

Fakultät für Bauingenieur- und Vermessungswesen
Lehrstuhl für Computergestützte Modellierung und Simulation
Prof. Dr.-Ing. André Borrmann

Automatisierte Planableitung anhand eines parametrischen Brückenmodells mit Autodesk® Inventor®

Manuel Fiermonte

Bachelorthesis

für den Bachelor of Science Studiengang Bauingenieurwesen

Autor:	Manuel Fiermonte
Matrikelnummer:	██████████
1. Betreuer:	Prof. Dr.-Ing. André Borrmann
2. Betreuer:	Mathias Obergrießer, M.Eng.
3. Betreuer:	Markus Hochmuth, Dipl.-Ing.
Ausgabedatum:	31. Oktober 2012
Abgabedatum:	31. März 2013

Danksagung

An dieser Stelle möchte ich mich ganz herzlich bei Mathias Obergrießer für die zahlreichen, hilfreichen Anregungen und Anmerkungen, vor allem bei der schriftlichen Erstellung der Arbeit, bedanken, sowie allgemein für die Betreuung der Bachelorarbeit.

Weiter danke ich Markus Hochmuth für die freundliche Betreuung bei der Firma Obermeyer Planen + Beraten GmbH, wo ich nicht nur Kenntnisse über die Planung von Straßenbrückenbauwerke erlangte, sondern unabhängig von der Bachelorarbeit einen Einblick in die tägliche Arbeit eines Bauingenieurs in dem spezifischen Zweig der Brückenplanung erhielt.

Mein besonderer Dank gilt meinen Eltern, die mich stets motiviert und tatkräftig unterstützt haben und sich auch nicht scheuten die Arbeit als Fachfremde mehrmals Korrektur zu lesen.

Zusammenfassung

Die hohe Komplexität eines Brückenbauwerks ist durch die Vielzahl unterschiedlicher Randbedingungen unumstritten. Durch einen parametrischen 3D-Ansatz wurde versucht mit Hilfe der Software Autodesk® Inventor® ein flexibles 3D-Brückenmodell zu erschaffen, das die eingegebenen konstruktiven, topographischen und fahrdynamischen Randbedingungen berücksichtigt. Die vorliegende Arbeit wurde speziell dem Thema der Ableitung von Plänen auf Basis eines solchen 3D-Modells gewidmet. Dabei ist eine erfolgreiche Planerstellung stellvertretend für den Erfolg des gesamten Ansatzes.

Zuerst wird ein kurzer Überblick über die normativen Grundlagen der Planableitung in Deutschland, allem voran der DIN 1356-1 über Bauzeichnungen, vermittelt. Außerdem wird in diesem allgemeinen Teil die besondere Problematik des Längsschnittes erläutert. Nicht nur Inventor sondern auch die meisten anderen 3D-CAD-Anwendungen bieten keine gekrümmten Schnittführungen an, so dass für den Längsschnitt einer gekrümmten Brücke ein approximierter Linienzug als Schnittlinie verwendet werden muss, wodurch gewisse Approximationsfehler entstehen.

Als Einstieg in die Zeichnungserstellung mit Inventor, werden anfangs die unterschiedlichen Funktionen, die später zur Anwendung kommen, allgemein vorgestellt. Weitere Themen, darunter die Stilverwaltung in Inventor, werden bei der Beschreibung, wie eine eigene Zeichnungsvorlage erstellt werden kann, abgedeckt.

Die Modellierung des Oberbaus wird mit einer bestimmten Funktion durchgeführt (siehe Römer, 2013), die diesen allerdings nicht ganz korrekt modelliert. Diese Funktion stellt aber momentan die beste Lösung dar. Durch Versuche mit gewissen Randbedingungen, die aus logischer Sicht mit die größten Abweichungen produzieren und als Minimalwerte für die meisten Straßenbrücken gelten dürften, wurden die Abmessungsfehler kurz analysiert und eine erste Einschätzung wiedergegeben.

Im Anschluss daran wird die manuelle Erstellung einer Draufsicht, eines Regelquerschnittes, einer Quer- und einer Längsschnittansicht schrittweise beschrieben und durch Abbildungen veranschaulicht.

Der Hauptteil der Arbeit beschäftigt sich mit dem Thema der Automatisierung. Dabei wurde ein Grundkonzept entwickelt, das beschreibt wie unabhängig von der Planart eine Zeichnung unterteilt werden kann, um so ein Gitternetz zu schaffen, das wiederum den Rahmen für unterschiedliche Ansichten oder Plandetails bildet. Auf dieser Basis wurde ein Modul in der Programmiersprache VB.NET entwickelt, das es ermöglicht ein bestimmtes Layout für eine Zeichnung zu erstellen und über die Inventor API als Schnittstelle dieses Layout automatisch in Inventor zu generieren. Neben der Struktur lassen sich die Ansichtstypen und weitere Einstellungsmöglichkeiten in einem solchen Layout im Vorfeld festlegen, wodurch viele Arbeitsschritte auf eine einfache Art und Weise gebündelt werden können.

Unabhängig vom Modul wurde noch ein VBA-Makro entwickelt, welches in Inventor ausgeführt werden kann und durch Angabe eines Toleranzwertes eine Kurvenlinie als Polygonzug approximiert. Der Anwendungszweck dieses Makros ist vor allem in der Approximation der Gradient für den Längsschnitt zu sehen.

Schlussfolgernd lässt sich behaupten, dass Inventor trotz einiger Schwächen durchaus für den Einsatz im Bauwesen herangezogen werden kann. Die automatisierte bauspezifische Planableitung steht erst am Anfang der Entwicklung. Das vorgestellte Modul bietet jedoch eine gute Grundbasis, die durchaus genügend Potential zur Weiterentwicklung bietet, wodurch dem Anwender in Zukunft viele Arbeitsschritte abgenommen werden können.

Abstract

The high complexity of bridge structures is due to a wide range of boundary conditions unquestioned. A parametric 3D approach was used to create a flexible 3D bridge model with the software Autodesk® Inventor®, which took account of the given constructive, dynamic and topographical boundary conditions. The present bachelor thesis covers the specific topic about drawing derivation of such a 3D model. A successful creation of a drawing represents at the same time the success of the whole 3D approach.

First a short overview about the normative foundations of drawing derivation in Germany, especially about the DIN 1356-1 covering civil drawings, is imparted. Furthermore the problem with a longitudinal section is explained. Not only Inventor, but most of the 3D-CAD-Softwares are not able to perform curved sections, so that for the longitudinal section view of a curved bridge the section line needs to be approximated, which results in certain approximation errors.

As an introduction to the creation of drawings in Inventor different functions, which will be used later, are presented in a general manner. Moreover some topics, including the style management in Inventor, are covered in the context of the creation of an own template.

The modelling of the superstructure of the bridge is accomplished by a certain function, that does not shape correctly, but which is at the moment the best solution. Tested with defined boundary conditions, which probably produce the maximum deviation and are at the same time the minimum values for most road bridges, the dimension errors are analysed and a first estimation is given.

Furthermore the manual creation of a top view, a view of the standard cross-section of the superstructure, a cross section view and a longitudinal section view is described step by step and illustrated by figures.

The main part of the paper is about the automation. A basic concept was developed, which describes how a drawing can be divided independently of the planning type, to create a grid, which in turn forms frames for different views and plan details. On this base, using as programming language VB.NET, a module was developed, which allows to configure a drawing layout and over the interface provided by the Inventor API that layout can be automatically generated in Inventor. Besides the structure, the view types and other settings can be set and defined in advance in such a layout, so that many individual operations can be bundled in a simple way.

Independently of the module a VBA macro was developed, which can be executed within Inventor and approximates a curve segment as lines after a tolerance value was entered. The intended use of this macro is the approximation of a curved bridge path for the longitudinal section.

As a conclusion it can be said that Inventor is usable in the civil engineering field despite some weak points. The automated drawing derivation is at the beginning of its development. However the introduced module provides a good ground base with enough potential for further development and in the future a lot of the user's operations can be reduced.

Inhaltsverzeichnis

Zusammenfassung	3
Abstract	4
1 Einführung und Motivation	8
1.1 Einführung	8
1.2 Ziel der Arbeit	9
1.3 Aufbau der Arbeit	9
2 Zeichnungsableitung	10
2.1 Normative Grundlagen	10
2.1.1 Ansichten	11
2.1.2 Pläne	11
2.1.2.1 Entwurfsplan	12
2.1.2.2 Ausführungsplan	13
2.1.3 Liniendicken und Schraffuren nach DIN 1356-1	14
2.1.4 Falten nach DIN 824	15
2.2 Längsschnittproblematik	16
2.2.1 Erster Approximationsfehler	16
2.2.2 Zweiter Approximationsfehler	18
3 Zeichnungserstellung mittels Autodesk Inventor	19
3.1 Allgemeines zu Zeichnungen in Inventor	19
3.1.1 Neue Zeichnung erstellen	19
3.1.2 Ansichten im Überblick	19
3.1.3 Erstsicht	20
3.1.4 Schnittansicht	20
3.1.5 Zusätzliche Skizzen	21
3.2 Zeichnungsableitung des Oberbaus	22
3.2.1 Modellierung	22
3.2.2 Schnitte und Abmessungsfehler	23
3.2.2.1 Erste Trasse	23
3.2.2.2 Zweite Trasse	23
3.2.2.3 Dritte Trasse	23
3.2.3 Fazit	24
3.3 Eigene Vorlage erstellen	25

3.3.1	Rahmen.....	25
3.3.2	Plankopf	26
3.3.2.1	iProperties	26
3.3.3	Arbeitsblattformate.....	26
3.3.4	Stile und Normen	27
3.3.4.1	Kategorie „Norm“	28
3.3.4.2	Kategorie „Bemaßung“	30
3.3.4.3	Kategorie „Layer“	31
3.3.4.4	Kategorie „Objektvorgaben“	31
3.3.4.5	Kategorie „Text“	31
4	Beispiel: Ausführungsübersichtsplan	32
4.1	Draufsicht	32
4.1.1	Drehen der Draufsicht	32
4.1.1.1	Erste Methode.....	32
4.1.1.2	Zweite Methode.....	33
4.1.1.3	Nachteile beider Methoden	33
4.1.1.4	Fazit	34
4.2	Regelquerschnitt.....	34
4.3	Querschnittsansicht	34
4.4	Längsschnitt	35
5	Automatisierung	36
5.1	Konzept	36
5.1.1	Konzeptprogramm.....	38
5.2	Umsetzung des Konzeptes.....	39
5.2.1	Programmiertechnische Anbindung an Inventor.....	39
5.2.2	Aspekte der Erweiterungsprogrammierung.....	40
5.2.3	Konzept der drei Ebenen	43
5.2.4	Modul <i>InventorLayout</i>	44
5.2.4.1	Enumeratoren	44
5.2.4.2	Klasse „templateDocument“	44
5.2.4.3	Klasse „LayoutID“	45
5.2.4.4	Überblick: Hauptklassen zum Erstellen eines Layouts	45
5.2.4.5	Initialisierung.....	45
5.2.4.6	Methode „CreateLayout“	46

5.2.4.7 Inhalt eines LayoutBox-Objektes.....	46
5.2.4.8 Integritätsüberprüfung.....	47
5.2.4.9 Darstellung des Layouts	47
5.2.4.10 Layout in Inventor generieren.....	48
5.2.4.11 Layout für den Ausführungsübersichtsplan	50
5.2.4.12 Änderung der Breite einer Spalte.....	50
5.2.5 Umsetzung der Konzeptanwendung.....	50
5.3 Bekannte Probleme und einige Ideen zur Erweiterung	52
5.3.1 Problem: Erkennung der Gradienten in Inventor	52
5.3.2 Automatische Bemaßung	53
5.3.3 Gitternetz des Layouts komplett frei manipulieren.....	53
5.3.4 Speichern von Layouts	53
5.4 Weitere Entwicklungen	54
5.4.1 Makro zur Polygonapproximation von Splines.....	54
5.4.2 VB.NET Funktion zur Schnittpunktermittlung im Lageplan.....	55
6 Zeichnungsexport	57
6.1 Exportieren einer Zeichnung in Inventor	57
7 Schlussfolgerung	59
Anhang A	61
Anhang B.....	62
Anhang C.....	64
Literaturverzeichnis.....	65
Allgemeine Literatur	65
Literatur zur Software <i>Autodesk Inventor</i>	66
Literatur zur Programmierung.....	67
Abbildungsverzeichnis	68
Tabellenverzeichnis.....	69

1 Einführung und Motivation

Durch die fortschreitende Entwicklung in der Computerbranche und der damit verbundenen Technologiezweige hat sich, vor allem auch durch die stark angestiegene Rechenleistung, der Bereich der 3D Modellierung sehr weit ausgedehnt. Dabei hängt die Bauindustrie allerdings eher noch am unteren Zweig dieses aufsteigenden Astes.

Die traditionellen Handzeichner arbeiten längst am Rechner und für Präsentationen werden auch seit einiger Zeit aufwendige 3D Modelle erstellt, um den Kunden mit diesen gerenderten Bildern zu beeindrucken. Was allerdings bisher noch in den Kinderschuhen steckt, ist die Möglichkeit bei komplexen Ingenieurbauten wie beispielsweise Brücken 3D-Modelle nicht nur zu visuellen Zwecken zu verwenden, sondern um effizient damit zu arbeiten. Was bisher fast völlig fehlt in dem Bereich, sind Modelle, die erstens parametrisch sind und somit flexibel an die jeweils spezifischen Randbedingungen angepasst werden können, und zweitens die Möglichkeit von so einem parametrischen 3D Modelle zumindest bis zu einem gewissen Grad zweidimensionale Pläne automatisch ableiten zu können.

Der momentane Arbeitsprozess bei der Erstellung von Plänen für gewisse Bauprojekt sieht so aus, dass zum Teil schon parametrische 3D Modelle zum Einsatz kommen, die Planerstellung selbst, also eigentlich die entscheidende Arbeit, allerdings häufig zu einem großen Teil ohne direkte Abhängigkeit zum Modell erfolgt. Dadurch sind bei Änderungen der Randbedingungen alle erstellten Pläne im Detail zu kontrollieren und anzupassen, was natürlich sowohl Zeit als auch Geld kostet und größeren Raum für mögliche Fehler zulässt.

1.1 Einführung

Im Bauwesen ist Autodesk vor allem durch die AutoCAD (Architecture) Produkte bekannt. Mit Inventor hat die Softwareschmiede eine parametrische 3D-CAD-Software in ihrer Produktlinie, die hauptsächlich für mechanische Konstruktionen genutzt wird, also im Maschinen- und Anlagebau Verwendung findet. Da es für den Infrastrukturbau kein spezifisches Produkt gibt, welches die gleiche Modelliermöglichkeiten, die mit einer Parametrik verbunden sind, im dreidimensionalen Raum bietet, wurde entschieden Inventor auf die Tauglichkeit hinsichtlich der im Bauwesen vorliegenden Randbedingungen und Anforderungen hin zu überprüfen. Der Anstoß für diese Untersuchung ging nicht nur vom Lehrstuhl für computergestützte Modellierung und Simulation der TU München aus, sondern wurde auch in enger Kooperation mit der Abteilung für Verkehrsbauwerke und Brücken der Firma OBERMEYER Planen + Beraten GmbH durchgeführt, um deren Entscheidungsfindung bei der Auswahl des richtigen Produktes zu unterstützen.

In diesem Zusammenhang entstand gleichzeitig eine weitere Bachelorarbeit am gleichen Lehrstuhl über die Modellierung eines parametrischen 3D-Brückenmodells mit Hilfe von Autodesk Inventor von Herrn Fritznel L.-L. Römer (2013). Das entstandene 3D-Modell wird als Basismodell in der vorliegenden Arbeit verwendet.

1.2 Ziel der Arbeit

In dieser Arbeit wird speziell das Thema der Planableitung mit Autodesk Inventor behandelt. Durch den Umstand, dass der Hauptanwendungsbereich der Software im Maschinenbausektor liegt, sind die Funktionen zur Zeichnungserstellung auch dementsprechend ausgerichtet.

Des Weiteren soll die Frage erörtert werden, welcher Grad der automatisierten Plangenerierung erreicht werden kann.

1.3 Aufbau der Arbeit

In Kapitel 2 wird zunächst ein kleiner Überblick über die normativen Grundlagen der Zeichnungsableitung in Deutschland vermittelt, wobei die Betonung auf „Überblick“ liegt, da dieses Thema eine eigene Bachelorarbeit füllen könnte. Des Weiteren wird die Problematik der Längsschnittführung theoretisch erläutert und eine erste Fehlereinschätzung gegeben.

Anschließend wird in Kapitel 3 die Thematik der Zeichnungserstellung mit Hilfe der Software Inventor von Autodesk behandelt, wobei das Hauptaugenmerk auf den für die Zeichnungsableitung relevanten Funktionen liegt. Ein Teil der Funktionalität wird dabei im Zusammenhang mit dem praktischen Beispiel der Erstellung einer eigenen Vorlage erläutert.

Darauf aufbauend wird in Kapitel 4 als Beispiel ein Ausführungsübersichtsplan auf Basis eines 3D-Brückenmodells generiert. Dieser Plan beinhaltet vier verschiedene Ansichten, die im Detail erklärt und durch Abbildungen einen ersten Eindruck vermitteln sollen, was in Inventor ohne weitere CAD-Kenntnisse möglich ist.

Das 5. Kapitel ist der eigentlich Hauptteil der Arbeit und behandelt die automatisierte Planableitung mit Inventor. Zuerst wird ein Konzept vorgestellt, welches anschließend mit der Programmiersprache Visual Basic .NET und der Inventor API umgesetzt wird. Außerdem wird noch eine zusätzliche Entwicklung beschrieben, die aus einem Makro zur Approximation eines gekrümmten Schnittes besteht.

Zum Schluss wird in Kapitel 6 noch kurz auf den Zeichnungsexport aus Inventor eingegangen.

2 Zeichnungsableitung

Zeichnen im Bauwesen ist als Lehre so alt wie der konstruktive Ingenieurbau selbst und reicht in der Geschichte weit zurück. Auch die normativen Grundlagen entstanden nicht erst Ende des letzten Jahrhunderts, sondern haben sich im Laufe des 20. Jahrhunderts stark entwickelt und obwohl heutzutage viele Normen durch die Einführung europäische Pendants ändern, ist deren Basis trotzdem meist alles andere als neu.

So wurde zum Beispiel schon 1926 eine Ausgabe der DIN 206 "*Zeichnungen; Schraffuren und Farben zur Kennzeichnung von Werkstoffen*" veröffentlicht. Mittlerweile wurde sie zwar offiziell 2002 durch die DIN ISO 128-50 ersetzt¹, jedoch basiert der Teil über Schraffuren in der DIN 1356-1 weiterhin auf der DIN 206. Deswegen wird in aktueller Literatur empfohlen eine Legende zu den Kennzeichnungen von Schnittflächen anzugeben.²

Ein weiteres Beispiel für die zeitliche Beständigkeit von Normen in diesem Bereich wäre die DIN 824, deren letzter Ausgabedatum der 3.1981 ist und die das Falten von technischen Zeichnungen auf ein bestimmtes Format, meistens auf das A4-Format zum Einheften, normativ regelt. In diesem Fall scheint es selbstverständlich zu sein, dass diese Norm sich nicht großartig ändern kann, da sich die Blattformate auch nicht geändert haben. Allerdings würde es genauso wenig Sinn machen, die allseits bekannte Schraffur für Stahlbeton zu ändern oder die festgelegten Liniendicken. Der Spielraum unter 0,5 cm ist so gering, dass auch in dem Bereich in Zukunft keine weiteren Änderungen zu erwarten sind.

2.1 Normative Grundlagen

Die **DIN 1356-1** legt in Deutschland die Arten und Inhalte von Bauzeichnungen für die Objekt- und Tragwerksplanung sowie Grundregeln für die Darstellung in Bauzeichnungen fest.³ Neben dieser thematisch sehr umfassenden Norm existiert noch eine Vielzahl weiterer Normen. Diese decken gewisse Bereiche in einem größeren Umfang ab, behandeln sehr spezifische Themen, die eine eher allgemeingehaltene Norm sprengen würden, oder haben einen ganz anderen geografischen Gültigkeitsbereich als die DIN Normen. Die DIN 1356-1 selbst basiert auf über zehn anderen Normen, die am Ende der Norm aufgelistet sind.

DIN ISO 128	Technische Zeichnungen, Grundlagen der Darstellung
DIN 406	Maßeintragungen, Ausführungsregeln
DIN 476	Papierendformate, technische Zeichnungen
DIN 824	Technische Zeichnungen, Faltung auf Ablageformat
DIN 5 Teil 1,2,10	Projektionsarten
DIN 1301-1	Einheiten, Einheitenamen, Einheitenzeichen
DIN 1356 Teil 1-10	Bauzeichnungen, Grundregeln, Objektplanung usw.
DIN ISO 5455	Technische Zeichnungen, Maßstäbe

¹ <http://de.wikipedia.org/wiki/Schraffur>

² BATRAN B., A. BORN, V. FREY, B. GUSTAVUS, H.-J. HANSEN, K. DR. KÖHLER, H. SCHLIEBNER, J.-P. SCHULLER, H. SOMMER, F. WEIDNER, R. WENZKE & J. DR. ZWANZIG 2010. *Bauzeichnen: Architektur, Ingenieurbau, Tief-, Straßen- und Landschaftsbau*. Holland & Josenhans. Seite 1-13

³ Teil 1 "Anwendungsbereich und Zweck", DIN 1356-1.

DIN EN ISO 5457	Technische Produktdokumentation, Formate und Gestaltung von Zeichnungsvordrucken im Bauwesen
ISO 2594	Bauzeichnungen, Projektionsmethoden
DIN ISO 7518	Zeichnen im Bauwesen, Darstellung von Abriss und Wiederaufbau
DIN ISO EN 4157	Bezeichnungssysteme, Anforderungen für Bezeichnungssysteme

Tabelle 2-1: Normenliste

Diese kurze Liste^{4 5} an Normen soll zeigen, wie umfangreich die Normierung zum Thema Bauzeichnungen ist.

2.1.1 Ansichten

Das Abbildungsprinzip für Zeichnungen in der Architektur und auch im konstruktiven Ingenieurbau ist üblicherweise die senkrechte Parallelprojektion.

„Man stellt sich das Objekt in einem Raumeck vor, dessen drei Ebenen senkrecht aufeinander stehen. Parallele Projektionsstrahlen, die senkrecht auf die drei Bildebenen treffen, bilden alle Strecken und Flächen parallel zur Bildebene in wahrer Größe ab. Werden die Ebenen auseinandergeklappt, so erhält man Grundriss oder Draufsicht, Aufriss oder Ansicht und Seitenriss oder Seitenansicht in der einen Fläche einer üblichen Zeichnung.“ (Prenzel, 1982)

Liegt eine Strecke parallel zur Projektionsebene im Raum, wird ihre wahre Länge abgebildet, andernfalls nicht. Dies ist vor allem bei Schnitten zu berücksichtigen.

„Ein Schnitt entsteht, indem man eine vertikale Schnittebene durch ein Gebäude legt und diese in Parallelprojektion als Ansicht betrachtet.“ (Bielefeld und Skiba, 2006)

2.1.2 Pläne

Im Bauwesen werden verschiedene Arten von Plänen unterschieden, die jeweils einen bestimmten Zweck verfolgen und deren Maßstab vorgegeben ist. Dabei gibt die Namensbezeichnung meist schon den Inhalt oder den Zweck des Planes wieder.

So stellt zum Beispiel ein Entwurfsplan das durchdachte Planungskonzept des Bauwerkes dar. Ausführungspläne, auch bekannt als Werkpläne, beinhalten hingegen alle für die Ausführung relevanten Detailangaben. Schalpläne sind spezielle Werkpläne, auf deren Basis später die Schalung entsteht und die deshalb für die Schalung wichtige Details angeben. Des Weiteren gibt es noch Bewehrungspläne, die eine hohe Bedeutung im Bauwesen haben, da sie die Bewehrungsführung zeichnerisch darstellen und unter anderem mit genauen Angaben über die Anzahl und den Durchmesser von Baustählen oder Matten versehen sind. Jeder dieser Pläne kann zusätzlich Detailzeichnungen enthalten.

Die Planarten sind eng mit den Leistungsphasen eines Bauprojektes verbunden. In der Honorarordnung für Architekten und Ingenieure (HOAI, 2009) sind für unterschiedliche

⁴ http://www.vbv.baden-wuerttemberg.de/servlet/PB/show/1237914/A1_Allgemeine_Planvorgaben_V02_112008.pdf

⁵ <http://de.wikipedia.org/wiki/Bauzeichnung>

- die Höhenlage des Bauwerks über NN anhand von Höhenkoten,
- Lage der Schnittebenen,
- Bauwerksfugen,
- Angabe der Bauart und der wesentlichen Baustoffe,
- die bestehenden und zu berücksichtigenden baulichen Anlagen (falls notwendig) und die zu berücksichtigende anschließende Bebauung,
- der vorhandene und geplante Geländeverlauf,
- den zu erhaltenen Baumbestand,
- Darstellungen des Abrisses, des Wiederaufbaues oder der Erneuerung von Bauteilen gemäß DIN ISO 7518,
- in den Schnitten: konstruktive Angaben zur Gründung,
- bei Straßenbrücken: Angabe der lichten Weite, Höhe und Stützweiten, sowie Straßen- bzw. Gleisachsenbezeichnungen.

(Schneider, 2010; Holschemacher und Ackermann, 2009; Werner und Pastor, 2010)

2.1.2.2 Ausführungsplan

Die zeichnerische Darstellung eines Bauobjektes mit allen für die Ausführung notwendigen Einzelangaben und ergänzenden Detailzeichnungen sind sogenannte Ausführungszeichnungen. Diese basieren auf den vorher erstellten Entwurfsplänen, werden meist im Maßstab 1:100, 1:50 oder 1:20 dargestellt und enthalten:

- ausführliche Bemaßung aller Bauteile, die zur Erstellung des Bauwerks erforderlich sind,
- Höhenkoten, die Lage des Bauwerks über NN,
- konstruktive Fugen,
- Lage und Verlauf von Abdichtungen,
- Angabe der Bauart und der verwendeten Baustoffe,
- konstruktive Hinweise,
- Verweise auf andere, zugehörige Pläne,
- Angabe über die Dränung,
- in den Schnitten: Angaben zur Gründung,
- in den Schnitten: Höhenkoten und Maßen, die in keiner anderen Ansicht angetragen oder sichtbar sind,
- bei Straßenbrücken: Angabe der lichten Weite, Höhe und Stützspannweiten, sowie Straßen- bzw. Gleisachsenbezeichnungen,
- der geplante Geländeverlauf,
- die Anordnung von betrieblichen Anlagen,
- die bestehenden und zu berücksichtigenden baulichen Anlagen (falls notwendig) und die zu berücksichtigende anschließende Bebauung.

(Schneider, 2010; Holschemacher und Ackermann, 2009; Werner und Pastor, 2010)

2.1.3 Liniendicken und Schraffuren nach DIN 1356-1

Liniendicken in Planzeichnungen sind nach folgender Tabelle aus der DIN 1356-1 und gelten sowohl für Handzeichnung als auch für am Rechner erstellte Pläne.

Tabelle 2: Liniendicken Maße in Millimeter

Spalte	1	2	3	4	5	6
Zeile	Linienart	Anwendungsbereich	Liniengruppe			
			I	II	III ¹⁾	IV ²⁾
			Zuordnung zu Maßstab			
			≤ 1 : 100		≥ 1 : 50	
Liniendicke						
1	Volllinie	Begrenzung von Schnittflächen	0,5	0,5	1,0	1,0
2	Volllinie	Sichtbare Kanten und sichtbare Umrisse von Bauteilen, Begrenzung von Schnittflächen von schmalen oder kleinen Bauteilen	0,25	0,35	0,5	0,7
3	Volllinie	Maßlinien, Maßhilfslinien, Hinweislinien, Lauflinien, Begrenzung von Ausschnittdarstellungen, vereinfachte Darstellungen	0,18	0,25	0,35	0,5
4	Strichlinie	Verdeckte Kanten und verdeckte Umrisse von Bauteilen	0,25	0,35	0,5	0,7
5	Strichpunktlinie	Kenzeichnung der Lage der Schnittebenen	0,5	0,5	1,0	1,0
6	Strichpunktlinie	Achsen	0,18	0,25	0,35	0,5
7	Punktlinie	Bauteile vor bzw. über der Schnittebene	0,25	0,35	0,5	0,7
8	Maßzahlen	Schriftgröße	2,5	3,5	5,0	7,0

1) Die Liniengruppe I ist nur dann anzuwenden, wenn eine Zeichnung mit der Liniengruppe III angefertigt, im Verhältnis 2 : 1 verkleinert wurde und die Verkleinerung weiterbearbeitet werden soll. In der Zeichnung mit der Liniengruppe III ist dann die Schriftgröße 5,0 mm zu wählen. Die Liniengruppe I erfüllt nicht die Anforderungen der Mikroverfilmung.

2) Die Liniengruppe IV ist für Ausführungszeichnungen anzuwenden, wenn eine Verkleinerung z. B. vom Maßstab 1 : 50 in den Maßstab 1 : 100 vorgesehen ist und die Verkleinerung den Anforderungen der Mikroverfilmung zu entsprechen hat. Die Verkleinerung kann dann gegebenenfalls mit den Breiten der Liniengruppe II weiterbearbeitet werden.

Abbildung 2-2: Liniendicken nach DIN 1356-1 1995-02, Seite 6

Die Vorgaben für Liniendicken haben sich über längere Zeit kaum verändert und lassen sich 30 Jahre alter Literatur mit fast identischen Angaben wiederfinden.⁶

Die DIN 206 über Schraffur wurde 2002, wie am Anfang des Kapitels schon einmal erwähnt, durch die DIN ISO 128-50 ersetzt. In der folgenden Abbildung ist eine Vergleichstafel beider Normen zu sehen.

Baustoff	Schraffur DIN ISO 128-50	Schraffur DIN 1356-1
Boden	gewachsen geschüttet	
Kies		
Sand		
Beton – unbewehrt		
Beton – bewehrt		
Mauerwerk		
Mauerwerk – erhöhte Festigkeit		
Holz – quer zur Faser		
Holz – parallel zur Faser		
Dämmstoff		
Dichtstoff		Dichtstoff Abdichtung
Metalle		

Abbildung 2-3: Schraffuren der DIN ISO 128-50 und der DIN 1356-1 im Vergleich (Holschemacher K., 2009, S. 12.25)

⁶ Prenzel (1982), Seite 26

2.1.4 Falten nach DIN 824

Das Falten von Plänen wird in der DIN 824 festgelegt und ist für die Verwaltung der Planunterlagen nicht zu unterschätzen. Der Plankopf befindet sich immer rechts unten in dem A4-großen Bereich, welcher sich nach dem Falten außen befindet und somit beim Durchblättern der eingehafteten Unterlagen alle Informationen des vorliegenden Planes auf einen Blick zur Verfügung stellt. Bei der Positionierung von verschiedenen Ansichten wird zum Teil darauf geachtet, dass der Plan später so wenig wie möglich wieder entfaltet werden muss, um die ganze Ansicht zu zeigen.

DIN A4	DIN A3	DIN A2	DIN A1																		
DIN A0																					
			<table border="1"> <thead> <tr> <th>Format</th> <th>Blattgröße</th> <th>Zeichenfläche</th> </tr> </thead> <tbody> <tr> <td>A4</td> <td>210 x 297</td> <td>180 x 277</td> </tr> <tr> <td>A3</td> <td>297 x 420</td> <td>277 x 390</td> </tr> <tr> <td>A2</td> <td>420 x 594</td> <td>400 x 564</td> </tr> <tr> <td>A1</td> <td>594 x 841</td> <td>574 x 811</td> </tr> <tr> <td>A0</td> <td>841 x 1189</td> <td>821 x 1159</td> </tr> </tbody> </table>	Format	Blattgröße	Zeichenfläche	A4	210 x 297	180 x 277	A3	297 x 420	277 x 390	A2	420 x 594	400 x 564	A1	594 x 841	574 x 811	A0	841 x 1189	821 x 1159
Format	Blattgröße	Zeichenfläche																			
A4	210 x 297	180 x 277																			
A3	297 x 420	277 x 390																			
A2	420 x 594	400 x 564																			
A1	594 x 841	574 x 811																			
A0	841 x 1189	821 x 1159																			

Abbildung 2-4: Falten nach DIN 824 (aus *Entwurfs- und Berechnungstabellen für Bauingenieure*, Tafel 12.23a, Seite 12.23)

2.2 Längsschnittproblematik

Es gibt viele unterschiedliche 3D-Modellierungsprogramme auf dem Markt, doch zum jetzigen Zeitpunkt (Ende 2012) werden gekrümmte Schnittführungen nicht oder nur ansatzweise angeboten. Dies ist also kein Inventor spezifisches Problem, sondern bereitet momentan vielen Softwareentwicklern Kopfzerbrechen. Der Längsschnitt durch eine gekrümmte Brücke stellt eine Abwicklung dieses Schnittes in eine Ebene dar. Die gemessene Länge der Brücke im Längsschnitt entspricht der wahren Länge der Brücke.

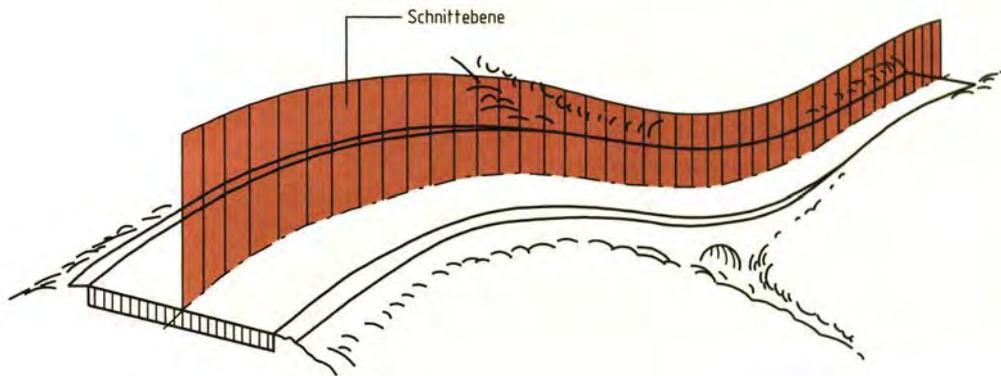


Abbildung 2-5: Schnittführung beim Längsschnitt (aus Batran B. et al., 2010, S.14)

Der momentan einzige Lösungsansatz mit den vorhandenen Mitteln ist eine Approximation der Fahrbahnlängsachse, welche auch noch als Gradiente bezeichnet wird, durch eine Polylinie. Dabei enthält jede Approximation gewisse Approximationsfehler. Bei dieser Approximation gibt es zwei wesentliche Fehler, die im Folgenden näher analysiert werden.

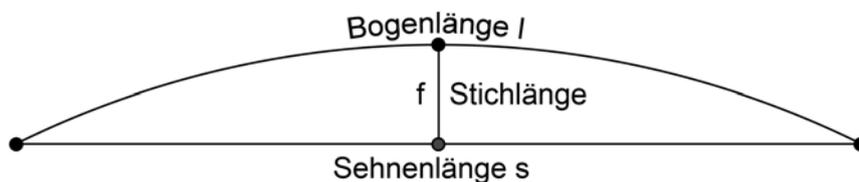


Abbildung 2-6: Begriffe des Kreisbogens

2.2.1 Erster Approximationsfehler

Der erste Approximationsfehler besteht in der Verkürzung der wahren Länge der Brücke. Die Länge der Bogensehne ist immer geringer als die Kreisbogenlänge. Folgende Formeln gelten für die Beiden Längen:

- Länge eines Kreisbogens: $L = \pi \cdot r \cdot \frac{\alpha}{180}$ [Grad] mit $r =$ Bogenradius
- Länge einer Kreissehne: $2 \cdot r \cdot \sin\left(\frac{L}{2r}\right)$

Die Abweichung ergibt sich also aus der Differenz dieser beiden Längen, oder in Prozent ausgedrückt:

$$Fehler(r, L) = 1 - \frac{L}{2 \cdot r \cdot \sin\left(\frac{L}{2r}\right)} [\%]$$

Da diese Abweichung von zwei Parametern abhängt, nämlich der Länge und des Radius der Gradiente, ist es schwierig diese in einem planaren Diagramm darzustellen. Deswegen wird in der Abbildung 2-7 als Beispiel der Sehnensfehler bei einem Radius von 300m als Diagramm dargestellt. Anschließend folgt noch ein Flächendiagramm, welches als horizontale Achse die Länge der Brücke (in Meter) und als vertikale Achse mögliche Bogenradien (in Meter) besitzt. Der Approximationsfehler wird farblich in 3 Bereiche 0%-0,05%, 0,05%-0,10% und 0,10%-0,50% eingeteilt und dargestellt.

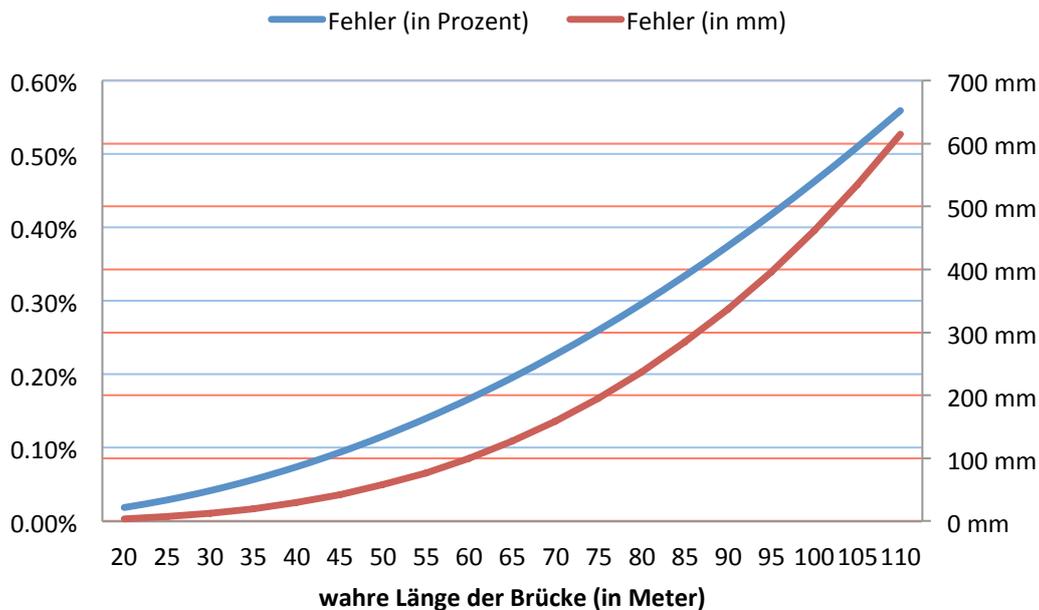


Abbildung 2-7: Diagramm: Sehnensfehler bei einem Radius von 300m

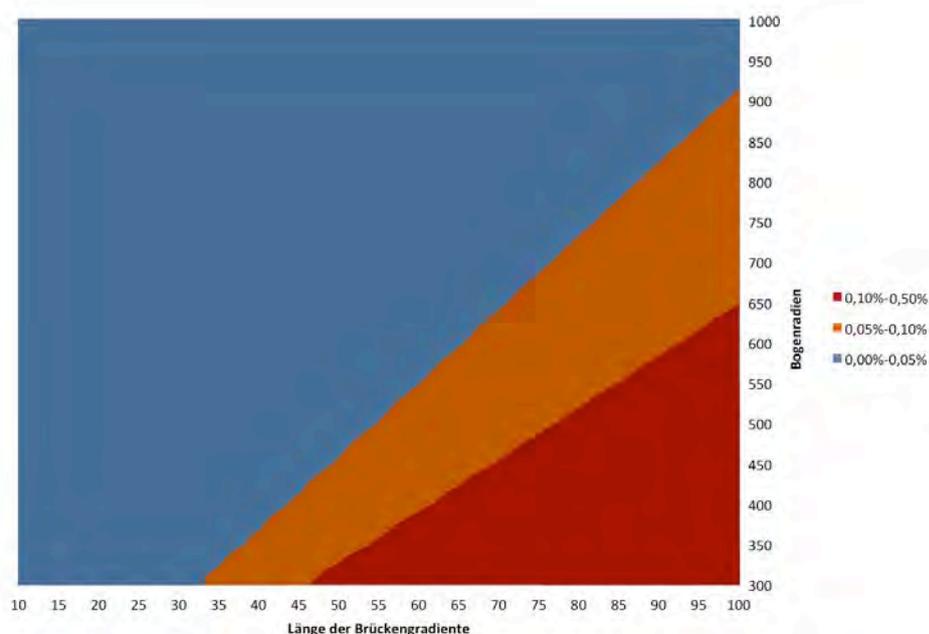


Abbildung 2-8: 1. Approximationsfehler als Flächendiagramm

2.2.2 Zweiter Approximationsfehler

Neben der verkürzten Länge der Brücke, gibt es durch diese Approximation noch einen weiteren Fehler in der Querrichtung. Der Schnitt folgt immer senkrecht zur Schnittlinie. Die Normale zur Sehne und die Normale zum Kreisbogen sind nur auf der Winkelhalbierenden des Bogens gleich. An jedem anderen Ort auf dem Kreisbogen weichen beide Normalen voneinander ab und somit entsteht überall ein Abweichungsfehler, der am Anfang und am Ende vom Kreisbogen am größten ist.

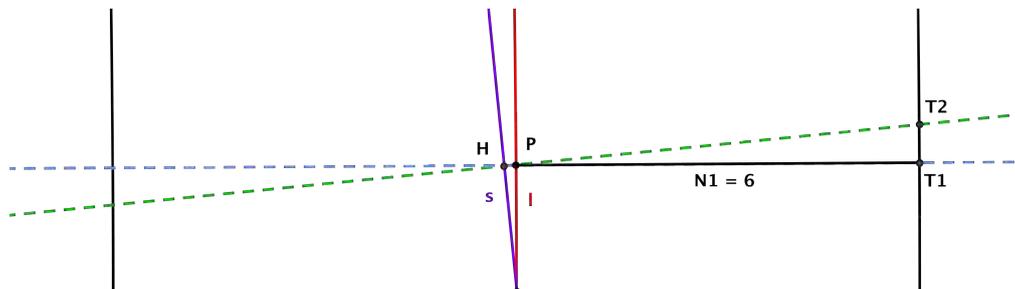


Abbildung 2-9 : zweiter Approximationsfehler

Die Abbildung 2-3 veranschaulicht den zweiten Approximationsfehler anhand einer Grafik. Folgende Linien sind zu unterscheiden:

- Die durchgezogene, rote Linie „l“ ist die Tangente.
- Die durchgezogene, violette Linie „s“ ist die Sehne.
- Die gestrichelte, blaue Linie ist die Normale zur Tangente durch den Punkt P.
- Die gestrichelte, grüne Linie ist die Normale zur Sehne durch den Punkt H.

Beide Punkte P und H haben die gleiche Distanz zum Endpunkt der Tangente, weswegen die Normale zur Sehne (grün) nicht durch den Punkt P geht. Auf der Abbildung hat es dennoch den Anschein als wäre dies der Fall. Das kommt aber nur durch den Umstand, dass die Distanz zum Endpunkt der Tangente, welcher sich am unteren Rand der Abbildung befindet, sehr gering ist.

Der zweite Approximationsfehler besteht in der Abweichung zwischen den zwei Punkten T_1 und T_2 , als Distanz ausgedrückt durch die Länge $\overline{T_1T_2}$. Dabei wird der Fehler größer je weiter die Punkte T_1 und T_2 von P weg sind, also je breiter die Brücke ist. Dieser Fehler ist deswegen hauptsächlich bei Längsschnitten zu berücksichtigen, wo äußere Kanten der Brücke durch gestrichelte Linien angedeutet werden. Bei einem Schnitt mit der Schnitttiefe 0 dürfte der Fehler weniger relevant sein.

Bei der Approximation durch eine Polylinie addiert sich der erste Approximationsfehler über die gesamte Länge der Approximationslinie, wohingegen der zweite Approximationsfehler sich pro Segment der Polylinie wiederholt.

3 Zeichnungserstellung mittels Autodesk Inventor

Die Zeichnungserstellung in Inventor wird im folgenden Kapitel zuerst einmal allgemein erläutert und daran anschließend wird in Kapitel 4 kurz ein Ausführungsübersichtsplan zu dem 3D-Brückenmodell schrittweise aufgebaut.

3.1 Allgemeines zu Zeichnungen in Inventor

Es gibt zwei unterschiedliche Formate für Zeichnungen in Inventor:

- .IDW - Format: *Inventor Drawing*: Das Inventor-spezifische Format für Zeichnungen
- .DWG - Format: *Autodesk Drawing*: Das bekannte DWG-Format, welches auch in anderen Autodesk Produkten Verwendung findet und für Inventor erweitert wurde.

Die beiden Formate haben nur einen wirklich wesentlichen Unterscheidungspunkt: Zeichnungen, die als .dwg-Dateien gespeichert wurden, können ohne Konvertierung in AutoCAD geöffnet werden, .idw-Dateien hingegen nicht. Dies ermöglicht es, Inventor Zeichnungen schnell intern oder an Kunden weiterzureichen, die selbst kein Inventor im Einsatz haben.

Die Dateigröße von .idw-Dateien soll allerdings geringer sein als die von .dwg-Dateien, was bei der heutigen Größe von Festplatten jedoch nicht das Hauptargument bei der Wahl zwischen den beiden Formaten sein wird. Deswegen wird im weiteren Verlauf der Arbeit auf das .dwg-Format gesetzt.

3.1.1 Neue Zeichnung erstellen

Neue Zeichnungen werden in Inventor wie jede andere neue Datei über den selben Menüpunkt erstellt. Dabei kann der Benutzer zwischen mehreren vordefinierten Vorlagen wählen. Die Vorlagen selbst befinden sich standardmäßig im Ordner „C:\Users\Public\Documents\Autodesk\Inventor 2013\Templates\“. Dieser kann aber in Ordneroptionen der Projekteinstellungen (zu finden über die Programmschaltfläche ►Verwalten ►Projekte) geändert werden.

Die Zeichnungsumgebung bietet oben die aus dem Modellierungsbereich bekannte Multifunktionsleiste, die alle Funktionen gruppiert, und links einen Übersichtsbrowser. Dieser enthält einen Zeichnungsressourcen-Ordner, auf den in einem späteren Kapitel noch ausführlich eingegangen wird, und unter diesem Ordner werden die Blätter mit den jeweiligen Ansichten in einer Hierarchie angezeigt.

3.1.2 Ansichten im Überblick

Es gibt unterschiedliche Typen von Ansichten. Die sogenannte *Erstansicht* ist dabei die Grundlage für jede andere Ansicht. So wird auch bei jeder neuen Zeichnung zuerst eine Erstansicht platziert, bevor weitere Ansichten wie zum Beispiel Schnittansichten hinzugefügt werden.

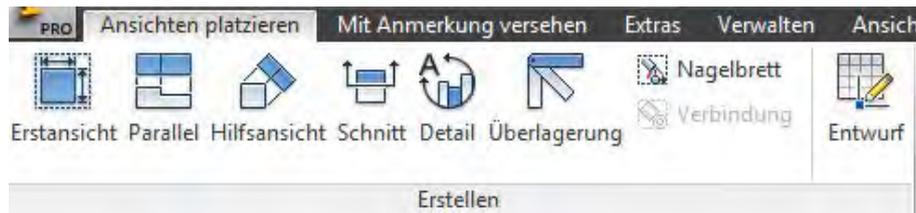


Abbildung 3-1: Auswahl der zur Verfügung stehenden Ansichtstypen

Von den 6 Ansichtstypen sind vor allem zwei für das spätere Vorhaben relevant: Die **Erstansicht** und die **Schnittansicht**.

3.1.3 Erstansicht

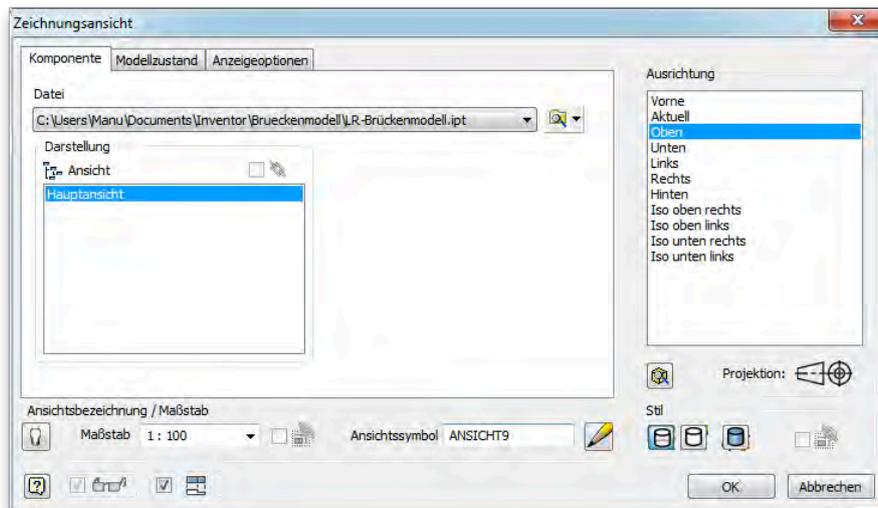


Abbildung 3-2: Einstellungsfenster für eine Ansicht

Bei der Erstellung einer Erstansicht wird das oben gezeigte Zeichnungsansichtseinstellungsfenster geöffnet. Neben der Auswahl des anzuzeigenden Bauteils, kann der Maßstab angegeben, der Linienstil geändert und eine Ausrichtung gewählt werden.

Beim Maßstab ist wichtig darauf zu achten in welcher Einheit das Bauteil erstellt und welche Grundeinheit in der Zeichnung festgelegt wurde.

Die drei möglichen Linienstile sind:

- mit verdeckten Linien,
- ohne verdeckte Linien,
- Schattiert.

Die auswählbare Ausrichtung ist gleich der Ausrichtung des Bauteils aus dem Modellierbereich. Sofern dort die Ausrichtung nicht geändert wurde, zeigt die Ausrichtung „Oben“ die XZ-Ebene und nicht die im Bauwesen übliche XY-Ebene.

3.1.4 Schnittansicht

Schnittansichten beziehen sich immer auf eine andere Ansicht. Die Schnittlinie wird dabei in einer separaten Skizze gezeichnet und muss geradlinig verlaufen. Gekrümmte Schnitte sind

leider derzeit (noch) nicht möglich. Die Ansichtseinstellungen sind nicht großartig anders als die der Erstanischt.

In den Schnitteigenschaften kann angegeben werden, ob es ein Vollschnitt sein soll oder ob der Schnitt eine gewisse Tiefe hat. Eine Tiefe von 0 gibt es nicht, sondern ein 0-Schnitt hat eine tatsächliche Tiefe von 0,000012, was aber in der Zeichnung genau wie ein 0-Schnitt angezeigt wird.⁷ Dadurch kann es aber beim nachträglichen Verdecken von Linien dazu führen, dass zwei gleiche Linien übereinanderliegen. Zur Schnittführung stehen zwei Methoden bereit: *projiziert* und *ausgerichtet*. Die Methode ist relevant sobald die Schnittlinie aus mehreren Segmenten besteht.

3.1.5 Zusätzliche Skizzen

In der Zeichnungsumgebung gibt es außerdem noch die Möglichkeit weitere Skizzen hinzuzufügen. Durch die Projektion von Geometrien aus einer Ansicht, können diese durch zusätzliche Skizzen auf eine einfache Art und Weise erweitert werden.

Die grundlegende Vorstellung bei der Verwendung von 3D Modellen geht davon aus, dass alle Details, die nachher auch auf dem Plan sichtbar sein sollen, modelliert werden, um den Aufwand einer Nachbearbeitung so gering wie möglich zu halten. Allerdings steigt dadurch die Komplexität der Modellierung sehr schnell an, sodass das Hinzufügen von Detailstandards aus einer erweiterbaren Bibliothek zielführender ist.

Solche zusätzlichen Skizzen bieten, wie jede andere Skizze in Inventor auch, die Möglichkeit geometrischen Abhängigkeiten und zusammenhängende Bemaßungen einzufügen. Parameter sind jedoch nicht abrufbar. Dies gilt im Allgemeinen für Skizzen in der Zeichnungsumgebung. Obwohl Inventor intern scheinbar mit Parametern arbeitet, da voneinander abhängige Bemaßungen erstellt werden können, wird die aus dem Modellierbereich bekannte Parameterfunktionalität nicht angeboten, obschon das Parameterfenster auch in der Zeichnungsumgebung zu finden ist.

Zusätzliche Skizzen besitzen noch zwei weitere entscheidende Nachteile. Die Funktion zur Projektion von Geometrien ist nicht anwendbar auf andere Skizzen sondern nur auf Ansichtsgeometrien. Außerdem können Skizzen nicht aktiviert oder deaktiviert werden. Ihre Sichtbarkeit kann also nicht geändert werden. Dies hat zur Folge, dass Skizzen schlecht aufgeteilt und keine zwei unterschiedliche Varianten erstellt werden können, die je nach Bedarf angewendet werden könnten.

Des Weiteren muss darauf geachtet werden, welche Geometrien bei der Projektion verwendet werden, um zu vermeiden, dass sich bei der Veränderung des 3D Modells die Skizze nicht anpasst, da eine projizierte Geometrie nicht mehr gefunden wird.

⁷<http://wikihelp.autodesk.com/Inventor/deu/2013/Help/1309-Autodesk1309/1813-Zeichnung1813/1823-Erstelle1823/1830-Schnitt1830>

3.2 Zeichnungsableitung des Oberbaus

Die Oberbau Modellierung wird momentan durch die Funktion **Erhebung** mit Hilfe von zwei Oberbauskizzen und der Oberbautrasse durchgeführt. Bei dem aktuellen Modell besteht die Trasse im Lageplan aus einer Kurve und im Höhenplan aus einer Geraden. In einer Inventor Schulung, welche von einer auf Autodesk Produkten spezialisierte Firma durchgeführt wurde, wurde gezeigt, dass diese Modellierungsart Abweichungen enthält und der Oberbau eigentlich nicht korrekt modelliert wird. Es gibt scheinbar so direkt keine Möglichkeit einen Querschnitt entlang einer Trasse zu sweepen mit der Kondition, dass die Z-Achse der Querschnittsskizze parallel zur globalen Z-Achse bleibt ohne sich zu verdrehen.

Durch Zwischenschnitte lassen sich bei starken Krümmungen die Abweichungen im Querschnitt schnell feststellen.

Folglich wird anhand von einem realistischen Gesamtbeispiel die Problematik noch einmal schrittweise veranschaulicht und im Detail erläutert. Anschließend wird eine erste Fehlereinschätzung durchgeführt.

3.2.1 Modellierung

Für diesen Versuch wurden drei Trassen erstellt, deren Komplexität von Trasse zu Trasse zunimmt. An jeder Raumkurve wurde ein Rechteck (10 Meter breit und 4 Meter hoch) mit seinem Mittelpunkt auf den jeweiligen Endpunkten der Raumkurve platziert und zwar so, dass das Rechteck sich lotrecht zur XY-Ebene befindet und die Normale des Rechtecks parallel zum Richtungsvektor der Raumkurve an dem angesetzten Punkt in der XY-Ebene ist. Das Rechteck wurde also zusätzlich um seine Z-Achse gedreht und senkrecht zur Raumkurve ausgerichtet. Anschließend wurde die Erhebung mit den zwei Rechtecken als Querschnitte und der Raumkurve als Mittellinie durchgeführt. Bei der dritten Trasse wurde ein zweiter Versuch mit einer weiteren Skizze, die in der Mitte der Trasse und ansonsten gleich wie oben beschrieben platziert wurde, unternommen.

Die drei Trassen selbst basieren auf einen Kreisbogen in der XY-Ebene mit dem Radius 300m, welcher der minimale Radius darstellt, der im Straßenbrückenbau zu finden ist und wodurch der wahrscheinlich größte Fehler entsteht. (Es gibt noch Ausnahmefälle, wo kleinere Radien verwendet werden. Diese werden aber hier nicht berücksichtigt und sind auch nicht der Regelfall.) Die Länge der Brücke wurde mit ca. 150 Meter angegeben. Da die Erhebung von Querschnittsskizze zu Querschnittsskizze stattfindet, dürfte der Fehler kleiner sein, je kleiner die Brücke ist, also je näher die Skizzen sich zueinander befinden.

Im Detail unterscheiden sich die Trassen wie folgt:

1. Trasse: Liegt planar in der XY-Ebene. (→ 2D Kurve)
2. Trasse: Wurde als Schnittlinie zwischen dem Kreisbogen und einem linearen, projizierten Höhenplan, welcher in der YZ-Ebene liegt, erstellt. (→ 3D Raumkurve)
3. Trasse: basiert auf folgender 3D-Gleichungskurve:

$$\begin{cases} z(t) &= \frac{8}{100} \cdot t \\ x(t) &= r \cdot \cos\left(\frac{t}{r}\right) \\ y(t) &= r \cdot \sin\left(\frac{t}{r}\right) \end{cases}$$

Mit r als Bogenradius und t als Parameter. Im Beispiel $r = 100$ und t von 0 bis 150.

Die Kurve hat in der Höhe, also im realen Höhenplan, eine konstante Steigung von 8%. Projiziert auf die ZY-Ebene wäre der Höhenplan nicht linear wie bei der 2. Trasse.

Die dritte Trasse stellt dabei die im Verkehrswegebau realistischste Trasse dar.

3.2.2 Schnitte und Abmessungsfehler

Nach der Modellierung wurde eine neue Zeichnung angelegt, eine Draufsicht erstellt und zwei Querschnitte, einer ungefähr in der Mitte der Strecke und einer in einem Viertel der Strecke, erstellt. Die Schnitte wurden dabei immer senkrecht zur Trasse angelegt.

Die Querschnitte wurden gemessen und dabei wurde folgendes festgestellt:

3.2.2.1 Erste Trasse

Der planare Oberbau zeigt weder in der Höhe noch in der Breite Abmessungsfehler an. Die Genauigkeit geht in dem Fall bis auf Hundertstel Millimeter.

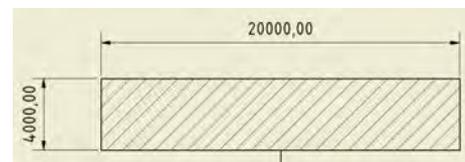


Abbildung 3-3 Resultat des 1. Versuchs

3.2.2.2 Zweite Trasse

Bei der zweiten Trasse sind erste Fehler zu erkennen. Bei den beiden getätigten Schnitten waren der maximale Längsfehler 5,81mm und der maximale Höhenfehler 0,15mm. Es wurde allerdings durch verschieben der Schnittlinien nicht versucht den maximalen Fehler zu finden. Dies sollte nur als erste Einschätzung dienen.

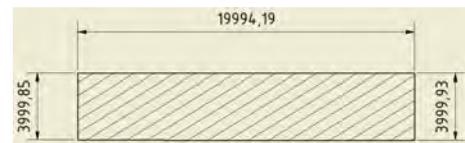


Abbildung 3-4: Resultat des 2. Versuchs

Die Abweichung von 5,81mm erscheint auf den ersten Blick recht viel. Es muss aber bedacht werden, dass die Breite des Querschnitts eine Abmessung von 20 Meter hatte, der Fehler also bezogen auf die Breite nur ca. 0,03 % beträgt. In der Höhe beträgt der Fehler sogar weniger als 0,01%.

3.2.2.3 Dritte Trasse

Auch bei der dritten Trasse treten, wie nach dem Ergebnis der zweiten Trasse zu erwarten, Fehler auf. Die gemessenen Fehler sind in der Breite etwas geringer und in der Höhe etwas größer. Der Längsfehler beträgt 3,8mm, der Höhenfehler 0,76mm. In Prozent ausgedrückt wären es jeweils circa 0,02%. Wie auch beim zweiten Beispiel wurde hier nicht der Versuch unternommen den globalen Maximalfehler zu suchen. Durch die Bogenform und durch die Platzierung der Schnitte ungefähr mittig und in einem Viertel der Trasse dürfte der maximale Fehler allerdings nicht viel größer sein als die gemessenen Werte. Beim 2. Versuch mit einer weiteren Querschnittsskizze in der Mitte konnte

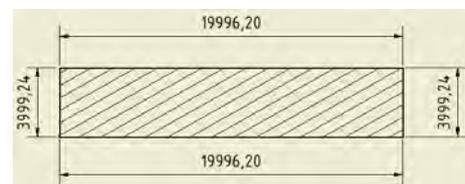


Abbildung 3-5: Resultat des 3. Versuchs

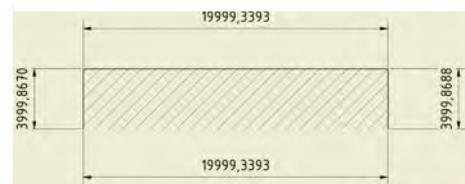


Abbildung 3-6: Resultat des 4. Versuchs

der Höhenfehler auf 0,133mm und der Längsfehler auf 0,661mm reduziert werden.

3.2.3 Fazit

Es ist momentan nicht möglich mit Inventor den Oberbau einer Brücke korrekt zu modellieren und es entstehen immer Abmessungsfehler. Diese sind allerdings, sofern die Trasse keine außergewöhnlichen Krümmungen beinhaltet oder extrem lang ist, sehr gering und soweit eigentlich vernachlässigbar. Je größer der Radius und je geringer die Steigung, desto geringer müssten die Fehler sein. Die Bemaßungen werden normalerweise bis auf eine gewisse Stelle gerundet und dadurch dürften die Fehler am Ende eigentlich nicht einmal sichtbar sein. Dabei müsste nur drauf geachtet werden ob automatisch auf- oder abgerundet wird. Werden die normalen Rundungsregeln angewendet, würde im zweiten Beispiel anstatt 20,00m 19,99m angezeigt werden. Des Weiteren kann durch etwas Mehraufwand mit Hilfe zusätzlicher Querschnittsskizzen als Zwischenschritte die Genauigkeit sehr stark erhöht werden.

Der Winkelfehler wurde nicht tiefgründig überprüft. Nur für das dritte Beispiel wurden die vier Eckwinkel einmal gemessen und es wurde jedes mal $90,0^\circ$ angezeigt. Ob im Zehntel Gradbereich vielleicht ein Fehler vorhanden ist wurde nicht überprüft.

Eine weitere Anmerkung, die zwar mit der Thematik zu tun hat, sich allerdings nicht nur darauf beschränkt, ist, dass die Schnittkanten keine Geraden sondern Splines sind. Dies wird erst sichtbar beim Exportieren und Öffnen in AutoCAD oder bei der Bemaßung, da sich die Kanten nicht mit einem Klick auf die Kantenlinie bemaßen lassen, sondern nur von Eckpunkt zu Eckpunkt. Der Grund dafür dürfte der selbe sein wie für die Abmessungsfehler, nämlich ein leichtes Verdrehen des Querschnittes beim Sweeping. Durch diese Verdrehung wird der Querschnitt gekrümmt.

3.3 Eigene Vorlage erstellen

Im folgenden Kapitel wird eine eigene Vorlage schrittweise erstellt und gleichzeitig die Themen Rahmen, Plankopf und Stile in Inventor besprochen ohne diese im Vorfeld allgemein anzuschneiden, um Wiederholungen zu vermeiden. Das Festlegen verschiedener Planelemente und Stile in einer Vorlage macht Sinn, da gewisse Elemente wie beispielsweise der Rahmen bei jedem A0 Plan prinzipiell der selbe ist. Durch eine Vorlage lässt sich also Wiederholungsarbeit vermeiden und eine gewisse Vereinheitlichung der Pläne kann leichter garantiert werden.

3.3.1 Rahmen

Über den Zeichnungsressourcenordner im Objektbrowser lassen sich einzelne Rahmen⁸ definieren und je nach Bedarf einfügen. Die Rahmendefinition an sich passiert in einer Skizzenumgebung und lässt sich wie von normalen Skizzen gewohnt bearbeiten. Als Grundlage für jede Rahmendefinition dienen die vier Eckpunkte eines Blattes. Dadurch können die Rahmen unabhängig von der eigentlichen Blattgröße gestaltet werden.

Neben gewöhnlichen Rahmen existiert auch noch die Möglichkeit spezielle Zonenrahmen zu erstellen. Diese Rahmen besitzen vordefinierte Zonen, die bei der Anzahl der Zonen und der Zonenbezeichnungen flexibel sind. Die Werte können in einem besonderen Einstellungsfenster festgelegt und geändert werden. Allerdings lässt sich die Geometrie der Zoneneinteilung nicht ändern und auch andere Elemente sind von Inventor fest vorgegeben, so dass sich diese Art von Rahmen eher schlecht an persönliche Bedürfnisse anpassen lässt.

In den Rahmen kann auch ein sogenannter Verbindungspunkt eingefügt werden. Dieser legt fest, wo später das Schriftfeld eingefügt wird. Falls dieser Verbindungspunkt nicht gesetzt wird, platziert Inventor das Schriftfeld automatisch in die rechte untere Ecke, so dass der einzige Zweck dieses Punktes ist, das Schriftfeld mit einem gewissen Abstand zum Rahmen zu platzieren.

Bei getriebenen Bemaßungen, also Bemaßungen die nur als Maßanzeige dienen und die Maße nicht durch Eingabe des Benutzers geändert werden kann, da die bemaßenden Elemente durch andere Konditionen komplett bestimmt sind, ist Vorsicht geboten. Hier scheint es in Inventor momentan (Stand Februar 2013) noch einen Softwarefehler zu geben, der bei erneuter Bearbeitung des Rahmens dazu führt, dass keine Bemaßung, die in direkter Verbindung mit der getriebenen Bemaßung steht, verändert werden kann ohne, dass ein Fehler auftritt. Getestet wurde speziell mit den getriebenen Bemaßungen der fixen Eckpunkt, welche die Blatthöhe und Breite widerspiegeln. Durch diesen Bug kann zum Beispiel kein Einteilungszeichen auf ein Drittel der Blatthöhe gesetzt werden, da die Blatthöhe nur über die Bemaßung der fixen Eckpunkte erhältlich ist. Der Fehler fällt erst auf, wenn der Rahmen erneut bearbeitet wird, aber auch wenn die Blattgröße geändert wird oder beim Versuch einem Blatt mit anderer Größe den Rahmen hinzuzufügen.

⁸<http://wikihelp.autodesk.com/Inventor/deu/2013/Help/1309-Autodesk1309/1813-Zeichnung1813/1814-Übersicht1814/1821-Zeichnung1821>

Für die Vorlage wurden unterschiedliche Rahmen erstellt. Der für das erste Blatt, welches A0 groß ist, ausgewählter Rahmen besitzt unten und rechts Faltmarkierungen, um das Blatt später entsprechend der Norm DIN 824 falten zu können.

3.3.2 Plankopf

Planköpfe werden in Inventor als *Schriftfelder*⁹ bezeichnet und befinden sich wie die Rahmen ebenfalls im Zeichnungsressourcen-Ordner. Es können mehrere Schriftfelder definiert werden und pro Blatt kann ein Schriftfeld eingefügt werden. Auch Schriftfelder sind gewöhnliche Skizzen. Die Platzierung des Schriftfeldes ist beim Erstellen irrelevant, da, wie im Kapitel über Rahmen schon erwähnt, das Schriftfeld später mit dem rechten, untersten Eck an dem im Rahmen definierten Verbindungspunkt eingesetzt wird.

Sofern schon ein Plankopf als AutoCAD-Datei vorhanden ist, kann dieser ganz leicht über den entsprechenden Einfügebefehl aus der Multifunktionsleiste eingefügt werden. Für die Vorlage wurde ein Plankopf der Firma *Obermeyer* eingefügt, der freundlicherweise zur Verfügung gestellt wurde.

3.3.2.1 iProperties

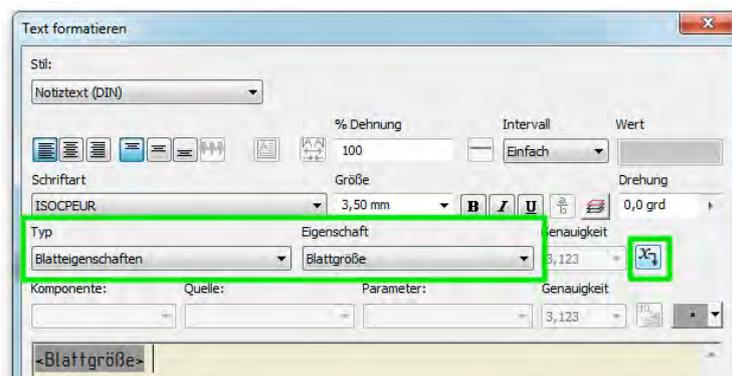


Abbildung 3-7: iProperties als Text einfügen

„iProperties“ ist im Prinzip einfach nur ein anderer Name für die Eigenschaften von Inventor Dateien. Es gibt eine riesige Anzahl an iProperties. Einige davon werden automatisch erstellt und ausgefüllt (z.B. *Blattgröße*), andere (z.B. *Autor*, *Firma*, *Beschreibung*, ...) können vom Benutzer bearbeitet werden. Neben den von Inventor vordefinierten iProperties können auch noch benutzerdefinierte hinzugefügt werden. Das Thema wird hier nicht weiter im Detail beschrieben sondern nur kurz angesprochen, da für den Plankopf relevante Informationen in den iProperties stecken. Diese können, wie auf der Abbildung 3-7 zu sehen ist, mit dem Text-Befehl eingefügt werden.

Für die Vorlage wurden einige iProperties exemplarisch in das Schriftfeld eingefügt.

3.3.3 Arbeitsblattformate

Arbeitsblattformate stellen eine Art Vorlage für die Platzierung von Ansichten dar, allerdings nur für Erst- und Parallelansichten, also nicht für assoziative Ansichten wie zum Beispiel Schnittansichten. Dabei merkt sich Inventor die Platzierung, Ausrichtung, den Linienstil und

⁹<http://wikihelp.autodesk.com/Inventor/deu/2013/Help/1309-Autodesk1309/1813-Zeichnung1813/1814-Übersicht1814/1822-Schrift1822>

den Maßstab. Sofern ein Blatt mit einigen Ansichten vorhanden ist, kann über das Kontextmenü des Blattes ganz einfach ein solches Arbeitsblattformat gespeichert und später in dem gleichnamigen Ordner in den Zeichnungsressourcen wieder abgerufen werden.

Sie haben hier allerdings nur vollständigshalber Erwähnung gefunden und werden aus zwei Gründen in der Vorlage nicht verwendet. Im Prozess der (Teil-)Automatisierung, wo die Vorlage später benutzt werden soll, werden die Ansichten relativ zu einem gewissen Layout platziert, welches im Vorfeld nicht bekannt ist. Außerdem kann sich der Linienstil und der Maßstab ändern, so dass es wenig Sinn macht im Vorfeld eine Ansicht, welche nachher möglicherweise wieder komplett geändert wird, über ein Arbeitsblattformat festzulegen.

3.3.4 Stile und Normen

Inventor arbeitet mit vielen sogenannten *Stilen*, welche nicht nur das Aussehen von Bemaßungen festlegen, sondern unter anderem auch Schraffuren für unterschiedliche Materialien definieren oder einem Layer eine spezifische Linienstärke und eine Farbe zuordnen.

Für jede Kategorie können mehrere Stile angelegt und beim Arbeiten zwischen den Stilen gewechselt werden. Als Beispiel könnten zwei Bemaßungsstile mit unterschiedlichen Einheiten verwendet werden, wenn die Einheiten der angezeigten Elemente sich um einen hohen Faktor unterscheiden.

Eine Ansammlung von Stilen wird in Inventor als **Norm** bezeichnet und diese kann entweder lokal in einer Zeichnungsdatei gespeichert, als *Stildefinitionsdatei* mit der Endung *.styxml (also eine gewöhnliche XML Datei) exportiert/importiert oder in einer Stilbibliothek verwaltet werden. Für kleinere Vorhaben eignet sich die Speicherung der Norm in einer Vorlage, allerdings nur solange diese nicht großartig verändert wird und gleichzeitig zwischen vielen Zeichnungen ein einheitliches Design gewährleistet werden soll. Sobald Inventor in einer Unternehmensumgebung zur Anwendung kommt, sollte auf jeden Fall die Methode der Stilbibliothek gewählt werden. Diese kann zentral auf dem firmeninternen Server von einem einzigen Zuständigen verwaltet und schreibegeschützt in jeweilig unterschiedliche Projekte eingebunden werden. Nur so lässt sich ein einheitliches Erscheinungsbild für Zeichnungen aufrecht erhalten, wenn viele Mitarbeiter involviert sind.

Die Standardnormen, die Inventor von Haus aus mitbringt, zielen hauptsächlich auf den Maschinenbausektor und können nicht eins zu eins im Bauwesen verwendet werden. Deswegen wurde extra für die Zeichnungen des Brückenmodells eine neue Norm namens **Brückenmodell-Norm** angelegt, basierend auf der Standardnorm **DIN**.



Abbildung 3-8: Brückenmodell-Norm als Stilnorm anlegen

Wie oben schon erwähnt, gibt es in Inventor viele unterschiedliche Kategorien, in denen Stile angelegt werden können. Dabei haben die Kategorien im Bauwesen eine andere Relevanz als im Maschinenwesen. So gibt es z.B. die Kategorien „Schweißsymbol“ und „Schweißnaht“, welche speziell bei Schweißkonstruktionen Anwendung finden, die im Maschinenbau recht häufig vorhanden sind, wohingegen sie im konstruktiven Ingenieurbau weniger häufig und erst bei einer sehr detailreichen Modellierung auftreten werden. Folgende Tabelle soll eine beispielhafte Aufzählung von Kategorien darstellen, die im Bauwesen von Bedeutung sind:

Wichtige Kategorien:

Bemaßung

Schraffur

Objektvorgaben

Revisionstabelle

Weniger wichtige Kategorien:

Bezugsstelle

ID

Ansichtsanmerkung

Die einzelnen Kategorien werden im nachfolgenden etwas näher erläutert und für die Vorlage entsprechend geändert, falls Änderungen nötig sind.

Inventor bietet mit dem Stil- und Normeneditor eine zentrale Stelle im Programm an, an der alle Änderungen, die die Stilnorm betreffen, vorgenommen werden können. Der Editor besitzt links eine Spalte mit den jeweiligen Kategorien und rechts das Hauptfenster. Zu jeder Kategorie können mehrere Stile festgelegt werden, welche durch Aufklappen der Kategorie in der linken Spalte aufgelistet werden.

3.3.4.1 Kategorie „Norm“

Die oberste Kategorie listet unterschiedliche Normen auf, von denen eine für die aktuelle Zeichnung aktiv ist. Je nach Vorgaben von externen Stellen, können mehrere angepasste Normen existieren und schnell zwischen ihnen gewechselt werden.

Nach Auswahl einer Norm lassen sich folgende Registerkarten im Hauptfenster des Editors finden:

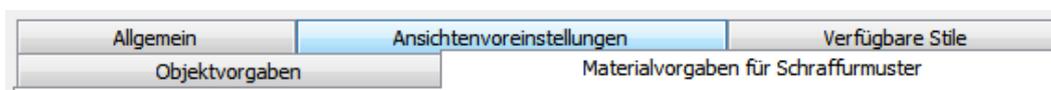


Abbildung 3-9: Registerkarten der Stileinstellungen einer Norm

Unter der Registerkarte **Allgemein** lässt sich ein globaler Linienmaßstab angeben. Nebenbei gibt es die vier verschiedenen Listen *Linienstärke*, *Texthöhe*, *Skalieren* und *Schnittschraffurwinkel* mit voreingestellten Werten, die hier geändert werden können. Des Weiteren gibt es eine Liste mit ausgenommenen Zeichen, die auf die Ansichtsbezeichnungen und Revisionstabellen angewendet wird. Zum Beispiel der Buchstabe „O“ wird ausgenommen, damit dieser nicht mit der Ziffer 0 verwechselt werden kann. Nicht in der Dokumenteneinstellung, wie es im Modellierbereich der Fall ist, sondern in dieser Registerkarte wird die Einheit für das Dokument festgelegt. Da es die Überlegung gab einfach

in mm zu modellieren und nachher die Zeichnung anzupassen, muss diese Einstellung dem entsprechenden Modell angepasst werden. Für die Vorlage wird Meter als Einheit gewählt.

Unter **Materialvorgabe für Schraffurmuster** lassen sich Schraffuren für bestimmte Materialien festlegen. Zum Beispiel für das Material "Stahlbeton" kann hier die allgemein bekannte Schraffur mit abwechselnd einer durchgezogenen Linie und einer gestrichelten mit einem Winkel von 45° verwendet werden. Im Dropdown-Menü der vorgegebenen Schraffurmuster ist keine solche Schraffur direkt zu finden, allerdings lassen sich über den Dropdown-Menüpunkt „Andere...“ weitere Muster aktivieren. Das Schraffurmuster ANSI 33 entspricht dem gesuchten Muster.

Dies ist das einzige Material, welches für die Vorlage definiert wird und sollte nur das Prinzip erläutern. Natürlich muss im Modell den entsprechenden Bauteilen aus Stahlbeton dieses Material auch zugewiesen werden. (Der Materialien-Browser ist im Modellierungsbereich zu finden unter *Extras* ► *Material und Darstellung*)

Unter **Objektvorgaben** wird eine Übersicht über alle Objekttypen (z.B. benutzerdefinierter Symboltext), sowie den jeweiligen zugewiesenen Objektstil und den Layer angezeigt. Durch einen Klick auf Vorgabestil des Objekts bearbeiten, springt die Auswahl in der linken Leiste sofort zu Objektvorgaben, wo sich der Objektstil und der Layer durch Dropdown-Menüs auswählen lässt. Außerdem kann ein neuer, lokaler Objektvorgabestil angelegt und zur Norm hinzugefügt werden.

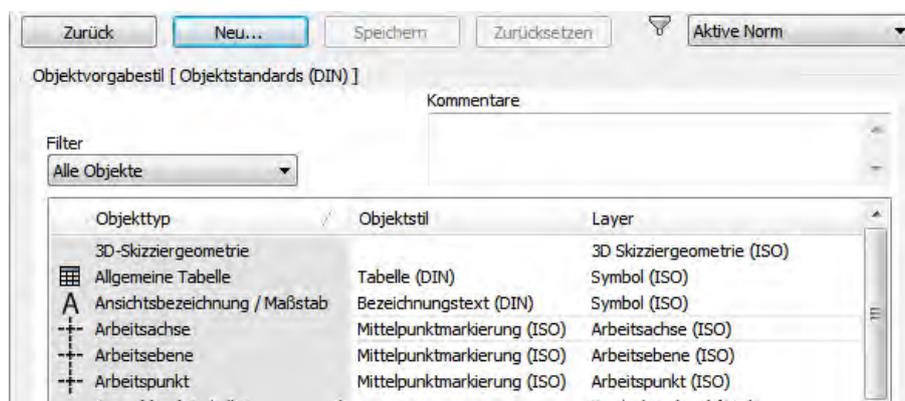


Abbildung 3-10: Fenster der *Objektvorgabe*-Einstellungen

In der Registerkarte **Verfügbare Stile** lassen sich mehrere Stile zu unterschiedlichen Stiltypen (Art Kategorien, wie z.B. Bemaßung, Text, Tabelle, Layer, usw.) an- und ausschalten.

Unter **Ansichtenvoreinstellungen** (siehe Abbildung 3-11) lässt sich unter anderem für verschiedene Ansichtstypen jeweils die Anzeige ändern.

Das Bearbeiten der Anzeige passiert in dem für Inventor typischen "Text formatieren"-Fenster. Hier lassen sich Modell- und Beschriftungseigenschaften einfügen, sowie die Größe, Farbe und die Auszeichnungsarten (fett, kursiv, unterstrichen) der Schrift ändern. Die Schriftart und die Ausrichtung werden allerdings dem Stil zugeschrieben und sind an dieser Stelle nicht änderbar. Der Stil an sich wird an einer anderen Stelle festgelegt und wird hier nur angezeigt.

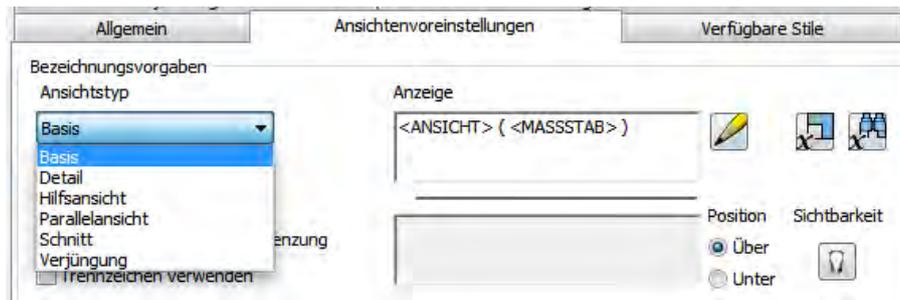


Abbildung 3-11: Anzeigeeinstellungen

Falls ein neuer Bezeichnungstextstil für die Ansicht hinzugefügt werden soll, muss dieser unter **Objektvorgaben** für das Objekt „Ansichtsbezeichnung/Maßstab“ ausgewählt werden, um als Textstil für die Anzeige der Ansicht zu wirken.

Was also nicht geht ist zum Beispiel für die Basis- und die Schnitt-Ansichtsanzeigen unterschiedliche Ausrichtungen im Vorfeld festzulegen. Nachträglich lässt sich die Ausrichtung aber individuell ändern.

Für den Ansichtstyp *Basis* wurde die Anzeige gemäß dem vorliegenden Ausführungsübersichtsplan geändert.

3.3.4.2 Kategorie „Bemaßung“

Hier lassen sich, wie der Kategorietitel schon vermuten lässt, die Bemaßungsstile¹⁰ einstellen. Wer allerdings die Einstellungsmöglichkeiten von Bemaßung in AutoCAD oder AutoCAD Architecture kennt und in dem Umfang auch in Inventor erwartet, wird enttäuscht. Inventors Anwendungsgebiet liegt im Maschinenbauwesen und so finden sich Einstellungen zu Bohrungs- und Stanzinformationen, oder für Biegungshinweise. Andere Einstellungen lassen sich hingegen nur bedingt vornehmen. Da die Bemaßungen auf einem eigenen Layer liegen, dürfte es keine großen Schwierigkeiten bereiten in einem möglichen späteren Weiterverarbeitungsprozess in AutoCAD die Bemaßung an einen anderen Bemaßungsstil anzupassen.

Nichtsdestotrotz sind die grundlegenden Einstellungen möglich. In der Registerkarte „Einheiten“ wurde für die Vorlage die Einheit auf Meter umgestellt, als Dezimalmarkierung „Punkt“ gewählt und die Genauigkeit auf 2 Nachkommastellen gesetzt.

Die alternative Einheit ist eine Einheit, die neben der Primäreinheit angezeigt werden kann. Dies ist nicht zu verwechseln mit der Möglichkeit eine andere Einheit anzuzeigen, wenn die Bemaßung einen bestimmten Wert unterschreitet. Die Möglichkeit gibt es in Inventor so direkt nicht. Dieses Problem könnte zum Beispiel durch einen zweiten Bemaßungsstil mit der entsprechenden Einheit und einem Makro, welches automatisch der Bemaßungsstil einer Bemaßung ab einem vorgegebenen Wert ändert, gelöst werden.

Unter „Anzeige“ lässt sich das Endezeichen der Bemaßung als auch das interne einer Kettenbemaßung auswählen. Als Endezeichen wurde der im Bauwesen übliche Schrägstrich gewählt.

¹⁰ http://wikihelp.autodesk.com/Inventor/deu/2012/Help/0073-Autodesk73/0285-Arbeitsu285/0292-Style_un292/0300-Bemaßung300

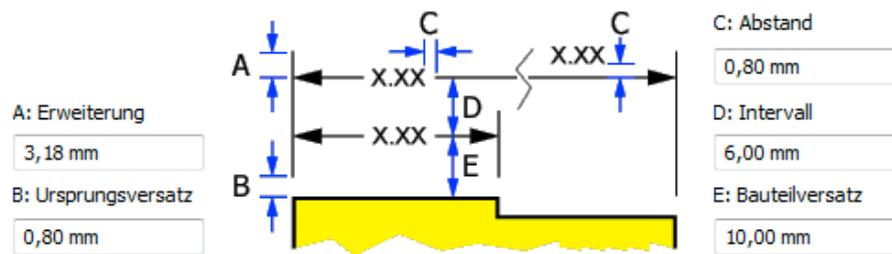


Abbildung 3-12: Detailsinstellungen der Bemaßungslinien

Des Weiteren lassen sich unterschiedliche Abstände festlegen. Der Ursprungsversatz (B) wurde etwas erhöht, damit die Bemaßung nicht zu sehr an der zu bemaßenden Linie klebt. Anders als in AutoCAD lässt sich die Länge der Hilfslinie nicht auf einen fixen Wert begrenzen. Weder an dieser Stelle im Stileditor noch an einer anderen wurde eine solche Möglichkeit gefunden.

Unter „Text“ und „Optionen“ lassen sich noch Ausrichtungsarten, Platzierungen und weitere Einstellungen finden, die aber zum Teil für Bauwerke nicht relevant sind. Es lassen sich also die Voreinstellungen der Norm „DIN“, auf welcher die Brückenmodell-Norm basiert, ohne weitere Änderungen an dieser Stelle übernehmen.

3.3.4.3 Kategorie „Layer“

In dieser Kategorie lassen sich die Layer erstellen, umbenennen und für jeden Layer Linientyp und Linienstärke festlegen. Der Layer „Bemaßung (ISO)“, welcher von der „DIN“-Norm übernommen wurde, hat zum Beispiel eine Linienstärke von 0,25 mm, was der Vorgabe aus der DIN 1356-1 für Maßlinien entspricht.

3.3.4.4 Kategorie „Objektvorgaben“

Diese Kategorie wurde oben schon einmal erwähnt und spiegelt die zentrale Stelle wieder, wo pro Objekttyp, jeweils der Objektstil und das Layer festgelegt werden. Zu jedem Element, welches ein Stil besitzen kann, gibt es ein entsprechender Objekttyp. Titelblocktext, Revisionstabelle, Schwerpunkt, Linearbemaßung sind einige dieser Objekttypen. Des Weiteren gibt es Objekttypen, wie zum Beispiel „Sichtbare Ansichtskanten“ oder „Skizziergeometrie“, denen kein Stil zugewiesen werden kann, dafür aber jeweils ein Layer, welches die Linienstärke festlegt.

3.3.4.5 Kategorie „Text“

Hier lassen sich Vorgaben für unterschiedliche Texte festlegen. Der als „Notiztext (DIN)“ übernommene Textstil wird unter anderem von der Bemaßung benutzt und hat eine Texthöhe von 3,5 mm. Dies entspricht der Textgröße, die in der DIN 1356-1 für Maßzahlen angegeben ist.

4 Beispiel: Ausführungsübersichtsplan

Als Beispiel einer Planableitung mit Inventor wird im Folgenden die manuelle Erstellung eines Ausführungsübersichtsplanes schrittweise durchgenommen und die Resultate der einzelnen Ansichten anhand von Abbildungen gezeigt. Das 3D-Brückenmodell, welches von Herrn Römer (2013) im Zusammenhang mit dessen Bachelorarbeit zum Thema der parametrischen 3D-Modellierung von Brückenbauwerken mit Autodesk Inventor entwickelt wurde, wird als Basismodell benutzt.

4.1 Draufsicht

Die Draufsicht ist eine Erstante mit der Ausrichtung „Oben“ und als Stil „mit verdeckten Linien“. Sie dient als Basisansicht für drei unterschiedliche Schnittansichten.

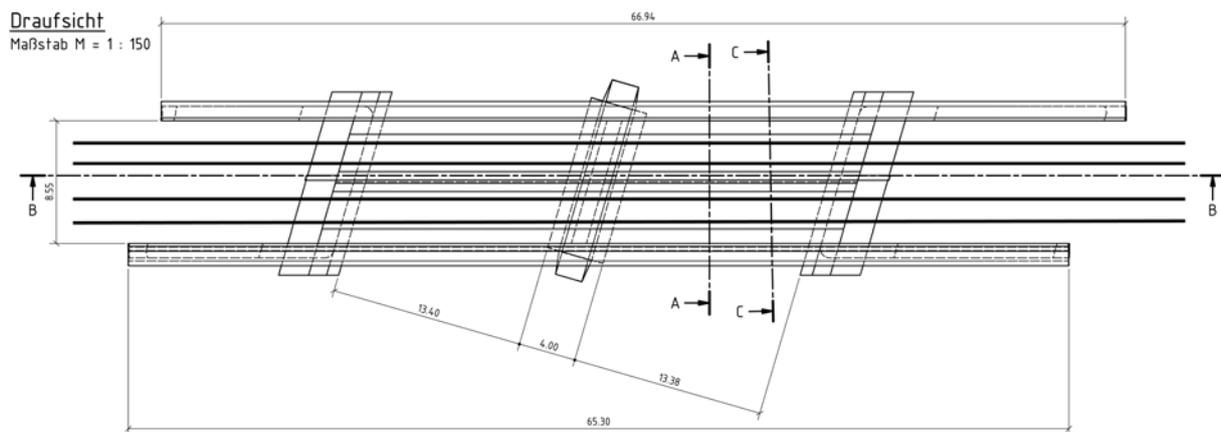


Abbildung 4-1: Draufsicht-Beispiel

4.1.1 Drehen der Draufsicht

Bei einer Draufsicht wird meist versucht die darzustellende Brücke möglichst horizontal auszurichten, sofern die Abmessungen der Brücke in dem ausgewählten Maßstab dies zulässt. Es existieren im Prinzip zwei Möglichkeiten die Ansicht richtig zu drehen, in anderen Worten horizontal auszurichten.

4.1.1.1 Erste Methode

Bei der Erstellung der Erstante kann neben den vordefinierten Ausrichtungen auch eine benutzerdefinierte Ausrichtung gewählt werden. Allerdings sind die Bearbeitungsmöglichkeiten in der Oberfläche, wo diese benutzerdefinierte Ansicht geändert werden kann, sehr beschränkt. Es lässt sich zwar die Funktion **Ausrichten nach...** benutzen, doch klappt diese bei gekrümmten Brücken eher schlecht, da die sichtbaren Bauteile fast allesamt eine Krümmung haben und diese Funktion nicht auf gekrümmte Flächen anwendbar ist. Deswegen wurde als Notlösung in der Modellierungsumgebung ein Rechteck, welcher tangential zur Trasse ist, in einer neuen Skizze gezeichnet und anschließend ein Volumenkörper durch Extrusion erstellt.

Dieser Körper wurde zum Ausrichten verwendet und zum Schluss wurde noch die aktuelle Ansicht als **Oben** festgelegt.

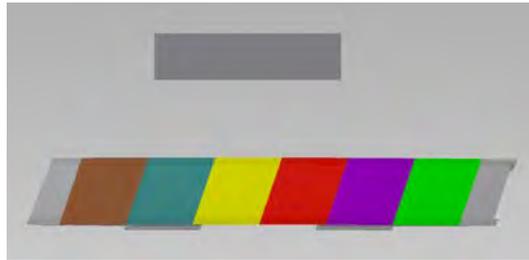


Abbildung 4-2: Drehen mit Hilfe eines Referenzkörpers

In einem Test hat sich herausgestellt, dass sich nach einer Aktualisierung der Zeichnung eine neu erstellte Erstsicht (als Draufsicht) entsprechend gedreht hat. Wohingegen eine alte Erstsicht, welche schon einen Schnitt enthielt, sich nicht gedreht hat. Der absolute Drehwinkel dieser Ansicht blieb ebenfalls unverändert, was das vorherige nur bestätigt. In der Ansichteneinstellung lässt sich die Ausrichtung auch nicht mehr neu auswählen, also ist an dieser Stelle eine Nachbearbeitung durch diese Methode unmöglich.

Eine weitere Anmerkung: Die Ausrichtung nach einem Element beinhaltet nicht, dass eine Abhängigkeit zwischen Element und Ausrichtung hergestellt wird. Wird nämlich das Element anschließend gedreht, ändert sich die Ausrichtung nicht mit. Dies ist ein großer Nachteil für die Flexibilität des parametrischen Modells und der Automatisierung.

4.1.1.2 Zweite Methode

Die zweite Methode besteht darin, die Ansicht in der Zeichnungsumgebung selbst zu drehen. Dabei lässt sie sich entweder um eine **Kante**, einen **absoluten** oder **relativen Winkel** drehen. Vom Prinzip her scheint diese Methode recht simpel zu sein und auch den Anforderungen für diesen Fall gerecht zu werden. Doch es gibt ein paar entscheidende Nachteile.

Durch den Umstand, dass die vorliegende Brücke gekrümmt ist, existieren wenig gerade Kanten, die sich auswählen lassen, um die Option **Um Kante drehen** zu benutzen. Des Weiteren scheint bei der Auswahl einige gerade Kanten nicht beachtet zu werden, wobei nicht ersichtlich ist, nach welchen Kriterien dies geschieht. Dadurch wird das Ganze ein eher willkürliches Prozedere und lässt sich schwer in ein strukturiertes Vorgehen einbinden.



Abbildung 4-3: „Ansicht Drehen“-Fenster

Einfacher ist es mit einem **absoluten Winkel** zu arbeiten.

Dafür muss zuerst in der Modellierungsumgebung der Winkel zwischen der X-Achse und der Brückentrasse am Kreuzungspunkt gemessen werden. Dabei muss, sofern noch nicht vorhanden, in eine (neue) Skizze eine tangentielle Linie zur Trasse gezeichnet werden. Anschließend kann der Winkel gemessen und sein negativer Betrag zum Drehen der Zeichnung verwendet werden.

4.1.1.3 Nachteile beider Methoden

Späteres Drehen, egal mit welcher Methode, kann ungewollte Auswirkungen auf nachträglich erstellte Skizzen haben. Skizzenelemente ohne Abhängigkeiten werden korrekt gedreht, bei Elementen mit Abhängigkeiten können Abweichungen auftreten. Dies wurde mit einem

Kreis, der keine Abhängigkeiten hatte und einer Linie, die rechtwinklig zu einer projizierten Gerade, die zur Brücke gehört, getestet. Dabei hat die Linie zwar die Rechtwinkligkeit beibehalten, allerdings hat sie sich verschoben. Eine Abstandsabhängigkeit war nicht vorhanden. Der Kreis war auch nach der Drehung an der richtigen Stelle.

Durch das Zuschneiden der Zeichnung kann es vorkommen, dass vorher verdeckte Elemente auf einmal wieder auftauchen oder ein Teil der Brücke außerhalb der Zuschneidegrenze landet und weggeschnitten wird.

4.1.1.4 Fazit

Beide Methoden eignen sich auf den zweiten Blick gleichermaßen gut, doch sind sie beide nicht ideal. Welche sich für die spätere Automatisierung besser eignet, muss sich erst noch herausstellen. Was zum jetzigen Zeitpunkt auf jeden Fall wichtig ist, ist dass das Drehen eines der ersten Schritte bei der Erstellung einer Ansicht ist. Dadurch lassen sich spätere Anpassungen unter anderem der zusätzlichen Skizzen, der Zuschneidung und der Bemaßung vermeiden.

Was an dieser Stelle wahrscheinlich offensichtlich erscheint und trotzdem erwähnt werden muss, ist der Umstand, dass nach einer Änderung der Trasse, je nachdem welche Änderungen vorgenommen wurden, die Ausrichtung wieder aufs Neue angepasst werden muss.

4.2 Regelquerschnitt

Der Regelquerschnitt ist ein Schnitt mit der Tiefe 0, der nur durch den Oberbau geht. Die Schnittlinie liegt senkrecht zur Gradienten. Jedes geschnittene Bauteil bekommt automatisch eine Schraffur zugewiesen, die bei dem Beispiel zum Teil manuell ausgeblendet wurden. Zur Bemaßung wurden mit Hilfe von einer zusätzlichen Skizze noch die Mittellinien eingezeichnet.

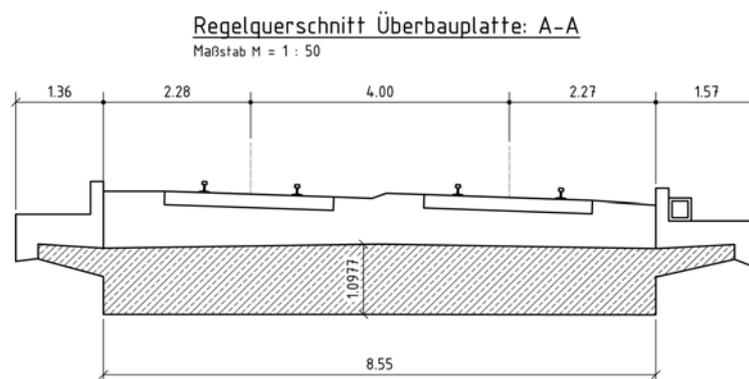


Abbildung 4-4: Regelquerschnitt-Beispiel

4.3 Querschnittsansicht

Die Querschnittsansicht ist ein Vollschnitt kurz vor einem der beiden Widerlager. Durch den Vollschnitt werden die Flügel und das Fundament der Widerlager angezeigt. Dies hat allerdings den Nachteil, dass die geschnittenen Objekte nach hinten dargestellt werden und nicht nur die Schnittfläche. Es gibt auch keine Option, um dies zu verhindern. Die überflüssigen Linien können allerdings „verdeckt“ werden. Die Schubfundamente, der Boden,

sowie die seitliche Böschung wurden als zusätzliche Skizze manuell gezeichnet. Die dargestellte, schräge Böschung ist allerdings nicht die Böschung in Endlage, sondern wurde als Beispiel aus dem vorliegenden Ausführungsübersichtsplan, welcher als Grundlage dient, übernommen. Die eigentliche Böschung müsste horizontal über die ganze Breite gehen und ist in der Abbildung 4-5 grau angedeutet. Eine Böschungsschraffur gibt es nicht, deswegen wurde diese manuell gezeichnet. In AutoCAD werden dafür meist LISP-Routinen verwendet, allerdings können diese in Inventor nicht eingesetzt werden.

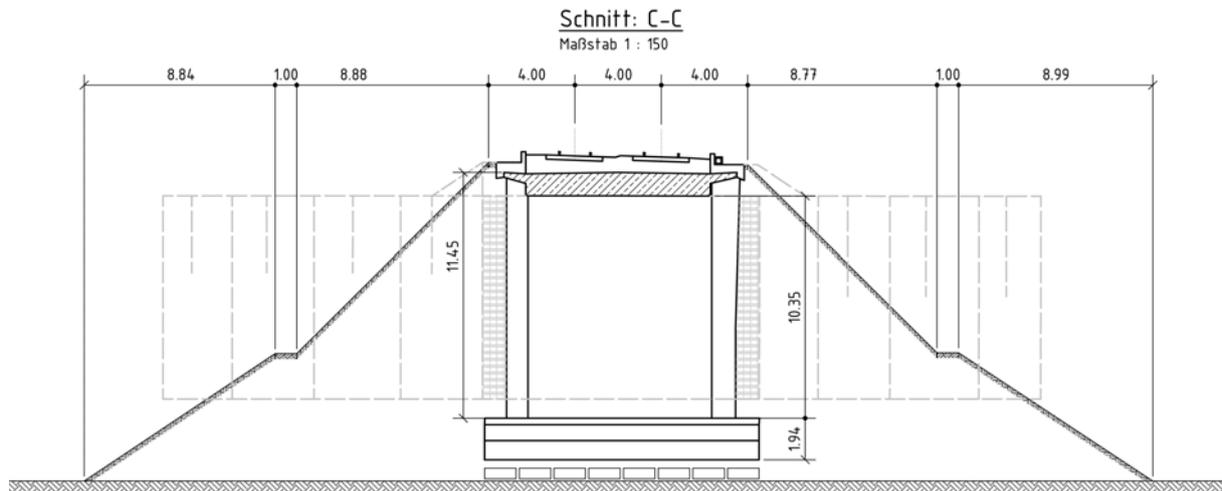


Abbildung 4-5: Querschnitt-Beispiel

4.4 Längsschnitt

Auch der Längsschnitt kommt nicht ohne zusätzliche Skizzen aus. Unter anderem wurden die gestrichelten Verlängerungen des Oberbaus manuell hinzugefügt, da der Oberbau selbst im Modell am Widerlager aufhört und durch eine Schiene schneiden keine Option ist.

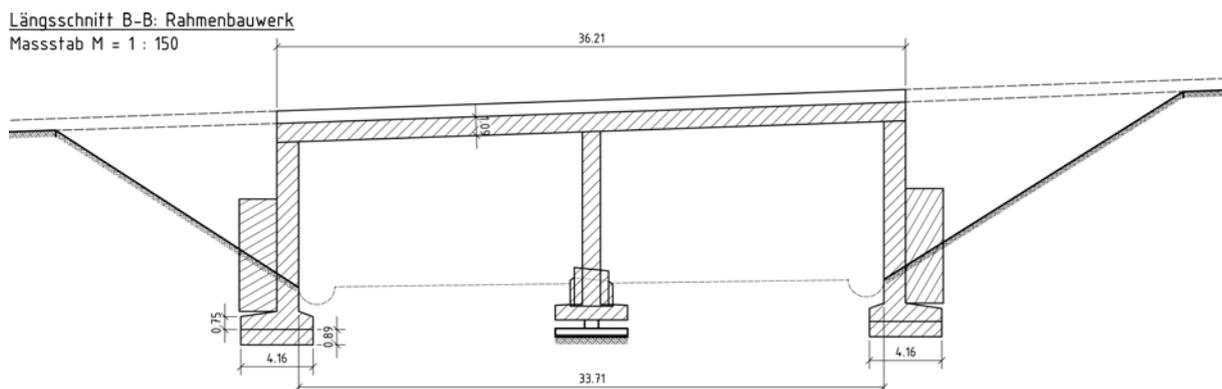


Abbildung 4-6: Längsschnitt-Beispiel

Allgemein wurde drauf geachtet, dass die manuell hinzugezeichneten Elemente durch entsprechende geometrische Bedingungen bei Veränderung verschiedener Parameter wie unter anderem der Höhe sich mitverändern. Sobald allerdings beispielsweise die Gradienten ändert und eine Krümmung aufweist, durch die der Längsschnitt approximiert werden müsste, ist die oben gezeigte Längsschnittansicht hinfällig. Dieses Beispiel ist also von der Zeichnung her leider nicht sehr flexibel und soll nur zeigen was theoretisch schon in einem gewissen Grenzbereich möglich ist.

5 Automatisierung

Im vorherigen Teil der Arbeit wurde auf Basis eines 3D Brückenmodells in Inventor ein Ausführungsübersichtsplan manuell abgeleitet. Ziel dieser manuellen Ableitung war es unter anderem herauszufinden, welche einzelne Schritte in Inventor von Nöten sind, um einen solchen Plan zu erhalten und ob eine gewisse Abfolge verschiedener Schritte eingehalten werden muss. Da für jedes andere Brückenmodell in Inventor und auch unabhängig davon ob es sich um einen Ausführungsübersichtsplan oder einen Schalplan handelt, sich einige dieser Schritte immer aufs neue wiederholen, ist es zielführend diese in einem (teil-)automatisierten Prozess zu bündeln, um in Zukunft unnötige Arbeit zu vermeiden.

Zur Planerstellung in Inventor wurde zuerst ein Konzept entwickelt, welches anschließend mit Hilfe der Inventor API und VB.NET in einem Modul und einem Konzeptprogramm umgesetzt wurde. Dabei ist das Konzept als Grundkonzept anzusehen, welches noch durchaus genügend Potential zur Erweiterung bietet. Diese Arbeit soll zeigen, dass ein solches Konzept Sinn macht und leicht umsetzbar ist.

5.1 Konzept

Grundlage für das nachfolgende Konzept ist ein zur Verfügung gestellter Ausführungsübersichtsplan, welcher schematisch dargestellt so aussieht:

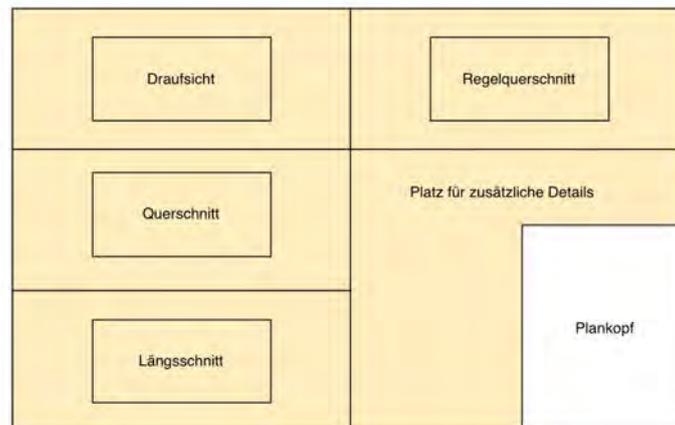


Abbildung 5-1: konzeptueller Ausführungsübersichtsplan

Dieser beispielhafte Ausführungsübersichtsplan besitzt eine Hauptansicht, die Draufsicht, und dazu jeweils drei unterschiedliche Schnitte, die sich auf die Draufsicht beziehen. Nebenbei befindet sich rechts unten im Eck der Plankopf, dessen Größe maximal die eines DIN A4 Blattes entspricht. Links und oberhalb dieses Plankopfes können noch zusätzliche Detailzeichnungen platziert werden. Dabei ist noch anzumerken, dass nicht jeder Ausführungsübersichtsplan gleich aussieht und die Anzahl der Ansichten und Schnitte oft variieren, allerdings hat dies auf das nachfolgende Konzept keine Auswirkungen.

Die in Abbildung 5-1 gezeigte schematische Darstellung, lässt sich theoretisch sehr vereinfachen in dem das Blatt grob in Quadranten eingeteilt wird. Dabei sind die Dimension

dieses Rechteckes, also Breite und Länge, im Grunde erst einmal irrelevant solange die Blattgrenzen eingehalten werden. Außerdem sollen sich die Rechtecke nicht überschneiden. Diese wenigen Randbedingungen ermöglichen eine Vielzahl an unterschiedlichen Layouts. Hier einige Beispiele:

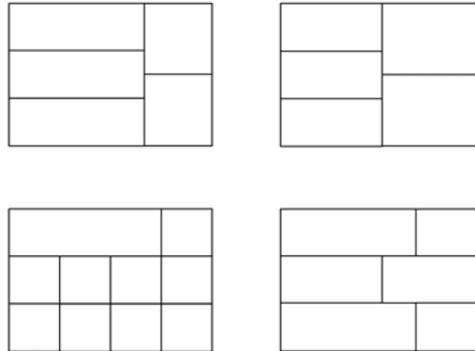


Abbildung 5-2: Verschiedene Layoutvarianten

Diese vier Beispiele veranschaulichen wie vielfältig die Möglichkeiten sind ein Layout mit nur Rechtecken zu gestalten. Durch die Form und Anordnungsart bietet sich zur Identifizierung jeder dieser Rechtecke eine zweidimensionale Matrix an. Des Weiteren könnte durch Rekursion mehrere Matrizen ineinander verschachtelt werden, was sicherlich auch einige Vorteile hätte. Jedoch erhöht sich dadurch die Komplexität der Umsetzung, weswegen dieses Konzept sich auf eine einzige Matrix für das gesamte Layout beschränkt.

Da der Ursprung eines Blattes in Inventor in der linken unteren Ecke liegt, wird das Zählen der Indizes für die Matrix auch in dieser Ecke begonnen. In den meisten Programmiersprachen besitzt das erste Element einer Matrix den Index 0, weswegen bei diesem Konzept der erste Rechteck den Index (0,0) erhält.

Um spätere Verwirrungen zu vermeiden und um gedanklich von der reinen geometrischen Form wegzukommen, wird folglich nicht mehr von Rechtecken gesprochen, sondern von sogenannten „Layoutboxen“. Jede Layoutbox enthält dabei unterschiedliche Informationen, die sich zum Beispiel in ihren Ansichtsarten unterscheiden. Dadurch lassen sich die Layoutboxen in verschiedene Kategorien unterteilen. Standardmäßig enthält der anfangs beschriebene Ausführungsübersichtsplan drei grundlegende Elemente: Eine Draufsicht, die als Basis zur Ableitung der Schnitte dient, mehrere Schnitte sowie einen Plankopf.

Folgende Layoutboxen-Kategorien sind denkbar:

- „**Basisansicht**“: Die Basisansicht ist eine nicht-assoziative Ansicht, die das Bauwerk von einer definierten Perspektive aus zeigt (z.B. Draufsicht).
- „**Schnittansicht**“: Schnittansichten beziehen sich immer auf eine referenzierte Ansicht in der die Definition des geradlinigen Schnittes erfolgte.
- „**Reserviert**“: Da nicht nur der Plankopf eine Fläche darstellt, die freigehalten werden muss, sondern auch anderswo eine solche freie Fläche für Details oder Zusatzinformationen vorbehalten werden kann, werden diese Flächen allesamt als *reserviert* bezeichnet.

- „**Kombiniert**“: Der Zweck dieser besonderen Kategorie ist es mehrere Layoutboxen zu einer Layoutbox auf eine einfache Art und Weise zusammenzufassen. Dies soll die Möglichkeiten der Gestaltung erweitern.

Die beiden Layoutbox-Kategorien „Basisansicht“ und „Schnittansicht“ müssen alle nötigen Informationen zur Erstellung dieser Ansichten in Inventor beinhalten. Im späteren Kapitel 5.2 zur Umsetzung des Konzeptes werden die benötigten Informationen im Detail vorgestellt, weswegen an dieser Stelle nicht weiter darauf eingegangen wird.

Bei der oben gezeigten schematischen Darstellung eines Planes fehlt der Rahmen. Die eigene, firmenspezifische Zeichnungsvorlage, die vorher erstellt wurde und später Verwendung findet, beinhaltet sowohl Plankopf als auch einen Rahmen. Dabei stellen die inneren Grenzen des Rahmens die Grenzen für das Layout dar.

5.1.1 Konzeptprogramm

Die Anwendung, die im Rahmen der Umsetzung geschrieben wurde, bietet zur Interaktion mit dem Anwender eine graphische Benutzeroberfläche (GUI, engl. „Graphical User Interface“) an, zu der im Folgenden kurz erläutert wird, auf welchen Ideen diese beruht. Im späteren Kapitel wird speziell auf die Umsetzung mit VB.NET eingegangen.

Die Grundidee war es möglichst einfach und mit wenigen Schritten einen ersten Entwurf für einen Ausführungsübersichtsplan zu generieren. Im Grunde sind dafür nur drei Schritte seitens des Anwenders von Nöten:



Abbildung 5-3: Diagramm der drei Schritte, die der Anwender durchgehen muss

Dabei soll entweder eine Inventor-Datei geladen werden, die das Bauwerk beinhaltet, oder es soll die Möglichkeit geben ein aktives Dokument in Inventor zu benutzen. Die Wahl des Maßstabes könnte später durch einen Automatisierungsprozess ersetzt werden, wird aber in dieser Anwendung vorerst noch vom Anwender selbst ausgewählt.

Des Weiteren soll der Anwender auch andere Layouts erstellen und benutzen können, weswegen die Anwendung in zwei Fenster, einem Hauptfenster und einem Layoutfenster, unterteilt wurde. Im Layoutfenster sind alle Möglichkeiten zur Gestaltung eines Layouts gruppiert, die implementiert wurden. Folgendes bietet dieses Layoutfenster an:

- Eine Darstellung des Layouts.
- Das Erstellen eines Layouts mit x Reihen und y Spalten.
- Die Eingabe aller relevanten Information zur Erstellung der Ansichten.
- Änderung der Breite einer Spalte.

An dieser Stelle soll noch einmal erwähnt werden, dass dies nur eine Konzeptanwendung ist, die ein paar Möglichkeiten aufweisen soll und alles andere als vollständig ist. Deswegen wurde unter anderem nur die Änderungen der Breite einer Spalte und nicht die Änderung der Höhe einer Reihe implementiert, da diese sich im Wesentlichen nicht wirklich unterscheiden.

5.2 Umsetzung des Konzeptes

Die Umsetzung des Konzeptes wurde mittels der Programmiersprache VB.NET und der Inventor API durchgeführt. Folglich wird das Thema der Erweiterung von Inventor etwas allgemeiner angesprochen und einige Aspekte der Programmierung von Erweiterungsanwendungen erläutert, bevor auf die eigentliche Arbeit eingegangen wird. Dabei wird zuerst das *InventorLayout*-Modul im Detail beschrieben und anschließend die Einbindung in das Konzeptprogramm.

5.2.1 Programmiertechnische Anbindung an Inventor

Um die Funktionalität von Inventor zu erweitern, bietet Autodesk die Inventor API an, eine Programmierschnittstelle (engl. *Application Programming Interface, API*) die auf der von Microsoft entwickelten COM (*Component Object Model*) Technologie basiert.¹¹ Diese Technik ermöglicht es dynamische Objekte zu erzeugen, die unter anderem sprachunabhängig sind, weswegen auch mit unterschiedlichen Programmiersprachen auf die Inventor API zugegriffen werden kann.

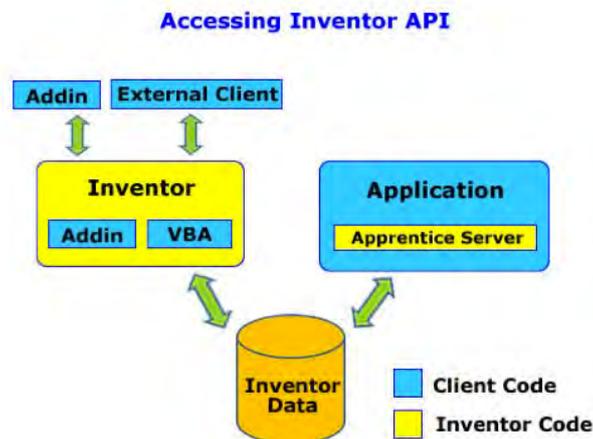


Abbildung 5-4: Diagramm: Zugriff auf die Inventor API (Autodesk, o.J.)

Neben den verschiedenen Programmiersprachen gibt es folgende vier Möglichkeiten um mit Inventor und den Inventor Daten zu interagieren:

- Mit Hilfe von VBA in Inventor selbst: Es gibt einen VBA Editor und es können einzelne Makros erstellt und über ein Auswahlfenster ausgelöst werden.
- Sogenannte Addins, also Erweiterungen, können für Inventor erstellt werden. Diese werden beim öffnen von Inventor geladen und in die Benutzeroberfläche von Inventor eingebettet. Dabei gibt Autodesk Richtlinien vor, wie diese Erweiterungen aussehen sollen.¹²
- Neben den Addins gibt es aber auch die Möglichkeit externe Anwendungen zu schreiben, die über die API auf Inventor zugreifen können.
- Bei den 3 vorherigen Methoden muss immer eine Instanz von Inventor geöffnet sein. Deswegen gibt es noch den Apprentice Server, der ohne Inventor, direkten Zugriff auf gewisse Inventor-Daten ermöglicht. Der Apprentice Server ist allerdings sehr

¹¹ http://de.wikipedia.org/wiki/Component_Object_Model

¹² Third-Party Design Guidelines.doc – wird mit dem SDK mitgeliefert und ist nicht online verfügbar

eingeschränkt, hauptsächlich nur leseberechtigt und nicht alle Informationen sind abrufbar. Er bietet sich vor allem bei der Bearbeitung von iProperties an.

Um in einer VB.NET-Anwendung auf die Inventor API zugreifen zu können, muss mit Hilfe der *Imports*-Anweisung auf den „Inventor“ Namespace verwiesen werden und das COM-Objekt selbst muss ebenfalls als Verweis in der IDE hinzugefügt werden. (Dadurch übernimmt die IDE einige Schritte, die sonst manuell vollzogen werden müssten.¹³)

5.2.2 Aspekte der Erweiterungsprogrammierung

Über die Inventor API kann direkt auf die Inventor Objekte und deren Funktionen zugegriffen werden. Deswegen wird diese Art und Weise im Folgenden als der direkte Weg bezeichnet. Dieser eignet sich vor allem bei kleinen Zusatzanwendungen oder bei Makros mit geringer Komplexität.

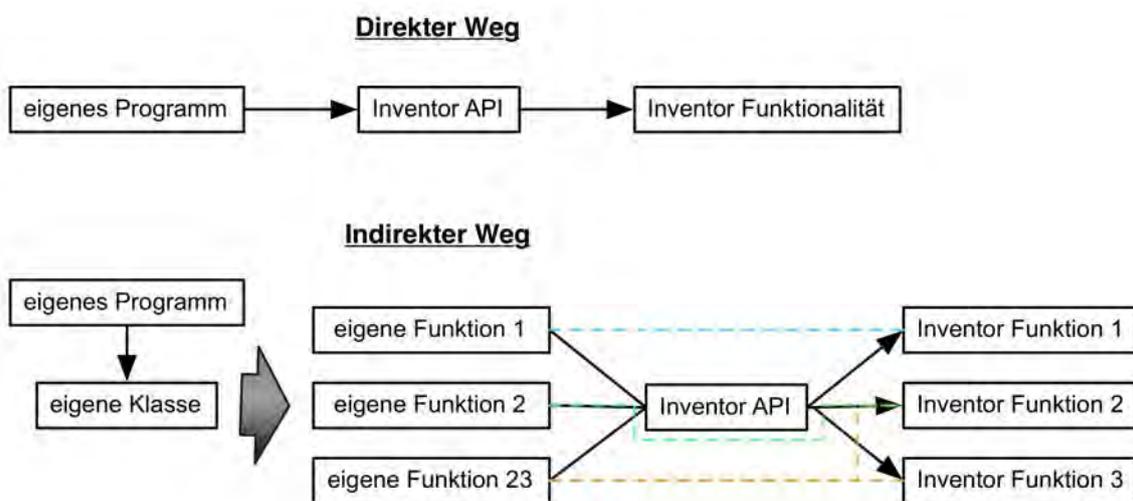


Abbildung 5-5: Diagramm: Direkter/Indirekter Weg

Neben dem direkten gibt es auch noch den indirekten Weg über eine eigene Klasse. Sofern nur der indirekte Weg benutzt werden soll, also von der Anwendung nie direkt auf die Inventor API zugegriffen werden soll, sondern immer nur über Funktionen aus der eigenen Klasse, scheint der indirekte Weg auf den ersten Blick einiges an überflüssiger Arbeit zu beinhalten. Für jede *Inventor Funktion x* müsste eine entsprechende *eigene Funktion x* in der Klasse definiert werden. Der große Vorteil des indirekten Weges ist allerdings, dass Inventor Funktionen (z.B. die Funktionen 2 und 3) in einer eigenen Funktion gebündelt werden können. Anders ausgedrückt ist es möglich ein Prozess, welcher sich aus einer Abfolge von mehreren Inventor Funktionen zusammensetzt, als eine Funktion über die Klasse anzubieten.

Das Konzept der objektorientierten Programmierung sieht vor, dass Funktionen und Eigenschaften in einem Objekt zusammengefasst werden. So lässt sich nie direkt auf eine Funktion zugreifen sondern nur über das Objekt. Jedes Objekt stellt eine Instanz einer Klasse dar, die im eigentlichen Sinne beschreibt, welche Eigenschaften ein Objekt hat und welche Funktionen das Objekt anbietet. Darüber hinaus existieren in VB.NET sogenannte Schnittstellen (engl. *Interfaces*), die unter anderem in der COM Technologie Anwendung

¹³ <http://msdn.microsoft.com/de-de/library/vstudio/cwys3b23.aspx>

finden. Diese stellen auch eine Art Form dar, jedoch im Vergleich zu Klassen ohne Inhalt, weswegen sie sich von Klassen in einigen Punkten unterscheiden. An dieser Stelle soll nur der in diesem Fall relevanteste Punkt Erwähnung finden: Es können keine Objekte von einer Schnittstelle sondern nur von einer Klasse erstellt werden.

Die Inventor API stellt im Prinzip nicht anders als ein einziges Objekt zur Verfügung, welches eine wüste Hierarchie mit einer Vielzahl von Unterobjekten beinhaltet. Dabei besitzt jedes Unterobjekt unterschiedliche Eigenschaften und Funktionen. Dieses Objektmodell wird von Autodesk zur Übersicht auch als PDF Datei angeboten.¹⁴

Dies hat zur Folge, dass es nicht möglich ist eins dieser Unterobjekte einzeln zu instanzieren. Dazu am besten ein Beispiel:

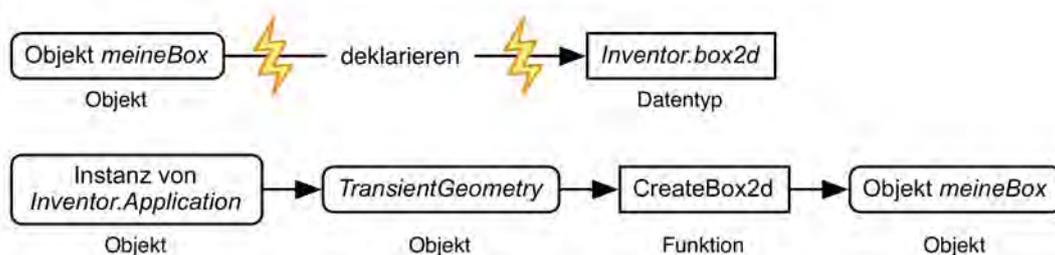


Abbildung 5-6: Diagramm: Beispiel zur Instanzierung von Objekten aus der Inventor API

Inventor benutzt intern öfters eine sogenannte RangeBox als Eigenschaft von einem Objekt. Diese RangeBox definiert ein Rechteck in einem gewissen Koordinatensystem anhand eines minimalen und eines maximalen Punktes und ist vom Datentyp *Inventor.box2d*. Solch eine Box könnte z.B. auch in Kombination mit einer GDI+ Grafik nützlich sein. *Inventor.box2d* stellt allerdings nur eine Schnittstelle dar und keine Klasse, so dass es nicht möglich ist eine Variablendeklaration mit diesem Datentyp durchzuführen. Deswegen muss zuerst eine Instanz vom Typ *Inventor.Application* erstellt werden und anschließend lässt sich über die Funktion *CreateBox2d* des Unterobjektes *TransientGeometry* ein Objekt vom Typ *box2d* erzeugen.

Durch diesen Umstand muss der eigenen Klasse eine Instanz vom Typ *Inventor.Application* übergeben werden. Da im Vorfeld nicht bekannt ist, ob Inventor schon geöffnet ist, oder ob Inventor erst durch das Zusatzprogramm geöffnet werden soll, sollte diese Instanzierung und die auch damit verbundene Fehlerbehandlung am Besten außerhalb der Klasse stattfinden und Teil des Zusatzprogramms sein.

Eine Instanz des Apprentice Servers wäre zwar innerhalb einer Klasse recht simpel, stellt dabei allerdings durch die hohen Einschränkungen keine Alternative dar.

Das Gruppieren von Inventor Funktionen und damit verbundene Abläufe in eigene Funktionen resultiert in einer langen Parameterliste für die Inventor Funktionen. Die Herkunft der Parameter und deren Weitergabe beziehungsweise die Frage "Wie gelangen die Parameter zu der Funktion, die diese als Eingabe verlangt?" spielt dabei eine wichtige Rolle.

¹⁴ <http://images.autodesk.com/adsk/files/inventor2013model.pdf>

Es gibt eigentlich zwei verschiedene Methoden:

- 1) Die Parameterliste kann als Parametereingabe für die eigene Funktion dienen.
- 2) Die Parameter werden als eigene Eigenschaft des Objektes abgelegt.

Die erste Methode hat den Nachteil, dass Parameterlisten sehr schnell sehr groß werden können. Dies kann zur Unübersichtlichkeit führen, Fehler passieren schneller und die Parameterübergabe muss bei jedem Funktionsaufruf stattfinden.

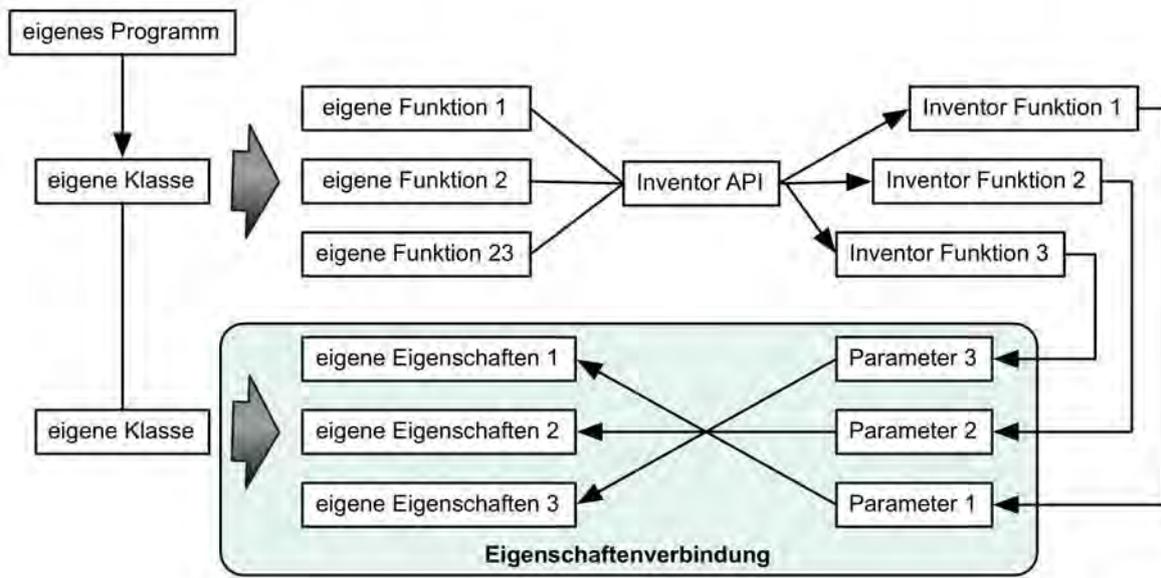


Abbildung 5-7: Diagramm: Eigenschaftsverbindung

Bei der zweiten Methode werden Parameter mit Eigenschaften verknüpft (siehe Abbildung 5-7) und müssen nur einmal gesetzt werden. Anschließend kann jede Funktion der Klasse darauf zugreifen. Der Parameterzugriff passiert dabei in der Funktion selbst und der Benutzer der Klasse muss sich nicht darum kümmern. Allerdings muss dieser wissen welche Eigenschaften von welcher Funktion benutzt werden, damit er sicher gehen kann, dass diese vor dem Aufruf der Funktion gesetzt sind.

Der einzige wirklich Nachteil ist folgender: Je vielfältiger die Klasse, desto größer wird natürlich auch die Anzahl der Eigenschaften und für jede Eigenschaft wird ein gewisser Speicherraum freigehalten. Sobald das Speichermanagement eine wichtige Rolle einnimmt, muss darüber nachgedacht werden, wie so eine Klasse sinnvoll aufgeteilt werden könnte.

Diese drei genannten Aspekte sind essentiell wichtig für die Modulentwicklung.

5.2.3 Konzept der drei Ebenen

Die Konzeptanwendung kann gedanklich in drei verschiedene Ebenen unterteilt werden.

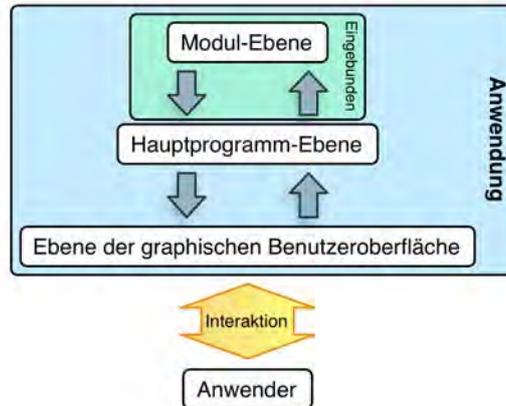


Abbildung 5-8: Diagramm: Anwendungsaufbau in 3 Ebenen gegliedert

Als vierte, virtuelle Ebene steht der Anwender selbst. Die Modul-Ebene, obschon sie in Abbildung 5-8 als Bestandteil der Anwendung dargestellt wird, gehört eigentlich nicht direkt zur Anwendung, sondern existiert extern und wird nur eingebunden, deswegen auch die zusätzliche hellgrüne Markierung. Dies ist allerdings nur zweitrangig. Viel wichtiger ist, dass jeder dieser Ebenen direkt mit der darüber liegenden oder der darunterliegenden Ebene interagiert und keine Ebene übersprungen wird. Dazu kurz ein Beispiel:

Drückt der Benutzer den „Plan erstellen“-Button, interagiert er direkt mit der GUI. Das Drücken des Buttons löst dabei auf Programm-Ebene die Prozedur *ButtonCreateInventorLayout_Click*, die unter anderem die Memberfunktion *DrawLayoutInventor* des Layout-Objektes aufruft, woraufhin das gesamte Layout erstellt wird. Dabei findet der letzte Schritt auf Modul-Ebene statt. So wurden alle Ebenen durchlaufen.

Die Hauptprogramm-Ebene kann noch weiter unterteilt werden, da sie im Prinzip zwei Aufgaben hat. Zum einen muss sie erstens den Informationsfluss und die Steuerung der graphischen Benutzeroberfläche übernehmen. Das Auslesen des Maßstabes wäre ein Beispiel dafür. Zum anderen besteht ihre Aufgabe in der Verarbeitung der Informationen und der Benutzung des Layout-Objektes. Das heißt schlicht, dass es nicht ausreichend ist das *InventorLayout*-Modul einzubinden und ein Layout-Objekt zu erstellen, sondern die Prozeduren des Objektes müssen auch auf Programmebene aufgerufen werden.

Dies scheint auf den ersten Blick alles recht trivial, spielt allerdings bei der Programmierung eine wichtige Rolle. Damit das Modul weitestgehend unabhängig von der Anwendung bleibt, muss darauf geachtet werden, welche Bestandteile auf welche Ebene, nämlich Modul-Ebene oder Hauptprogramm-Ebene, gehören. Des Weiteren ist es vorteilhaft für die Übersicht im Quellcode die Interaktion mit der GUI-Ebene vom Rest zu trennen.

5.2.4 Modul *InventorLayout*

Das *InventorLayout*-Modul beinhaltet vier verschiedene Klassen und drei Enumeratoren. Davon sind zwei Klasse kleine Nebenklassen, die mit den Enumeratoren im Folgenden zuerst beschrieben werden.

Um Verwechslung zu vermeiden werden Klassennamen, Funktionen und Objektnamen immer kursiv geschrieben.

5.2.4.1 Enumeratoren

In einer Enumeration sind Member definiert, denen jeweils ein Wert zugeschrieben wird. Die Member an sich besitzen eine logische Zusammengehörigkeit und die Enumeration ist deshalb wie eine Gruppe. Ziel von Enumerationen ist es Werten aussagekräftige Namen zu geben und die Lesbarkeit sowie die Verständlichkeit im Quellcode zu erhöhen. Die Tabelle 5-1 listet die im *InventorLayout*-Modul definierten Enumeratoren auf. Die Werte an sich sind irrelevant, da nie mit den Werten gearbeitet wird sondern immer nur mit den Member. Deshalb wurde als kleinster Wert einfach 1 gewählt (Nicht Null, um später bei der Überprüfung, ob eine Variable überhaupt gesetzt wurde, keine Fehler zu erhalten.) und die Werte steigern sich sukzessiv um den Wert 1.

<u>LayoutBoxLabelPositionEnum</u>	<u>LayoutBoxTypeEnum</u>	<u>LayoutBoxSectionTypeEnum</u>
kTop	kReserved	kHorizontal
kBottom	kBase	kVertical
kLeft	kSection	kDifferent
kRight	kCombined	
kTopLeft		
kTopRight		
kBottomLeft		
kBottomRight		

Tabelle 5-1: Enumerationen aus dem *InventorLayout*-Modul

Die Namenskonvention wurde aus der *Inventor* API übernommen. So wird jede Enumeration durch den Zusatz „Enum“ hinten im Namen als solche kenntlich gemacht. Jedes Member einer Enumeration beginnt mit dem Buchstaben „k“. Der Zweck der drei Enumerationen ist durch deren Namen eigentlich schon fast selbsterklärend.

- *LayoutBoxLabelPositionEnum*: Definiert die Stelle an der die Ansichtsbezeichnung innerhalb der Layoutbox hinkommen soll.
- *LayoutBoxTypeEnum*: Definiert die vier verschiedenen Typen einer LayoutBox.
- *LayoutBoxSectionTypeEnum*: Speziell für Schnitte werden wiederum drei unterschiedliche Typen definiert.

5.2.4.2 Klasse „*templateDocument*“

Die Layoutgrenzen werden jeweils durch den Zeichenbereich eines Blattes definiert. Der Zeichenbereich selbst kann durch die Blattgrenzen oder einen Rahmen festgelegt werden. Des Weiteren sind die Dimensionen des Schriftfeldes wichtig, da in dem Bereich nicht gezeichnet werden darf. Obschon beim Erstellen des Planes eine Vorlage geladen werden muss, müssen die genannten Informationen schon beim Erstellen des Layouts vorhanden sein. In *Inventor*

gibt es die Möglichkeit, Dokumente nicht sichtbar zu öffnen, was das Auslesen der Informationen im Hintergrund ermöglicht. Allerdings sollte das Dokument nicht geöffnet bleiben, da vielleicht nur an einem Layout gearbeitet, jedoch kein Plan erstellt wird. Deswegen bietet die *templateDocument*-Klasse die Möglichkeit zur Erstellung eines Vorlagenobjektes an, welches die nötigen Informationen speichern kann.

5.2.4.3 Klasse „LayoutID“

Da jede Layoutbox durch Spalten- und Zeilenangabe eindeutig identifizierbar ist, macht es Sinn diese Angaben in einem ID-Objekt zu bündeln. Genau solch ein Objekt wird in der LayoutID-Klasse definiert. Des Weiteren wird eine Memberfunktion namens „identical“ bereitgestellt, um zwei IDs miteinander vergleichen zu können.

5.2.4.4 Überblick: Hauptklassen zum Erstellen eines Layouts

Das grundlegende Konzept, welches ausführlich im vorherigen Kapitel beschrieben wurde, ist die Basis für zwei Hauptklassen. Das gesamte Layout als ein Objekt wird von der Klasse *Layout* abgeleitet und besitzt als Membervariable eine Matrix zur Speicherung der einzelnen Layoutboxen.

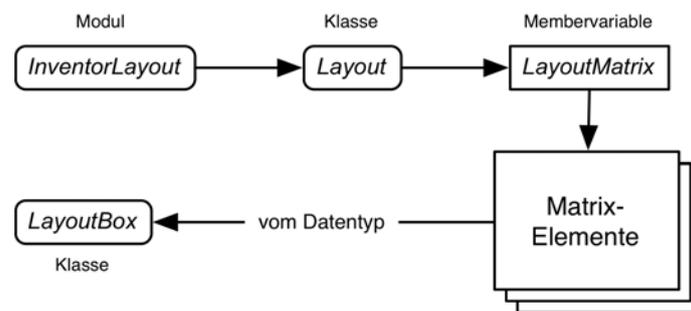


Abbildung 5-9: Diagramm: Zusammenhang der Klassen *Layout* und *LayoutBox*

Jede Layoutbox ist wiederum ein Objekt für sich und von der Klasse *LayoutBox* abgeleitet. Die *Layout*-Klasse kann deshalb auch als die übergeordnete Klasse bezeichnet werden.

Es wäre unsinnig alle Methoden der beiden Klasse schrittweise zu erläutern, da dies einer Dokumentation des Quellcodes gleich kommen würde. Aus diesem Grund werden im Folgenden nur der Aufbau und die Ideen zu den wichtigsten Prozeduren erläutert und relevante Details näher erklärt. Im Anhang sind Tabellen vorhanden, die zur Übersicht alle Variablen und Prozeduren der beiden Klassen auflisten. Der gesamte Quellcode des Modules und der Konzeptanwendung sind auf der CD.

5.2.4.5 Initialisierung

Wie im vorherigen Kapitel 5.2.2 „Aspekte der Erweiterungsprogrammierung“ erläutert, wird in jeder Hauptklasse eine Referenz auf eine Instanz vom Typ *Inventor.Application* benötigt, um z.B. *Point2d*-Punkte erstellen zu können. Das Setzen dieser Referenz wird als Initialisierungsprozess bezeichnet, da dieser Schritt zwingend notwendig ist, damit das Modul richtig arbeiten kann und somit auch gleich nach Erstellung des Objektes stattfinden soll. Die entsprechende Prozedur für diesen Prozess trägt den Namen *Initialisation* und existiert in beiden Hauptklassen. Der Aufruf der Initialisierung des *Layout*-Objektes passiert dabei auf

Hauptprogramm-Ebene, wohingegen die Initialisierung der einzelnen *LayoutBox*-Objekte bei deren Deklaration innerhalb der entsprechenden Prozedur in dem Layout-Objekt stattfindet. Der Name der Referenzvariable lautet „*iApp*“.

5.2.4.6 Methode „CreateLayout“

Wird eine neue Instanz der Klasse *Layout* erstellt, kann dem Konstruktor eine *RangeBox* übergeben werden, die den Zeichenbereich definiert und damit die Abmessungen des Layouts festlegt. Anschließend wird das Objekt wie oben beschrieben initialisiert. Das Layout an sich besitzt zu diesem Zeitpunkt also nur einen Rahmen, ist ansonsten aber noch leer.

Die Prozedur *CreateLayout* der *Layout*-Klasse erstellt als nächsten Schritt die Grundstruktur des Layouts. Dabei werden als Parameter die Anzahl der Reihen und Spalten übergeben.

Zuerst wird die Breite und Höhe einer Layoutbox berechnet, falls diese Werte kleiner als die Minimalwerte sind, wird anhand der Blattabmessungen und den Minimalwerten die maximale Anzahl an Spalten bzw. Reihen berechnet. Da es keine halben Layoutboxen gibt, wird auf einen ganzzahligen Wert abgerundet und der Rest wird der letzten Reihe bzw. der letzten Spalte beigefügt.

Anschließend wird eine Rangebox mit den zuvor berechneten Werten als *box2d*-Objekt mit Hilfe der Funktion *CreateBox2d()* des *TransientGeometry*-Objektes aus der Inventor API erstellt. Diese RangeBox wird der neuen Instanz der *LayoutBox*-Klasse übergeben. Bevor dieses Objekt in die *LayoutMatrix* eingefügt wird, wird es noch initialisiert und bekommt eine ID.

5.2.4.7 Inhalt eines LayoutBox-Objektes

Nachdem das Grundlayout und für jede Layoutbox ein Objekt erstellt wurde, ist der nächste logische Schritt den Inhalt jeder Layoutbox zu definieren.

Zuerst wird dabei der Typ der Layoutbox festgelegt. Die 4 verschiedenen Typen wurden im Kapitel über das Konzept schon einmal näher beschrieben. Dabei ist der Typus der kombinierten Layoutbox in der momentanen Version des Moduls noch nicht weiter implementiert und stellt nur eine Idee dar. Deshalb sind die 3 relevanten Typen „kBase“, „kSection“ und „kReserved“. Dabei müsste „kReserved“ als Typ eigentlich nie vom Benutzer selbst gesetzt werden. Falls eine freie Layoutbox gewünscht wird, kann die *Type*-Variable einfach leer bleiben. Der Zweck dieses Typs ist die Unterscheidung zwischen einer wirklich freien Layoutbox, in der zum Beispiel eine Ansicht platziert werden kann oder einer reservierten Layoutbox. Deswegen wird empfohlen diesen Typ trotzdem zu setzen, wenn eine Layoutbox für einen bestimmten Zweck leer bleiben soll. Die Prozedur *checkReservedSpace()* überprüft ob eine Layoutbox mit der reservierten Fläche für den Plankopf in Berührung kommt und falls dies der Fall ist, wird die *Type*-Variable automatisch auf „kReserved“ gesetzt.

Neben der *Type*-Variable sind viele weitere Variablen definiert, die an dieser Stelle aber nicht extra aufgelistet werden. Wichtig ist, dass es Kontrollvariablen gibt, die im Modul zur Steuerung gewisser Funktionen benutzt werden, sowie Referenzvariablen, die jeweils eine Referenz zu einem Inventor-Objekt speichern. Des Weiteren existieren Variablen, die gewisse

Inventor-Eigenschaften wie z.B. den Ansichtsstil oder die Ansichtsausrichtung speichern. Als modul-spezifische Variable könnte unter anderem die *labelPosition*-Variable bezeichnet werden, die die Position des Bezeichnungstextes der Ansicht festlegt.

Für viele Variablen, die vom Benutzer gesetzt werden können, sind auch Standardwerte vorgesehen. Ein Beispiel wäre die *sectionAngle*-Variable, die den Schnittwinkel speichert und standardmäßig den Wert 0 besitzt. Andere Variablen wie die *sectionBaseViewLayoutBoxID*-Variable, die bei einer Schnittansicht die ID der Basisansicht enthält, müssen unbedingt gesetzt werden.

5.2.4.8 Integritätsüberprüfung

Damit das gesamte Layout fehlerfrei ist, ist es wichtig, dass die benötigten Variablen gesetzt und bei anderen Variablen gewisse Werte eingehalten werden. Zur Überprüfung der Integrität eines Layouts gibt es die entsprechende Prozedur *CheckIntegrity()* in der *Layout*-Klasse. In der jetzigen Implementierung werden drei Bedingungen überprüft:

1. Für jede Basisansicht muss eine Referenz auf das abzubildende Modell in Inventor vorhanden sein.
2. Für jede Schnittansicht muss die *sectionBaseViewLayoutBoxID*-Variable gesetzt werden und diese ID muss einer Layoutbox gehören, welche eine Basisansicht oder eine Schnittansicht enthält.
3. Der Skalierungsfaktor jeder Layoutbox, die eine Ansicht enthält, darf nicht 0 sein.

Die Integrität wird erst kurz vor dem Erstellen des Layouts in Inventor überprüft, da die Referenz zum Inventor-Modell auch erst kurz davor gesetzt wird.

5.2.4.9 Darstellung des Layouts

Um dem Benutzer vor dem Erstellen des Planes eine Möglichkeit zu geben, das Layout in irgendeiner Form visuell zu betrachten, wurde eine Methode implementiert, die das Layout als 2D Grafik mit Hilfe von Microsofts Grafikbibliothek GDI+ darstellen kann. Diese Grafik kann, muss allerdings nicht, auch dazu verwendet werden, um mit dem Anwender zu interagieren, in dem er z.B. auf eine Layoutbox klickt und anschließend in einem Eingabefenster die Informationen dieser Layoutbox editieren kann.



Abbildung 5-10: Ursprünge und Achsenrichtungen

Der *DrawLayoutGraphics*-Methode des *Layout*-Objektes muss zur Darstellung lediglich ein entsprechendes *System.Drawing.Drawing2D.Graphics*-Objekt als Referenzparameter übergeben werden. Dabei ist allerdings zu beachten, dass bei einem solchen Grafikobjekt der Ursprung in der linken, oberen Ecke ist und die Achsen ihre positive Richtung nach unten

bzw. nach rechts haben, wohingegen der Ursprung des Koordinatensystems eines Blattes in Inventor sich in der linken unteren Ecken befindet, die positive Y-Achse nach oben und die positive X-Achse nach rechts zeigen. (siehe Abbildung 5-10)

Da im *Layout*-Objekt die Abmessungen des Blattes nicht gespeichert werden, muss auf Hauptprogramm-Ebene der Ursprung in Y-Richtung um den Wert der Höhe des Blattes verschoben und die Y-Achse durch eine Skalierung von -1 umgedreht werden. Es wäre aber sinnvoll diesen Prozess bei der Verbesserung des Moduls in das Modul selbst zu integrieren.

Allgemein wurde versucht die Höhen und die Breiten der Elemente in mm als Pixel zu übernehmen. Durch den Umstand, dass die Abmessungen der Layoutboxen durch die Divisionsrechnung oft keine ganzzahligen Millimeterwerte besitzen, wird bei der Umrechnung einfach abgerundet. Durch das Abrunden fehlen allerdings oft ein paar Pixel am oberen und am rechten Rand, die besonders bei dünnen Rahmen auffallen oder die Ränder der Layoutboxen gehen nicht bis zum Rand des Blattes, wenn kein Rahmen vorhanden ist.

Dieses Problem wurde gelöst indem diese fehlenden Pixel in beide Richtungen berechnet und anschließend über die Reihen und die Spalten jeweils verteilt wurden. Diese Aufteilung wurde in jeweils einer 2-dimensionalen Matrix pro Richtung gespeichert.

1	1	1	1
1	1	1	1
2	2	2	2

Abbildung 5-11: Beispiel: 4 fehlende Pixel in y-Richtung verteilen

Zum Zeichnen der Layoutboxen wird die *LayoutMatrix* über eine doppelte Iteration durchlaufen. Dabei wird zuerst ein *box2d*-Objekt mit den Abmessungen der Layoutbox erstellt, wobei die fehlenden Pixel einfach hinzugerechnet werden. Dieses Objekt wird als *DrawingRangeBox*-Variable in der *LayoutBox*-Instanz gespeichert. Anschließend wird die *DrawLayoutBoxGraphics*-Methode des *LayoutBox*-Objektes aufgerufen. In der jetzigen Version werden neben der ID auch der Layoutbox-Typ und der Schnitttyp in der Layoutbox angezeigt. Dabei passt der Text seine Größe automatisch an die Breite der Layoutbox an.

5.2.4.10 Layout in Inventor generieren

Nachdem das Layout ausreichend definiert und der Inhalt der Layoutboxen festgelegt wurde, soll das Layout in einer Inventor-Zeichnung erstellt werden. Dies passiert durch den Aufruf der Methode *DrawLayoutInventor* des *Layout*-Objektes, der als Parameter lediglich eine Referenz zu einem Blatt einer Zeichnung in Inventor übergeben werden muss. Ob dieses Blatt einer schon geöffneten Zeichnung angehört oder ob zuerst eine neue Zeichnung anhand einer Vorlage erstellt wird, wird unabhängig vom Modul und dem *Layout*-Objekt geregelt, sprich auf Anwendungsebene.

Zuerst müssen die Hauptansichten (Basisansichten) erstellt werden, da diese als Basis für die Schnittansichten dienen. Durch doppelte Iteration über die *LayoutMatrix* wird bei jedem *LayoutBox*-Objekt, die vom Typ „kBase“ ist, die *DrawLayoutBoxInventor*-Methode

aufgerufen. Anschließend wird die *LayoutMatrix* noch einmal durchlaufen und für jedes *LayoutBox*-Objekt vom Typ „kSection“ zuerst eine Referenz zur Basisansicht in der *sectionBaseView*-Variable abgelegt, bevor auch für die Schnittansichten die entsprechende Methode zum Erstellen aufgerufen wird. In der folgenden Abbildung wird der Aufbau der Methode anhand eines Diagramms veranschaulicht:

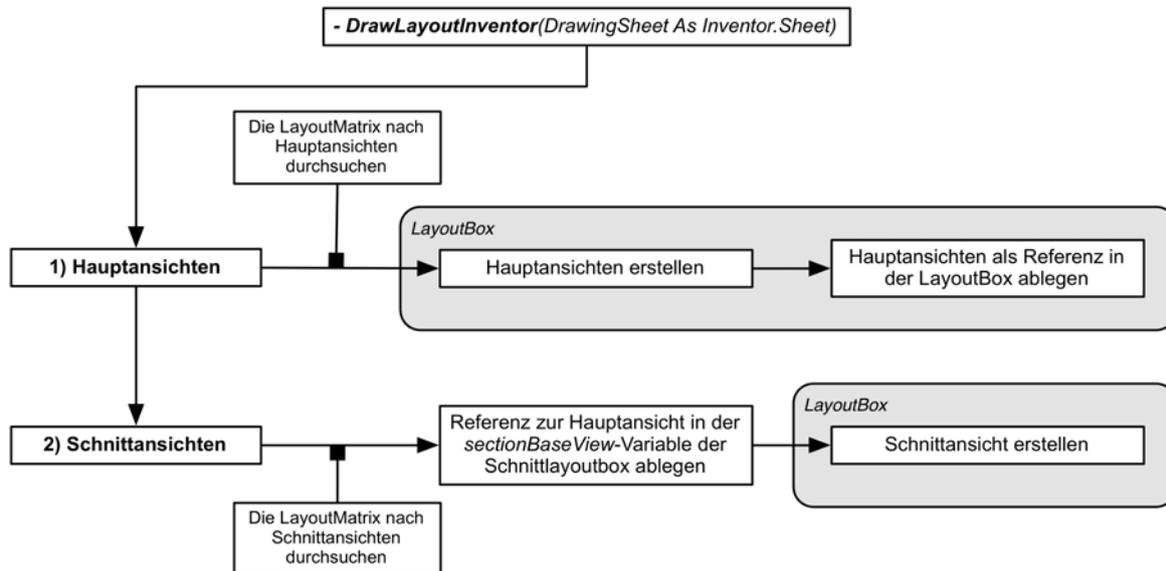


Abbildung 5-12: Diagramm: Überblick zur Vorgehensweise der Methode *DrawLayoutInventor*

Diese Vorgehensweise hat allerdings einen entscheidenden Nachteil: Als Basis für Schnittansichten können keine anderen Schnittansichten verwendet werden. Sinnvoll wäre es die *DrawLayoutInventor*-Methode umzuschreiben und die Reihenfolge des Erstellens der Ansichten beispielsweise durch eine Prioritätenmatrix, die zuerst erstellt werden muss, zu steuern.

Die *DrawLayoutBoxInventor*-Prozedur der *LayoutBox*-Klasse erstellt die Ansicht in Inventor, speichert diese als Referenz in der entsprechenden Variable, verändert die Ansicht je nachdem welche Eigenschaften gesetzt wurden und platziert den Beschriftungstext. Bei der Änderung der Richtung, der Vollschnittoption und der Schnitttiefe einer Schnittansicht wird jeweils 0,2 Sekunden gewartet, bevor die nächste Operation durchgeführt wird. Befindet sich die Schnittlinie nämlich außerhalb eines Bauteils, stürzt Inventor ab, wenn diese Operationen sofort nacheinander durchgeführt werden.

Für das Zuschneiden existiert in der Inventor API von 2013 noch keine entsprechende Methode. Deshalb muss über ein *ControlDefinition*-Objekt direkt auf die Funktion aus der Multifunktionsleiste von Inventor zugegriffen werden. Zuerst wird allerdings eine Skizze mit der Umrandung erstellt und diese wird ausgewählt.

```

1. Dim oDef As ControlDefinition
2. oDef = iApp.CommandManager.ControlDefinitions("DrawingCropViewCmd")
3. oDef.Execute()
  
```

Listing 5-1: Zugriff auf die Zuschneidefunktion von Inventor

5.2.4.11 Layout für den Ausführungsübersichtsplan

Das Layout für den Ausführungsübersichtsplan, wie es im Kapitel „Konzept“ dargestellt wurde, wurde als eigene Methode namens *CreateStandard32Layout* in das Modul implementiert. Wie der Name schon andeutet, basiert das Layout auf einer 3-2-Matrix, dessen Grundkonstruktion durch die Methode *CreateLayout()* als erstes erstellt wird. Als nächster Schritt wird der Inhalt der einzelnen Layoutboxen definiert. Zum Schluss wird die Breite der zweiten Spalte noch reduziert und die *checkReservedSpace()*-Methode aufgerufen.

5.2.4.12 Änderung der Breite einer Spalte

Exemplarische wurde die Methode *changeColWidth* implementiert mit deren Hilfe sich die Spaltenbreite einer bestimmten Spalte ändern lässt. Das Prinzip wäre für eine Methode, die das Ändern der Reihenhöhe ermöglicht, natürlich gleich.

Bei einer solchen Änderung ist nicht nur die ausgewählte Spalte betroffen, sondern vor allem auch die Spalte direkt daneben. Außerdem muss die minimale Spaltenbreite immer eingehalten werden. Soll die Spaltenbreite der letzten Spalte geändert werden, geschieht die Ausdehnung nach links und betrifft die zweitletzte Spalte direkt. Bei einer anderen Spalte passiert die Ausdehnung in der jetzigen Implementation nach rechts. Es fehlt also im Moment noch der Fall, dass wenn die Ausdehnung nach rechts durch die einzuhaltende minimale Breite pro Layoutbox nicht möglich ist, über die gesamte Breite jedoch eingehalten werden könnte und somit eine Ausdehnung in beide Richtungen stattfinden müsste.

5.2.5 Umsetzung der Konzeptanwendung

Die Konzeptanwendung „InventorDrawingLayout“ ist eine gewöhnliche *Windows Application* mit einem einzigen Fenster, also einer *Form*. Beim Öffnen der Anwendung wird zuerst versucht eine Referenz zu Inventor mit der Funktion *GetObject* zu erhalten. Falls dies klappt wird als nächstes die eigene Vorlage, die im Kapitel 3.3 beschrieben und im Standardordner für metrische Vorlagen¹⁵ abgelegt wurde, eingelesen indem sie in Inventor als nicht sichtbar geöffnet wird. Je nach Systemsprache und Installation von Inventor heißt dieser Ordner allerdings anders, weswegen ein Dialogfenster zur Auswahl einer alternativen Vorlage geöffnet wird, falls die eigene Vorlage nicht gefunden wird. Die Blatinformationen werden in einem *templateDocument*-Objekt gespeichert und benutzt, um eine neue *Layout*-Instanz zu erzeugen.

Bei dem erstmaligen Laden des Fensters werden unter anderem auch die *ComboBox*-Elemente mit Hilfe von *DataTable*-Objekten gefüllt.

Um kein zweites Fenster zu haben und trotzdem den Hauptteil der Anwendung vom Layout-Editor zu trennen, wurde auf Registerkarten (*TabPage*-Steuerelemente) zurückgegriffen.

Das Hauptfenster, zu sehen in Abbildung 5-13, ist sehr schlicht gehalten und in drei Bereiche untergliedert, die der Benutzer alle eins nach dem anderen durchgehen soll. Als oberster Bereich ist die Wahl des Bauwerks, gefolgt vom Maßstab und schließlich noch die Wahl des Layouts. Dabei sind jeweils zwei unterschiedliche Möglichkeiten gegeben.

¹⁵ \Users\Public\Documents\Autodesk\Inventor 2013\Templates\Metric\

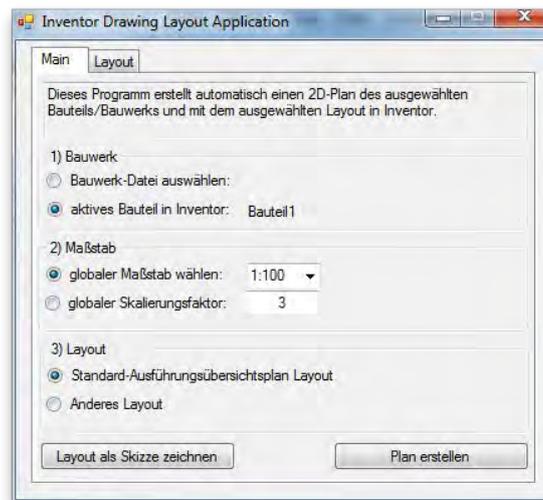


Abbildung 5-13: Hauptfenster der Konzeptanwendung

Der „Layout als Skizze zeichnen“-Button erstellt im aktiven Dokument, sofern es sich um ein Zeichnungsdokument handelt, eine Skizze und zeichnet das Gitternetz des Layouts. Dabei wird die eigens dafür implementierte Methode *DrawLayoutSketchInventor* des *Layout*-Objektes aufgerufen. Dies ist ein kleiner Zusatz und soll dem Anwender als Orientierung in Inventor helfen.

Wird als Layout „anderes Layout“ ausgewählt, springt die Anwendung automatisch in den Layouteditor.

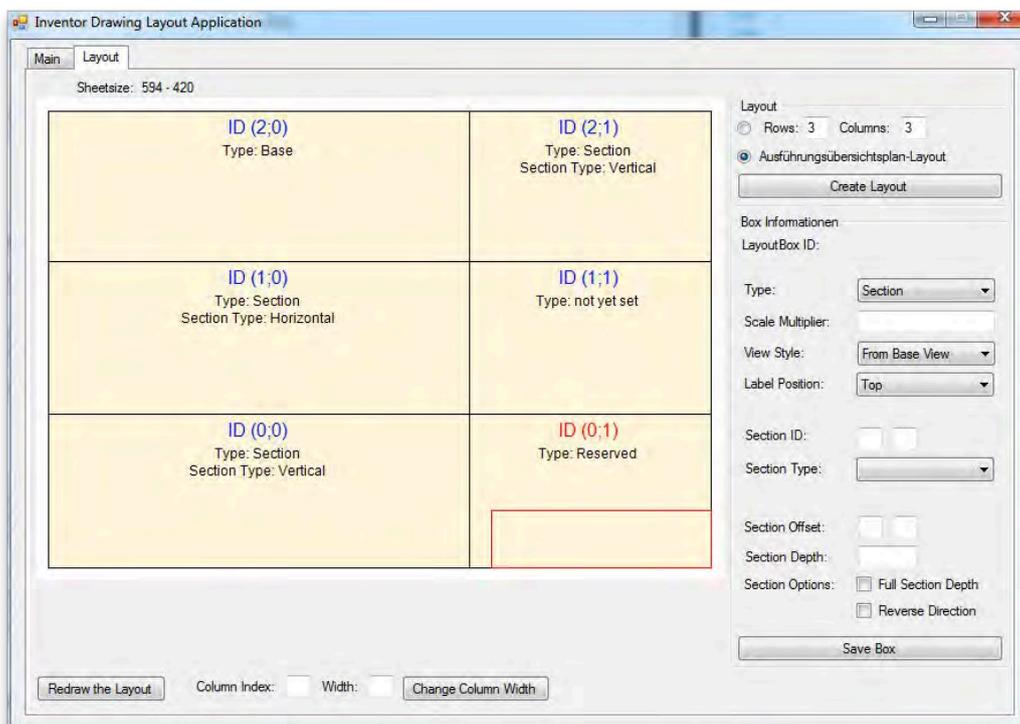


Abbildung 5-14: Der Layouteditor – als Beispiel zu sehen ist das Ausführungsübersichtsplan-Layout

Dieser gliedert sich grob in 3 Bereiche. Rechts neben der graphischen Darstellung des Layouts, gibt es oben die Möglichkeiten ein neues Layout zu erstellen, in dem die Anzahl der Spalten und der Reihen angegeben werden oder aber das fertige Layout für einen Ausführungsübersichtsplan zu generieren, wie es auch schon im Hauptfenster zur Auswahl steht. Diese zweite Möglichkeit wurde extra hinzugefügt, damit noch Änderungen an dem vordefinierten Layout vorgenommen werden können. Direkt unterhalb werden die Informationen zu der ausgewählten Layoutbox angezeigt und können editiert sowie gespeichert werden.

Unter der Grafik sind noch zwei weitere Funktionen zu finden. Neben der Möglichkeit die Spaltenbreite einer Spalte zu ändern, gibt es noch einen Button um die Grafik neu zu zeichnen. Dieser Button wurde vor allem deswegen hinzugefügt, weil bis zur Fertigstellung dieser Arbeit ein bekanntes Problem noch nicht gelöst werden konnte: Bei der Benutzung der TAB-Taste „verschwindet“ die Grafik immer.

Zur Programmierung gibt es an dieser Stelle nichts besonderes mehr zu sagen. Die wichtigsten Funktionalitäten sind alle im Modul selbst ausgelagert. Die Anwendung übernimmt nur noch die direkte Kommunikation mit dem Benutzer und deshalb besteht der Quellcode hauptsächlich aus der Steuerung der gesamten GUI und dem Informationsfluss, sowie der Interaktion mit dem Anwender.

5.3 Bekannte Probleme und einige Ideen zur Erweiterung

Sowohl die Konzeptanwendung als auch das *InventorLayout*-Modul stellen einen Anfang in der automatisierten Planableitung mit Inventor dar und besitzen neben einigen Kinderkrankheiten vor allem noch sehr viel Potential zur Weiterentwicklung. Da beides auf gewisser Art und Weise Hand in Hand gehen, wurde dieses Kapitel absichtlich in ein einziges zusammengestellt und nicht unterteilt.

5.3.1 Problem: Erkennung der Gradienten in Inventor

Die Erstellung des Planes passiert momentan voll automatisch ohne dass der Benutzer in irgendeiner Weise interagieren muss. Beim Ausführungsübersichtsplan-Layout wird beispielsweise eine vertikale Schnittansicht erzeugt, die den Regelquerschnitt repräsentieren soll. Ein solcher Schnitt kann in der Brückenmitte geführt werden. Es kann allerdings auch vorkommen, dass die Brücke einen mittleren Pfeiler besitzt, der nicht in dieser Ansicht gezeigt werden soll, weswegen der Schnitt wo anders geführt oder der Pfeiler ausgeblendet werden muss. Dies stellt kein Problem dar solange die Brücke gerade ist. Ist sie jedoch gekrümmt, muss die Schnittlinie senkrecht zur Gradienten sein, damit unabhängig von der Lage des Schnittes auf der Gradienten immer der gleiche Schnitt angezeigt wird. Einzelne Elemente in einer Skizze können keine Namen zugewiesen werden und programmiertechnisch gibt die Inventor API einfach nur eine Liste wieder mit beispielweise allen Linien aus einer Skizze. Dies führt dazu, dass die Erkennung der Gradienten programmiertechnisch alles andere als einfach und trivial ist. Im Prinzip könnte über die Erhebung des Oberbaus herausgefunden werden, welche Linie als Führungslinie dient. Allerdings muss dann die richtige Erhebung gefunden werden. Eine zweite Möglichkeit wäre

der Gradiente eine bestimmte Farbe zuzuweisen. Auf jeden Fall müssen im Vorfeld gewisse Konventionen festgelegt werden, damit dies nicht einem komplexen Suchspiel endet.

Als Beispiel wurde jetzt ein Schnitt genannt. Die beschriebene Problematik wirkt sich aber auch zum Beispiel auf das Drehen der Ansicht aus. Wie im Kapitel 4.1.1 beschrieben, sollte das Drehen der Ansicht natürlich auch möglichst automatisch durchgeführt werden. Dies konnte in entsprechenden Programmiersuchen auch umgesetzt werden, nur musste die Gradiente manuell vom Anwender ausgewählt werden. Alternativ konnte auch die Skizze, die die Gradiente enthielt, angegeben werden, doch dann musste die Gradiente sich durch eine bestimmte Farbe von allen anderen Linien unterscheiden.

5.3.2 Automatische Bemaßung

Wünschenswert wäre neben der Erstellung der einzelnen Ansichten auch eine automatische Bemaßung in einem gewissen Rahmen, allerdings ist auch hier wieder das Problem der Erkennung von Elementen, nur auf einem anderen Niveau. Sogar wenn durch Namenskonventionen recht einfach auf die Zeichnungselemente eines bestimmten Bauteils zurückgegriffen werden kann, ist ein komplexer Algorithmus nötig der jedes Element, ob Linie, Spline oder Halbbogen, sowie die gesamte Ansicht genau analysiert, damit nicht einfach alles bemaßt und die Bemaßung auch richtig platziert wird.

5.3.3 Gitternetz des Layouts komplett frei manipulieren

In der *Layout*-Klasse wurde bis dato nur die Möglichkeit implementiert die Breite einer ganzen Spalte zu ändern. Das eigentlich Ziel ist die freie Manipulation des gesamten Gitternetzes und somit auch jeder einzelnen Layoutbox. Durch die Veränderung der Abmessungen einer einzelnen Layoutbox sind alle angrenzenden Layoutboxen betroffen, was die Komplexität einer solchen Implementierungen um einiges erhöht gegenüber der Methode *changeColWidth*.

Des Weiteren müsste der Layouteditor diese Funktionalität auch leicht bedienbar anbieten, am besten durch die direkte Manipulation mit der Maus in der Darstellung des Layouts und nebenbei noch die Möglichkeit einer genauen Eingabe.

5.3.4 Speichern von Layouts

Dem Modul fehlen momentan noch die Methoden zum Speichern und Laden von Layouts. Durch die hierarchische Struktur würde sich XML hervorragend zur Speicherung eignen. Dabei hätte XML den Vorteil, dass es beim Öffnen des Dokuments in einem normalen Texteditor leicht verständlich lesbar wäre und auch das Auslesen durch andere Programme wäre dementsprechend einfach.

5.4 Weitere Entwicklungen

5.4.1 Makro zur Polygonapproximation von Splines

Die Polygonapproximation wurde im allgemeinen Teil der Arbeit (Kapitel 2.2) auf theoretischer Basis weitestgehend durchgesprochen. An dieser Stelle soll nun der Aufbau eines VBA-Makros für eine solche Approximation in Inventor erläutert werden.¹⁶

Jede Kurve besitzt eine Eigenschaft über die auf ein geometrisches Auswertungsobjekt (engl. *Evaluator*) zugegriffen werden kann. Je nach Kurventyp wäre dies *Kurve.Geometry.Evaluator* oder direkt *Kurve.Evaluator2D*. Jede Kurve wird in Inventor als parametrische Kurve hinterlegt. Deswegen ist der erste Schritt mit Hilfe der Funktion *GetParamAtExtents* des Auswertungsobjektes den Start- und Endparameter der Kurve zu ermitteln. Des Weiteren wird ein Toleranzwert als *Double* benötigt, der nichts weiter als die Länge des Sticks darstellt.

Die Polygonapproximation erfolgt mit der Funktion *GetStrokes* des Auswertungsobjektes, welche eine Liste der Eckpunkte für die Polylinie zurückgibt, sowie die Anzahl an Unterteilungen.

Bevor die Polylinie gezeichnet wird, werden zuerst die Skizzenpunkte erstellt. Bei den Koordinaten der Skizzenpunkte muss unbedingt die *SheetToDrawingViewSpace*-Prozedure der Ansicht, die die Kurve enthält, angewendet werden, um die Koordinaten der Punkte vom Blattkoordinatensystem in das Ansichtskordinatensystem zu transformieren.

Durch den Umstand, dass die Eckpunkte des Polygonzuges ziemlich exakt auf der Kurvenlinie liegen, erscheint in der Schnittansicht eine vertikale Linie. Diese lässt sich selbstverständlich im Nachhinein manuell unsichtbar machen, doch auf Programmiererebene ist dies etwas schwieriger.

Diese Problematik lässt sich durch das Verschieben der Eckpunkte um einen minimalen Wert umgehen. Dabei müssen die Eckpunkte allerdings auf die gleiche Seite der Kurve verschoben werden. Ausgehend davon, dass sich auf der Brücke keine Unendlichkeitsstelle befindet, werden die Eckpunkt mit Hilfe der zweiten Ableitung immer auf die Innenseite der Kurve verschoben.

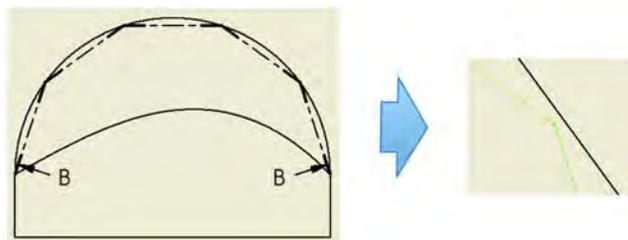


Abbildung 5-15: Beispiel der Approximation – Die Eckpunkte liegen nicht genau auf der Kurve

Durch Ausprobieren hat sich herausgestellt, dass eine Verschiebung um den Wert von 0.00001 cm ausreichend ist, damit in der Schnittansicht keine vertikalen Schnittlinien wegen den Eckpunkten mehr sichtbar sind.

¹⁶ Der komplette Quellcode befindet sich auf der CD.

Falls die Polygonapproximation nachträglich getätigt wird, sollte außerdem die Schnittmethode der Schnittansicht kontrolliert werden. Dort muss **ausgerichtet** ausgewählt sein.

Es wurde auch ein Test mit einem halben Kreis durchgeführt, welcher einen Durchmesser von 50 oE¹⁷ hatte und somit ein Länge von $50 \cdot \text{PI} = 157,079633$ oE. Folglich sind die Resultate dieses Testes in einer Tabelle aufgelistet:

Toleranzwert	Länge	Anmerkungen
0.005	157.0298	mit Skizzierpunktverschiebung von 0.000001 ; Linie in der Mitte noch sichtbar
0.005	157.0292	mit Skizzierpunktverschiebung von 0.00001
0.01	156.98	
0.1	156.07	
0.2	155.29	
0.3	154.51	6 Punkte, 5 Linien
0.4	153.07	5 Punkte, 4 Linien
0.5	153.07	5 Punkte, 4 Linien
1	150.00	4 Punkte, 3 Linien

Tabelle 5-2: Ergebnisse der Versuche mit dem Approximationsmakro

5.4.2 VB.NET Funktion zur Schnittpunktermittlung im Lageplan

Linienobjekte besitzen die Memberfunktion *IntersectWithCurve* zur Ermittlung der Schnittpunkte mit einer Kurve, egal von welchem Typ. Kurvenobjekte hingegen fehlt eine solche Funktion. In der Discussion Group „Autodesk Inventor Customization“ auf der Autodesk Seite ist von 2009 ein Thread zu finden, der sich mit dem Thema der Schnittpunktermittlung zwischen zwei Splines beschäftigt.¹⁸ Auf Basis des dort veröffentlichten Quellcodes und der Methode wurde eine Funktion *FindIntersectionPoints* geschrieben, die Schnittpunkte zwischen Splines, Linien und Bögen findet.

Allerdings wurde jetzt im Nachhinein zufällig auf einer Folie einer Präsentation von Autodesk über die Inventor 2012 API entdeckt, dass das *TransientGeometry*-Objekt seit 2012 eine *CurveCurveIntersection*-Methode anbietet. Trotzdem soll im nachfolgenden die Herangehensweise der *FindIntersectionPoints*-Funktion kurz erläutert werden.

Falls eine der beiden Kurven eine Linie ist, wird die *IntersectWithCurve*-Methode benutzt. Falls dies nicht der Fall ist, wird wie folgenden vorgegangen: Zuerst werden beide Kurven fixiert. Anschließend wird auf einer Kurve eine gewisse Anzahl an Skizzenpunkten verteilt, denen zu jeder Kurve die Abhängigkeit „koinzident“ vergeben wird. Falls die Vergabe dieser zwei Abhängigkeiten fehlschlägt, schneiden sich die Kurven nicht. Andernfalls muss nur noch

¹⁷ oE : ohne Einheit – Die Einheit spielt bei dem Test im Prinzip keine Rolle.

¹⁸ Discussion Groups: „**Finding Intersection-Points of sketchSplines**“:

<http://forums.autodesk.com/t5/Autodesk-Inventor-Customization/Finding-Intersection-Points-of-sketchSplines/td-p/2431130>

überprüft werden, welche Skizzenpunkte aufeinander liegen, um einen Schnittpunkt nicht doppelt zu erhalten. Zum Schluss muss das Fixieren der Kurven wieder aufgehoben werden.

Bei Kreisbogen muss allerdings noch überprüft werden ob der vermeintliche Schnittpunkt wirklich auf dem Bogen liegt, denn intern wird der Kreisbogen wie ein normaler Kreis behandelt, so dass bei einem Kreisbogen, der sich mit einem normal kurvigen Spline schneidet, der beispielsweise eine Gradienten einer Brücke darstellt, immer zwei Schnittpunkte herauskommen werden, sofern der Kreisbogen nicht tangential zum Spline verläuft, da der Kreis den Spline zwei mal schneidet. Kreisbögen haben gegenüber Kreisen noch die Definition des Anfangswinkel (=StartAngle-Eigenschaft) des Bogens und der Länge in Bogenmaß (=SweepAngle-Eigenschaft). Um herauszufinden ob der Kandidatenpunkt tatsächlich auf dem Kreisbogen liegt, wird dessen Winkel zum Kreisbogen gemessen.

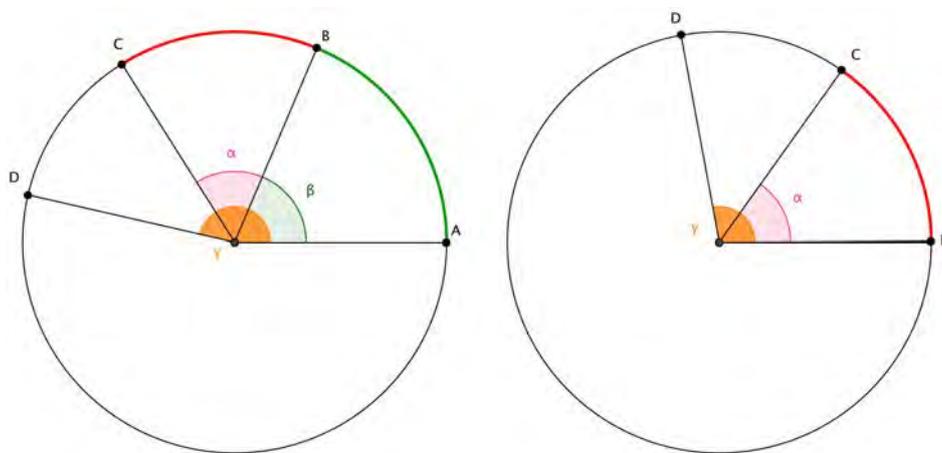


Abbildung 5-16: Darstellung der Kreisbogenproblematik

In der Abbildung 5-16 ist auf der linken Seite die Ausgangslage dargestellt. Der Punkt **D** stellt den Kandidatenpunkt dar. Der zu testende Kreisbogen **BC** hat den *SweepAngle* α und den *StartAngle* β . Die einfachste Methode zum Testen ob der Kandidatenpunkt auf dem Kreisbogen liegt, ist den Winkel γ des Punktes mit dem Wert $2\text{PI} - \beta$ zu addieren und gegebenenfalls 2PI wieder subtrahieren, falls das Resultat größer als 2PI ist. Dies entspricht einer Verschiebung des Punktes B auf den Anfangspunkten A (rechts in der Abbildung 5-16 zu sehen). So kann der neu errechnete Winkel γ direkt mit dem *SweepAngle* α verglichen werden. Falls γ kleiner als α ist, befindet sich der Kandidatenpunkt auf dem Kreisbogen. (In der Abbildung ist dies nicht der Fall.)

6 Zeichnungsexport

Bei der Erstellung von Planzeichnungen wird es in absehbarer Zeit nicht möglich sein nur mit Inventor zu Arbeiten. Das normale AutoCAD und dessen Anbindungen sind momentan ein fester Bestandteil des Arbeitsablaufes bei der Planerstellung. Somit ist es wichtig, dass der Export aus Inventor ein Dokument liefert, welches sich ohne größere Umstände und ohne Einschränkungen gut mit AutoCAD weiterverarbeiten lässt.

Folglich wird der Export samt seiner unterschiedlichen Optionen beschrieben.

6.1 Exportieren einer Zeichnung in Inventor

In dem rechteckigen Menü in der linken oberen Ecke befindet sich das Menü **Exportieren**, welches wiederum den Menüpunkt **In DWG exportieren** erhält.

Durch einen Klick auf diesen Menüpunkt erscheint das typische Speicherfenster von Windows.

Als Speichertypen gibt es zwei Möglichkeiten: Entweder als **Inventor-Zeichnungsdateien (*.dwg)** oder als **AutoCAD DWG-Dateien (*.dwg)**. Trotz gleicher Endung unterscheiden sich diese beiden Formate und für die Weiterverarbeitung in AutoCAD muss deshalb auch „*AutoCAD DWG-Dateien (*.dwg)*“ ausgewählt werden.

Über den „Optionen...“-Button lassen sich die Exportoptionen für DWG-Dateien aufrufen. Diese sind essentiell wichtig, wie im folgenden Abschnitt verdeutlicht wird.

Autodesk versucht die Kompatibilität zwischen ihren Produkten und den unterschiedlichen Versionen möglichst hochzuhalten indem sie das Dateiformat nur alle drei Jahre aktualisieren. Die neuste DWG Version ist von 2013. Zeichnungen, die in der Version gespeichert werden, können nicht mit älteren AutoCAD Versionen geöffnet werden, weswegen im Folgenden die Dateiversion auf AutoCAD 2010-Zeichnung umgestellt wurde. Hier sei natürlich anzumerken, dass je nach Versionenbestand im Unternehmen auf eine noch ältere Version zurückgegriffen werden muss.

Eine weitere Möglichkeit ist die Datei später mit *TrueDWG*, einer kostenlosen Software von Autodesk, in ein älteres Format zu konvertieren. Dies ist bei Dokumenten, die mit einer Software für Lernzwecke erstellt wurden, unterbunden.

Über **Next** > sind noch weitere Optionen zu finden. Unter anderem lassen sich auch nur einzelne Blätter für den Export auswählen, falls nicht alle benötigt werden - sofern mehrere vorhanden sind.

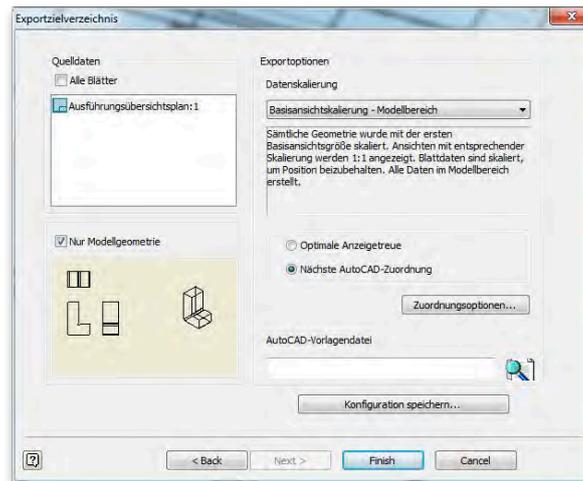


Abbildung 6-1: Exportoptionen-Fenster

Ein ganz wichtiger Punkt an dieser Stelle ist die Datenskalierung. Es gibt grob drei unterschiedliche Skalierungsoptionen, wobei jeweils auch noch Layout- oder Modellbereich unterschieden wird:

- **Basisansichtsskalierung:** *Sämtliche Geometrie wurde mit der ersten Basisansichtsgroße skaliert. Ansichten mit entsprechender Skalierung werden 1:1 angezeigt. Blattdaten sind skaliert, um Position beizubehalten. Alle Daten im Modellbereich/Papierbereich (Layout) erstellt.*
- **Blattskalierung:** *Keine Geometrieskalierung vorhanden. Blattdaten sind dimensional in Originalgröße. Alle Daten im Modellbereich/Papierbereich(Layout) erstellt.*
- **Maßstab 1:1:** *Sämtliche Geometrie im Maßstab 1:1 skaliert. Alle Daten im Modellbereich erstellt. Vorsicht: Einige Ansichten können aufgrund verschiedener Skalierungen und Positionen auf dem Blatt überlappen.*

Für die Weiterverarbeitung wäre es wichtig ein Dokument im Maßstab 1:1 zu erhalten, allerdings ist bei der dementsprechenden Datenskalierungsoption der Haken bei **Nur Modellgeometrie** automatisch gesetzt und es ist nicht möglich diesen zu entfernen. So wird eine DWG-Zeichnung erstellt, welche weder Plankopf noch Schnitte mit einbezogene Skizzen oder Bemaßungen beinhaltet. Deshalb stellt dies definitiv keine Option dar, die benutzt werden kann, ohne die eigentlichen Zeichnungsarbeiten alle in AutoCAD selbst zu erstellen.

Alle anderen Datenskalierungsoptionen skalieren die jeweiligen Ansichten nach ihren Einstellungen.

7 Schlussfolgerung

Zusammenfassend lässt sich behaupten, dass Autodesk Inventor durchaus das Potential hat sinnvoll für die Modellierung und die Planableitung von Brückenbauwerken eingesetzt zu werden. Allerdings besitzt die Software einige Schwächen, darunter eine gravierende, die tief in ihrem Kern verankert sind. Diese Nachteile beschränken Inventor für den speziellen Anwendungszweck wiederum ein.

Die nicht vorhandene Möglichkeit bei der Modellierung des Oberbaus den Querschnitt immer parallel zur globalen Z-Achse auszurichten, was zu einer Verdrehung des Oberbaus führt, ist wohl der Hauptnachteil von Inventor. Dabei lässt sich die Funktion „Erhebung“, die zur Erstellung des 3D-Körpers verwendet wird, nicht einfach um diese Option durch externe Programmierung erweitern, sondern müsste seitens des Herstellers angepasst werden. Die erwähnte Verdrehung hat in der Planableitung zur Folge, dass Zeichnungselemente durch Splines abgebildet werden, da sie gekrümmt sind, obwohl sie eigentlich gerade sein müssten. Außerdem entstehen durch diese Art der Modellierung Abmessungsfehler. Obschon durch gewisse Modelliermethoden und den fahrdynamischen Randbedingungen aufgezeigt wurde, dass diese Abmessungsfehler sich in einem gewissen Rahmen befinden und durch ihre geringe Größe durchaus in Kauf genommen werden können, müssen sie dennoch immer berücksichtigt und schließlich vom Anwender toleriert werden.

Durch die sehr umfangreiche Inventor API war es leicht möglich ein gewisses Konzept zur automatisierten Planableitung in Inventor in ein funktionierendes Modul und einer dazugehörigen Konzeptanwendung umzusetzen. Dabei ist das entwickelte Konzept ein Grundkonzept und im Prinzip vom Modell und von der Planart unabhängig. Obwohl beispielsweise das Platzieren einer Ansicht in Inventor über die API ziemlich einfach ist, gibt es einerseits gewisse Schwierigkeiten, die mit der komplexen Art der Modellierung einer Brücke zusammenhängen, andererseits sind scheinbar triviale Prozesse, wie unter anderem die automatische Bemaßung von gewissen Elementen, nur schwer auf Programmiererebene zu realisieren und in irgendeiner Weise zu standardisieren. Hierfür wären umfassende Algorithmen nötig, was wiederum zu einer langen und intensiven Entwicklungszeit führen würde. Geschuldet ist dies unter anderem in der Art wie Inventor intern die Daten speichert.

Unabhängig von diesen genannten Schwierigkeiten ist eine Teilautomatisierung, vor allem um schnell Übersichtspläne zu generieren, sehr wohl leicht anwendbar. Es wird in naher Zukunft nicht möglich sein ohne gewisse Vor- und Nacharbeit vom Anwender komplette Pläne zu erstellen. Das Grundgerüst kann mit Hilfe des selbstgeschriebenen Moduls mit relativ wenig Aufwand aufgebaut werden. Jede Verfeinerung, die über das Grundgerüst hinausgeht, wird programmiertechnisch sehr schnell ziemlich komplex. Als kurzes Beispiel könnte an dieser Stelle noch einmal die Linienführung des Längsschnittes erwähnt werden. Bei einer gekrümmten Brücke, könnte die Gradiente mit Hilfe des Approximationsmakros mit wenigen Eingaben vom Anwender recht schnell als Polygonzug approximiert werden. Dabei wäre die Auswahl der Gradiente mit einem Klick getätigt, wohingegen auf Programmiererebene diese Auswahl alles andere als banal ist und einiges an Entwicklungszeit in Anspruch nehmen

würde. Es ist deshalb sinnvoll je nach bevorstehendem Aufwand für die komplette Automatisierung eines Prozesses stattdessen lieber auf interaktive Makros zu setzen.

Der Punkt ist also sehr schnell erreicht, wo entschieden werden muss, ob in Entwicklungszeit investiert wird, oder ob der Anwender einige Schritte weiterhin manuell tätigen muss. Davon mal abgesehen ist Inventor anderen CAD-Programmen wie beispielsweise AutoCAD im Funktionsumfang für die Zeichnungserstellung einfach unterlegen. Hinzu kommt, dass viele Unternehmen AutoCAD schon Jahre lang im Einsatz haben, die Software ein fester Bestandteil des Erstellungsprozesses darstellt und auch dort viel Zeit in die Entwicklung von Erweiterungen gesteckt wurde. Deswegen muss sich immer wieder die Frage gestellt werden „Was soll in Inventor selbst realisiert werden und was sollte besser erst nach einem Export in AutoCAD hinzugefügt werden?“. Dabei darf der Hauptfokus nicht darin bestehen, den jetzigen Arbeitsprozess möglichst unverändert zu lassen, sondern aufgrund der gestiegenen Anforderungen sind neue, wohldurchdachte Konzepte nötig, die den ganzen Arbeitsprozess an sich neu gestalten. Außerdem ist zu beachten, dass eine Konvertierung immer mit einem Informationsverlust behaftet ist und im Falle von Inventor nach AutoCAD deshalb auch unidirektional verläuft. Hinzu kommt noch eine zum Teil redundante Datenhaltung, die am Besten vermieden werden soll.

Der 3D-Ansatz mit Inventor sollte weiterhin verfolgt werden. Um jedoch produktiv in einem Unternehmensumfeld mit dieser Software bei der Erstellung von Plänen eines Brückenbauwerkes zu arbeiten, müssen im Gespräch mit den verantwortlichen Ingenieuren, Zeichnern, Modellierern und Programmierern gemeinsam Konventionen festgelegt und das Einsatzgebiet der Software genau definiert werden.

Anhang A

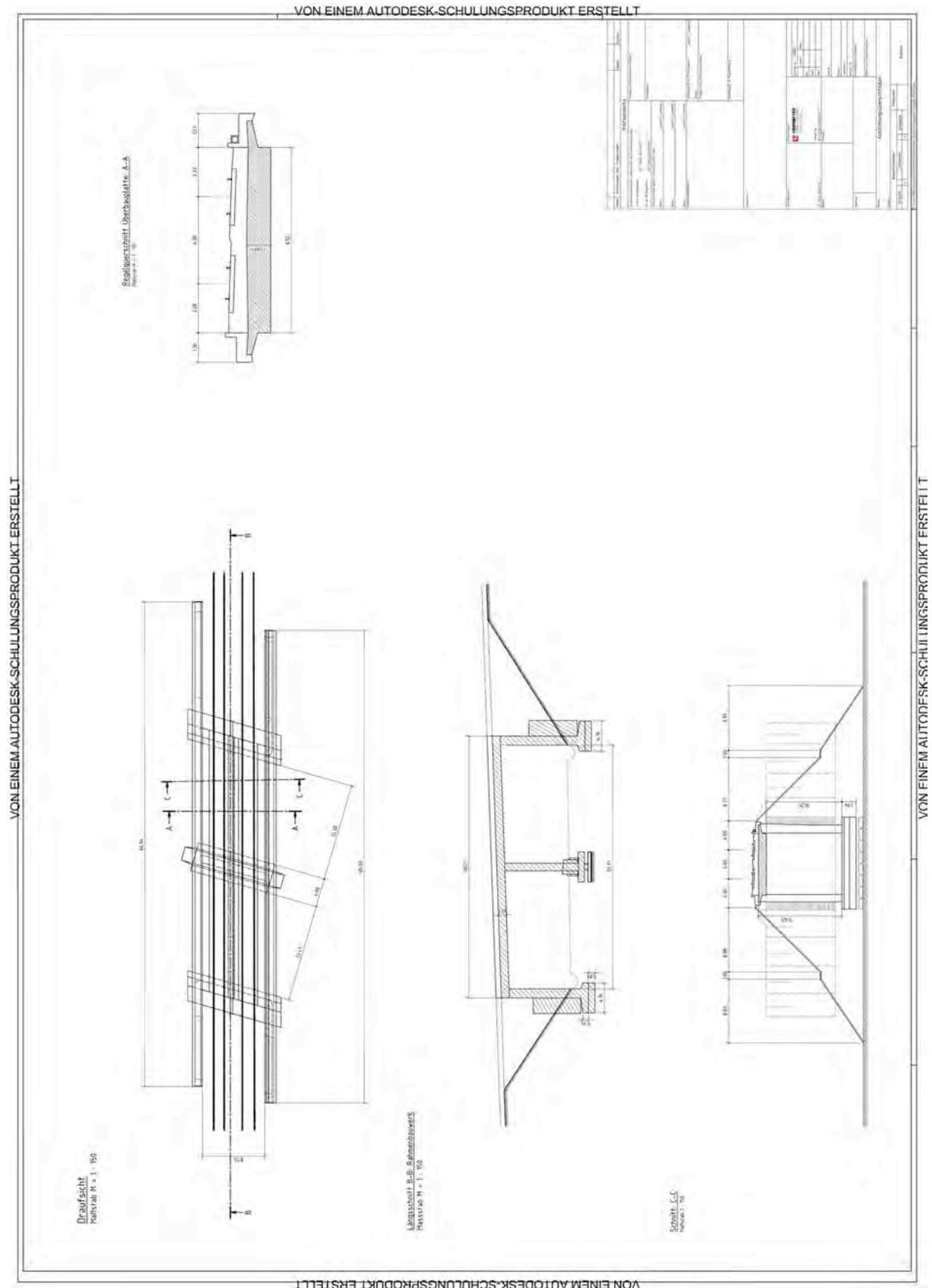


Abbildung A-1: Manuell erstellter Ausführungsübersichtsplan

Anhang B

Variablenamen	Variabletyp	Zweck
LayoutMatrix(.)	LayoutBox	Speicherung der einzelnen Layoutboxen
RangeBox ReservedSpace	Inventor.Box2d Inventor.Box2d	Definition des Zeichenbereiches Schriftfeldbereich
scaleFactor	Double	Globaler Skalierungsfaktor
iApp	Inventor.Application	Referenzobjekt zur Inventor-Anwendung
Integrity minimalBoxWidth minimalBoxHeight	Boolean Double Double	Interne Integritätsvariable Minimale Layoutbox-Breite (10 cm) Minimale Layoutbox-Höhe (8 cm)

Tabelle 1-1: Variablenauflistung für die Klasse *Layout*

Typ	Methodenname	Parameter	Rückgabewert
Prozedur	New	Optional ByVal Dimensions As Inventor.Box2d = Nothing, Optional ByRef oSheet As Inventor.Sheet = Nothing	
Prozedur	Initialisation	ByRef oApp As Inventor.Application	
Prozedur	Release		
Funktion	Width		Double
Funktion	Height		Double
Funktion	rowsIndex		Integer
Funktion	colsIndex		Integer
Prozedur	EndLayout		
Funktion	getLayoutBoxByID	ByVal id As InventorLayout.LayoutID	InventorLayout.LayoutBox
Prozedur	changeColWidth	ByVal colIndex As Integer, ByVal changeWidth As Double	
Prozedur	setGlobalScaleFactor	ByVal globalScaleFactor As Double	
Prozedur	setGlobalInventorModel	ByRef globalModel As Object	
Private			
Prozedur	CheckIntegrity		
Prozedur	CreateLayout	ByVal row As Integer, ByVal col As Integer	
Prozedur	DrawLayoutGraphics	ByRef g As Graphics, Optional ByVal DrawReservedSpaceBoolean As Boolean = True	
Prozedur	DrawLayoutInventor	ByVal DrawingSheet As Inventor.Sheet	
Prozedur	CreateStandard32Layout	Optional ByRef model As Inventor.Document = Nothing	
Prozedur	checkReservedSpace		
Prozedur	setReservedSpace	ByVal reservedRangeBox As Inventor.Box2d	
Prozedur	DrawLayoutSketchInventor	ByRef oDrawingDoc As Inventor.DrawingDocument	

Tabelle 1-2: Auflistung der Methoden der Klasse *Layout*

Variablenamen	Variabletyp	Zweck
ID	InventorLayout.LayoutID	ID der Layoutbox
Name	String	
InventorName	String	
InventorModel	Inventor.Document	Referenzobjekt zum Modell
iApp	Inventor.Application	Referenzobjekt zur Inventor-Anwendung
RangeBox	Inventor.Box2d	Definition der Layoutbox-Abmessungen auf dem Blatt in cm
DrawingRangeBox	Inventor.Box2d	Definition der Layoutbox-Abmessungen auf dem Blatt in cm
Type	LayoutBoxTypeEnum	Layoutbox Typ
RotateView	Double	Drehwinkel in Bogenmaß für die Ansicht
RotateCamera	Double	Drehwinkel in Bogenmaß für die Kamera
scaleFactor	Double	Skalierungsfaktor
scaleFactorMulitplier	Double	Wert, der mit dem Skalierungsfaktor multipliziert wird
ViewStyle	Inventor.DrawingViewStyleEnum	Ansichtsstil
labelPosition	LayoutBoxLabelPostionEnum	Position für die Ansichtsbezeichnung
autoResize	Boolean	Boole'scher Wert, der die automatische Verkleinerung steuert
cropView	Boolean	Zuschneiden der Ansicht, ja/nein
Basisansichtsvariablen:		
BaseView	Inventor.DrawingView	Referenz zur Basisansicht
Orientation	Inventor.ViewOrientationTypeEnum	Ausrichtung der Ansicht
Schnittansichtsvariablen:		
sectionBaseView	Inventor.DrawingView	Referenz zur Basisansicht für die Schnittansicht
sectionView	Inventor.SectionDrawingView	Referenz zur Schnittansicht
sectionBaseViewLayoutBoxID	LayoutID	ID der Basisansicht
sectionType	LayoutBoxSectionTypeEnum	Schnitttyp
sectionAngle	Double	Schnittwinkel
sectionOffset	Inventor.Point2d	Offset-Wert für den Mittelpunkt der Schnittlinie
FullSectionDepth	Boolean	Vollschnitt, ja/nein
SectionDepth	Double	Schnitttiefe
ReverseDirection	Boolean	Schnitttrichtung ändern, ja/nein

Tabelle 1-3: Variablenauflistung für die Klasse *LayoutBox*

Typ	Methodenname	Parameter	Rückgabewert
Prozedur	New	ByVal Dimensions As Inventor.Box2d	
Prozedur	Initialisation	ByRef oApp As Inventor.Application	
Funktion	Width		Double
Prozedur	SetDrawingRangeBox	ByVal Dimensions As Inventor.Box2d	
Funktion	Height		Double
Funktion	Check	ByVal reservedBox As Inventor.Box2d	Boolean
Prozedur	DrawLayoutBoxGraphics	ByRef g As Graphics	
Prozedur	DrawLayoutBoxInventor	ByRef DrawingSheet As Inventor.Sheet	

Tabelle 1-4: Auflistung der Methoden der Klasse *LayoutBox*

Anhang C

Compact Disc

Auf der beigefügten Compact Disc sind folgende Daten enthalten:

- Der schriftliche Teil der Arbeit als Word-Dokument und als PDF-Dokument.
- Die Inventor dwg.-Vorlage.
- Ein Excel-Dokument zum 1. Approximationsfehler beim Längsschnitt.
- Der manuell und der automatisch erstellte Ausführungsübersichtsplan als Inventor-dwg.-Datei.
- Das gesamte Visual Studio Projekt „InventorDrawingLayout“.
- Die kompilierte InventorDrawingLayout-Anwendung als Installer.
- Das VBA Approximationsmakro als Textdatei.
- Linksammlung als PDF oder/und HTML.

Literaturverzeichnis

Allgemeine Literatur

- Batran, Balder, Alexandra Born, Volker Frey, Beatrix Gustavus, Hans-Jürgen Hansen, Klaus Dr. Köhler, Heinz Schliebner, Jens-Peter Schuller, Helmut Sommer, Frank Weidner, Rüdiger Wenzke, und Joachim Dr. Zwanzig. (2010). *Bauzeichnen: Architektur, Ingenieurbau, Tief-, Straßen- Und Landschaftsbau*. Holland & Josenhans.
- Bielefeld, Bert, und Isabella Skiba. (2006). *Basics Technisches Zeichnen*. 1. Aufl. Birkhäuser Verlag AG.
- Bindseil, Peter. (1998). *Stahlbetonfertigteile : Unter Berücksichtigung Von Eurocode 2 ; Konstruktion - Berechnung - Ausführung*. Edited by Peter Bindseil. 2. Aufl. Düsseldorf: Werner.
- Brühwiler, Eugen, und Christian Menn. (2003). *Stahlbetonbrücken*. Dritte, aktualisierte und erweiterte Auflage. Springer-Verlag Wien New York.
- Goldau, Richard. (1978). *Zeichnen Im Konstruktiven Ingenieurbau*. Bauverlag.
- Holschemacher, Klaus, und Thomas Ackermann. (2009). *Entwurfs- Und Berechnungstafeln Für Bauingenieure*. 4., aktualisierte und erweiterte Auflage. Bauwerk Verlag GmbH.
- Holst, Karl Heinz. (1993). *Brücken Aus Stahlbeton Und Spannbeton. Entwurf, Konstruktion Und Berechnung*. 3. A. ed. Ernst W. + Sohn Verlag.
- Mittag, Martin, und Rosemarie Tietze-Mittag. (1974). *Bauzeichnungslehre*. 8. Aufl., 71. - 80. Tsd. Limburg (Lahn): Starke.
- o.V. (2008). *Pläne Und Daten Richtliniensammlung*. URL: http://www.vbv.baden-wuerttemberg.de/servlet/PB/show/1237914/A1_Allgemeine_Planvorgaben_V02_112008.pdf (aufgerufen am 11.3.2013).
- Pfeifer, Rolf H., und Tristan M. Mölter. (2008). *Handbuch Eisenbahnbruecken. Grundsätze Fuer Planung Und Konstruktion Sowie Hinweise Auf Bauverfahren*.
- Prenzel, Rudolf. (1982). *Bauzeichnung Und Darstellungstechnik*. 2., überarb. und wesentl. erw. Auflage. Stuttgart: Krämer.
- Wikipedia. (2013). *Bauzeichnung*. URL: <http://de.wikipedia.org/wiki/Bauzeichnung> (aufgerufen am 11.3.2013).
- Wikipedia. (2010). *Schraffur*. URL: <http://de.wikipedia.org/wiki/Schraffur> (aufgerufen am 11.3.2013).
- Schneider, Klaus-Jürgen. (2010). *Bautabellen Für Ingenieure: Mit Berechnungshinweisen Und Beispielen*. 19. Auflage. Werner, Neuwied.
- Werner, Ulrich, und Walter Pastor. (2010). *VOB - Vergabe- Und Vertragsordnung Für Bauleistungen. HOAI - Honorarordnung Für Architekten Und Ingenieure*. 28. Auflage, August 2010. Deutscher Taschenbuch Verlag.

Literatur zur Software *Autodesk Inventor*

- Autodesk. (2011). *Bemaßungsstile in Zeichnungen (WIKIhelp)*. URL:
http://wikihelp.autodesk.com/Inventor/deu/2012/Help/0073-Autodesk73/0285-Arbeitsu285/0292-Stile_un292/0300-Bemaßung300 (aufgerufen am 11.3.2013).
- Autodesk. (2012). *Schnittansichten (WIKIhelp)*. URL:
<http://wikihelp.autodesk.com/Inventor/deu/2013/Help/1309-Autodesk1309/1813-Zeichnun1813/1823-Erstelle1823/1830-Schnitta1830> (aufgerufen am 11.3.2013).
- Autodesk. (2012). *Schriftfelder (WIKIhelp)*. URL:
<http://wikihelp.autodesk.com/Inventor/deu/2013/Help/1309-Autodesk1309/1813-Zeichnun1813/1814-Übersich1814/1822-Schriftf1822> (aufgerufen am 11.3.2013).
- Autodesk. (2012). *Zeichnungsrahmen (WIKIhelp)*. URL:
<http://wikihelp.autodesk.com/Inventor/deu/2013/Help/1309-Autodesk1309/1813-Zeichnun1813/1814-Übersich1814/1821-Zeichnun1821> (aufgerufen am 11.3.2013).
- o.V. (2011). *Autodesk Inventor 2012. Trainingshandbuch, Grundlagen*. Mensch und Maschine Systemhaus GmbH.
- Scheuermann, Günter. (2010). *Inventor 2011 Bauteile, Baugruppen, Zeichnungen*. Editiert von Günter Scheuermann. 3., aktualisierte Aufl. *Cad to Go*. München: Hanser.
- Scheuermann, Günter. (2011). *Inventor 2012*. Hanser Fachbuchverlag.
- Tremblay, Thom. (2012). *Autodesk Inventor 2013 and Autodesk Inventor Lt 2013 Essentials*. John Wiley & Sons.
- Waguespack, Curtis. (2012). *Mastering Autodesk Inventor 2013 and Autodesk Inventor LT 2013*. John Wiley & Sons.

Literatur zur Programmierung

- Autodesk. (2012). *Autodesk Inventor 2013 Api Object Model*. URL:
<http://images.autodesk.com/adsk/files/inventor2013model.pdf> (aufgerufen am 11.3.2013).
- Ekins, Brian. (2011). *Understanding What Autodesk® Inventor® Apprentice Can Do for You*. URL: http://aucache.autodesk.com/au2011/sessions/5606/class_presentations/v1_CP5606-Apprentice.pdf (aufgerufen am 20.3.2013).
- Ekins, Brian, und Brill, Wayne. (o.J.). *Inventor Customization Blog: Mod the Machine*. URL: <http://modthemachine.typepad.com>
- Autodesk Discussion Groups. (2009). *Finding Intersection-Points of Sketchsplines*. URL: <http://forums.autodesk.com/t5/Autodesk-Inventor-Customization/Finding-Intersection-Points-of-sketchSplines/td-p/2431130> (aufgerufen am 11.3.2013).
- Kühnel, Andreas, und Stephan Leibbrandt. (2009). *Visual Basic 2008: Das Umfassende Handbuch (Galileo Computing)*. 3 Aufl. Galileo Computing.
- Microsoft. (o.J.). *Gewusst Wie: Verweisen Auf COM-Objekte Aus Visual Basic*. URL: <http://msdn.microsoft.com/de-de/library/vstudio/cwys3b23.aspx> (aufgerufen am 11.3.2013).
- Wikipedia. (2013). *Component Object Model*. URL: http://de.wikipedia.org/wiki/Component_Object_Model (aufgerufen am 11.3.2013).

Abbildungsverzeichnis

ABBILDUNG 2-1: BEISPIEL EINER AUSFÜHRUNGSZEICHNUNG (GOLDAU R., 1978, S.45)	12
ABBILDUNG 2-2: FALTEN NACH DIN 824 (AUS ENTWURFS- UND BERECHNUNGSTAFELN FÜR BAUINGENIEURE, TAFEL 12.23A, SEITE 12.23)	15
ABBILDUNG 2-3: LINIENBREITEN NACH DIN 1356-1 1995-02, SEITE 6	14
ABBILDUNG 2-4: SCHRAFFUREN DER DIN ISO 128-50 UND DER DIN 1356-1 IM VERGLEICH (HOLSCHEMACHER K., 2009, S. 12.25).....	14
ABBILDUNG 2-5: SCHNITTFÜHRUNG BEIM LÄNGSSCHNITT (AUS BATRAN B. ET AL., 2010, S.14)	16
ABBILDUNG 2-6: BEGRIFFE DES KREISBOGENS.....	16
ABBILDUNG 2-7: DIAGRAMM: SEHNENFEHLER BEI EINEM RADIUS VON 300M.....	17
ABBILDUNG 2-8: 1. APPROXIMATIONSFEHLER ALS FLÄCHENDIAGRAMM	17
ABBILDUNG 2-9 : ZWEITER APPROXIMATIONSFEHLER.....	18
ABBILDUNG 3-1: AUSWAHL DER ZUR VERFÜGUNG STEHENDEN ANSICHTSTYPEN	20
ABBILDUNG 3-2: EINSTELLUNGSFENSTER FÜR EINE ANSICHT	20
ABBILDUNG 3-7: IPROPERTIES ALS TEXT EINFÜGEN	26
ABBILDUNG 3-8: BRÜCKENMODELL-NORM ALS STILNORM ANLEGEN.....	27
ABBILDUNG 3-9: REGISTERKARTEN DER STILEINSTELLUNGEN EINER NORM	28
ABBILDUNG 3-10: FENSTER DER <i>OBJEKTVORGABE</i> -EINSTELLUNGEN.....	29
ABBILDUNG 3-11: ANZEIGEEINSTELLUNGEN	30
ABBILDUNG 3-12: DETAIL-EINSTELLUNGEN DER BEMAßUNGSLINIEN.....	31
ABBILDUNG 4-1: DRAUFSICHT-BEISPIEL	32
ABBILDUNG 4-2: DREHEN MIT HILFE EINES REFERENZKÖRPERS	33
ABBILDUNG 4-4: REGELQUERSCHNITT-BEISPIEL.....	34
ABBILDUNG 4-5: QUERSCHNITT-BEISPIEL.....	35
ABBILDUNG 4-6: LÄNGSSCHNITT-BEISPIEL	35
ABBILDUNG 5-1: KONZEPTUELLER AUSFÜHRUNGSÜBERSICHTSPLAN	36
ABBILDUNG 5-2: VERSCHIEDENE LAYOUTVARIANTEN	37
ABBILDUNG 5-3: DIAGRAMM DER DREI SCHRITTE, DIE DER ANWENDER DURCHGEHEN MUSS.....	38
ABBILDUNG 5-4: DIAGRAMM: ZUGRIFF AUF DIE INVENTOR API (AUTODESK, O.J.)	39
ABBILDUNG 5-5: DIAGRAMM: DIREKTER/INDIREKTER WEG	40
ABBILDUNG 5-6: DIAGRAMM: BEISPIEL ZUR INSTANZIERUNG VON OBJEKTEN AUS DER INVENTOR API.....	41
ABBILDUNG 5-7: DIAGRAMM: EIGENSCHAFTENVERBINDUNG	42
ABBILDUNG 5-8: DIAGRAMM: ANWENDUNGS-AUFBAU IN 3 EBENEN GEGLIEDERT.....	43
ABBILDUNG 5-9: DIAGRAMM: ZUSAMMENHANG DER KLASSEN <i>LAYOUT</i> UND <i>LAYOUTBOX</i>	45
ABBILDUNG 5-10: URSPRÜNGE UND ACHSENRICHTUNGEN	47
ABBILDUNG 5-11: BEISPIEL: 4 FEHLENDE PIXEL IN Y-RICHTUNG VERTEILEN	48
ABBILDUNG 5-12: DIAGRAMM: ÜBERBLICK ZUR VORGEHENSWEISE DER METHODE <i>DRAWLAYOUTINVENTOR</i>	49
ABBILDUNG 5-13: HAUPTFENSTER DER KONZEPTANWENDUNG	51
ABBILDUNG 5-14: DER LAYOUTEDITOR – ALS BEISPIEL ZU SEHEN IST DAS AUSFÜHRUNGSÜBERSICHTSPLAN- LAYOUT.....	51
ABBILDUNG 5-15: BEISPIEL DER APPROXIMATION – DIE ECKPUNKTE LIEGEN NICHT GENAU AUF DER KURVE... ..	54
ABBILDUNG 5-16: DARSTELLUNG DER KREISBOGENPROBLEMATIK.....	56
ABBILDUNG 6-1: EXPORTOPTIONEN-FENSTER	58
ABBILDUNG A-1: MANUELL ERSTELLTER AUSFÜHRUNGSÜBERSICHTSPLAN.....	61

Tabellenverzeichnis

TABELLE 2-1: NORMENLISTE	11
TABELLE 2-2: MAßSTÄBE VERSCHIEDENER PLANARTEN.....	12
TABELLE 5-1: ENUMERATIONEN AUS DEM <i>INVENTORLAYOUT</i> -MODUL	44
TABELLE 5-2: ERGEBNISSE DER VERSUCHE MIT DEM APPROXIMATIONSMAKRO	55
TABELLE 1-1: VARIABLENAUFLISTUNG FÜR DIE KLASSE <i>LAYOUT</i>	62
TABELLE 1-2: AUFLISTUNG DER METHODEN DER KLASSE <i>LAYOUT</i>	62
TABELLE 1-3: VARIABLENAUFLISTUNG FÜR DIE KLASSE <i>LAYOUTBOX</i>	63
TABELLE 1-4: AUFLISTUNG DER METHODEN DER KLASSE <i>LAYOUTBOX</i>	63

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Bachelor-These selbstständig erarbeitet habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

Ich versichere außerdem, dass die vorliegende Arbeit noch nicht einem anderen Prüfungsverfahren zugrunde gelegen hat.

München, 25. März 2013

Unterschrift des Verfassers

Manuel Fiermonte

[Redacted signature area]