,2/45,7/42,6/50,4/55,2/47,9/53,4/48,9/44,9	49,1
$0,95 \cdot 270 \approx 257$	55,5/55,3/55,4/54,2/57,4/55,4/61,0/57,1/50,1/56,6	55,8
$0,95 \cdot 540 \approx 513$	70,6/73,5/74,3/73,9/75,5/71,9/79,5/73,7/75,8/72,3	74,1
$0,95 \cdot 810 \approx 770$	101,1/97,3/99,0/99,7/98,8/102,7/103,5/105,7/ 102,3/101,9	101,2

Tabelle 12: Dauer von Simulationsläufen für den Ausstiegsvorgang an der U-Bahnhaltestelle Theresienstraße bei unterschiedlichen Anzahlen von aussteigenden Personen.

Anzahl der Personen	Dauer des Ausstiegsvorgangs [s] bei zehn Simulationsläufen	Mittelwert [s]	Zunahme gegenüber Tabelle 12 [%]
$0,95 \cdot 90 \approx 86$	42,1/45,4/41,5/46,5/42,7/46,3/ 40,8/44,2/41,2/47,9	43,9	0,46
$0,95 \cdot 180 \approx 171$	56,0/46,9/48,1/47,6/50,8/49,9/ 49,7/46,5/48,4/56,5	51,0	3,87
$0,95 \cdot 270 \approx 257$	60,9/59,2/57,6/59,3/57,8/59,9/ 55,9/55,5/61,1/54,7	58,2	4,30
$0,95 \cdot 540 \approx 513$	87,3/93,7/89,4/93,2/89,1/96,0/ 90,5/87,9/93,1/86,5	90,7	22,4
$0,95 \cdot 810 \approx 770$	130,1/135,1/133,6/128,7/134,8/ 136,2/126,9/131,1/130,2/131,2	131,8	30,2

Tabelle 13: Dauer von Simulationsläufen für den Ausstiegsvorgang an der U-Bahnhaltestelle Theresienstraße nach Sperrung der mittleren Treppen in Folge von durchzuführenden Wartungsarbeiten.

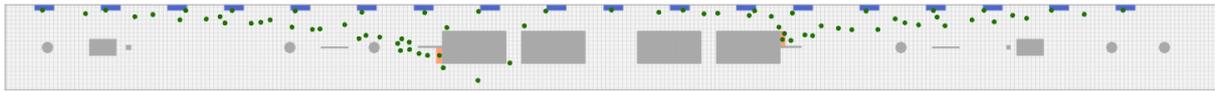


Abbildung 84: Nach einer Sperrung der beiden mittleren Rolltreppen verlagert sich die Bahnen der Fußgänger auf die beiden äußeren Ausgänge (Vergleiche Abbildung 80).

In einem weiteren Simulationsszenario des U-Bahnhofs Theresienstraße wurde eine Sperrung der beiden mittleren Treppen etwa infolge von durchzuführenden Wartungsarbeiten angenommen. Abbildung 84 zeigt die sich ändernden Bahnen der Fußgänger, denen jetzt nur noch die beiden äußeren Rolltreppen als Ausgang zur Verfügung stehen. Wie zuvor wurden für verschiedene Personenanzahlen die Ausstiegsvorgänge simuliert. Die Ausstiegszeiten in Abhängigkeit der Anzahl der austiegenden Personen sind in Tabelle 13 dargestellt, so dass diese Ergebnisse direkt mit denen der vorangegangenen Simulationen (ohne Rolltreppensperrung – siehe Tabelle 12) verglichen werden können.

Bei einer geringen Anzahl an austiegenden Personen ist zunächst kaum eine Änderung in den Ausstiegszeiten zu beobachten, da es hier zu keinerlei Staus vor den Rolltreppen kommt. Denn die zwei äußeren Rolltreppen können die zusätzlichen Personen ohne weitere Probleme aufnehmen. Bei der Interpretation der Ergebnisse ist insbesondere zu beachten, dass der Großteil der Fußgänger den Bahnhof über die äußeren Treppen verlässt wie man es auch ganz deutlich obigen Abbildungen entnehmen kann. Erhöht man jedoch die Anzahl an austiegenden Personen, so nehmen die simulierten Ausstiegzeiten stetig zu. Beim simulierten Maximalwert von 810 Personen, ergibt sich so eine um knapp ein Drittel erhöhte Ausstiegsdauer, da es zu deutlichen Stauungen vor den Rolltreppenzugängen kommt. Diese Situation ist in Abbildung 85 klar zu beobachten.

Das zuletzt gewählte Beispiel demonstriert die Stärke des präsentierten Ansatzes im auf die Bauphase folgenden Alltagsbetrieb des Bahnhofs.

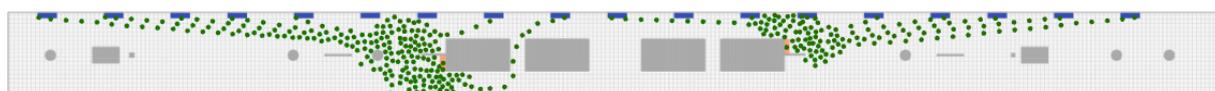


Abbildung 85: Bei einer hohen Anzahl an austiegenden Personen kommt es zu deutlichen Staus vor den zwei (noch) begehbaren Rolltreppen.

6.2.5. Zusammenfassung zur Integration von Personenstromsimulatoren

Bei der Bewertung der Alltagstauglichkeit von U-Bahnhöfen können Simulationen von Fußgängerströmen den planenden Ingenieur sehr gut unterstützen. Die Integration der Modelldaten in die spezifischen Werkzeuge ist meist ein komplexer und fehleranfälliger Prozess.

Deshalb wurde ein Konzept entwickelt, mit dessen Hilfe man die dreidimensionalen prozeduralen Modelle, die über die Kollaborationsplattform erstellt wurden, direkt in die Eingangsformate typischer Werkzeuge zur Personenstromsimulation transformieren kann. Die zugrunde liegenden Techniken und Algorithmen wurden detailliert besprochen und einige Beispiele zur Plausibilisierung präsentiert.

Mit diesem Konzept ist es nun im Gegensatz zu anderen Ansätzen möglich, direkt die Simulationsergebnisse aus den Modellen zu gewinnen, die auf der Kollaborationsplattform hinterlegt sind, und das Ergebnis dieser Simulationen dann direkt in die weitere Modellierung einfließen zu lassen. Die Alltagstauglichkeit dieses Ansatzes konnte abschließend in einigen Beispielsimulationen eindrucksvoll belegt werden.

7. Zusammenfassung und Ausblick

In dieser Arbeit wurden Möglichkeiten für die Verbesserung der Zusammenarbeit verschiedener Experten untersucht, die an der Planung von trassenbasierten Infrastrukturbauwerken beteiligt sind. Dabei wurden zwei Schwerpunkte gesetzt: zum einen die synchrone, geometrische Modellierung, zum anderen die simulationsgestützte Evaluierung so entstandener Modelle.

Synchrone Modellierung über die Plattform Kollabserver

Für beide Punkte grundlegend war die Konzeption der Plattform *Kollabserver*. Diese wurde als zentrale Instanz für einen Modellierungsprozess entwickelt, in dem die verschiedenen Experten die CAD-Werkzeuge ihrer Wahl wie etwa Siemens NX oder Autodesk Inventor verwenden können.

Die Basis für diesen gemeinsamen Prozess bildet ein multi-skaliges prozedurales Geometriemodell. In dieses wurde fundamental das Konzept verschiedener, flexibel definierbarer Levels of Detail (LoDs) integriert, welche die Sichtweisen der unterschiedlichen Experten abbilden. Die Plattform bietet den verschiedenen Teilnehmern Zugriff auf das Gesamtmodell, ermöglicht aber auch eine Extraktion von Teilmodellen sowie die Betrachtung des Modells in verschiedenen Detaillierungsstufen (LoDs). Insbesondere ist sie für die Sicherung der Konsistenz zwischen dem zentral hinterlegten Modell und seinen lokalen Replikaten auf der Clientseite verantwortlich. Hierzu wurden Möglichkeiten der Nebenläufigkeitskontrolle diskutiert und ein Algorithmus entworfen, der die geforderte Konsistenz durch *Sperren* gewährleistet. Dieser Algorithmus operiert direkt auf dem Sperr-Abhängigkeitsgraphen, der sich aus dem bipartiten Modellierungsgraphen ableiten lässt. Hier konnte gezeigt werden, dass bei der Modifikation eines Elements nur die Elemente des Graphen für den Zugriff durch andere Benutzer gesperrt werden müssen, die sich in der Zusammenhangskomponente des Graphen befinden, die das zu modifizierende Element enthält. So können andere Benutzer gleichzeitig Elemente des Gesamtmodells ändern, solange diese nur anderen Zusammenhangskomponenten angehören. Außerdem wurde dargelegt, dass dieses Sperrkonzept auch gültig bleibt, wenn einzelne Benutzer nur auf extrahierten Teilmodellen operieren oder Modelle in ganz bestimmten LoDs betrachten.

Um die Praxistauglichkeit des präsentierten Ansatzes zu belegen, wurde das ursprünglich für die Modellierung von Tunnelbauwerken geschaffene Konzept bei der Modellierung von Bahnhofsmoellen erfolgreich eingesetzt, wobei eine Reihe detaillierter Modelle konkret erstellt wurde.

Der *Design Intent* des modellierenden Ingenieurs kann mit Hilfe von Constraints zumindest teilweise in prozeduralen Modellen abgebildet werden. Um hier vorhandene Defizite zu verringern, wurde eine Strategie zur Integration von kollaborativen Regeln vorgestellt. Diese

erlaubt es dem Benutzer, spezifische Beziehungen zwischen einzelnen Elementen des Modells zu definieren und systemneutral abzuspeichern, so dass diese Regeln im Gegensatz zu bisherigen Ansätzen insbesondere in einem Prozess verwendet werden können, in welchem Benutzer unterschiedlicher CAD-Werkzeuge teilnehmen.

Bei der Modellierung von Infrastrukturbauwerken sind die geographischen und geotechnischen Rahmenbedingungen von entscheidender Bedeutung. Informationen über diese werden heute häufig von *Geo Web Services* zur Verfügung gestellt, ihre Integration in gängige CAD-Werkzeuge jedoch nur unzureichend unterstützt.

Zur Beschreibung dieser Services bieten sich *Web Service Context Documents* an, die auf der Plattform hinterlegt werden können. Im Rahmen dieser Arbeit wurde eine Methode zur einheitlichen, dynamischen und generischen Einbindung von den durch *Geo Web Services* zur Verfügung gestellten Informationen entwickelt. Die Grundlage dieser Methode ist eine Containerstruktur, die eine einheitliche Behandlung von prozeduralen Geometriemodellen und Geodaten zulässt. So werden Möglichkeiten geschaffen, verschiedene Daten simultan zu visualisieren und zu bearbeiten, sowie *Web Processing Services* zu integrieren, die Analysen auf den zugrunde liegenden Daten erlauben.

Die Grundlage für die Modellierung eines U-Bahn-Tunnels bildet das Trassenmodell. Die Verwaltung dieser Modelle wurde als zentrale Komponente in die Plattform integriert und bewusst von der Verwaltung der darauf basierenden Geometriemodelle getrennt. Im Gegensatz zu anderen Ansätzen lässt es die hier präsentierte Methode zu, Änderungen an der Trasse vorzunehmen, selbst wenn bereits mit der Modellierung von Bauwerken begonnen wurde, die auf dieser basieren.

Zudem wurde eine Zerlegung des Gesamtmodells gemäß der einzelnen Trassenabschnitte vorgeschlagen, so dass eine synchrone Zusammenarbeit auf den so extrahierten Teilmodellen möglich wird. Auch hier kann die Konsistenz zwischen Teilmodellen und dem Gesamtmodell von der Plattform mit Hilfe des präsentierten Algorithmus gewahrt werden.

Einbindung von Simulationssoftware

Bei der Bewertung von Bauwerksmodellen können die Ergebnisse von Simulationen durch den Experten häufig hilfreich verwendet werden. Anhand zweier Beispiele wurden Strategien diskutiert, Simulationssoftware über die Plattform direkt in den Planungs- und Modellierungsprozess einzubinden. Zum einen wurde eine Energiesimulation entwickelt und integriert, zum anderen eine Methode geschaffen, Fußgängersimulationen zur Bewertung von Bahnhofsbauwerken an die Plattform zu koppeln. In beiden Fällen können die aus den Berechnungen resultierenden Ergebnisse unmittelbar im Modellierungsprozess gewinnbringend genutzt werden.

Das Alignmentmodell ist nicht nur Grundlage für die Modellierung eines Tunnels, sondern auch für den Energieverbrauch der später auf dieser Strecke verkehrenden Züge: So bestimmt der Höhenplan den Bedarf an potentieller Energie, während der Lageplan ursächlich für den Kurvenwiderstand ist. Im Rahmen dieser Arbeit wurden ein physikalisches Modell zur Berechnung des Energieverbrauchs und eine darauf basierende Softwarekomponente entwickelt. Ein Konzept für deren Integration in die Plattform wurde besprochen, welches es dem Planer erlaubt, den Einfluss seiner Änderungen am Trassenverlauf auf den Energieverbrauch unmittelbar einzusehen und zu interpretieren. Hier wurde zudem gezeigt, dass bereits geringe Modifikationen im Trassenverlauf, insbesondere im Höhenprofil zu signifikanten Änderungen im Energieverbrauch führen können. Die Energieberechnung ohne spürbare Verzögerung ist für den Ingenieur grundlegend, um manuelle Optimierungen an der Trasse vorzunehmen. Um außerdem eine strukturierte Vorgehensweise bei der Suche nach einem optimalen Trassenverlauf zu etablieren, wurde eine Methode vorgestellt, die auf der automatischen Variation der *Points of Vertical Intersection* (PVI) basiert. Da die Berechnung aller sich durch die Variation der PVI ergebenden Höhenprofile im Allgemeinen sehr zeitaufwendig ist, wurde zum Auffinden von Näherungslösungen die Verwendung eines Optimierungsalgorithmus vorgeschlagen. Eine Adaption eines solchen Algorithmus wurde diskutiert, wobei die Berechnungszeiten der energieoptimalen Profile von mehreren Stunden auf einige Sekunden reduziert werden konnten.

Bei der Bewertung von Bahnhofsbauwerken spielt die Simulation des Fußgängerverhaltens eine zentrale Rolle. Zum einen sollen der Alltagsbetrieb, zum anderen Evakuierungsszenarios untersucht und insbesondere die notwendigen Evakuierungszeiten abgeschätzt werden. Üblicherweise arbeiten Fußgängersimulatoren auf zweidimensionalen Eingangsdaten, die ein Bauwerk Etage für Etage abbilden. Um dreidimensionale prozedurale Bahnhofsmo-delle direkt den Simulationswerkzeugen zuführen zu können, wurde ein Algorithmus präsentiert, der durch die Projektion der einzelnen Volumenkörper des Bahnhofsmo-dells (halb-)automatisiert ein zweidimensionales Eingangsszenario für diese Simulatoren generiert. So wird eine unmittelbare Analyse der Bahnhofsmo-delle durch den Experten möglich, der diese Ergebnisse dann wiederum direkt in den Modellierungsprozess mit einbeziehen kann.

Ausblick

Hinsichtlich der präsentierten Plattform ergeben sich eine Vielzahl von weiteren Einsatzmöglichkeiten und Verbesserungen: In Bezug auf die prozedurale synchrone Modellierung bietet sich die Untersuchung und Integration weiterer Operationen sowie eine Erweiterung des regelbasierten Ansatzes in Richtung eines *Knowledge Based Engineering Systems* an. Im Bereich der Tunnelmodellierung wurde konkret die Integration fünf verschiedener Detaillierungsstufen (LoDs) in das prozedurale Modell vorgeschlagen, das die Grundlage des synchronen Modellierungsprozesses bildet. Hinsichtlich der Modellierung von Fernstraßen und

Brücken, sowie bei Bahnhofsgebäuden bietet sich eine dementsprechende Untersuchung und Definition sinnvoller Detaillierungsstufen unter Durchführung entsprechender Tests an. Bei der Nebenläufigkeitskontrolle ist eine möglichst geringe Anzahl der zu sperrenden Elemente für eine erfolgreiche Zusammenarbeit von großem Vorteil. Wie in Kapitel 3.7 angedeutet, ist es eine interessante Frage, wie man diese Menge reduzieren und allgemein eine untere Schranke für sie angeben kann. Bei der Modellierung des Alignments können sich systematische Untersuchungen hinsichtlich der dabei zulässigen Variationen in einer späten Modellierungsphase als gewinnbringend für den diskutierten Modellierungsprozess erweisen. Die vorgestellte Energiesimulation versucht, eine Optimierung der Trasse durch eine Verbesserung des Höhenplans zu erreichen. Im Gegensatz dazu wurden bereits zahlreiche Vorschläge hinsichtlich der Optimierung des Geschwindigkeitsprofils gemacht. Eine Kombination der hier präsentierten Ansätze mit den Ansätzen zur Optimierung des Geschwindigkeitsprofils sollte in Betracht gezogen werden und dürfte zu insgesamt noch besseren Resultaten führen. Bei der Bewertung von Bahnhofsbauelementen spielen zudem Entrauchungssimulationen eine entscheidende Rolle, aber auch Überflutungssimulationen scheinen sinnvoll, wenn man etwa an die Überflutung der New Yorker U-Bahn im Jahr 2012 denkt. Eine Integration dieser Art von Strömungssimulationen, die direkt auf den Daten des dreidimensionalen Modells operieren, wäre sicherlich eine interessante Herausforderung. Als nächsten Schritt könnte man sich dann eine Kopplung von Entrauchung, Flutung und Fußgängersimulation vorstellen. Auch eine Kopplung von Fußgänger und Energiesimulation ist möglich: So kann man sich vorstellen, dass eine Verzögerung beim Ausstiegsvorgang zu einem Bremsvorgang des nachfolgenden Zuges führt, der zu einer erneuten (energieintensiven) Beschleunigungsphase führt.

Anhang

Konstruktionsbeispiel für einen typischen Trassenabschnitt

Im Folgenden geben wir exemplarisch ein Verfahren zur Konstruktion eines aus der Elementfolge Gerade, Klothoide, Kreisbogen, Klothoide und Gerade bestehenden Trassenabschnitts an. Dabei wird konkret gezeigt, wie man bei zwei vorgegebenen Geradenstücken passende Klothoiden an einen Kreisbogen legen kann, dessen Radius und Öffnungswinkel vorgegeben werden, während seine konkrete Lage zunächst unbestimmt ist. Dieses Vorgehen kann ebenso zur Anpassung des Radius eines bereits existierenden Kreisbogens innerhalb dieser Elementfolge verwendet werden. Diese Art von Modifikation ist typisch für die in Kapitel 5 beschriebene Modellierungsphase, in der ein Planer noch Anpassungen an der (bereits weitgehend bestimmten) Trasse vornehmen darf, obwohl schon mit der Modellierung der auf der Trasse basierenden Bauwerke begonnen wurde. Sowohl in dem in Abbildung 37 als auch in Abbildung 44 dargestellten Beispielen tritt diese Situation mehrfach auf.

Zudem wird durch das vorzustellende Verfahren klar, dass eine Veränderung des Radius oder des Klothoidenparameters (bei passender Wahl der Parameter) möglich ist, ohne dabei andere Elemente der Gesamttrasse als die fünf Elemente dieser Gruppe zu beeinflussen.

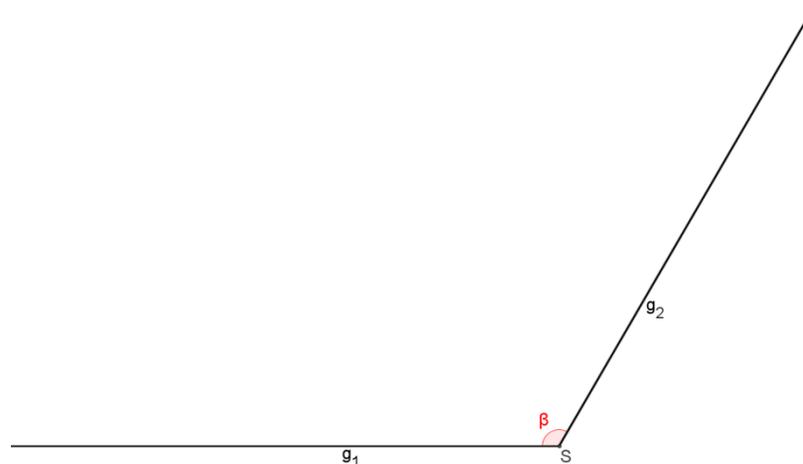


Abbildung 86: Ausgangssituation zur Konstruktion des Übergangs zwischen zwei Geradenstücken und einem Kreis durch zwei Klothoiden

Die Ausgangssituation für diese Problemstellung ist beispielhaft in Abbildung 86 dargestellt, in der sich die zwei Geradenstücke g_1 und g_2 im Punkt S unter dem Winkel β , in der Abbildung konkret mit $\beta = 80^\circ$, schneiden. O. B. d. A. wurde die Richtung der Geraden g_1 horizontal gewählt, so dass diese später mit der Richtung der x-Achse des Koordinatensystems übereinstimmt, in dem die erste Klothoide der Elementfolge konstruiert werden soll. Der Klothoidenstartpunkt O fällt dann nach Konstruktion mit dem Koordinatenursprung zusammen. Die allgemeine Lösung erhält man hieraus durch eine Rotation (der Gesamtanordnung) um den Ursprung und eine sich daran anschließende Translation. Wir geben hier zur konkre-

ten Berechnung den Öffnungswinkel θ des einzufügenden Kreisbogens mit $\theta = 60^\circ$ und seinen Radius mit $R = 200m$ vor und zeigen dabei allgemein, dass alle weiteren Größen bereits durch diese drei Angaben festgelegt sind. Das Ergebnis der Konstruktion und alle relevanten Bezeichnungen sind in Abbildung 87 dargestellt. Schritt für Schritt wird nun die zugrunde liegende Konstruktionsfolge angegeben.

Zunächst sind β und θ vorgegebenen, wobei nur Angaben mit $\frac{\theta}{2} + \frac{\beta}{2} \leq 90^\circ$, also mit $\theta + \beta \leq 180^\circ$ sinnvoll sind, da im Dreieck ΔASM aufgrund der Innenwinkelsumme $90^\circ + \frac{1}{2} \cdot \beta + \frac{1}{2} \cdot \theta + \tau = 180^\circ$ gelten muss.

Damit folgt unmittelbar $\tau = \sphericalangle AMP = 180^\circ - 90^\circ - \frac{\theta}{2} - \frac{\beta}{2} = 90^\circ - \frac{\theta}{2} - \frac{\beta}{2}$. Bei der konkret

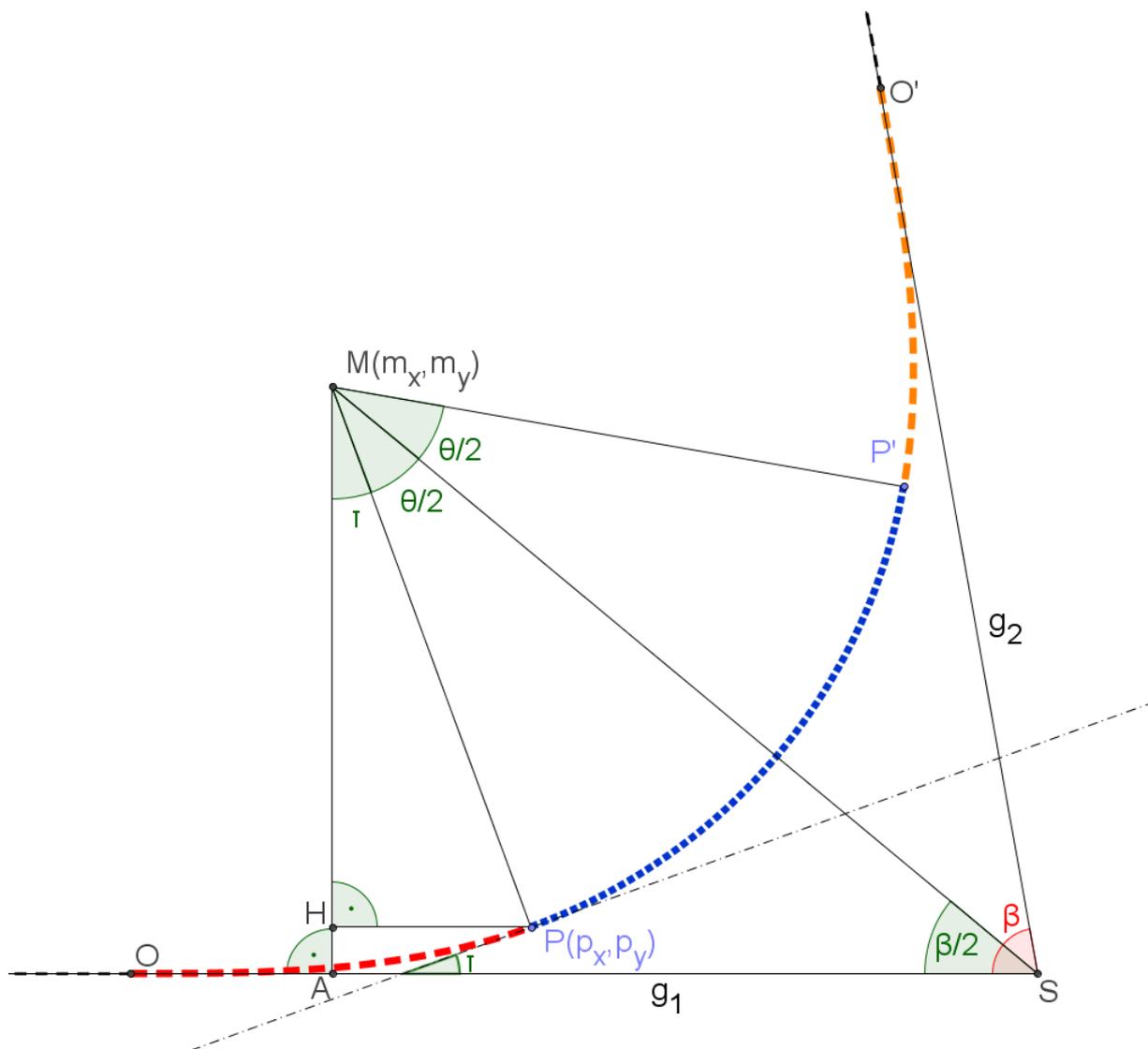


Abbildung 87: Zwischen den zwei Geradenstücken $[SO]$ und $[SO']$ wurde eine Folge aus Klothoide, Kreisbogen und Klothoide bei vorgegebenen Radius R und Öffnungswinkel θ des Kreisbogens eingefügt. Die erste Klothoide ist rot-gestrichelt, der Kreisbogen blau-gepunktet und die zweite Klothoide orange-gestrichelt dargestellt.

vorgegebenen Wahl von $\beta = 80^\circ$ und $\theta = 60^\circ$ ergibt sich $\tau = 20^\circ$. Dieser Winkel stimmt mit dem Schnittwinkel der Tangente an die Klothoide in ihrem Endpunkt P und der Achse OS überein, da diese beiden spitzen Winkel paarweise aufeinander senkrecht stehen.

Aus der Grundgleichung für eine Klothoide lassen sich für den Tangentenschnittwinkel die Beziehung $\tau_{rad} = \frac{L}{2 \cdot R}$ und für den Klothoidenparameter die Beziehung $A = \sqrt{R \cdot L}$ herleiten. Eine detaillierte Herleitung der Parameterdarstellung einer Klothoide und der grundlegenden Beziehungen zwischen den einzelnen charakteristischen Größen findet man etwa bei (Brown 2008) oder (Kaczmarczyk et al. 2010). Damit ergibt sich für die Länge L des Bogens zwischen den Punkten O und P allgemein

$$L = \tau_{rad} \cdot 2 \cdot R = \frac{2 \cdot \pi \cdot \tau_{deg}}{360^\circ} \cdot 2 \cdot R, \quad (23)$$

im gegebenen Beispiel der Wert $L \approx 419m$, und für den Klothoidenparameter A der Wert $A = \sqrt{R \cdot L} \approx 501m$.

Hiermit berechnen wir nun die x - und y -Koordinate p_x bzw. p_y des Klothoidenendpunkts $P(p_x; p_y)$ in einem Koordinatensystem mit Ursprung O und x -Achse in Richtung der Tangente am Klothoidenstartpunkt (siehe auch Abbildung 88). Die Grundlage dafür ist die Parameterdarstellung einer Klothoide in Abhängigkeit des Parameters l , der die Kurvenlänge angibt

$$b(l) = \begin{pmatrix} x(l) \\ y(l) \end{pmatrix} = A \cdot \pi \cdot \int_0^l \begin{pmatrix} \cos\left(\frac{\pi \cdot t^2}{2}\right) \\ \sin\left(\frac{\pi \cdot t^2}{2}\right) \end{pmatrix} dt. \quad (24)$$

Die gesuchten Koordinaten ergeben sich für die zuvor bestimmte Länge L des Übergangsbogens zu

$$\begin{pmatrix} p_x \\ p_y \end{pmatrix} = b(L) = \begin{pmatrix} x(L) \\ y(L) \end{pmatrix} \approx \begin{pmatrix} 414m \\ 48m \end{pmatrix}. \quad (25)$$

Die konkrete Berechnung des Integrals kann dabei durch die Taylornäherungen für die im Integranden auftretenden *cosinus* und *sinus* Funktionen erleichtert werden (Kaczmarczyk et al. 2010).

Hieraus lassen sich die Koordinaten des Mittelpunkts des Kreisbogens berechnen, indem man mit den üblichen trigonometrischen Beziehungen für das bei H rechtwinklige Dreieck ΔHPM die Streckenlängen $\overline{HP} = R \cdot \sin(\tau)$ und $\overline{HM} = R \cdot \cos(\tau)$ verwendet und die Beziehungen

$$m_x = p_x - \overline{HP} \text{ und } m_y = p_y + \overline{HM} \quad (26)$$

bildet. Im Rechenbeispiel liefert dies $M(209m; 612m)$. Damit ergibt sich nun final der gesuchte Abstand s_x des Klothoidenstartpunkts zum Scheitelpunkt zu

$$s_x = m_x + \frac{m_y}{\tan\left(\frac{\beta}{2}\right)} \approx 169m. \quad (27)$$

Die Koordinaten von Startpunkt und Endpunkt der zweiten Klothoide, die Punkte P' bzw. O' , ergeben sich zu guter Letzt durch einen Spiegelung der Punkt O bzw. P an der Achse SM .

Im zuvor beschriebenen Verfahren haben wir den Öffnungswinkel θ und Radius R vorgegeben. Völlig analog verläuft die Konstruktion bei Vorgabe des Öffnungswinkels θ und Klothoidenparameters A , wenn man zusätzlich beachtet, dass $L^2 = 2 \cdot \tau_{rad} \cdot A^2$ gilt.

Unterteilt man den Öffnungswinkel θ nicht gerade in zwei gleiche Teile, sondern lässt eine unterschiedliche Unterteilung dieses Winkels vor, so ergibt sich ähnlich dem zuvor beschriebenen Fall der allgemeinere Fall einer Folge aus Gerade, Klothoide, Kreisbogen, Klothoide und Gerade mit zwei nicht deckungsgleichen Klothoiden.

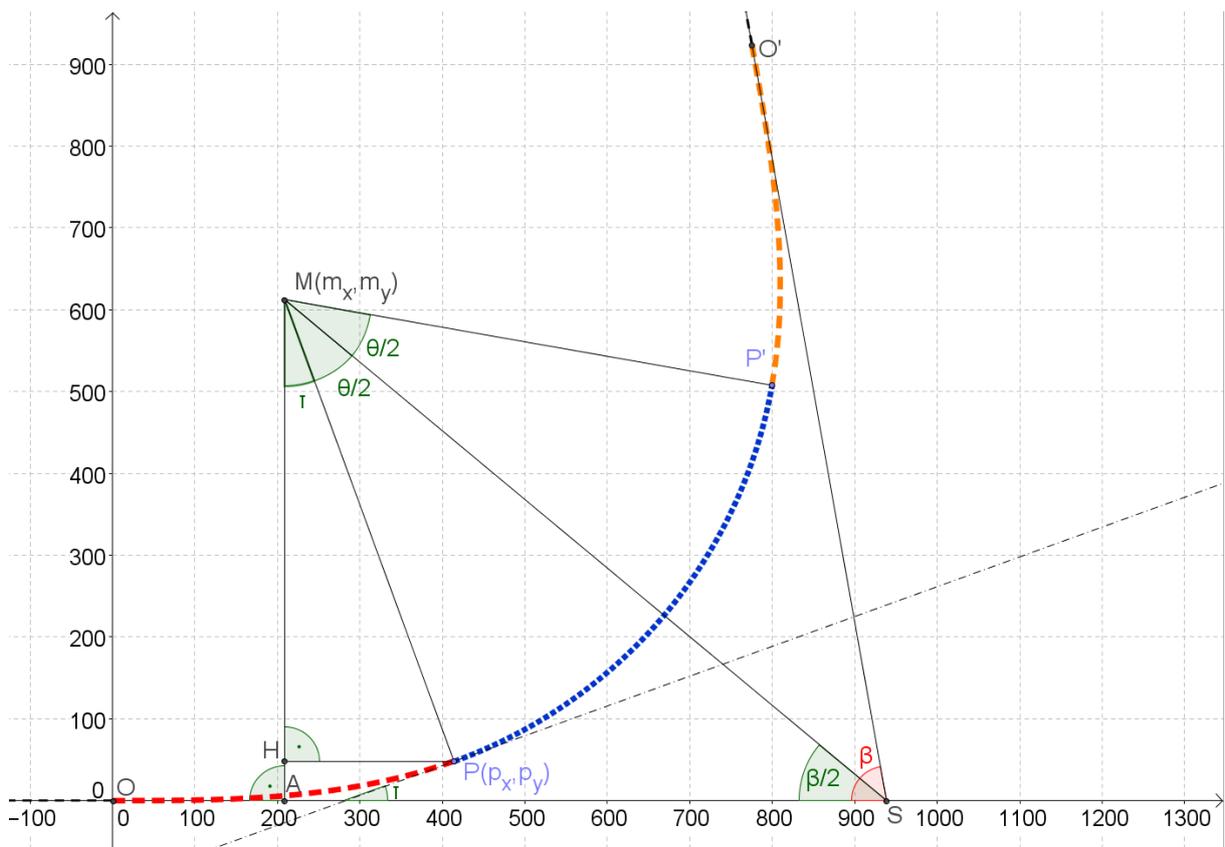


Abbildung 88: Das in Abbildung 87 gegebene Beispiel unter Hinzunahme des Koordinatensystems, dessen genaue Position sich erst im Laufe der Berechnungen ergibt

Ein Algorithmus zum Auffinden von Zyklen im Skizzengraphen

Im Folgenden wird in Pseudocode der Algorithmus angegeben, dessen Verwendung kurz im Rahmen der Transformation der dreidimensionalen prozeduralen Modelle in Kapitel 6.2.3.3 angesprochen wurde, konkret beim Auffinden von Zyklen im Skizzengraphen.

Hauptfunktion

Für alle Knoten im Graphen

Falls Findezyklus (Knotenindex) hat Zyklus gefunden **Dann**

Verlasse Für alle Schleife

Ende Falls

Ende Für

Ende Hauptfunktion

Funktion FindeZyklus(Knotenindex)

Falls Markierung[Knotenindex] ist *Start* **Dann**

Setze Markierung[Knotenindex] auf *InBearbeitung*

Füge Knotenindex dem Zyklusarray hinzu

Für alle Nachbarknoten von Knotenindex

Falls Findezyklus(Nachbarknotenindex) liefert Rückgabe *Wahr* **Dann**

Verlasse Funktion mit Rückgabewert *Wahr*

Ende Falls

Ende Für

Setze Markierung[Nachbarknotenindex] auf *Abgearbeitet*

Entferne Nachbarknotenindex aus dem Zyklusarray

Sonst Falls Markierung[Knotenindex] ist *InBearbeitung* **Dann**

Zyklus, beginnend bei Knotenindex, wurde gefunden und im Array gespeichert

Verlasse Funktion mit Rückgabewert *Wahr*

Ende Falls

Verlasse Funktion mit Rückgabewert *Falsch*

Ende Funktion

Literatur

- Abulawi, J. (2012). *Ansatz zur Beherrschung der Komplexität von vernetzten 3D-CAD-Modellen*. Dissertation. Universität der Bundeswehr Hamburg.
- Adeli, H. (1988). *Expert systems in construction and structural engineering*. United Kingdom: Chapman & Hall.
- Aho, A.; Sethi, R.; Ullmann, J. (1999). *Compilerbau Teil 1*. Deutschland: Oldenbourg.
- Albahari, J.; Albahari, B. (2012). *C# 5.0 in a Nutshell. The Definitive Reference*. USA: O'Reilly.
- Allen, L.; Chien, S. (2016). *Examination of the Contributions of Coasting and Regenerative Braking to the Efficiency of Electric Rail Vehicles*. In Proc. the Third International Conference on Railway Technology: Research, Development and Maintenance, Cagliari, Italy.
- Amann, J.; Flurl, M., Jubierre, J.R.; Borrmann, A. (2014). *A proposal for an improved IFC 4 based alignment model for infrastructure design*. In: Proc. of the 10th European Conference on Product & Process Modeling (ECPPM), Vienna, Austria.
- Anderson, B.; Ansaldi, S. (1998). *ENGEN Data Model: a neutral model to capture design intent*. In Proc. 10th International IFIP WG 5.2/5.3 Conference PROLAMAT 98, Trento, Italy.
- AREMA – The American Railway Engineering and Maintenance-of-Way Association. *Manual for Railway Engineering*, AREMA, 2015. USA.
- Baker, C. (2014). *A review of train aerodynamics Part 2 – Applications*. *Aeronautical Journal*, Volume 118, Number 1202.
- Batita, W.; Roche, S.; Bédard, Y.; Caron, C. (2014). *Towards a Conceptual Framework for WikiGIS*. *future internet*, Volume 6, Issue 4, pp. 640-672.
- Baumgart, B. (1975). *A polyhedron representation for computer vision*. In Proc. AFIPS '75 national computer conference and exposition, USA.
- Bernstein, P.; Goodman, N. (1981). *Concurrency Control in Distributed Database Systems*. *ACM Computing Surveys*, Volume 13, Number 2, pp. 185-221.
- Bidarra, R. (1999). *Validity Maintenance in Semantic Feature Modeling*. Dissertation. Technische Universiteit Delft.

- Bidarra, R.; van der Berg, E.; Broonsvoort, W. (2001). Interactive Facilities for Collaborative Feature Modeling on the Web. In Proc. 10th Portuguese Conference on Computer Graphics, Portugal.
- Bidarra, R.; Berg van den, E.; Broonsvoort, W. (2002). A Collaborative Feature Modeling System. *Journal of Computing and Information Science Engineering*, Volume 2, Issue 3, pp. 192-198.
- Bianconi, F. (2005). Towards a procedural CAD model for data exchange: problems and perspectives. 5th workshop on Design Tools and Methods in Industrial Engineering, Pisa, Italy.
- Bianconi, F.; Paolo, C.; Di Angelo, L. (2006). Interoperability among CAD/CAM/CAE systems: a review of current research trends. In Proc. Proceedings of the Geometric Modeling and Imaging — New Trends (GMAI'06), Washington D.C., USA.
- Biedermann, D.; Kielar, P.; Handel, O.; Borrmann, A. (2014). Towards TransiTUM: A Generic Framework for Multiscale Coupling of Pedestrian Simulation Models based on Transition Zones. In Proc. The Conference on Pedestrian and Evacuation Dynamics 2014 (PED 2014), Delft, Netherlands.
- Bonabeau, E.; Dorigo, M.; Theraulaz, G. (1999). *Swarm Intelligence from Natural to Artificial Systems*. USA: Oxford University Press.
- Borghoff, U.; Schlichter, J. (1995). *Rechnergestützte Gruppenarbeit*. Deutschland: Springer.
- Borrmann, A. (2007). *Computerunterstützung verteilt kooperativer Bauplanung durch Integration interaktiver Simulationen und räumlicher Datenbanken*. Dissertation. Technische Universität München.
- Borrmann, A.; Ji, Y.; Jubierre, J.; Flurl, M. (2012). Procedural Modeling: A new approach to multi-scale design in infrastructure projects. In Proc. 19th EG-ICE International Workshop on Intelligent Computing in Engineering, Herrsching, Germany.
- Borrmann, A.; Ji, Y.; Jubierre, J. (2012). Multi-scale geometry in civil engineering models: Consistency preservation through procedural representations. In Proc. of the 14th International Conference on Computing in Civil and Building Engineering (ICCCBE), Moscow, Russia.
- Borrmann, A.; Flurl, M.; Jubierre, J.R., Mundani, R.-P.; Rank, E. (2014). Synchronous collaborative tunnel design based on consistency-preserving multi-scale models. *Advanced Engineering Informatics*, Volume 28, Issue 4, pp. 499-517

- Boschetti, G.; Mariscotti, A. (2012). The parameters of motion mechanical equations as a source of uncertainty for traction systems simulation. In Proc. XX IMEKO World Congress, Busan, Republic of Korea.
- Bouma, W.; Fudos, I.; Hoffmann, C.; Cai, J.; Paige, R. (1995). A Geometric Constraint Solver. Technical Report. Department of Computer Science, Purdue University.
- Bretschneider, D. (1998). Modellierung rechnerunterstützter, kooperativer Arbeit in der Tragwerksplanung. Dissertation. Ruhr-Universität Bochum.
- Breunig, M.; Rank, E.; Schilcher, M.; Borrmann, A.; Hinz, S.; Mundani, R.-P.; Ji, Y.; Menninghaus, M.; Donaubaue, A.; Steuer, H.; Vögtle, T. (2011). Towards Computer-Aided Collaborative Subway Track Planning in Multi-Scale 3D City and Building Models. In Proc. 6th 3D Geoinfo Conference, Wuhan, China.
- Brown, H. (2008). The History of the Derivation of the Arema Spiral. In Proc. AREMA Annual Conference, Salt Lake City, Utah, USA.
- Brüderlin, B.; Roller, D. (1998). Geometric Constraint Solving and Applications. Deutschland: Springer.
- Bungartz, H.-J.; Zimmer, S.; Buchholz, M.; Pflüger, D. (2013). Modellbildung und Simulation. 2. Auflage. Deutschland: Springer Spektrum.
- Cai, G. (2005). Extending Distributed GIS to Support Geo-Collaborative Crisis Management. Geographic Information Sciences, Volume 11, Special Issue on Distributed GIS, pp. 4-14.
- Cai, X.; Li, X.; He, F.; Han, S.; Chen, X. (2012). Flexible Concurrency Control for Legacy CAD to Construct Collaborative CAD Environment. Journal of Advanced Mechanical Design, Systems, and Manufacturing, Volume 6, Number 3, pp. 324-339.
- Chaplier, J.; Nguyen That, T.; Hewatt, M.; Gallee, G. (2010). Toward a standard: RoadXML, the road network database format. In Proc. Driving Simulation Conference, Paris, France.
- Chen, Y.; Cai, Y.; Li, P.; Zhang, G. (2015). Study on Evacuation Evaluation in Subway Fire Based on Pedestrian Simulation Technology. Mathematical Problems in Engineering.
- Choi, G.-H.; Mun, D.; Han, S. (2002). Exchange of CAD Part Models Based on the Macro Parametric Approach. International Journal of CAD/CAM, Volume 2, Issue 1, pp. 13-21.

- Chu, C.-H.; Chan, Y.-H.; Wu, P. (2008). 3D streaming based on multi-LOD models for networked collaborative design. *Computers in Industry*, Volume 59, Issue 9, pp. 863-872.
- Cormen, T.; Leiserson, C.; Rivest, R.; Stein, C. (2010). *Algorithmen - Eine Einführung*. Deutschland: Oldenbourg.
- Demtröder, W. (2013). *Experimentalphysik 2: Elektrizität und Optik*. 6. Auflage. Deutschland: Springer.
- Demtröder, W. (2015). *Experimentalphysik 1: Mechanik und Wärme*. 7. Auflage. Deutschland: Springer.
- Denver, T.; Pan, L.; Benson, D. (2013). Analysis of railroad energy efficiency in the united states. MPC Report No. 13-250.
- Dixon, J.; Libardi, E.; Nielsen, E. (1990). Unresolved research issues in development of design-with-features systems. In: *Geometric Modeling for Product Engineering*, (hrsg.) Wozny, M.; Turner, J.; Preiss, K. Netherlands: Elsevier Science.
- Donaubauer, A. (2004). *Interoperable Nutzung verteilter Geodatenbanken mittels standardisierter Geo Web Services*. Dissertation. Technische Universität München.
- Dorigo, M.; Colorni, A.; Maniezzo, V. (1991). Distributed Optimization by Ant Colonies. In *Proc. ECAL91 – European Conference on Artificial Life*, Paris, France.
- Dorigo, M.; Colorni, A.; Maniezzo, V. (1992). An Investigation of some Properties of an Ant Algorithm. In *Proc. Parallel Problem Solving from Nature Conference (PPSN 92)*, Brussels, Belgium.
- Duives, D.; Daamen, W.; Hoogendoorn, S. (2013). State-of-the-art crowd motion simulation models. *Transportation Research Part C 37*, Elsevier, pp. 193-209.
- Dym, C. (1985). *EXPERT SYSTEMS: New Approaches to Computer-aided Engineering*. *Engineering with Computers*, Springer, pp. 9-25.
- Dym, C.; Levitt, R. (1991). *Knowledge-Based Systems in Engineering*. USA: McGraw-Hill Book Company.
- Eastman, C.M. (1999): *Computer Environments, Supporting Design and Construction*. USA: CRC Press.
- Ellis, C.; Gibbs, S. (1989). Concurrency Control in Groupware Systems. In *Proc. ACM SIGMOD '89 Conference on the Management of Data*, Seattle, Washington, USA.

- Ellis, C.; Gibbs, S.; Rein, G. (1991). Groupware Some Issues and Experiences. *Communications of the ACM*, Volume 34, Issue 1, pp. 38-58.
- Fan, H.; Sun, C. (2012). Dependency-based Automatic Locking for Semantic Conflict Prevention in Real-Time Collaborative Programming. In *Proc. SAC'12*, Riva Del Garda, Italy.
- Firmenich, B.; Rank, E. (2007). Überblick zum Themenbereich verteilte Produktmodelle. In: *Vernetzt-kooperative Planungsprozesse im konstruktiven Ingenieurbau*, hrsg. Uwe Rüppel. Deutschland: Springer.
- Firmenich, B. (2002). CAD im Bauplanungsprozess: Verteilte Bearbeitung einer strukturierten Menge von Objektversionen. Dissertation. RWTH Aachen.
- Flurl, M., Mundani, R.-P., Borrmann, A., Rank, E. (2012). A Collaborative Multi-Scale Planning Platform: Concept and Implementation Approach. In: *Proc. of the 14th International Conference on Computing in Civil and Building Engineering (ICCCBE)*, Moscow, Russia.
- Flurl, M., Mundani, R.-P., Rank, E. (2014). Graph-based Concurrency Control for Multi-Scale Procedural Models. In: *Proc. of the 10th European Conference on Product & Process Modeling (ECPMP)*, Vienna, Austria.
- Flurl, M., Jubierre, J.R., Monjarez, C., Morelos, R., Watson, E. (2014). Interaktive Planung von U-Bahn Tunnels: Automatisierte Berechnung der Änderung des normierten Energieverbrauchs eines U-Bahn Zuges bei Änderung des zugrunde liegenden Trassenverlaufs. In: *Beiträge zum 26. Forum Bauinformatik*, Darmstadt, Deutschland.
- Flurl, M.; Singer, D.; Mundani, R.-P.; Rank, E.; Borrmann, A. (2015). A Rule-based Collaborative Modelling System for Infrastructure Design. In *Proc. 22nd EG-ICE International Workshop: Intelligent Computing In Engineering*, Eindhoven, Netherlands.
- Flurl, M. (2015). Interaktive Planung von U-Bahn-Tunnels: Integration von Fußgängersimulationen in den geometrischen Modellierungsprozess von Tunneln und Bahnhofsgebäuden. In: *Beiträge zum 27. Forum Bauinformatik in Aachen*, Deutschland.
- Flurl, M., Morelos, R., Mundani, R.-P., Rank, E. (2016). Energy optimization for subway trains by interactive track alignment planning. In: *Proc. of Railway 2016 – The Third International Conference on Railway Technology: Research, Development and Maintenance*. Cagliari, Sardinia, Italy.
- Foley, J.; Van Dam, A.; Feiner, S.; Hughes, J. (1996). *Computer Graphics Principles and Practice*. USA: Addison-Wesley.

- Fook, C.; Monteiro, A.; Câmara, G.; Casanova, M.; Amaral, S. (2009). Geoweb Services for Sharing Modelling Results in Biodiversity Networks. *Transaction in GIS*, Volume 13, Issue 4, pp. 379-399.
- Freystein, H.; Muncke, M.; Schollmeier, P. (2015). *Entwerfen von Bahnanlagen: Regelwerke, Planfeststellung, Bau, Betrieb, Instandhaltung*. Deutschland: Eurailpress in DVV Media Group.
- Frischmann, F., Kollmannsberger, S., Rabold, A. (2013). Prä-Prozessor Framework für BIM-gekoppelte vibroakustische Simulationen im Holzbau. In: *Beiträge zum 25. Forum Bauinformatik*, München, Deutschland.
- Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. (1995). *Design Patterns Elements of Reusable Object-Oriented Software*. USA: Addison-Wesley.
- Gerdes, I.; Klawonn, F.; Kruse, R. (2004). *Evolutionäre Algorithmen*. Deutschland: Vieweg.
- Griffin, N.; Lewis, F. (1989). A rule-based inference engine which is optimal and VLSI implementable. In *Proc. IEEE International Workshop on Tools for Artificial Intelligence*, Fairfax, USA.
- Grudin, J.; Poltrock, S. (2014). Computer Supported Cooperative Work. In: Soegaard, M.; Dam, R. F. (eds.). *The Encyclopedia of Human-Computer Interaction*, 2nd Ed., https://www.interaction-design.org/encyclopedia/cscw_computer_supported_cooperative_work.html.
- Gujarathi, G.; Ma, Y.-S. (2011). Parametric CAD/CAE Integration Using a Common Data Model. *Journal of Manufacturing System*, Volume 30, Issue 3, pp. 118-132.
- Gutwin, C.; Greenberg, S. (1998). Effects of Awareness Support on Groupware Usability. In *Proc. CHI '98, SIGCHI Conference on Human Factors in Computing Systems*, Los Angeles, USA.
- Hamacher, H.; Tjandra, S. (2001). Mathematical Modelling of Evacuation Problems: A State of the Art. *Berichte des Fraunhofer ITWM*, Nummer 24.
- Hao, S.-Q.; Huang, H.-W.; Yuan, Y. (2009). Railway Station Pedestrian Simulation on Fire Smoke Based on Smartfire. In *Proc. International Forum on Information Technology and Applications*, 2009. IFITA '09, Chengdu, China.
- Hao, S.-Q.; Huang, H.-W.; Yuan, Y. (2009). Fire Evacuation of Underground Tunnel Based Building EXODUS. In *Proc. CSIE 2009*, Los Angeles, USA.

- Hartmann, D.; Lehner, K. (1990). Technische Expertensysteme. Deutschland: Springer.
- Hauschild, T. (2003). Computer Supported Cooperative Work - Applikationen in der Bauwerksplanung auf Basis einer integrierten Bauwerksmodellverwaltung. Dissertation. Bauhaus-Universität Weimar.
- Hampe, M.; Intas, S. (2006). Extension of the OGC Web Feature Services Standard for Multiple Representation Data. In Proc. ISPRS Technical Commission II Symposium, Vienna, Austria.
- Hay, W. (1982). Railroad Engineering. Canada: John Wiley & Sons.
- Hayes-Roth, F.; Jacobstein, N. (1994). The state of knowledge-based systems. Communications of the ACM, Volume 37, Issue 3, pp. 26-39.
- Helbing, D.; Molnár, P. (1995). Social force model for pedestrian dynamics. Physical Review, Volume 51, Issue 5, pp. 4282-4286.
- Henderson, L. (1971). The Statistics of Crowd Fluids. Nature, Volume 229, Issue 5284, pp. 381-383.
- Hepworth, A. (2014). Conflict Management and Model Consistency in Multi-User CAD. Dissertation. Brigham Young University.
- Hoffmann, C. (1992). Erep An Editable High-Level Representation for Geometric Design and Analysis. In Proc. IFIP WG5.2 Workshop on Geometric Modeling, Rensselaerville, New York, USA.
- Howlett, P. (1990). An Optimal Strategy for the Control of a Train. The ANZIAM Journal, Volume 31, Issue 4, pp. 454-471.
- Howlett, P.; Milroy, I.; Pudney, P. (1994). Energy-efficient train control. Control Engineering Practice, Volume 2, Number 2, pp. 193-200.
- Howlett, P.; Cheng, J. (1997). Optimal driving strategies for a train on a track with continuously varying gradient. The Journal of the Australian Mathematical Society. Series B. Applied Mathematics, Volume 38, Issue 3, pp. 388-410.
- Howlett, P. (2000). The Optimal Control of a Train. Annals of Operations Research, Volume 98, Issue 1, pp. 65-87.
- Howlett, P.; Pudney, P.; Vu, X. (2009). Local energy minimization in optimal train control. Automatica, Volume 45, Issue 11, pp. 2692-2698.

- Hull, G.; Roberts, C.; Hillmansen, S. (2010). Simulation of Energy Efficiency Improvements on Commuter Railways. In Proc. IET Conference on Railway Traction Systems, Birmingham, United Kingdom.
- Iwnicki, S. (2006). Handbook of Railway Vehicle Dynamics. USA: CRC Press, Taylor & Friends.
- James, R.; Jungbauer, R.; Abrams, B. (2005). Train Accident Reconstruction and FELA and Railroad Litigation. 4. Auflage. Lawyers & Judges Publishing Company, Inc.
- Janicki, J.; Reinhard, H.; Rüffer, M. (2008). Rail Vehicle Engineering. 3. Auflage. Deutschland: Bahn Fachverlag.
- Ji, Y. (2014). Durchgängige Trassen- und Brückenplanung auf Basis eines integrierten parametrischen 3D-Infrastrukturbauproduktmodells. Dissertation. Technische Universität München.
- Johansen, R. (1988). GroupWare: Computer Support for Business Teams. USA: The Free Press.
- Johnson-Lenz, P.; Johnson-Lenz, T. (1982). Groupware: the process and impacts of design choices. In: Computer-Mediated Communication Systems, hrsg. Kerr, E. B., Hiltz, S. R. USA: Academic Press.
- Jones, J.; Kuhn, W.; Keßler, C.; Schneider, S. (2014). Making the Web of Data Available Via Web Feature Services. In Proc. 17th AGILE Conference on Geographic Information Science, Castellón, Spain.
- Jong, J.-C. (2003). Analytical Solutions for Predicting Train Coasting Dynamics. In Proc. Eastern Asia Society for Transportation Studies (EAST), Fukuoka, Japan.
- Jong, J.-C.; Chang, S. (2005). Algorithms for Generating Train Speed Profiles. Journal of the Eastern Asia Society for Transportation Studies, Volume 6, pp. 356-371.
- Jubierre, J., Borrmann, A. (2013). Cross-submodel consistency preservation in multi-scale engineering models. In: Proc. of the 14th International Conference on Civil, Structural and Environmental Engineering Computing (CSC2013), Cagliari, Sardinia, Italy.
- Jubierre, J.; Borrmann, A. (2015). Knowledge-based Engineering for Infrastructure Facilities: Assisted Design of Railways Tunnels based in Logic Models and Advanced Procedural Geometry Dependencies. Journal of Information Technology in Construction, ISSN 1874-4753.

- Kaczmarczyk, C.; Kuhr, H.; Strupp, P.; Schmidt, J.; Schmidt, A. (2010). Bautechnik für Bauzeichner. 2. Auflage. Deutschland: Vieweg + Teubner.
- Kao, Y.-C.; Lin, G. (1998). Development of a Collaborative CAD/CAM system. *Robots and Computer-Integrated Manufacturing*, Volume 14, Issue 1, pp. 55-68.
- Karimi, H.; Akinci, B. (2010). CAD and GIS Integration. USA: Auerbach Publishers Inc.
- Kay, M. (2008). XSLT 2.0 and XPath 2.0 Programmer's Reference (Programmer to Programmer). 4. Auflage. USA: Wiley Publishing, Inc.
- Kielar, P.; Handel, O.; Biedermann, D.; Borrmann, A. (2014). Concurrent Hierarchical Finite State Machines for Modeling Pedestrian Behavioral Tendencies. In Proc. The Conference on Pedestrian and Evacuation Dynamics 2014 (PED 2014), Delft, Netherlands.
- Kim, J.; Pratt, M.; Iyer, R.; Sriram, R. (2007). Data Exchange of Parametric CAD Models Using ISO 10303-108. NISTIR 7433.
- Kim, J.; Pratt, M.; Iyer, R.; Sriram, R. (2008). Standardized data exchange of CAD models with design intent. *Computer-Aided Design*, Volume 40, Issue 7, pp. 760-777.
- Kim, K.; Chien, S. (2010). Simulation-Based Analysis of Train Controls under Various Track Alignments. *Journal of Transportation Engineering*, Volume 136, Issue 11, pp. 937-948.
- Kinkeldey, C.; Rose, M. (2003). Fußgängersimulationen auf Basis sechseckiger zellulärer Automaten. Studienarbeit. Universität Hannover.
- Kirsch-Pinheiro, M.; de Lima, J.; Borges, M. (2003). A framework for awareness support in groupware systems. *Computers in Industry*, Volume 52, Issue 1, pp. 47-57.
- Klimentew, L.; Jenke, J.; Meyer, H. (2012). Comparison of energy efficiency by using simulation tools. *Landtechnik*, Volume 67, Number 4, pp. 256-259.
- Klimke, J.; Döllner, J. (2010). Combining Synchronous and Asynchronous Collaboration within 3D City Models. In: *Geographic Information Science*, hrsg. Fabrikant, S., Reichenbacher, T., Kreveld, M., Schlieder, C. (Eds.). Springer.
- Klingsch, W.; Rogsch, C.; Schadschneider, A.; Schreckenberg, M. (2010). *Pedestrian and Evacuation Dynamics 2008*. Deutschland: Springer.
- Kneidl, A. (2013). Methoden zur Abbildung menschlichen Navigationsverhaltens bei der Modellierung von Fußgängerströmen. Dissertation. Technische Universität München.

- Knothe, K.; Stichel, S. (2003). Schienenfahrzeugdynamik. Deutschland: Springer.
- Koch, C. (2009). Bauwerksmodellierung im kooperativen Planungsprozess: Mit der Objektorientierung zur Verarbeitungsorientierung. Dissertation. Bauhaus-Universität Weimar.
- Korte, B.; Vygen, J. (2012). Kombinatorische Optimierung. Theorie und Algorithmen. 4. Auflage. Deutschland: Springer.
- Kucera, P.; Bradacova, I. (2012). Modelling the evacuation of people from a train on fire in a railway tunnel. In Proc. European Conference of Civil Engineering, Paris, France.
- Kuhr, H. (1991). EDV/CAD für die Bautechnik. Deutschland, Stuttgart, B. G. Teubner.
- Lawler, E.; Lenstra, J.; Rinnooy Kann, A.; Shmoys, D. (1985). The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization (Wiley Interscience Series in Discrete Mathematics). USA: John Wiley & Sons Inc.
- Lee, K., Kim, J., Banerjee, A. (2010): Collaborative intelligent CAD framework incorporating design history tracking algorithm. Computer-Aided Design Volume 42, Issue 12, Elsevier, pp. 1125-1142.
- Li, M.; Gao, S.; Li, J.; Yang, Y. (2004). An Approach to Supporting Synchronized Collaborative Design Within Heterogeneous CAD Systems. In Proc. 24th Computers and Information in Engineering Conference, Salt Lake City, Utah, USA.
- Li, M.; Gao, S.; Wang, C. (2007). Real-Time Collaboration Design with Heterogeneous CAD Systems based on Neutral Modeling Commands. Journal of Computing and Information Science in Engineering, Volume 7, Number 2, pp. 113-125.
- Li, W.D.; Ong, S.; Fuh, J.; Wong, Y.; Lu, Y.; Nee, A. (2004). Feature-based design in a distributed and collaborative environment. Computer-Aided Design, Volume 36, Issue 9, pp. 775-797.
- Li, X.; He, F.; Cai, X.; Chen, Y.; Liu, H. (2009). Using Procedure Recovery Approach to Exchange Feature-based Data among Heterogeneous CAD Systems. In Proc. 13th International Conference on Computer Supported Cooperative Work in Design, Santiago, Chile.
- Li, Y.; Lei, B.; Ingason, H. (2013). Theoretical and Experimental Study of Critical Velocity for Smoke Control in a Tunnel Cross-Passage. Fire Technology, Volume 49, Issue 2, Springer, pp. 435-449.

- Lukaszewicz, P. (2001). Energy Consumption and Running Time for Trains – Modelling of Running Resistance and Driver Behaviour Based on Full Scale Testing. Dissertation. KTH Stockholm.
- Ma, Y.-S.; Chen, G.; Thimm, G. (2008). Paradigm shift: unified and associative feature-based concurrent and collaborative engineering. *Journal of Intelligent Manufacturing*, Volume 19, Issue 6, Springer, pp. 625-641.
- Maher, M.; Rutherford, J. (1997). A Model for Synchronous Collaborative Design Using CAD and Database Management. *Research in Engineering Design*, Volume 9, Issue 2, pp. 85-98.
- Maier, F.; Stumptner, M. (2007). Enhancements and Ontological Use of ISO-10303 (STEP) to Support the Exchange of Parametrised Product Data models. In *Proc. Intelligent Systems Design and Applications, 2007. ISDA 2007*, Rio de Janeiro, Brasilia.
- Maiyo, L.; Kerle, N.; Köbben, B. (2010). Collaborative Post-disaster Damage Mapping via Geo Web Services. In: *Geographic Information and Cartography for Risk and Crisis Management*, hrsg. Konecny, M., Zlatanova, S., Bandrova, T. Deutschland: Springer.
- Malone, T.; Crowston, K. (1994). The Interdisciplinary Study of Coordination. *ACM Computing Surveys*, Volume 26, Issue 1, pp. 87-119.
- Mäntylä, M. (1988). *Introduction to Solid Modeling*. USA: W. H. Freeman & Co.
- Menniger, K. (1954). *Mathematik in deiner Welt*. Deutschland: Vandenhoeck & Ruprecht.
- Miller, P. (2007). *Schwarmintelligenz*. National Geographic Deutschland, Heft 8.
- Morelos, R. (2016). *Track Alignment Optimization by Using Elitist Ant System and Simulated Annealing*. Masterarbeit. Technische Universität München.
- Müller, G.; Buchschmid, M. (2014). Hybrid approaches for vibroacoustical problems based on the finite element method and statistical energy analysis. *Wave Motion*, Volume 51, Issue 4, pp. 622-634.
- Mun, D.; Han, S.; Kim, J.; Oh, Y. (2003). A set of standard modeling commands for the history-based parametric approach. *Computer-Aided Design*, Volume 35, Issue 13, pp. 1171-1179.

- Mundani, R.-P. (2006). Hierarchische Geometriemodelle zur Einbettung verteilter Simulationsaufgaben. Dissertation, Universität Stuttgart.
- Munson, J.; Dewan, P. (1994). A Flexible Object Merging Framework. In Proc. ACM conference on Computer supported cooperative work, Chapel Hill, North Carolina, USA.
- Munson, J.; Dewan, P. (1996). A Concurrency Control Framework for Collaborative Systems. In Proc. 1996 ACM conference on Computer supported cooperative work, Boston, USA.
- Nagel, K.; Schreckenberg, M. (1992). A cellular automaton model for freeway traffic. *Journal de Physique I*, Volume 2, Numéro 12, pp. 2221-2229.
- Negulescu, S.; Oprean, C.; Kifor, C.; Carabulea, I. (2008). Elitist Ant System for Route Allocation Problem. In Proc. 8th conference on applied informatics and communications (AIC'08), Rhodes, Greece.
- Neis, P.; Singler, P.; Zipf, A. (2010). Collaborative mapping and Emergency Routing for Disaster Logistics – Case studies from the Haiti earthquake and the UN portal for Afrika. In Proc. of Geospatial Crossroads@ GI-Forum, Salzburg, Austria.
- Park, W.; Kim, D.; Chang, H. (2006). Numerical predictions of smoke movement in a subway station under ventilation. *Tunnelling and Underground Space Technology*, Volume 21, Issue 3-4.
- Peachavanish, R.; Karimi, H.; Akinci, B.; Boukamp, F. (2005). An ontological engineering approach for integrating CAD and GIS in support of infrastructure management. *Advanced Engineering Informatics*, Volume 20, Issue 1, Elsevier, pp. 71-88.
- Penichet, W.M.R., Marin, I., Gallud, J.A., Lozano M.D., Tesoriero, R. (2007). A Classification Method for CSCW Systems. *Electronic Notes in Theoretical Computer Science* 02/2007, Volume 168, pp. 237-247.
- Phillips, W. (1998). Architectures For Synchronous Groupware. Technical Report. Queens University, Kingston, Ontario, Canada.
- Popescu, S.; Hilbert, F.; Schapke, S.-E.; Scherer, R. (2014). Eine intelligente Plattform für die verteilte Zusammenarbeit. In: *Informationssysteme im Bauwesen 1*, hrsg. Scherer R. & Schapke S.-E. Deutschland: Springer
- Pospech, T. (2013). GML - Geography Markup Language. Deutschland: Grin Verlag GmbH.

- Pratt, M.; Anderson, B. (2001). A shape modelling applications programming interface for the STEP standard. *Computer-Aided Design*, Volume 33, Issue 7, pp. 531-543.
- Pratt, M.; Anderson, B.; Ranger, T. (2005). Towards the standardized exchange of parameterized feature-based CAD models. *Computer-Aided Design*, Volume 37, Issue 12, pp. 1251-1265.
- Pratt, M.; Kim, J. (2006). Experience in the Exchange of Procedural Shape Models using ISO 10303 (STEP). In *Proc. of the 2006 ACM symposium on Solid and physical modeling (SPM '06)*, Cardiff, Wales.
- Puppe, F. (1991). *Einführung in Expertensysteme*. Deutschland: Springer.
- Ramani, K.; Agrawal, A.; Babu, M.; Hoffmann, C. (2003). CADDAC: Multi-Client Collaborative Shape Design System with Server-based Geometry Kernel. *Journal of Computing and Information Science in Engineering*, Volume 3, Issue 2, pp. 170-173.
- Raymond, C. (1918). Euler's Spiral. *American Math Monthly*, Volume 25, pp. 276-282.
- Rebolj, D.; Tibaut, A.; Cus-Babic, N.; Magdid, A.; Podpreznic, P. (2008). Development and application of a road product model. *Automation in Construction*, Volume 17, Issue 6, Elsevier, pp. 719-728.
- Reed, C. (2006). Data Integration and Interoperability: OGC Standards for Geo-Information. In: *Large-scale 3D Data Integration Challenges and Opportunities*, hrsg. Sisi Zlantanova, David Proserpi. Taylor & Francis.
- Reijnders, A. (2012). *Integrating Knowledge Management and Knowledge-Based Engineering: Formal and Platform Independent Representation of Engineering Rules*. Masterthesis. TU Delft, Netherlands.
- Requicha, A. (1980). Representations of rigid solid objects. In: *Computer Aided Design Modeling, Systems Engineering, CAD-Systems*, hrsg. Jose Encarnacao. Springer.
- Ressel, M.; Nitsche-Ruhland, D.; Gunzenhäuser, R. (1996). An Integrating, Transformation-Oriented Approach to Concurrency Control and Undo in Group Editors. In *Proc. CSCW '96, Conference on Computer supported cooperative work*, Boston, USA.
- Rochard, B.; Schmid, F. (2000). Review of Methods to Measure and Calculate Train Resistances. *Journal of Rail and Rapid Transit*, Volume 415, Number 5, pp. 185-199.
- Romberg, R. (2005). *Gebäudemodell-basierte Strukturanalyse im Bauwesen*. Dissertation, Technische Universität München.

- Salvat, E.; Mugnier, M.-L. (1996). Sound and Complete Forward and Backward Chainings of Graph Rules. In Proc. ICCS 4th International Conference on Conceptual Structures, Sydney, Australia.
- Sandhu, R.; Coyne, E.; Feinstein, H.; Youman, C. (1995). Role-Based Access Control Models. IEEE Computer, Volume 29, Number 2, pp. 38-47.
- Schadschneider, A.; Klingsch, W.; Klüpfel, H.; Kretz, T.; Rogsch, C.; Seyfried, A. (2009). Evacuation dynamics: Empirical results, modeling and applications. Encyclopedia of Complexity and System Science. Encyclopedia of Complexity and Systems Science, pp. 3142-3176.
- Schatz, K. (2014). Bauwerksmodellbasierte Serious Games als Ingenieurmethoden im Brandschutz am Beispiel interaktiver Entfluchtungsanalysen. Dissertation. Technische Universität Darmstadt.
- Schöning, U. (2008). Theoretische Informatik - kurz gefasst. 5. Auflage. Deutschland: Spektrum.
- Scholz, D., Rank, E., Glueck, M., Breuer, M., Durst, F. (2004). Fully Three-Dimensional Coupling of Fluid and Thin-Walled Structures. In: Proc. of High Performance Computing in Science and Engineering, Munich, Germany.
- Schultze, C.; Buhmann, E. (2008). Developing the OKSTRA® Standard for the Needs of Landscape Planning in Context of Implementation for Mitigation and Landscape Envelope Planning of Road Projects. In Proc. 9th International Conference on Information Technologies in Landscape Architecture, Köthen, Anhalt, Deutschland.
- Schwabe, G.; Streitz, N.; Unland, R. (2001). CSCW-Kompodium. Deutschland: Springer.
- Schwietert, K. (2014). Parametrische Modellierung des Ostbahnhofs München unter Verwendung eines neutralen prozeduralen Modells. Bachelorarbeit. Technische Universität München.
- Sikder, I. (2009). Geospatial Web Services for Environmental Planning. International Technology Management Review, Volume 2, Number 1.
- Sorger, C.; Frischmann, F.; Kollmannsberger, S.; Rank, E. (2014). TUM.GeoFrame: Automated high-order hexahedral mesh generation for shell-like structures. Engineering with Computers, Volume 30, Issue 1, pp. 41-56.
- Starke, G. (2015). Effektive Softwarearchitekturen: Ein praktischer Leitfaden. Deutschland: Carl Hanser Verlag.

- Stefik, M.; Bobrow, D.; Foster, G.; Lanning, S.; Tatar, D. (1987). WYSIWIS Revised: Early Experiences with Multiuser Interfaces. *ACM Transactions on Office Information Systems*, Volume 5, Number 2, pp. 147-167.
- Steuer, H., Donaubaue, A., Kolbe, T.H., Flurl, M., Mundani, R.-P., Rank, E. (2013). Planning Inner-City-Railway-Tracks: Dynamic Integration of Geospatial Web Services in a Collaborative Multi-Scale Geometry Modelling Environment. In: *Proc. of the 20th EG-ICE International Workshop: Intelligent Computing In Engineering*, Vienna, Austria.
- Steuer, H., Donaubaue, A., Kolbe, T.H., Flurl, M., Mundani, R.-P., Rank, E. (2014). Collaborative Planning of Inner-City-Railway-Tracks: A Generic Description of the Geographic Context and Its Dynamic Integration in a Collaborative Multi-Scale Geometry Modelling Environment. *Advanced Engineering Informatics*, Volume 28, Issue 4, pp. 261-271.
- Stevens, R. (2004). *TCP/IP*. Deutschland: mitp-Verlag.
- Stouffs, R.; Krishnamurti, R.; Eastman, C. (1996). A Formal Structure for Nonequivalent Solid Representations. In *Proc. IFIP WG 5.2 Workshop*, Pittsburgh, USA.
- Su, S.; Li, X.; Tang, T.; Gao, Z. (2013). A Subway Train Timetable Optimization Approach Based on Energy-Efficient Operation Strategy. *IEEE Transactions on Intelligent Transportation Systems*, Volume 14, Issue 2, pp. 883-893.
- Sun, C.; Jia, X.; Zhang, Y.; Yang, Y.; Chen, D. (1998). Achieving Convergence, Causality Preservation, and Intention Preservation in Real-Time Cooperative Editing Systems. *ACM Transactions on Computer-Human Interaction*, Volume 5, Number 1, pp. 63-108.
- Sun, C. (2002). Optional and Responsive Fine-Grain Locking in Internet-based Collaborative Systems. *IEEE Transactions on Parallel and Distributed Systems*, Volume 13, Number 9, pp. 994-1008.
- Tanenbaum, A.; Van Steen, M. (2006). *Distributed Systems: Principles and Paradigms*. 2. Auflage. USA: Prentice Hall International.
- Teufel, S.; Sauter, C.; Mühlherr, T.; Bauknecht, K. (1995). *Computerunterstützung für die Gruppenarbeit*. Deutschland: Addison-Wesley.
- Trautwein, M. (1990). *CAD für Bauingenieure*. Deutschland: Vieweg + Teubner.
- Turau, V. (2004). *Algorithmische Graphentheorie*. Deutschland: Oldenbourg.

- Vajina, S.; Weber, C.; Schlingensiepen, J.; Schlottmann, D. (1994). CAD/CAM für Ingenieure. Deutschland: Vieweg.
- van Treeck, C. (2004) Gebäudemodell-basierte Simulation von Raumluftrömungen. Dissertation. Technische Universität München.
- Varduhn, V., Mundani, R.-P. (2014). Large-Scale Parallel Simulation of Urban Flooding Scenarios. In Proc. of the 16th SIAM Conference on Parallel Processing for Scientific Computing – SIAM PP2014.
- Weise, G.; Durth, W. (2005). Straßenbau- Planung und Entwurf. Deutschland: Verlag für Bauwesen.
- Wende, D. (2003). Fahrdynamik des Schienenverkehrs. Deutschland: Vieweg+Teubner Verlag.
- Xue, L.; Zhang, K.; Sun, C. (2000). Conflict Control Locking in Distributed Cooperative Graphics Editors. In Proc. 1st International Conference on Web Information Systems Engineering, Hong Kong.
- Yang, G.-s.; An, Y.-l.; Peng, L.-m.; Zhang, J.-h. (2006). Simulation of smoke flow and longitudinal ventilation in tunnel fire. Transactions of Nonferrous Metals Society of China, Volume 16, Issue 3, pp. 741-746.
- Yang, J.; Meng, L. (2011). Railway Alignment Optimisation for Specified Travel Time-saving. In Proc. 8th International Symposium on Location-Based Services, Vienna, Austria.
- Yang, J.; Nottbeck, A.; Murphy, C.; Jancke, M.; Meng, L.; Freudenstein, S. (2015). Pro-Zeit: An Automated Workflow of Optimising the Geometric Design of Railway Alignment for Travel Time Saving. In Proc. ICC 2015 – 27th International Cartographic Conference, Rio de Janeiro, Brasil.
- Zimbardo, P. (1992). Psychologie. Deutschland: Springer-Lehrbuch.
- Zlatanova, S.; Prospero, D. (2006). Large-scale 3D Data Integration: Challenges and Opportunities (Gisdata). USA: Taylor & Francis Group.