

Alexander Ott, BSc.

Integration der Virtual Reality Technologie in den Fahrzeugentwicklungsprozess

Master Thesis

zur Erlangung des akademischen Grades
eines Master of Science (MSc.)
der Studienrichtung Maschinenbau
an der Technischen Universität Graz

Institut für Fahrzeugtechnik

Institutsvorstand:

Univ.-Prof. Dipl.-Ing. Dr.techn. Peter Fischer

Betreuer:

Assoc.Prof. Dipl.-Ing. Dr.techn. Mario Hirz

Graz, Dezember 2018

Vorwort

Die vorliegende Arbeit entstand im Zuge meiner Tätigkeit bei der Firma MAGNA STEYR Fahrzeugtechnik AG & Co KG in Graz, im Bereich der virtuellen Vorentwicklung von computergestützten Arbeitsmethoden und dem Institut für Fahrzeugtechnik an der Technischen Universität Graz.

In diesem Zusammenhang gilt mein besonderer Dank Herrn DI Severin Stadler und Herrn DI Dr.techn. Wilhelm Dietrich für die kollegiale Betreuung und die vielen inspirierenden Gespräche und fachlichen Diskussionen zu dieser Arbeit.

Meinen Dank möchte ich auch an die Mitarbeiter des Instituts für Fahrzeugtechnik, insbesondere Herrn DI Robert Hofstadler und Herrn DI Markus Salchner, aussprechen, die immer ein offenes Ohr für meine Anliegen und Fragen hatten.

Mein allergrößter Dank gilt meinen Eltern und Familie. Ohne sie wäre ich nun nicht an diesem Lebenspunkt angelangt. Sie haben es mir ermöglicht, diesen Weg zu gehen und mich immer mit ihrer Liebe und ihrem Glauben an mich unterstützt. Vielen herzlichen Dank!

Kurzfassung

Die Automobilbranche ist ein Gewerbe, welches sich sukzessive mit schneller Geschwindigkeit weiterentwickelt. So wie sich die Anforderungen an die Produkte immer wieder verändern, müssen auch die internen Prozesse der Automobilhersteller und deren Zulieferer diesen neuen Anforderungen angepasst und weiterentwickelt werden. Zeitgleich wird versucht, die Prozesse immer weiter zu optimieren, um die Entwicklungszeit und somit auch die Entwicklungskosten zu reduzieren. Großes Potential versprechen in dieser Hinsicht die frühen Phasen des Fahrzeugentwicklungsprozesses. In diesen Arbeitsabschnitten bestehen noch keine physischen Prototypen und Versuchsträger; jedoch kann in diesen Phasen bereits auf moderne CAD-Programme zurückgegriffen werden, welche es ermöglichen, an virtuellen Modellen Auslegungs- und Absicherungsarbeiten durchzuführen.

Das zurzeit am meisten Erfolg versprechende Werkzeug, um die Effektivität dieser Arbeiten zu steigern, ist der Einsatz von Virtual Reality Methoden. Die Qualität der VR Geräte ist nun so weit fortgeschritten, dass der Nutzer vollständig in eine virtuelle Welt eintauchen und dort an einem virtuellen Fahrzeugmodell, als wäre es physisch vorhanden, arbeiten kann. Des Weiteren können mit neuen VR Softwarelösungen fotorealistic Renderings des virtuellen Fahrzeugmodells erzeugt werden. Durch deren geschickten Einsatz können viele Arbeitsanwendungen schon sehr früh im Entwicklungsprozess durchgeführt werden, was der Effektivität und der Stabilität des gesamten Fahrzeugentwicklungsprozesses zugutekommt. Hierzu ist jedoch viel Vorarbeit zu leisten, bzw. sind neue Arbeitsprozesse zu erstellen und bereits vorhandene Prozesse mit den neuen Möglichkeiten zu erweitern.

In der vorliegenden Arbeit wird eine mögliche Integration der Virtual Reality Methoden in den Fahrzeugentwicklungsprozess erarbeitet. Diesbezüglich wird der Fahrzeugentwicklungsprozess hinsichtlich des Einsatzes von VR Methoden untersucht und aufgezeigt, wo Potential dafür vorhanden ist. Den Hauptteil der Arbeit nimmt die Erstellung des Integrationsprozesses ein, welcher das nötige Umfeld und Arbeitsschritte erläutert, die zur Implementierung von VR Anwendungen in den Fahrzeugentwicklungsprozess notwendig sind. Durch die Erstellung des VR Integrationsprozesses und der Erstellung einer autonomen VR Autodesk VRED Arbeitsvorlage liefert diese Arbeit einen wesentlichen Beitrag für den gezielten Einsatz von Virtual Reality Methoden im Fahrzeugentwicklungsprozess.

Abstract

Automotive development is a constantly evolving industry. Just as the demands on the products change, the internal processes of automobile manufacturers and their suppliers must be adapted to these new requirements and constantly further developed. At the same time, efforts are being made to optimize the processes to reduce development time and therefore also development costs. Great potential in this regard promise the early stages of the vehicle development process. In these stages, there are no physical prototypes and experimental vehicles available, but modern CAD programs can already be used during these early phases to carry out design and validation tasks on virtual models.

The most promising tool to increase the effectiveness of this tasks includes the use of virtual reality methods. The quality of the VR devices is already so advanced that the user can fully immerse themselves in a virtual world and work on a virtual vehicle model as if it were physically present. Furthermore, photo-realistic renderings of the virtual vehicle model can be generated with the current VR software solutions. Their use enables many work applications to be carried out very early in the development process, which benefits the effectiveness and stability of the vehicle development process. For these possibilities many preliminary works have to be done and new work processes have to be created or already existing processes have to be expanded.

In the present thesis, a possible integration of virtual reality methods into the vehicle development process is developed. In this regard, the vehicle development process is examined and the potential for virtual reality is shown. The main part of the thesis includes the creation of a VR integration process, which explains the necessary environment in the company and necessary work steps to implement VR applications in the vehicle development process. With the creation of the VR integration process and creating a standalone VR Autodesk VRED template, this thesis provides a significant contribution to the use of virtual reality methods in the vehicle development process.

Eidesstattliche Erklärung

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen nicht benutzt und die den Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen inländischen oder ausländischen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht. Die vorliegende Fassung entspricht der eingereichten elektronischen Version.

16.12.2018

A handwritten signature in blue ink, appearing to read 'Aldi Ols', is positioned above the printed word 'Unterschrift'.

Unterschrift

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ausgangssituation	1
1.2	Zielsetzung	1
2	Virtual Reality	3
2.1	Allgemeines	3
2.2	Die Anfänge von Virtual Reality	5
2.3	Stand der Technik	9
2.3.1	Technische Eigenschaften	9
2.3.2	VR-Geräte bei Magna Steyr	13
2.4	Das VR-System	15
2.4.1	Verzögerungen im VR-System	16
2.5	Das VR-Team	18
3	Der Fahrzeugentwicklungsprozess	21
3.1	Phasen des Fahrzeugentwicklungsprozesses	21
3.2	Meilensteine	24
3.3	Simultaneous Engineering	26
4	Potential im Fahrzeugentwicklungsprozess	28
4.1	Der Use Case - Anwendungsfall	29
4.1.1	Allgemeine Erläuterung	29
4.1.2	Erstellung eines Use Case	29
4.1.3	Das Use Case Diagramm	31
4.2	Anwendungsbezogene Use Cases	34
4.3	Hürden der Integration	38
5	Der Virtual Reality Integrationsprozess	40
5.1	Workflow zur Erstellung einer VR-Szene	40
5.2	Bereitstellung und Aufbereitung der Dateien	43
5.2.1	Modellvarianten im Fahrzeugentwicklungsprozess	43
5.2.2	Erstellung der Modellstruktur	46
5.2.3	Handhabung der Geometriedateiformate und Dateienexport	48

5.2.4	Einschub: Geometriedateiformat JT (Jupiter Tessellation).....	52
5.2.5	Konvertierung der Daten.....	56
5.2.5.1	Standardeinstellungen des Konverters	57
5.2.6	Einfügen der Dateien in die Visualisierungssoftware Autodesk VRED.	60
5.2.7	Vergleich der relevanten Aufbereitungsvarianten.....	61
5.3	Visualisierung - Modellaufbereitung	64
5.3.1	Die Autodesk VRED Arbeitsumgebung	66
5.3.1.1	Asset Manager.....	67
5.3.1.2	Material Editor	68
5.3.1.3	Kamera Editor	71
5.3.1.4	Licht Editor	73
5.3.1.5	Clip Maker	75
5.3.1.6	Curve Editor	76
5.3.1.7	Textur Editor	76
5.3.1.8	UV-Editor.....	77
5.3.1.9	Overlay Editor	78
5.3.1.10	Node Editor	78
5.3.2	Verwaltung der Projektdateien.....	79
5.3.2.1	Asset Manager Bibliothek.....	79
5.3.2.2	VISU Hauptbibliothek	81
5.3.2.3	VISU Projekt Bibliothek	82
5.3.2.4	Projektbezogene Ordnerstruktur	82
5.4	Der automatisierte Update Prozess	83
5.4.1	Definition der Vergleichsparameter	84
5.4.2	Erhaltung der Arbeitsstruktur.....	84
5.4.2.1	Dateienbereitstellung.....	85
5.4.2.2	Einfügen der Dateien in VRED.....	87
5.4.2.3	Unterteilung des Import-Knotens.....	88
6	Erstellung eines Visualisierungs-Template	90
6.1	Aufbau des VRED Master Template	90
6.2	Funktionen des Master Template, Werkzeuge und Plug-ins	93
6.2.1	Auswahl der Funktionen	93
6.2.1.1	Bedienung mittels Tastenkombinationen.....	93

6.2.1.2	Interaktive Bedienung des Master Template.....	94
6.2.2	Taschenlampe – „Flashlight“	95
6.2.3	Schneiden des Fahrzeugmodells – „Sectioning“	96
6.2.4	Vergleich von Fahrzeugmodellvarianten – „Comparison“	96
6.2.5	Positionswechsel - Viewpoints	97
6.2.6	Umgebungswechsel.....	97
6.2.7	Modellwechsel	98
6.2.8	Wechsel der Farbe des Fahrzeugmodells – Color Switch.....	98
6.2.9	Vordefinierte Schnitte - Sections	98
6.2.10	Zurücksetzen der Szene - Reset	98
6.2.11	Teleporting Plug-in	99
6.3	Performance des Master-Template	100
6.3.1	Beschreibung des Performance Tests.....	100
6.3.2	Testumgebung	101
6.3.3	Ergebnis und Auswertung	102
6.3.3.1	Versuch 1: Erhöhung der Anzahl der dargestellten Dreiecke	102
6.3.3.2	Versuch 2: Erhöhung der Anzahl der Struktur-	
	/Geometrieknoten.....	104
7	Zusammenfassung und Ausblick	106
	Literaturverzeichnis	108

Abbildungsverzeichnis

Abbildung 1	Kollaborative AR Bauraumbeurteilung (Körner, 2017)	4
Abbildung 2	Mixed Reality (Milgram, 1995)	5
Abbildung 3	Sensorama – Werbegrafik und Patent-Zeichnung (Computer History Museum)	6
Abbildung 4	HMD Prototyp von Ivan Sutherland (Computer History Museum).....	6
Abbildung 5	VPL Prototypen von Datenhandschuhen und Ganzkörper Anzügen (Robertson & Zelenko).....	7
Abbildung 6	VR-Brillen Marktanteil Q3 2017 (Canalys, 2017).....	8
Abbildung 7	Explosionsdarstellung einer Oculus Rift DK2 (Lang, 2017)	10
Abbildung 8	Darstellung des Sichtfelds (Kleinman, 2016)	11
Abbildung 9	HTC Vive Pro (links, HTC) und Oculus Rift (Oculus)	13
Abbildung 10	Darstellung einer VR-Szene auf einer Powerwall (3Dims)	14
Abbildung 11	Schematische Darstellung eines VR-Systems (Dörner, 2013).....	15
Abbildung 12	Latenzen im VR-System	17
Abbildung 13	Aufgabenbereiche des VR Teams	19
Abbildung 14	Phasen des Fahrzeugentwicklungsprozesses (Hirz, 2013).....	21
Abbildung 15	Schematisches Beispiel eines Fahrzeugentwicklungsprozesses	24
Abbildung 16	Zeitersparnis durch Simultaneous Engineering	27
Abbildung 17	Notation eines Use Case.....	31
Abbildung 18	Notation eines Akteurs	32
Abbildung 19	Assoziationen eines Use Case	32
Abbildung 20	Beziehungen zwischen Use Cases	33
Abbildung 21	Multiplikatoren an einem Use Case	33
Abbildung 22	VR-Darstellung eines CFD Ergebnisses in BETA CAE Systems ANSA	36
Abbildung 23	HMI Untersuchung mit Hilfe einer Sitzkiste (nViz, 2018).....	37
Abbildung 24	Vereinfachte Darstellung des Workflows	41
Abbildung 25	Magna Steyr Concept Car Mila Plus.....	42
Abbildung 26	Designprozess in der Fahrzeugentwicklung.....	44
Abbildung 27	Datenentwicklung der Modellvarianten	46
Abbildung 28	Schematischer Aufbau des PDM Systems	47

Abbildung 29	Dateistruktur vor und nach der Konvertierung in JT	48
Abbildung 30	Dateien Aufbereitungsprozess CAS.....	49
Abbildung 31	Dateien Aufbereitungsprozess CAD	50
Abbildung 32	PLMXML Konverter	51
Abbildung 33	JT Geometrieformat nach Siemens	53
Abbildung 34	Darstellung der Tesselierungsparameter (Unreal engine, 2018).....	54
Abbildung 35	Tesselierter Körper in Autodesk VRED	55
Abbildung 36	CAD to OSB Konverter	56
Abbildung 37	Einstellungsmenü „CAD to OSB“ - Konverter.....	57
Abbildung 38	Dreieck der Modellvisualisierung.....	65
Abbildung 39	Arbeitsumgebung Autodesk VRED.....	66
Abbildung 40	Autodesk Automotive Assets.....	67
Abbildung 41	VRED Material Editor	68
Abbildung 42	Interaktionen im zwischen VRED Asset Manager und Material Editor. 69	
Abbildung 43	VRED Kamera Editor	71
Abbildung 44	VRED Licht Editor	73
Abbildung 45	VRED Clip Maker.....	75
Abbildung 46	VRED Curve Editor	76
Abbildung 47	VRED Textur Editor	77
Abbildung 48	VRED UV-Editor.....	78
Abbildung 49	Ordnerstruktur einer Asset Bibliothek	80
Abbildung 50	Ordnerstruktur der VISU Hauptbibliothek.....	81
Abbildung 51	Vordefinierte projektbezogene Ordnerstruktur	82
Abbildung 52	PDM Versionsmanagement	83
Abbildung 53	Entwurf des Update-Prozesses.....	87
Abbildung 54	Importstruktur des VRED Master Template.....	89
Abbildung 55	Szenengraph des VRED Master Template.....	90
Abbildung 56	„T_Model“ Knoten.....	92
Abbildung 57	„T_Env_Switch“ Knoten	92
Abbildung 58	Tastenkombinationen zur Steuerung des Master-Template.....	94
Abbildung 59	Controller der HTC Vive VR Brille (HTC, 2018)	94
Abbildung 60	Interaktives Menü des Master Template	95
Abbildung 61	VR Taschenlampe	95
Abbildung 62	Schnitt des Fahrzeugmodells in der VR Szene	96

Abbildung 63	Vergleich zweier Modelle in der VR Szene.....	96
Abbildung 64	Kacheln in VR Szene	97
Abbildung 65	Benutzeroberfläche des Teleporting Plug-in.....	99
Abbildung 66	HTC Vive Pro Versuch 1	103
Abbildung 67	Desktop Bildschirm Versuch 1	103
Abbildung 68	HTC Vive Pro Versuch 2	105
Abbildung 69	Desktop Bildschirm Versuch 2	105

1 Einleitung

1.1 Ausgangssituation

Der Produktentwicklungsprozess entwickelt sich stetig weiter und so sind auch alle Automobilhersteller darum bemüht, ihre Prozesse bestmöglich zu optimieren und deren Effizienz zu steigern. Zu diesem Zweck werden auch neue Methoden untersucht und bei positiven Erkenntnissen in den Fahrzeugentwicklungsprozess aufgenommen. Eine dieser neuen Methoden ist die Anwendung von Virtual Reality. Virtual Reality verspricht ein großes Potential als Präsentationsmethode von virtuellen Fahrzeugprototypen, welche so realitätsnah aufbereitet werden können, dass für viele Arbeiten auf einen physischen Prototyp verzichtet werden kann. Des Weiteren ergeben sich durch Virtual Reality ganz neue Möglichkeiten, um Ergebnisse aus der virtuellen Absicherung und Simulation zu betrachten und Schlüsse daraus zu ziehen. So kann neben der Senkung von Kosten auch die Effektivität einzelner Prozesse gesteigert werden.

1.2 Zielsetzung

Das große Potential von Virtual Reality Anwendungen hat auch Magna Steyr erkannt und bereits vieles unternommen, um sie in ihre Prozessabläufe zu integrieren. So wurde bereits eine eigene Arbeitsgruppe gegründet, welche sich mit dieser Thematik beschäftigt und diese mit der nötigen Hardware, Software und Infrastruktur ausgestattet. Des Weiteren wurden bereits erste Anwendungsfälle für VR Methoden ausgearbeitet und diese, parallel zu den herkömmlichen Arbeitsprozessen, an vergangenen Projekten angewandt. Die daraus resultierenden Erfahrungen sollen nun genutzt werden, um einen vollständigen Prozessablauf bezüglich der Integration von Virtual Reality Methoden in den Fahrzeugentwicklungsprozess abzubilden.

Eine der größten Herausforderungen des VR Integrationsprozesses liegt in der Bereitstellung der Fahrzeugdateien. Bei der Durchführung vergangener Projekte wurde noch kein Augenmerk auf den Dateibereitstellungsprozess gelegt, da die virtuellen Fahrzeugmodelle nur für einmalige Untersuchungen dienten. Die Dateien wurden dazu von der zuständigen Fachabteilung zur Verfügung gestellt und von einem Mitarbeiter des VR Teams, für diese eine Anwendung, aufbereitet. Ein Ziel dieser Arbeit ist es nun, den gesamten Prozess zu beschrei-

ben und darzustellen, um in weiterer Folge den Prozessablauf zu verbessern und dessen Effektivität zu steigern. Des Weiteren soll die Möglichkeit eines automatisierten Update Prozesses, von der Erstellung der CAD-Datei bis zum Einfügen in die Visualisierungssoftware, betrachtet und im Prozess berücksichtigt werden.

Abschließendes Ziel ist es, eine Arbeitsvorlage für die Visualisierungssoftware Autodesk VRED (Autodesk) anzufertigen, welche eine schnelle Durchführung von ersten interaktiven Arbeiten an einem beliebigen virtuellen Fahrzeugmodell ermöglicht.

2 Virtual Reality

2.1 Allgemeines

Egal welches Wissenschaftsmagazin man aufschlägt, in vielen befindet sich ein Bericht über den Einsatz von Virtual Reality-Anwendungen (abgekürzt VR) in verschiedensten Einsatzgebieten. In den letzten Jahren bzw. Monaten ist die Entwicklung der Endprodukte so weit fortgeschritten, dass die Technik auch für Privatkunden leistbar wurde und so auch den Weg raus aus den Labors der Firmen hinein in die Wohnzimmer der Privatanutzer fand.

Aber was ist diese Virtual Reality? Um den Begriff besser zu verstehen, sollte man zuerst die Bedeutung der Wörter Virtual (deutsch „virtuell“) und Reality (deutsch „Realität“) einzeln betrachten. In unserem Sprachgebrauch wird als virtuell etwas bezeichnet, das „nicht echt“ bzw. „nicht in Wirklichkeit vorhanden“ ist. In technischer Hinsicht wird dieser Begriff öfters verwendet, um etwas zu beschreiben, das zwar nicht physisch, jedoch in seiner Funktionalität vorhanden ist - wie zum Beispiel ein virtuelles Laufwerk in der Informatik.

Als Realität verstehen wir die „Wirklichkeit“, die Welt die wir alle kennen und durch unsere Sinneseindrücke wahrnehmen. Der zusammengesetzte Begriff Virtual Reality beschreibt somit die Wahrnehmung einer virtuellen Welt, welche zwar nicht echt ist, jedoch in ihrer Funktionalität vorhanden ist und somit auch erlebt werden kann. (William R. Sherman, 2003)

Der Anwender taucht dabei vollständig in die virtuelle Welt ein und erlebt sie mit beinahe all seinen Sinnen, was es zu einer sehr immersiven Erfahrung macht. Die reale Welt um sich herum kann der Anwender nicht sehen. Der Grad der Immersion von VR hängt hierbei von vielen Faktoren ab, welche sich in eine technische und eine emotionale Seite unterteilen lassen. Zu den technischen Faktoren gehört hier zum Beispiel das Sichtfeld der VR-Brille, die Framerate der dargestellten Szene, die Genauigkeit der Standortbestimmung (Tracking), die Reaktionszeit der Hardware, Möglichkeiten der Interaktion usw., wobei die emotionalen Faktoren sich von Nutzer zu Nutzer unterscheiden, da jeder Mensch seine Umgebung auf verschiedene Weise wahrnimmt bzw. auf verschiedene Weise emotional auf seine Umgebung reagiert. So kann als Immersionsgrad nur ein theoretisches Potenzial verstanden werden, inwieweit es möglich ist, mit vorgegebener Hardware eine möglichst immersive VR Umgebung zu erzeugen. (Slater M., 2009)

Neben der Virtual Reality haben sich auch andere Formen der virtuellen Arbeitsunterstützung in der Forschung etabliert. Hier ist die „Augmented Reality“ (abgekürzt AR), übersetzt ins Deutsche „Erweiterte Realität“, zu erwähnen. AR erlaubt es den Nutzer die reale Welt zu sehen, welche durch virtuelle Objekte und Informationen ergänzt wird. Dies kann über einen transparenten Bildschirm oder über eine direkte Projektion der Objekte ins Auge erfolgen. Infolgedessen erweitert AR die Realität, während VR diese vollkommen ersetzt. In Abbildung 1 wird ein solches Szenario, anhand einer kollaborativen Beurteilung des Bauraums eines Fahrzeugmotorraumes, dargestellt.



Abbildung 1 Kollaborative AR Bauraumbeurteilung (Körner, 2017)

Einige Hersteller von VR-Brillen (auch HMD, für Head-Mounted Display) statten diese mit Kameras an der Frontseite aus, welche dadurch in der Lage sind, eine Mischung aus VR und AR darzustellen. Hierbei wird dem Nutzer die, durch die Kameras aufgenommene, reale Umgebung am Bildschirm dargestellt und diese mit virtuellen Objekten erweitert.

Neben AR wird diese Form der Darstellung häufig auch als „Mixed Reality“ (abgekürzt MR, auf Deutsch frei übersetzt „gemischte Realität“) bezeichnet. Der Begriff „Mixed Reality“ bedeutet, dass reale und virtuelle Inhalte miteinander vermischt werden. MR stellt jedoch in Wirklichkeit keine explizite Form der Visualisierungsart dar, sondern ist als ein Kontinuum, welches sich zwischen der Realität und der Virtualität erstreckt, zu verstehen. In der Literatur ist diesbezüglich die systematische Darstellung (Abbildung 2) nach Paul Milgram (1995) weit verbreitet.

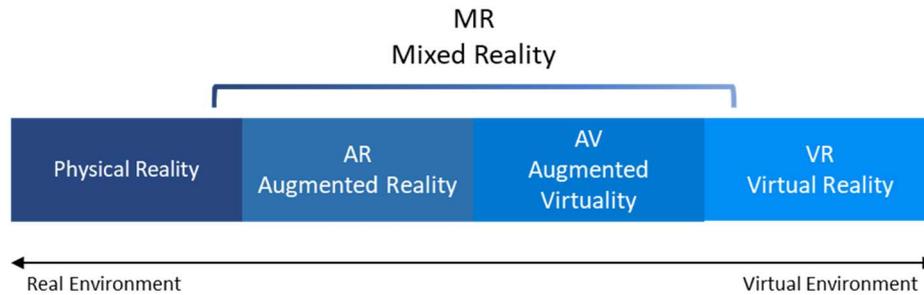


Abbildung 2 Mixed Reality (Milgram, 1995)

Im Realitäts-Virtualitäts-Kontinuum nach Paul Milgram ist die „Augmented Virtuality (AV, auf Deutsch „erweiterte Virtualität“) ein weiterer Teil der gemischten Realität (MR). Die AV besteht überwiegend aus virtuellen Räumen, in die reale physikalische Objekte und Personen eingebunden werden und diese in Echtzeit mit der virtuellen Welt interagieren können.

2.2 Die Anfänge von Virtual Reality

Die Anfänge von VR liegen bereits viel länger zurück als man glauben mag. Ihren Ursprung fand das Konzept in der Science-Fiction Literatur, welche bereits in den 1960er Jahren vielen Entwicklern den Anstoß gab, es auch in die Realität umzusetzen. Neben ersten Displays, die in der Lage waren Bild- und Videoinhalte dreidimensional darzustellen, gelang es dem amerikanischen Erfinder und Filmmacher Morton Leonard Heilig eines der ersten VR Endgeräte zu entwickeln, welches er unter den Namen „Sensorama“ im Jahre 1962 patentierte. Das System, welches in Abbildung 3 dargestellt ist, war in der Lage 3D Videos abzuspielen. Es war unter anderem mit einem vibrierenden Sitz, Stereo-Tonausgabe, einen Ventilator für Windeffekte und einer Vorrichtung zur Emittierung von Gerüchen ausgestattet. Als „Kino der Zukunft“ beworben konnte sich dieses System jedoch nie durchsetzen. (Robertson & Zelenko)

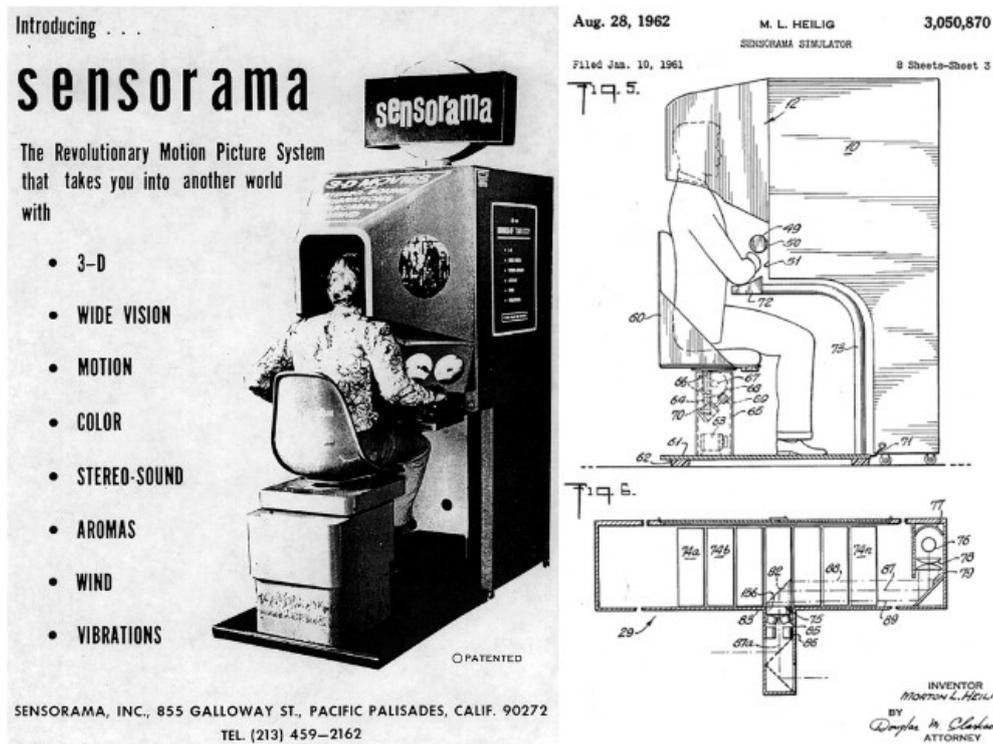


Abbildung 3 Sensorama – Werbegrafik und Patent-Zeichnung (Computer History Museum)

Im Jahre 1965 nahm sich Ivan Sutherland dieser Entwicklung an, der bereits zuvor als Entwickler der bahnbrechenden Computer Grafikanwendung „Sketchpad“ Erfolge feierte. Sutherland überdachte das Ausgabekonzept und stellte 1968 einen ersten Prototyp eines Head-Mounted Display (HMD) vor, welches er „Sword of Damocles“ nannte (Abbildung 4). Diese Vorrichtung kann als erstes HMD, inklusive Head-Tracking Funktionalität, bezeichnet werden, dessen Grundkonzept auch noch heute angewendet wird. (Robertson & Zelenko)



Abbildung 4 HMD Prototyp von Ivan Sutherland (Computer History Museum)

Ab den 1980er Jahren bekam die Entwicklung an VR einen neuen Aufschwung. Viele Unternehmen wurden in dieser Zeit gegründet, welche den kommerziellen Vertrieb von VR-Komponenten als Ziel hatten. Hierbei ist besonders das Unternehmen „VPL Research“ zu erwähnen, welches von Jaron Lanier und Thomas Zimmermann im kalifornischen Palo Alto im Jahre 1984 gegründet wurde. Das Pionierunternehmen von VR-Technologie und Network-3D-Graphics entwickelte eine VR-Brille mit dem Namen „EyePhone“, den ersten kommerziellen Datenhandschuh und einen VR-Anzug, der die Bewegungen des kompletten Körpers erfassen konnte (Abbildung 5). Alle VR-Unternehmen hatten jedoch dasselbe Problem. Ihre Entwicklungen waren ihrer Zeit voraus und extrem teuer, was einen kommerziellen Vertrieb, gerichtet an private Nutzer, unmöglich machte. 1990 meldete VPL Research Konkurs an und wurde von Sun übernommen. (Robertson & Zelenko)



Abbildung 5 VPL Prototypen von Datenhandschuhen und Ganzkörper Anzügen (Robertson & Zelenko)

Zu dieser Zeit entstanden auch viele günstige VR-Produkte für Konsolen- und Computerspieler. Aber auch diese Produkte scheiterten an der miserablen Auflösung, technischer Unzulänglichkeiten und zu hohen Preisen. Nur im professionellen Bereich wurden VR-Lösungen dagegen immer häufiger eingesetzt.

Bis zum nächsten Meilenstein in der VR-Entwicklung musste man bis ins Jahr 2012 warten. In diesem Jahr wurde der amerikanische Programmierer und Spieleentwickler John Carmack auf den Studenten Palmer Luckey aufmerksam, der in der Garage seiner Eltern an einem VR-Brillen Prototypen arbeitete. John Carmack zeigte diesen Prototyp auf der Spielemesse E3 in Los Angeles der Öffentlichkeit und sorgte damit für eine Wiedergeburt der VR-Brille

für private Anwendungen. Die in Folge dessen als Kickstarter-Projekt gegründete Firma Oculus wurde in kürzester Zeit weltweit bekannt und löste einen riesen Hype um die neuen Produkte aus. Dieser Hype animierte auch Hersteller, wie zum Beispiel HTC, Sony und Samsung, zur Entwicklung eigener VR-Geräte. (Länger, 2017)

Nach einigen Entwicklermodellen brachte Oculus im Jahr 2016 mit dem Modell Rift VR die erste VR-Brille für Endverbraucher auf den Markt. Ein Jahr später erschien mit der HTC Vive ein direktes Konkurrenzprodukt, welche zur offiziellen VR-Brille der Internet-Vertriebsplattform Steam wurde, da Oculus von Facebook übernommen wurde. Im selben Jahr startete auch Sony den Verkauf ihrer VR-Brille PlayStation VR. Diese kann jedoch nur in Kombination mit der hauseigenen Spielkonsolenplattform verwendet werden und ist somit für industrielle Anwendungen ungeeignet.

Im November 2017 berichtete die auf Marktanalysen spezialisierte Firma Canalys, dass im dritten Quartal des Jahres 2017 zum ersten Mal über eine Million Stück VR-Brillen weltweit verkauft wurden. Der summierte Marktanteil der oben genannten Firmen belief sich dabei auf 86% des Gesamtvolumens. (Canalys, 2017)

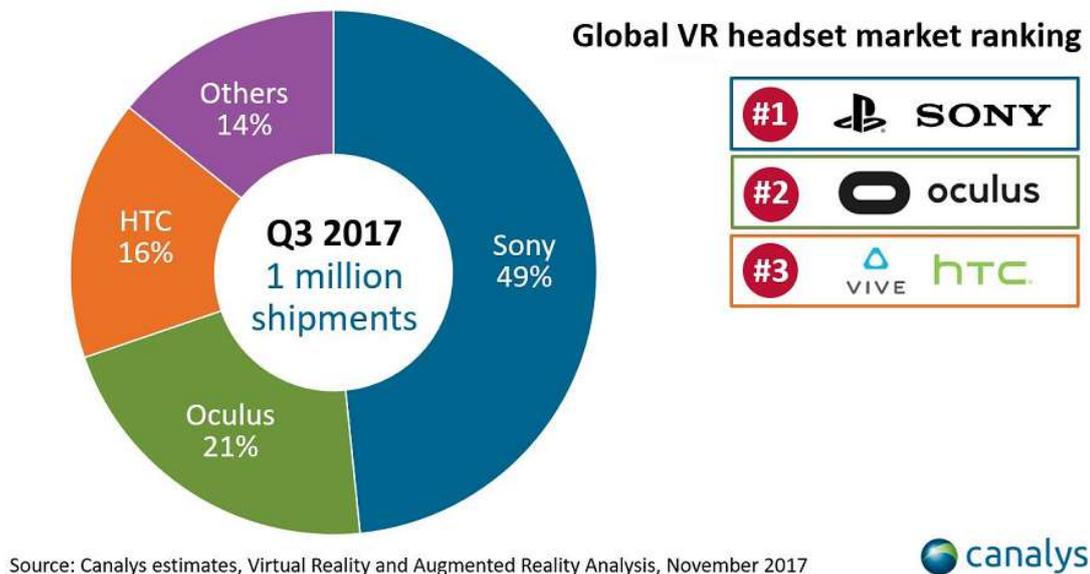


Abbildung 6 VR-Brillen Marktanteil Q3 2017 (Canalys, 2017)

2.3 Stand der Technik

In diesem Kapitel wird nun der aktuelle Entwicklungsstand der aktuellen VR Ausgabegeräte Generation beschrieben und näher auf die im Laufe dieser Arbeit verwendeten Geräte eingegangen. Wie bereits in Kapitel 2.2 erwähnt avancierten sich Oculus und HTC zu den marktführenden Herstellern von HMDs, welche für den Einsatz in der Produktentwicklung und Industrie anwendbar sind. Aber auch ihre Produkte hatten zu Beginn mit technischen Unzulänglichkeiten zu kämpfen. Dadurch entwickelte sich der Absatz der Geräte nicht wie erwartet, wodurch viele Hersteller, wie auch Kunden, an der Technologie zu zweifeln begannen. Die Hauptprobleme waren unter anderem die noch schlechte Bildschirmqualität, eine große Latenz in der Wiedergabe und eine dadurch hervorgerufene Übelkeit beim Anwender oder auch das mangelnde und viel zu teure Softwareangebot für Privatnutzer. (Mattgey, 2017)

2.3.1 Technische Eigenschaften

In den folgenden Punkten werden die technischen Eigenschaften der HMDs genauer erläutert und deren möglichen Probleme näher beschrieben:

Linsen:

Über spezielle Linsen (Hinweis 1 in Abbildung 7) wird sichergestellt, dass der Anwender das Bild, trotz der unmittelbaren Nähe zu den Augen, fokussieren und somit scharf sehen kann. Bei den modernen Geräten ist der Abstand zwischen Linse und Auge oder zwischen Linse und Display einstellbar (2) und dadurch eine leichte Kurz- oder Weitsichtigkeit des Anwenders ausgleichbar. Ebenso ist der Abstand zwischen den Linsen dem Augenabstand anpassbar. (Charara, 2017)

Die Einstellmöglichkeiten der Linsen sind sehr wichtig, da es modellabhängig nur einen schmalen Bereich gibt, indem der Anwender das dargestellte Bild optimal sieht – der sogenannte „Sweet Spot“. Es ist hierbei wichtig genau mittig und gerade durch die Linsen zu sehen, um die beste Bildschärfe und Bildqualität zu erreichen. (Albrand, 2016)

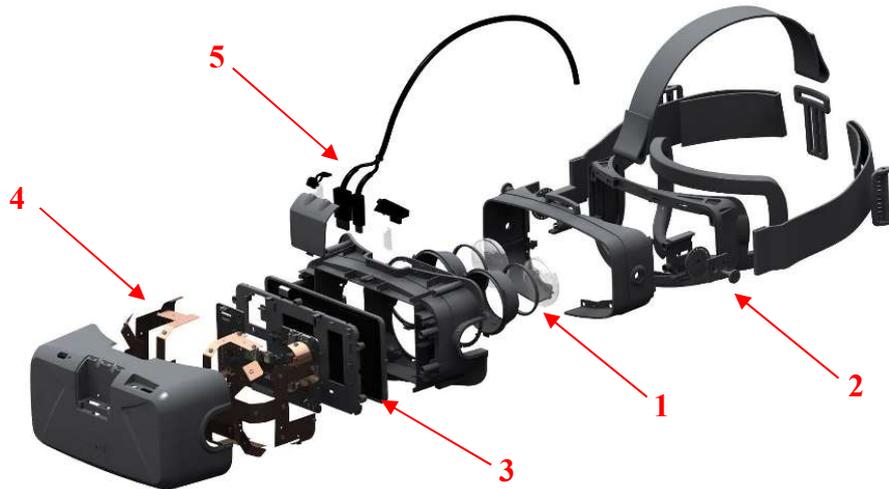


Abbildung 7 Explosionsdarstellung einer Oculus Rift DK2 (Lang, 2017)

Bildschirm:

Einen sehr großen Einfluss auf die Qualität der VR-Erfahrung hat die Auflösung des Bildschirms. So kann man bei aktuellen Modellen, obwohl die Auflösung bereits über Full-HD liegt, ein Gitter zwischen den Pixeln erkennen. Dieser Effekt ist auf die zu geringe Pixeldichte, dem sehr geringen Abstand zwischen Auge und Bildschirm und der starken Fokussierung durch die Linsen zurückzuführen. Die Hersteller versuchen dem Effekt durch eine Erhöhung der Pixeldichte entgegenzuwirken. Der limitierende Faktor ist hierbei jedoch nicht die Bildschirmauflösung und die damit einhergehende Pixeldichte, sondern die Leistung der Grafikkarten, welche die virtuelle Szene berechnen. So präsentierte Google in Zusammenarbeit mit LG bereits ein spezielles OLED Display für VR-Anwendungen, welches eine Auflösung von 18-Megapixel und eine Pixeldichte von 1443 Pixel pro Zoll aufweist. (Etienne, 2018) Nur leider sind die aktuellen Grafikkartengenerationen noch nicht in der Lage, eine etwas aufwendigere virtuelle Szene in dieser Auflösung performant zu berechnen.

Die Darstellung auf dem Bildschirm erfolgt stereoskopisch, was bedeutet, dass für jedes Auge ein eigenes Bild dargestellt wird. Die Bilder unterscheiden sich in der Perspektive um den Augenabstand, um somit das räumliche dreidimensionale Sehen zu simulieren. Dazu muss für jedes Auge ein eigenes Bild berechnet werden, worauf die hohe benötigte Grafikleistung zurückzuführen ist. (Nunez, 2015)

Bildwiederholrate:

Die Bildwiederholrate beschreibt die Anzahl an Bildern pro Sekunde, welche der Bildschirm der VR-Brille anzeigt. Aktuelle Modelle haben eine maximale Bildwiederholrate von 90 Hz, wobei 75 Hz ausreichen, um eine optimale virtuelle Erfahrung zu ermöglichen. Die letztendlich erreichte Bildwiederholrate ist jedoch wiederum vom Detailgrad und Größe der virtuellen Szene, der Grafikkartenleistung und der Verbindung zwischen Host-PC und VR-Brille abhängig.

Sichtfeld:

Das Sichtfeld, auch Field of View (FoV), beschreibt den Blickwinkel, den der Nutzer durch die VR-Brille einsehen kann (Abbildung 8). Ist das Sichtfeld zu schmal, so ist um das Bild der schwarze Displayrand zu erkennen, was die empfundene Immersivität sehr verschlechtert. Aktuelle Modelle weisen ein Sichtfeld von 110° auf, wodurch der Rand nicht sonderlich auffällt.



Abbildung 8 Darstellung des Sichtfelds (Kleinman, 2016)

Tracking:

Ein weiterer wichtiger Bestandteil von VR-Brillen ist das Head Tracking. Es ermöglicht eine räumliche Positionserkennung der Brille und übermittelt deren Bewegungen an die VR-Szene. Dadurch bewegt sich das Sichtfeld parallel zu den Kopfbewegungen des Anwenders und erlaubt eine Rundumsicht. (Nunez, 2015)

Neben den Bewegungen der Brille werden zusätzlich auch andere Eingabegeräte, wie zum Beispiel die Handcontroller, vom Tracking miteingeschlossen. So können auch die Bewegungen der Hände mit in die virtuelle Szene übertragen werden. An den Geräten von Oculus sind hierzu an den Geräten Infrarot-LEDs angebracht, welche von speziellen Kameras erfasst werden (Hinweis 4 in Abbildung 7). HTC hingegen verbaut in ihre Geräte Infrarot-Photosensoren, die von zwei Basisstationen durch Aussenden von Infrarot-Laserstrahlen getrackt werden. Zusätzlich sind in den Geräten beider Hersteller ein Gyroskop, ein Beschleunigungssensor und ein Magnetometer verbaut, um eine optimale Positionserkennung zu ermöglichen. (Charara, 2017)

Anschluss:

Die aktuellen VR-Brillen sind standardmäßig über ein HDMI-Kabel und ein USB-Kabel mit dem Host PC verbunden, um die nötigen Daten empfangen zu können. Diese sind von den Herstellern als kombiniertes Kabel ausgeführt und bieten eine Länge von fünf Metern. Da die Signalstärke mit der Länge des Kabels abnimmt ist dieses, in Verbindung mit einer Signal-Verstärkerbox, auch nur um weitere fünf Meter verlängerbar, was den Arbeitsraum der Brillen merklich limitiert. Seit September 2018 bietet HTC auch eine kabellose Verbindungsmethode für ihre Vive Produkte an. Dabei nutzen sie Intels 60-GHz-WiGig-Technologie, um die anfallenden Datenmengen ohne spürbare Latenzzeit übertragen zu können. (Lang, 2018)

Die weiteren Eingabegeräte, wie Handcontroller oder Tracker, sind mittels Bluetooth mit dem Host-PC verbunden.

Latenz:

Unter Latenz versteht man die Zeitspanne, welche ein Gerät braucht, um auf bestimmte Eingaben zu reagieren. Diese Verzögerung kann auf VR bezogen durch Laufzeiten von Signalen in Kabeln, durch die Verarbeitung von Daten durch Algorithmen oder durch die Zeit bis zur nächsten Abtastung beim Tracking verursacht werden. Vor allem beim Head Tracking ist es unausweichlich, die Latenz so gering wie möglich zu halten, da die Verzögerung sonst beim Anwender Übelkeit verursachen kann. In Abschnitt 2.4.1 wird das Thema Latenz, bezogen auf das gesamte VR-System, nochmals betrachtet.

2.3.2 VR-Geräte bei Magna Steyr

Zur VR-Ausrüstung von Magna Steyr gehören verschiedene Geräte von den Herstellern HTC und Oculus, welche auch im Laufe der Arbeit verwendet wurden. In der folgenden Tabelle sind diese Geräte mit ihren technischen Eigenschaften angeführt. Zur Ergänzung ist die VR-Brille von VRgineers angeführt. Die Brille von VRgineers weist eine höhere Bildschirmauflösung und ein größeres Sichtfeld auf und ist mit den Peripheriegeräten von HTC kompatibel. Diese Eigenschaften brachten die Überlegung nahe, die Ausrüstung bei Magna Steyr in naher Zukunft um eine VRgineers XTAL zu erweitern. Zu der Größe der verbauten Bildschirme bzw. zur Pixeldichte gibt es leider keine Angaben von Seiten des Herstellers. Es gilt abzuwarten, ob die Qualität wirklich besser ist als bei den vorhandenen Konkurrenzprodukten.

Tabelle 1 VR-Geräte bei Magna Steyr

Hersteller	HTC	HTC	Oculus	VRgineers
Modell	Vive	Vive Pro	Rift	XTAL
Erscheinungsjahr	2016	2018	2016	2018
Display	2x OLED	2x OLED	2x OLED	2x OLED
Auflösung (pro Auge)	1080x1200	1440x1600	1080x1200	2560x1440
Pixeldichte	ca. 456 ppi	ca. 615 ppi	ca. 456 ppi	k.A.
Sichtfeld	110°	110°	110°	170°
Framerate	90 Hz	90 Hz	90 Hz	70 Hz



Abbildung 9 HTC Vive Pro (links, HTC) und Oculus Rift (Oculus)

Vor der Verwendung von HMDs zur Darstellung und Beurteilung der Anwendungsfälle etablierten sich große, stereoskopische und hochauflösende Displays, die sogenannten Powerwalls. Über Rückprojektion auf eine Leinwand wird dem Benutzer hierbei ein immersiver Eindruck vermittelt. Die Projektion wird mit Beamern durchgeführt. Magna Steyr verfügt am Standort Graz über einen eigens errichteten Showroom, welcher mit einer ca. 600 cm x 300 cm (Breite x Höhe) großen Powerwall ausgestattet ist. So ist es möglich VR-Szenarien auch für mehrere Benutzer immersiv darzustellen. Da jedoch der Grad der Immersion sehr gering ist, ist diese Darstellungsform nicht für alle Anwendungsfälle ausreichend geeignet.



Abbildung 10 Darstellung einer VR-Szene auf einer Powerwall (3Dims)

2.4 Das VR-System

Ein VR-System besteht immer aus mehreren Teilsystemen, um die geforderten Anforderungen als Mensch-Maschine-Schnittstelle bestmöglich abdecken zu können. Vereinfacht kann das System in drei Teilsysteme unterteilt werden – in die Eingabe, die Ausgabe und die Datenverarbeitung. In der Mitte kann der Nutzer gesehen werden, der mit allen Teilsystemen interagiert. In Abbildung 11 ist das System schematisch dargestellt:

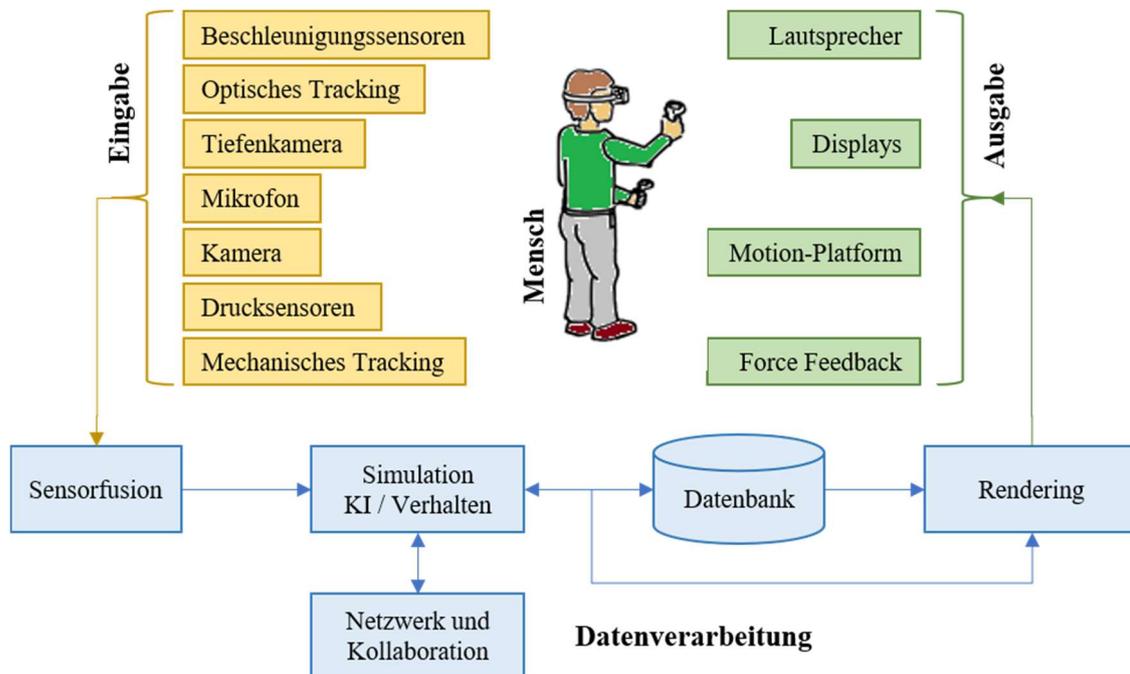


Abbildung 11 Schematische Darstellung eines VR-Systems (Dörner, 2013)

Die Eingabegeräte (hier Gelb dargestellt) sind für die Interaktion zwischen dem Menschen und der Maschine zuständig. Beim Aufbau des VR-Systems ist darauf zu achten, welche Aufgaben das System zu erfüllen hat, um die nötigen Eingabegeräte zu integrieren. Sind zu viele Eingabegeräte bzw. Sensorik in den Aufbau aufgenommen, könnte der Nutzer dadurch eingeschränkt und die aufgenommenen Daten nicht sinnvoll eingesetzt werden.

Die in Grün dargestellte Ausgabe dient dazu, dem Nutzer die virtuelle Welt so gut wie möglich darzustellen. Hier ist es sinnvoll möglichst viele unterschiedliche Wahrnehmungskanäle des Nutzers anzusprechen, um einen hohen Immersionsgrad zu erreichen. Aber auch hier gilt es herauszufinden, welche Ausgabegeräte für den jeweiligen Anwendungsfall überhaupt wichtig und von Nutzen sind, statt alles technologisch Mögliche umzusetzen.

Für die Datenverarbeitung (Blau dargestellt) dient ein leistungsstarker Host-PC, der für kollaborative Anwendungen auch mit dem Netzwerk verbunden sein kann. Dieser Host-PC verfügt über ein im Vorhinein erstelltes Modell der virtuellen Szene. Dieses Modell legt das Verhalten der virtuellen Realität fest, welches über die Eingabegeräte vom Nutzer beeinflusst werden kann. Interaktionen des Nutzers oder Einflüsse wie Lichtquellen, Spiegelungen und Schatten werden vom Computer in Echtzeit berechnet. Gegebenenfalls müssen hier Vereinfachungen vorgenommen werden, um Verzögerungen in der Berechnung in einem erträglichen Maß zu halten. Es kann auch notwendig sein, die einzelnen Rechenoperationen auf mehrere Computer zu verteilen. Bei sehr aufwendigen virtuellen Szenen werden oft sogenannte Rechen-Cluster eingesetzt. Im Folgenden Abschnitt werden diese Verzögerungen nochmals vertieft betrachtet.

2.4.1 Verzögerungen im VR-System

Eine Grundvoraussetzung an das VR-System ist seine Interaktivität. Für das immersive Empfinden des Nutzers ist es sehr wichtig, dass er die Konsequenzen seiner Handlungen in der virtuellen Welt wahrnehmen und seinem eigenen Handeln zuordnen kann. Dazu muss der Host-PC die Eingaben des Nutzers auswerten und auf die virtuelle Szene berechnen, um die Veränderungen für den Nutzer über die Ausgabe darstellen zu können. Diese Arbeitsschritte des VR-Systems erfolgen jedoch nicht ohne Verzögerungen. Die Zeitspanne, die das System für die Reaktion auf eine Eingabe des Nutzers benötigt, wird als Latenz bezeichnet. Je größer die Latenz des Systems ist, desto größer ist der zeitliche Abstand zwischen der Ausführung einer Handlung und der wahrnehmbaren Konsequenz in der virtuellen Szene. Eine besondere Anforderung an die Latenz im VR-System ist die Berechnung der dynamischen Perspektive des Nutzers, vor allem wenn eine VR-Brille mit Head-Tracking zum Einsatz kommt. Ist die Latenz hier zu hoch, so verändert sich die in der Brille angezeigte Perspektive merklich asynchron zu den Bewegungen des Nutzers, was bei vielen Personen schnell zu einem Schwindelgefühl oder Übelkeit führen kann. In verschiedenen Tests hat sich eine Latenz von unter 50 ms für die Anwendung mit VR-Brillen empfohlen (Brooks, 1999). Neben Latenzen beim Tracking können auch in Ausgabekanälen, wie zum Beispiel visuell, auditiv und haptisch, Latenzen entstehen. Diese werden vom Nutzer nicht unbedingt direkt wahrgenommen, können aber dennoch zu einem Unwohlsein führen. (Buhr, 2013)

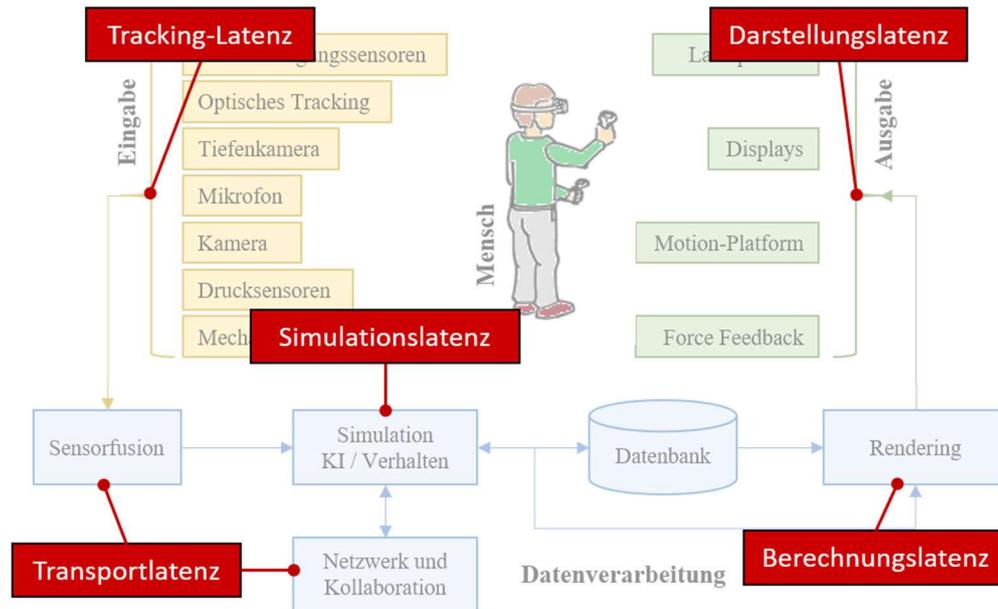


Abbildung 12 Latenzen im VR-System

Abbildung 12 zeigt anhand der schematischen Darstellung des VR-Systems, wo und welche Latenzen auftreten können. Bei der Eingabe kann die schon vorhin diskutierte Tracking-Latenz auftreten. Eine wichtige Aufgabe der Sensorfusion ist der Ausgleich bzw. die Berücksichtigung der Latenzunterschiede der einzelnen getrackten Eingabegeräte. Meist ist die Gesamtlatenz des Trackings durch den langsamsten Tracker im System bestimmt. Zwischen den einzelnen Komponenten der Teilsysteme kann es zu einer Transportlatenz kommen. In der Simulation werden die eingehenden Eingabebefehle verarbeitet und die Interaktionseffekte in die virtuelle Szene adaptiert. Die hier auftretenden Simulationslatenzen entstehen durch die hierzu erforderlichen Berechnungen und eventuelle Warteprozesse, wie zum Beispiel auf eingehende Tracking-Informationen. Nachdem der neue Zustand der virtuellen Szene simuliert wurde, müssen die erforderlichen Ausgaben, entsprechend dem jeweiligen Ausgabegerät, gerendert werden, wo es wiederum zu einer Berechnungslatenz kommt. Letztlich wird der neue Inhalt über die Ausgabegeräte dem Nutzer dargestellt (Darstellungslatenz). (Buhr, 2013)

Die Summe dieser einzelnen Latenzen ergibt die Gesamtlatenz des VR-Systems. Um ein interaktives VR-System betreiben zu können, ist es wichtig, alle möglichen Quellen von Verzögerungen zu kennen, um im Falle einer unzureichenden Darstellungsqualität darauf reagieren zu können.

2.5 Das VR-Team

Magna Steyr in Graz beschäftigt sich bereits seit geraumer Zeit mit Möglichkeiten, den Fahrzeugentwicklungsprozess durch den Einsatz von Virtual Reality Methoden zu optimieren. Für diese Aufgaben wurde im Firmenumfeld ein eigenes VR Team integriert. Dieses Team hat hierbei eine Fülle an Aufgabenbereichen zu erarbeiten. Abbildung 13 zeigt eine Übersicht über diese verschiedenen Aufgabenbereiche und erläutert die einzelnen Aufgaben der Mitarbeiter des VR Teams.

In der Mitte des Teams steht der Visualisierungs-Experte. Dieser Mitarbeiter verfügt über eine lange Arbeitserfahrung im Bereich der Visualisierung, welche es ihm ermöglicht, das VR-Team bestmöglich zu koordinieren. Des Weiteren ist er die erste Ansprechperson nach außen und repräsentiert die gesamte VR Abteilung gegenüber den anderen Fachabteilungen bzw. Kunden. Um ihn herum sind die verschiedenen Aufgabenbereiche angeordnet, welche von seinem Team betreut werden.

Visualisierungsarbeiten in Autodesk VRED

- Erstellung und Verwaltung von VR Applikationen und Templates
- Programmierung von Skripten zur VR Szenensteuerung
- Erstellung und Verwaltung von projektspezifischen Anwendungsfällen
- Erstellung von Animationen in der VR Szene
- Erstellung von Filmaufnahmen von VR Inhalten für Präsentationen
- Kontakt zu Autodesk bezüglich Updates und sonstigen Neuerungen

Visualisierungs- Datenmanagement

- Verwaltung der Visualisierungsbibliotheken
- Bereitstellung der Fahrzeugdateien
- Instandhaltung und sukzessive Verbesserung des Dateien Importprozesses
- Entwicklung eines automatisierten Updateprozesses
- Aufbau der Modellstruktur des virtuellen Fahrzeugs

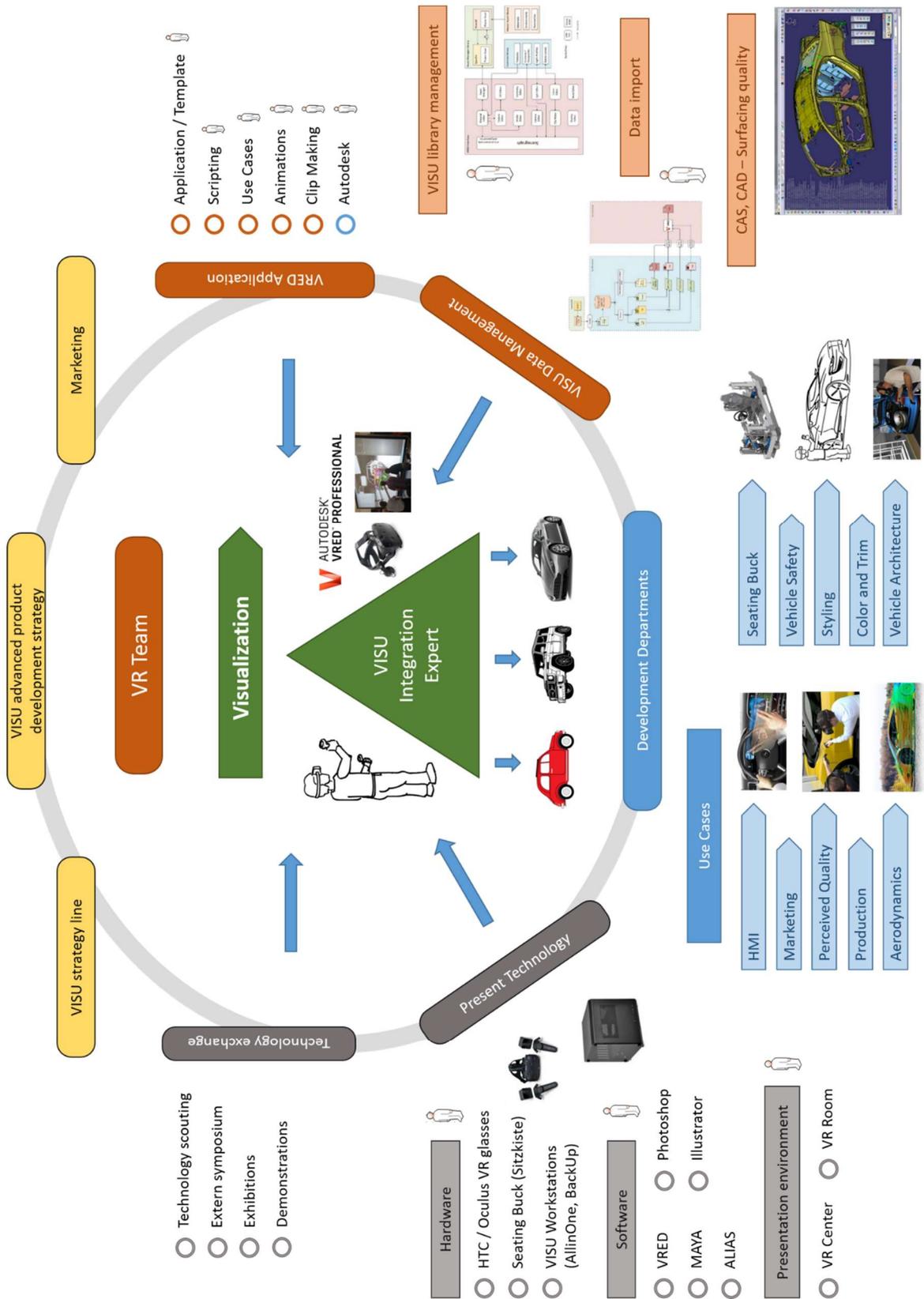


Abbildung 13 Aufgabenbereiche des VR Teams

Erhaltung des Arbeitsumfeldes

- Wartung der verwendeten Hardware
- Anpassung der Hardware an die steigenden Anforderungen
- Betreuung und Wartung der verwendeten Software
- Betreuung und Wartung der VR Infrastruktur und Präsentationsumfelds

Wissensaustausch

- Besuch von Messen und Konferenzen
- Durchführung von Demonstrationsveranstaltungen mit den Fachabteilungen

Die Ausarbeitung von Anwendungsfällen sollte in enger Zusammenarbeit mit den Entwicklungsfachabteilungen durchgeführt werden. So können die Anwendungsfälle optimal auf die Bedürfnisse der Anwender ausgelegt werden. Mittig in der unteren Bildhälfte sind die Entwicklungsfachabteilungen abgebildet.

Virtual Reality ist eine relativ neue Innovation im Automobilbau. Daher ist das Interesse daran dementsprechend hoch. Dazu steht der Leiter des VR Teams auch im engen Kontakt mit der Marketingabteilung des Unternehmens, um Messeauftritte und etwaige Werbemittel zu gestalten.

3 Der Fahrzeugentwicklungsprozess

Das folgende Kapitel befasst sich mit dem Produktentwicklungsprozess in der Automobilindustrie. Dabei handelt es sich um einen Prozess, in dem verschiedene Unternehmensbereiche unterschiedliche Aufgaben und Verantwortlichkeiten übernehmen. Dieser Prozess wird dauernd weiterentwickelt, um die bestmögliche Produktivität und Qualität bei der Entwicklung und Produktion eines Fahrzeugs zu erreichen. Der Ablauf des Prozesses ist durch mehrere Entwicklungsphasen definiert. Des Weiteren beinhaltet der Entwicklungsprozess festgelegte Meilensteine, zu denen sich die verschiedenen Unternehmensbereiche synchronisieren, um sicher zu gehen, dass bestimmte Entwicklungsziele erreicht wurden, um an der Entwicklung des Fahrzeugs weiterarbeiten zu können. Dabei werden voneinander unabhängige Vorgänge zur gleichen Zeit ausgeführt und abhängige Arbeitsvorgänge so weit wie möglich parallelisiert (Simultaneous Engineering). (Schäppi, 2005)

3.1 Phasen des Fahrzeugentwicklungsprozesses

Ein Fahrzeug durchläuft in seiner Entstehung bestimmte Planungs- und Entwicklungsphasen, welche in Abbildung 14 beispielhaft dargestellt sind.

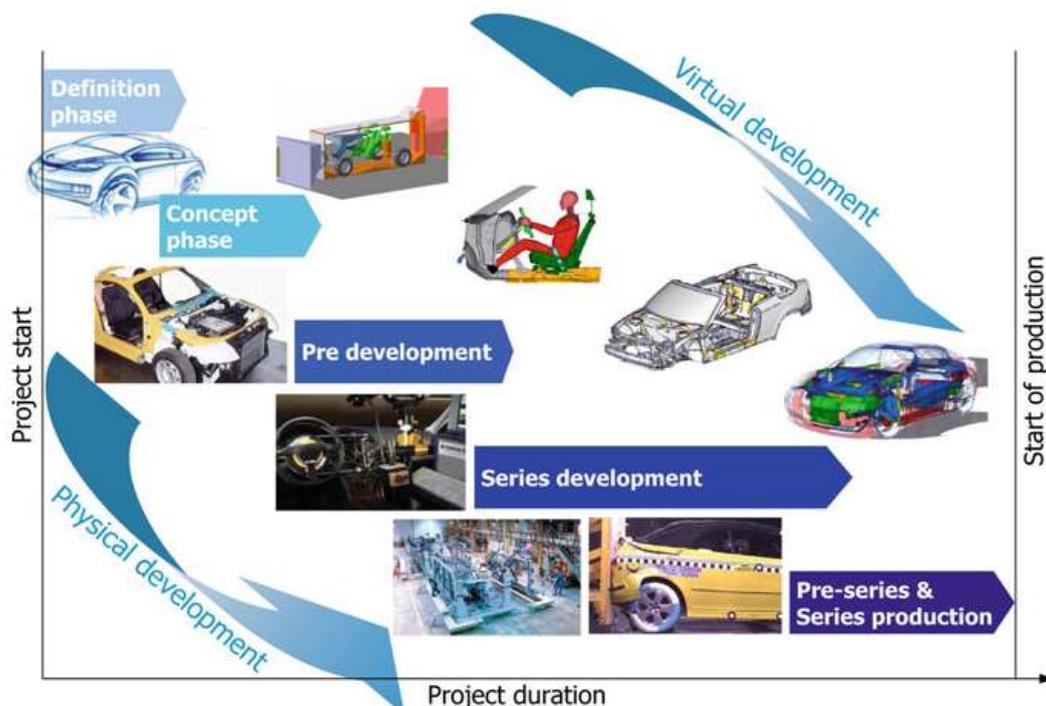


Abbildung 14 Phasen des Fahrzeugentwicklungsprozesses (Hirz, 2013)

Definitionsphase

Am Beginn der Fahrzeugentwicklung steht die Definitionsphase. In dieser Phase wird der Markt sondiert und eine Produktstrategie entwickelt. Hierbei werden die Anforderungen an das zu entwickelnde Fahrzeug definiert und in einem Zielkatalog zusammengefasst. Darunter fallen unter anderem die Definition des Fahrzeugtyps, Bestimmung des Zielmarktes, die groben Fahrzeugabmessungen, die Antriebsart, die Anzahl der Sitzplätze, das Fahrzeuggewicht aber auch Prozesseigenschaften wie der geplante Produktionsstart. Mit Hilfe einer Marktanalyse werden Kundenwünsche identifiziert, um eine Marktprognose abzugeben und die Bedürfnisse der zukünftigen Käufer befriedigen zu können. Des Weiteren werden in der Definitionsphase erste Designstudien angefertigt. (Gusig & Kurse, 2010)

Konzeptphase

In der Konzeptphase werden nun die in der Definitionsphase getroffenen Entscheidungen konkretisiert. Dabei müssen alle Produkteigenschaften so weit detailliert werden, dass alle Rahmenbedingungen für die späteren Entwicklungsphasen bestimmt sind. Da jedoch in dieser Phase viele Informationen noch nicht bekannt sind oder parallel dazu erstellt werden, muss häufig auf empirische Erfahrungswerte bzw. auf Daten von vorangegangenen Modellen zurückgegriffen oder es müssen Annahmen getroffen werden.

Während der Konzeptphase wird auch das Design des Fahrzeugs den technischen Anforderungen gerecht werdend weiterentwickelt. Zu diesem Zeitpunkt steht das Design bereits in Form von Geometriedaten zur Verfügung. Der Abgleich zwischen dem Design und den technischen Anforderungen wird auch als Design-Technik-Konvergenz (in Englisch *Styling-Technical-Convergence*, abgekürzt STC) bezeichnet. Dieser Vorgang dauert bis zum sogenannten *Styling Freeze*-Meilenstein am Ende der Konzeptphase, bei dem die Fahrzeuggestalt endgültig festgelegt wird. (Harrich, 2014)

Des Weiteren werden in der Konzeptphase Analysen von Konkurrenzprodukten durchgeführt, erste Verbindungen mit Lieferanten eingegangen und klare Zielkosten definiert. Am Ende der Konzeptphase liegt ein Fahrzeugkonzept vor, welches alle wesentlichen geometrischen und technischen Eigenschaften abbildet.

Vorentwicklungsphase

Die Aufgabe der Vorentwicklungsphase ist nun, die Lösungsvorschläge der Konzeptphase auszuarbeiten und bei Bedarf zu optimieren. Externe Entwicklungspartner und Lieferanten werden verstärkt mit in die Entwicklung eingebunden. In der Konstruktion werden Bauteile Bezug nehmend auf deren Produktion weiter detailliert und zeitgleich die Planung von Produktionsanlagen und Werkzeugen vorangetrieben.

In dieser Phase ist der Detailgrad der Geometriedaten schon so weit fortgeschritten, dass die Daten für virtuelle Entwicklungsmethoden verwendet werden können. So kann zu diesem Zeitpunkt zum Beispiel bereits das Crashverhalten des Fahrzeugs simuliert oder erste Qualitätsuntersuchungen am virtuellen Fahrzeug durchgeführt werden, obwohl noch keine Prototypen des Fahrzeugs aufgebaut wurden.

Mit dem Ende der Vorentwicklungsphase liegt ein detailliertes Fahrzeugkonzept vor und alle Anforderungen aus dem Lastenheft des Fahrzeugs wurden erfüllt. (Harrich, 2014)

Serienentwicklungsphase

In der Serienentwicklungsphase kommen nun auch Prototypenfahrzeuge zum Einsatz. Mit deren Hilfe werden die bereits in der Vorentwicklungsphase, durch den Einsatz virtueller Methoden, erhaltenen Ergebnisse überprüft und bei Bedarf Änderungen vorgenommen. Jedoch ist in dieser Entwicklungsphase die Möglichkeit für Veränderungen bereits sehr eingeschränkt bzw. mit sehr hohen Mehrkosten verbunden, da parallel bereits die notwendigen Produktionsanlagen entwickelt und gebaut werden. Des Weiteren können bereits kleinste Veränderungen an der Gestalt des Fahrzeugs einen Einfluss auf die geometrische Integration oder bereits festgelegte funktionale Eigenschaften haben. (Harrich, 2014)

Am Ende der Serienentwicklungsphase ist das Fahrzeug vollständig entwickelt und es folgt die Produktionsfreigabe.

Vorserien- und Serienproduktionsphase

Nun beginnt die Serienproduktion des Fahrzeugs. Als Vorserienproduktionsphase wird der Zeitraum bezeichnet, in dem die Produktion schrittweise hochgefahren wird, bis die geplante Auslastung erreicht ist. Die produzierten Fahrzeuge aus der Vorserienproduktion sind theoretisch bereits Kundenfahrzeuge, jedoch sind in der Praxis zu diesem Zeitpunkt noch nicht alle Entwicklungsziele erreicht. Dies führt zu Nacharbeiten und Ausbesserungen von

Schwachstellen, bevor das Fahrzeug verkauft werden kann. Aus diesem Grund wird die Vorserienproduktionsphase meist noch der Produktentwicklung zugeordnet und geht fließend in die Produktherstellung über.

3.2 Meilensteine

Wie bereits angeführt besteht der Fahrzeugentwicklungsprozess aus mehreren spezifischen Meilensteinen, um den Entwicklungsfortschritt zwischen den verschiedenen Unternehmensbereichen zu synchronisieren und so das gemeinsame Entwicklungsziel zu erreichen. Ein Meilenstein definiert hierbei einen Termin, an dem die erforderlichen Ergebnisse in vorgegebener Detaillierung und Vollständigkeit vorliegen müssen. Meilensteine befinden sich am Ende der einzelnen Entwicklungsphasen und können auf den gesamten Entwicklungsprozess oder aber auch nur auf eine einzelne Entwicklungsaufgaben bezogen sein. In Abbildung 15 ist ein exemplarischer Produktentstehungsprozess im Automobilbau mit seinen Phasen und Meilensteinen dargestellt.

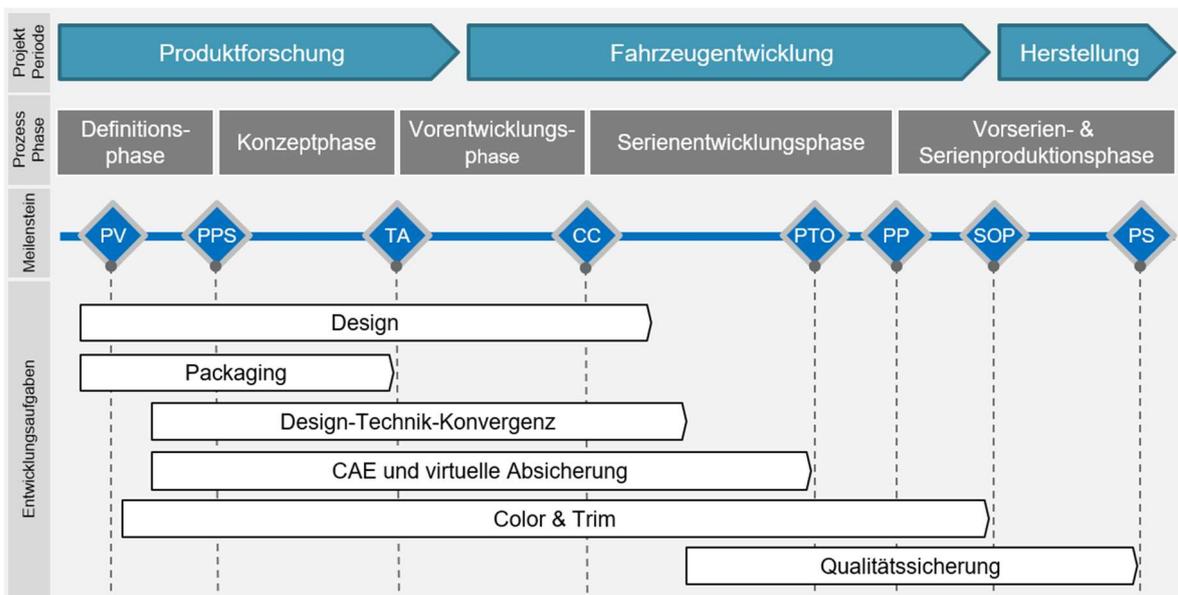


Abbildung 15 Schematisches Beispiel eines Fahrzeugentwicklungsprozesses

PV – Product vision

Der erste Meilenstein im Fahrzeugentwicklungsprozess ist der PV. Zu diesem Zeitpunkt ist die Innovations- und Strategiephase abgeschlossen und ein erster Zielkatalog liegt vor. Dieser Zielkatalog beinhaltet unter anderem eine ausreichende Beschreibung von Marketinganforderungen und Produktplatzierung, eine erste Festlegung vom Entwicklungsumfang und die wichtigsten Funktionsziele und Packageanforderungen des zu entwickelnden Fahrzeugs.

PPS – Preliminary product specification

Am Ende der Definitionsphase befindet sich der PPS Meilenstein. Der Zielkatalog wurde um Strategien zur Produktrealisierung aller Entwicklungsbereiche erweitert und alle Produktziele in einem Rahmenheft zusammengefasst. Des Weiteren liegen nun ein Businessplan, eine detaillierte Programmkostenschätzung und ein Finanzplan vor.

TA – Target agreement

Mit dem TA Meilenstein ist die Konzeptphase abgeschlossen. Das Konzept für das zu entwickelnde Fahrzeug ist nun vollständig definiert und die Aufteilung der Verantwortlichkeiten und der Arbeitsaufgaben auf die verschiedenen firmeninternen Abteilungen und externen Entwicklungspartner vergeben.

CC – Concept confirmation

Am Ende der Vorentwicklungsphase folgt die Konzeptbestätigung. Von den Abteilungen und Lieferanten wird bestätigt, dass sie ihre Projektziele erreichen und den Terminplan fristgerecht einhalten können.

PTO – Production try out

Zu diesem Zeitpunkt wurde das erste Versuchsfahrzeug auf der Serienfertigungsanlage montiert und für Versuchs- und Validierungszwecke in Betrieb genommen. Des Weiteren wurden alle Vorkehrungen bezüglich der Anlaufplanung getroffen und der Markteinführungstermin bestätigt.

PP – Pilot production

Wurde das erste Vorserienfahrzeug unter Serienbedingungen gefertigt und erfüllt auch die definierte Teile- und Produktionsqualität, so ist der PP Meilenstein erreicht. Damit ist die Serienentwicklungsphase abgeschlossen und es folgt die Freigabe der Vorserienproduktion.

SOP – Start of production

Mit dem SOP Meilenstein startet die Produktion und Auslieferung der ersten Kundenfahrzeuge. Die Qualität der Fahrzeuge entspricht den definierten Anforderungen.

PS – Process stability

Die Prozesssicherheit ist gegeben und die Erreichung der Produktentwicklungsziele bestätigt. Des Weiteren wurde eine ausführliche Bewertung der Teilprozesse durchgeführt, um den Ablauf des Entwicklungsprozesses sukzessive zu verbessern. Der Fahrzeugentwicklungsprozess ist mit diesem Meilenstein abgeschlossen.

3.3 Simultaneous Engineering

In den Anfängen der Produktentwicklung wurden die notwendigen Arbeitsschritte stets nach der Reihe abgearbeitet. Es wurde mit dem nächsten Arbeitsvorgang so lange gewartet, bis der vorherige vollständig abgeschlossen war. So begann der Entwicklungsprozess mit der Definition der Kundenanforderungen und Ideenauswahl, gefolgt von der Erstellung eines Konzepts und dessen Umsetzung in Form von Prototypen. Nach der darauffolgenden Testphase begann die Fertigungsplanung und schließlich der Start der Produktion. Diese Aneinanderreihung der Entwicklungsaufgaben führte zu langen Durchlaufzeiten und aufgrund der fehlenden Abstimmung zwischen den einzelnen Prozessschritten zu einem ineffizienten Ressourceneinsatz. (Burmester & Vahs, 2005)

Um diese ablaufbedingten Mängel zu beheben entwickelte sich sukzessive eine neue Vorgehensmethode – das Simultaneous Engineering. Hierbei werden die einzelnen Prozessschritte nicht mehr schrittweise ausgeführt, sondern so weit als möglich parallelisiert. Sobald in einem Einzelprozess genügend Informationen erarbeitet wurden, wird bereits mit dem nächsten Arbeitsvorgang begonnen und dadurch eine Verkürzung der Produktentwicklungs-

zeit erreicht (Abbildung 16). Neben der Zeitersparnis konnten auch eine Senkung der Entwicklungskosten und eine Verbesserung der Produktqualität, bezogen auf die Vorstellungen der Kunden, erzielt werden.

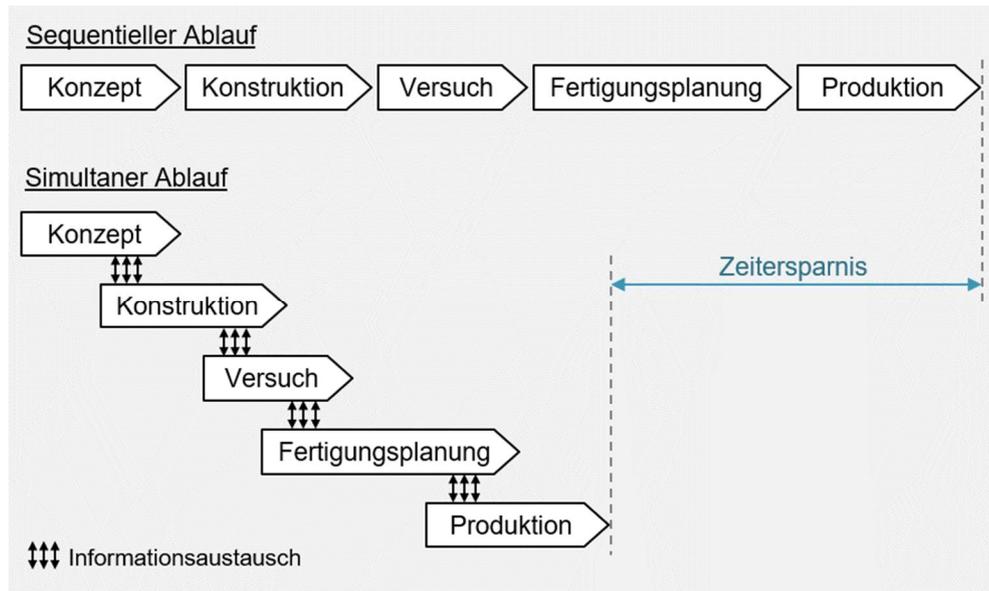


Abbildung 16 Zeitersparnis durch Simultaneous Engineering

Der vorzeitige Beginn von Arbeitsvorgängen kann jedoch auch teilweise zu Mehrarbeit führen, da zu diesem Zeitpunkt noch nicht mit dem endgültigen Informationsstand gearbeitet wird und sich dieser jederzeit verändern kann. Dafür können jedoch Fehler in der Entwicklung schneller erkannt und bereits in einer früheren Entwicklungsphase beseitigt werden.

4 Potential im Fahrzeugentwicklungsprozess

In diesem Kapitel betrachten wir nun, welches Potential der Einsatz von Virtual Reality im Fahrzeugentwicklungsprozess mit sich bringt. Welche Vor- und Nachteile verbergen sich im Einsatz dieser Technologie und wie kann die Zusammenarbeit zwischen einer Fachabteilung und dem VR-Team, welches im folgenden Kapitel genauer beschrieben wird, bestmöglich funktionieren.

Virtuelle Anwendungen sind schon seit längerer Zeit ein fester Bestandteil im Fahrzeugentwicklungsprozess. So erfolgt zum Beispiel die Konstruktion der Fahrzeuge in modernen CAD-Programmen. Die daraus entstehenden Geometriedaten können in CAE-Anwendungen weiterverwendet und Arbeitsfolgen wie Crash Tests oder Fußgängerschutz bis hin zu Aerodynamikuntersuchungen virtuell simuliert werden. Die grundlegenden Simulationstechniken haben sich hierbei in den letzten Jahren zu einem prognosesicheren und leistungsfähigen Auslegungswerkzeug entwickelt. Dank dieser virtuellen Werkzeuge kann heute in der Entwicklung auf ganze Prototypenbaureihen verzichtet werden, was eine große Kostenersparnis mit sich bringt. (Schulte-Frankenfeld, Brass, & Pieck, 2007)

Virtual Reality kann nun jedoch nicht als eigenständiges Entwicklungswerkzeug verstanden werden, sondern ist eine Methode, um die Interaktion zwischen dem Anwender und der Maschine (in diesem Fall das eigentliche virtuelle Entwicklungswerkzeug) zu verbessern. Durch die Verwendung einer VR-Brille erhält der Anwender ein vollwertiges Raumgefühl von dem Fahrzeug und kann dadurch am virtuellen Fahrzeug Analysen und Bewertungen durchführen, welche an einem normalen Bildschirm nicht möglich wären. Dazu muss jedoch das virtuelle Entwicklungswerkzeug über die Möglichkeit einer VR-Ausgabe verfügen bzw. der benötigte Anwendungsfall von einem Visualisierungsteam aufbereitet werden. Die Erstellung einer VR-Szene aus den vorhandenen Fahrzeugdaten und die Implementierung des Anwendungsfalles birgt jedoch einen großen Arbeitsaufwand und setzt eine enge Zusammenarbeit zwischen dem Visualisierungsteam und der Entwicklungsfachabteilung voraus.

4.1 Der Use Case - Anwendungsfall

4.1.1 Allgemeine Erläuterung

Ein Use Case (zu Deutsch Anwendungsfall) dokumentiert die Funktionalität eines Systems mit Hilfe eines einfachen Modells. In einem Use Case wird das nach außen sichtbare Verhalten eines Prozessablaufs aus Sicht des Nutzers beschrieben, um die folgende Durchführung zu erleichtern bzw. ohne größere Zwischenfälle durchzuführen. Die Aufgaben des Nutzers werden darin genau abgebildet, welcher hierbei als Person, eine Rolle oder als ein anderes System in Interaktion treten kann. Die Erstellung eines Use Case ist relativ unkompliziert, erleichtert das gemeinsame Verständnis des Vorhabens zwischen Nutzer und System jedoch enorm, was den Use Case in der Produktentwicklung und Softwareentwicklung sehr beliebt macht (Sparx Systems, 2018).

Die Stärke eines Use Case ist es, einen guten Überblick über das Vorhaben bzw. das Verhalten eines Systems zu geben. Des Weiteren dient der Use Case als Wegweiser, um das gewünschte Ziel eines Anwenders bestmöglich zu definieren. Zusätzliche Anforderungen an das System, mögliche Schwierigkeiten und Unterbrechungen im Entwicklungsprozess sollen somit so früh wie möglich erkannt werden. Je später dies geschieht, umso teurer und problematischer wird der Entwicklungsprozess.

4.1.2 Erstellung eines Use Case

Bevor man mit der Erstellung eines Use Cases beginnt, sollte man sich die Frage stellen, was das geplante System bzw. Vorhaben leisten soll, um nicht während der Erstellung im Detail zu versinken und dadurch das Wesentliche zu vergessen. Dabei können folgende Fragestellungen hilfreich sein (microTOOL, 2018):

Fragen zum Akteur:

- Wer nutzt das System?
- Welche anderen Systeme interagieren mit dem System?
- Wer oder was liefert dem System Informationen?
- Was ist das Ziel des Akteurs?
- Welche Aktionen muss der Akteur durchführen, um sein Ziel zu erreichen?

Fragen zum Auslöser:

- Welches Event löst den Use Case aus?
- Welcher Akteur initiiert den Use Case?

Fragen zu den Vorbedingungen:

- Welche Bedingung muss erfüllt sein, damit der Anwendungsfall eintritt?
- In welchem Zustand muss sich das System befinden, damit der Use Case eintritt?

Fragen zu den Nachbedingungen:

- Welche Bedingung muss erfüllt sein, dass der Use Case abgeschlossen wird?
- In welchem Zustand muss sich das System befinden, so dass der Use Case abgeschlossen wird?

Fragen zum Standardablauf:

- Wie interagiert der Akteur mit dem System?
- Was regiert das System darauf (bietet Optionen an, zeigt Informationen an, führt einen Prozess aus, etc.)?

Fragen zum alternativen Ablauf:

- Was passiert, wenn ...?
- Welche alternativen Aktionen kann der Akteur bei jedem Schritt durchführen?

Fragen zu Ausnahmen:

- Welche Fehler oder Unterbrechungen können bei jedem Schritt des Anwendungsfall auftreten?
- Was soll passieren, wenn der Akteur den Vorgang abbricht?

Um all diese Fragen zusammenzufassen, wurde ein interaktives Word - Dokument erstellt, welches vor der Erstellung des Use Case im Zuge einer Vorbesprechung zwischen dem Nutzer (Kunde bzw. Entwicklungsfachabteilung) und einem Mitglied des Visualisierungsteams gemeinsam ausgearbeitet wird.

4.1.3 Das Use Case Diagramm

Ergänzend zu einem Use Case kann ein Use Case-Diagramm (Anwendungsfalldiagramm) erstellt werden, welches einen grafischen Überblick über die Funktionalität des Systems, den Beziehungen von Komponenten untereinander und die Beziehung des Systems zu dessen Umgebung ermöglicht. Dabei wird jedoch nicht jeder Arbeitsschritt im Detail erläutert, sondern eine Art „Black Box“-Sicht auf das Gesamtsystem gewährt, um die Grundabläufe besser zu verstehen. Es ist nicht unüblich, dass ein Use Case Gesamtsystem aus vielen kleineren Use Cases besteht (Jeckle, Rupp, Hahn, & Zengler, 2003).

Die Hauptbausteine in einem Use Diagramm sind das „System“, der „Use Case“ und der „Akteur“, welcher immer außerhalb des Systems auftritt. Des Weiteren werden die Beziehungen zwischen Akteur und Use Case, der Akteure untereinander und der Use Cases untereinander dargestellt.

Im Standardfall werden Use Cases durch eine Ellipse dargestellt. Der Name des Use Case wird inner- oder unterhalb der Ellipse notiert (Abbildung 17). Die Benennung erfolgt hierbei immer aus der Sicht des Akteurs, nicht aus Sicht des Systems. Die Notation kann aber auch in Form eines Rechtecks erfolgen. Dies ist praktischer, wenn man dem Use Case Attribute oder Operationen zuordnen will.

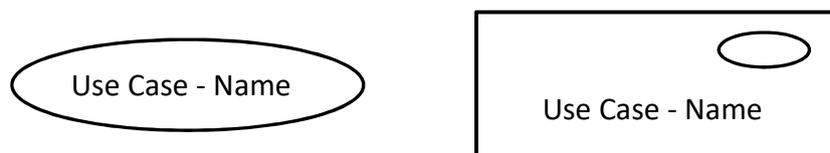


Abbildung 17 Notation eines Use Case

Um einen Use Case zu starten, muss er von einem Akteur ausgelöst werden. Dies kann als Person selbst, über einen Trigger oder durch das Anklicken eines Menüpunktes erfolgen. Die Norm (UML – Unified Modeling Language) gibt als Notation für den Akteur eine Strichfigur vor (Abbildung 18). Falls jedoch kein menschlicher Akteur gemeint ist, kann alternativ ein Rechteck mit der Bezeichnung „Akteur“ verwendet werden.

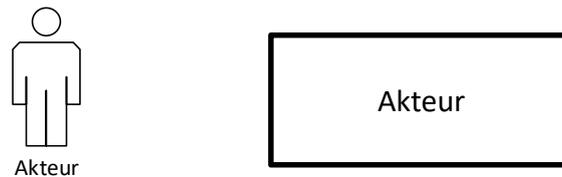


Abbildung 18 Notation eines Akteurs

Die Use Cases und Akteure stehen in Beziehungen zueinander, welche mit Linien dargestellt werden (Abbildung 19). Im Grunde soll ein Use Case Diagramm keine Navigation durch das Vorhaben darstellen, jedoch ist es erlaubt gerichtete Assoziationen einzufügen. Diese werden durch einen Richtungspfeil dargestellt. Die gerichteten Assoziationen bedeuten nun jedoch keine Navigationsvorgabe, sondern beschreiben, welcher Teil des Systems der Aktive und welcher der Passive ist. Wenn ein Akteur zu einem Use Case geführt wird, so ist der Akteur der Aktive, welcher den Vorgang auslöst. Im umgekehrten Fall, also wenn die gerichtete Assoziation vom Use Case zum Akteur führt, so wird der Akteur vom Use Case benötigt und aufgefordert teilzunehmen (Jeckle, Rupp, Hahn, & Zengler, 2003).

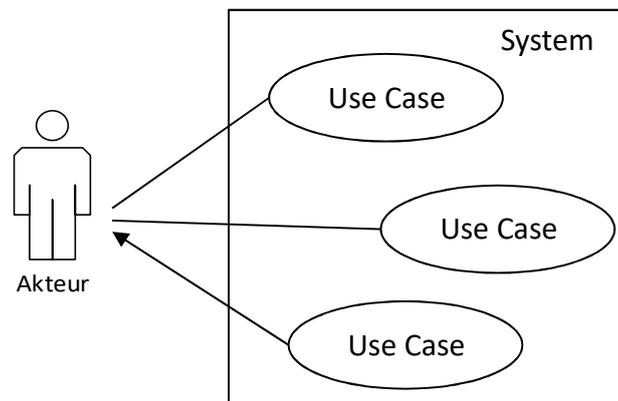


Abbildung 19 Assoziationen eines Use Case

Zusätzlich zu den Beziehungen zwischen Akteur und Use Case können auch zwei oder mehrere Use Cases zueinander in einer Beziehung stehen. Hierbei wird unterschieden, ob der weitere in Beziehung stehende Use Case den ursprünglichen „erweitert“ oder ob der ursprüngliche Use Case den zusätzlichen Use Case „enthält“. Bei einer Erweiterung, also wenn ein Use Case unter bestimmten Umständen durch einen Zweiten erweitert wird, wird dies mit einem strichlierten Pfeil mit dem Zusatz „extend“ dargestellt (Abbildung 20). Die Pfeilspitze zeigt hierbei auf den Use Case, der erweitert wird. Ist ein Use Case in einem Zweiten enthalten, also ein fester Bestandteil davon, so wird dies mit einem strichlierten Pfeil mit

dem Zusatz „include“ gekennzeichnet. In diesem Fall zeigt die Pfeilspitze auf den enthaltenen Use Case.

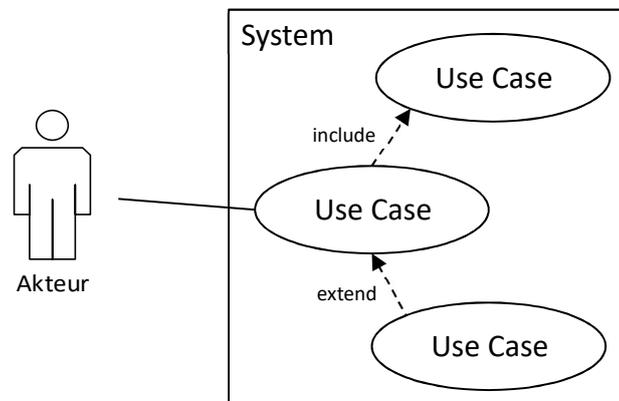


Abbildung 20 Beziehungen zwischen Use Cases

Des Weiteren ist es möglich, die Beziehungen zwischen einem Akteur und einem Use Case mit Multiplikatoren zu versehen (Abbildung 21). Auf der Seite des Use Cases gibt der Multiplikator an, wie oft dieser vom Akteur zeitgleich ausgeführt werden kann/darf. Wenn keine Angabe gemacht wird, ist der Zusatz immer 0..1. Auf Seiten des Akteurs bedeutet es, wie viele Akteure beteiligt sein müssen, um den Use Case durchführen zu können. Ohne Angabe ist es immer 1..1, bzw. vereinfacht 1 (Sparx Systems, 2018).

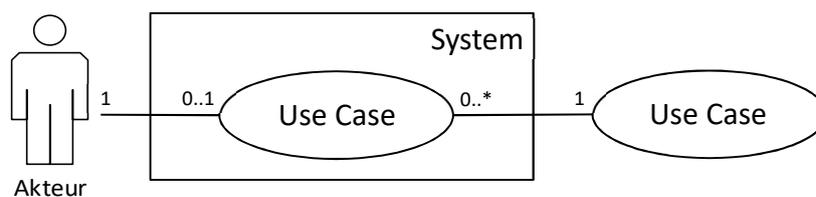


Abbildung 21 Multiplikatoren an einem Use Case

4.2 Anwendungsbezogene Use Cases

In diesem Kapitel wird näher auf Use Cases, die sich auf die Fachbereiche des Automobilbaus beziehen, eingegangen. Dabei wird ein größerer Fokus auf den Fahrzeugentwicklungsprozess gelegt und diskutiert, welche Bereiche ein hohes Potential, bezugnehmend auf die Prozessoptimierung, für eine VR-Einbindung aufweisen.

Die Fachbereiche im Automobilbau sind sehr weitschichtig und erstrecken sich in viele verschiedene Berufsgruppen. Nicht in jedem Fachbereich macht es daher Sinn, Abläufe durch eine VR-Anbindung zu optimieren, da schlicht kein Potential vorhanden ist bzw. der nötige Aufwand die Summe aller Erleichterungen in der späteren Anwendung übersteigen würde. Die folgenden Punkte sollen zeigen, in welchen Fachbereichen des Automobilbaus sehr wohl ein großes Potential für eine VR-Einbindung vorhanden ist:

Marketing

Im Marketing werden Virtual Reality Lösungen bereits von vielen Automobilherstellern eingesetzt. Hersteller wie der Volkswagen Konzern bieten bereits Fahrzeug-Konfiguratoren an, welche über eine VR-Schnittstelle verfügen. (La Rocco, 2016)

So werden zum Beispiel Autohäuser mit einem VR-System ausgestattet, um den Kunden sein individuell konfiguriertes Fahrzeug realitätsnah präsentieren zu können. (Audi AG, 2017)

Dokumentation

Hinsichtlich der Dokumentation des Fahrzeugs werden bereits Augmented Reality Lösungen angewendet. Hyundai präsentierte im Jahr 2016 eine Smartphone Applikation, welche als virtuelle Bedienungsanleitung fungiert. Die App nutzt die eingebaute Kamera des Smartphones, um die Komponenten des Fahrzeugs zu erkennen und fügt den Komponenten am Bildschirm Beschriftungen an, auf denen erklärt wird, was man sieht. Zusätzlich verfügt die App über ein Lernprogramm, in welchem dem Nutzer aktiv erklärt wird, wie man zum Beispiel den Ölstand kontrolliert oder den Luftfilter wechselt. (Gitlin, 2016)

Fabrikplanung, Anlagenprojektierung

Auch in der Fabrikplanung werden bereits VR-Systeme eingesetzt. So können virtuelle Techniken in der Detailplanung den Arbeitsablauf gut unterstützen, wenn sie zur Absicherung der Fördertechnik, Montage, Automatisierungstechnik, Robotik und Fertigungsverfahren durch Visualisierung und Simulation genutzt werden. Des Weiteren ist es möglich, vorab einen virtuellen Rundgang durch die geplante Fabrik zu unternehmen, um so die Arbeitsplatzgestaltung und die Planung der Montagesysteme realitätsnah zu prüfen.

In der Logistik werden AR-Lösungen angewendet, um den Arbeitern bei ihren Tätigkeiten zu unterstützen. (Hartmann, 2017)

Styling

Ein sehr großes Potential haben Virtual Reality Anwendungen im Styling. Durch den Einsatz einer VR-Brille kann das Fahrzeug bereits zu einem sehr frühen Entwicklungszeitpunkt realitätsnah betrachtet und damit auch bereits erste Designbewertungen durchgeführt werden. Dies vermindert die Entwicklungszeit und spart Kosten und Ressourcen, da in diesem Entwicklungsschritt weitestgehend auf physische Prototypen verzichtet werden kann. Hinzukommend ist durch die digitale Aufbereitung des Fahrzeugs immer der aktuellste Entwicklungsstand verfügbar.

Des Weiteren ist es möglich, das Fahrzeug virtuell zu skalieren. Clay-Modelle können somit in einem kleineren Maßstab erzeugt und nach dem Einscannen der Konturen virtuell vergrößert werden, um eine realitätsnahe Bewertung durchzuführen.

CAE

Virtuelle CAE-Entwicklungsmethoden gehören mittlerweile zu den Standardwerkzeugen in der Fahrzeugentwicklung. So werden zum Beispiel Mehrkörpersystem-Simulationen (MKS) eingesetzt, um einen Crash-Test oder die Kinematik mechanischer Systeme, wie das Fahrwerk, zu simulieren. Finite Elemente Methoden (FEM) werden eingesetzt um Spannungs- und Steifigkeitsanalysen an hoch belasteten Komponenten durchzuführen und Computational Fluid Dynamics (CFD) Anwendungen kommen zum Einsatz, um das Verhalten von Gasen, Flüssigkeiten oder Gemischen unterschiedlicher Fluide zu bestimmen. So können die Luftführungen im Fahrzeuginnenraum bis hin zur Aerodynamik des Gesamtfahrzeuges am Computer berechnet werden. Des Weiteren ist es möglich, die einzelnen Anwendungen zu

kombinieren, um die Aussagegenauigkeit der Simulationsergebnisse zu erhöhen. Dies ist bei vernetzten Aufgabenstellungen, wie zum Beispiel der Simulation der Bremskühlung oder Fahrdynamikuntersuchungen unter Seitenwindeinfluss, besonders wichtig. (Deuschl, 2006)

Wie bereits erwähnt, bietet hier VR jedoch kein vollwertiges Werkzeug, sondern kann nur zur Präsentation der Ergebnisse herangezogen werden. Es obliegt hierbei der Entscheidung des Entwicklungsingenieurs, ob er das Ergebnis bevorzugt an einem konventionellen Computerdisplay oder interaktiv mit einer VR-Brille betrachtet. Diese Entscheidung ist unter anderem auch stark von der Art des Anwendungsfalls abhängig. So ist es nicht sonderlich hilfreich, die Ergebnisse einer FEM Spannungsanalyse mit der VR-Brille zu betrachten, wobei es jedoch einen großen Mehrwert darstellen kann, das Ergebnis einer CFD Strömungsanalyse am Gesamtfahrzeug realitätsnah und interaktiv am virtuellen Fahrzeugmodell zu betrachten. Virtuelle Vorentwicklungsprogramme, wie BETA CAE Systems ANSA, bieten zum Beispiel die Möglichkeit mittels der VR Brille direkt in einem virtuellen Windkanal an der Strömungsanalyse teilzunehmen und selbständig die Stromlinien an das virtuelle Fahrzeugmodell zu legen (Abbildung 22). (BETA CAE Systems, 2018)

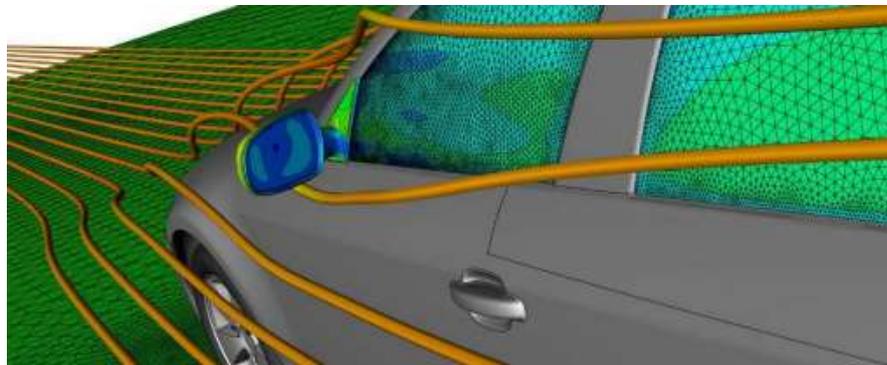


Abbildung 22 VR-Darstellung eines CFD Ergebnisses in BETA CAE Systems ANSA

HMI

Ebenso ein großes Potential haben VR Anwendungen bei HMI (Human Machine Interface) Untersuchungen. Mit Hilfe eines des virtuellen Fahrzeugmodells können bereits in einer frühen Entwicklungsphase Ergonomie-Untersuchungen durchgeführt werden, zu deren Zeitpunkt noch kein physisches Modell verfügbar ist. Des Weiteren können die Bedienbarkeit und Ablesbarkeit der Bedienelemente im Fahrzeuginnenraum überprüft werden. Um den

Realitätsgrad noch weiter zu erhöhen, wird bei vielen Untersuchungen eine sogenannte Sitzkiste eingesetzt (Abbildung 23). Hierbei handelt es sich um eine Nachbildung der Fahrgastzelle, um den Nutzer bestmöglich mit dem virtuellen Modell verschmelzen zu lassen. Die Komponenten der Sitzkiste, wie zum Beispiel der Sitz, das Lenkrad oder die Lage und Winkel der Armaturen, sind frei verstellbar und können dem Fahrzeug angepasst werden. Zusätzlich ist die Sitzkiste mit Sensoren und Trackern ausgestattet, um die Interaktionen des Nutzers auf das virtuelle Modell zu übertragen und für die Auswertung aufzuzeichnen.



Abbildung 23 HMI Untersuchung mit Hilfe einer Sitzkiste (nViz, 2018)

Neben Ergonomie-Untersuchungen kann auch eine Testfahrt mit dem virtuellen Fahrzeug ermöglicht werden. Dabei können äußere Einflüsse, wie störende Spiegelungen an den Fahrzeugkomponenten oder die Wahrnehmung und Einsicht der Umgebung während der Fahrt, beurteilt werden.

Qualitätsbewertung – Perceived Quality

Ein ebenfalls sehr hohes Potential hat die Einbindung von VR Ausgabegeräten in der Qualitätsbewertung. Mit Hilfe von Raytrace Rendermethoden können fotorealistische Abbildungen des virtuellen Fahrzeugmodells, mit einer realitätsnahen Schattenberechnung, erstellt werden, welche zur Beurteilung der allgemeinen Erscheinung des Fahrzeugs oder der Bewertung von Spaltmaßen bzw. Toleranzketten der Anbauteile herangezogen werden können. Bei einem sehr performanten Modell und genügend Rechenleistung des VR-Systems ist sogar Echtzeit-Raytracing möglich. Hierbei kann der Anwender vollständig in eine fotorealistische VR-Szene eintauchen und seine Bewertungsarbeit wie an einem physischen Fahrzeugmodell durchführen.

4.3 Hürden der Integration

Virtuelle Methoden sind schon lange im Fahrzeugentwicklungsprozess angekommen und ein fester Bestandteil der Fahrzeugentwicklung. Nur die Aufbereitung deren Ergebnisse für VR-Anwendungen hinkt immer noch stark hinterher, obwohl es technisch bereits für viele Anwendungsfälle in ausreichender Qualität möglich wäre, wie Martin Rademacher in einer empirischen Studie im Laufe seiner Dissertation aufzeigt. (Rademacher, 2014)

Aber warum werden VR-Lösungen noch in so wenigen Abteilungen eingesetzt? Um diese Frage zu beantworten führte ich im Laufe meiner Arbeit bei Magna Steyr Gespräche mit mehreren Mitarbeitern verschiedener Entwicklungsabteilungen. Aus diesen Gesprächen lassen sich folgende Aspekte zusammenfassen, warum in den Fachabteilungen eine Skepsis gegenüber VR-Lösungen vorherrscht:

- Kein bzw. noch zu wenig Vertrauen in die aktuelle Technik und Qualität der VR-Ausgabegeräte.
- Die Fachabteilungen sind vereinzelt nicht gewillt, neue Methoden in ihren Arbeitsprozess zu integrieren. Sie belassen es lieber bei Altvertrautem und wollen keinen wirklichen Nutzen in den neuen Methoden sehen.
- Überreaktion der Testperson, hervorgerufen durch die Variantenvielfalt. Es wird befürchtet, dass die Beurteilungszeit einer Variante zu kurz geraten könnte, um die richtigen Entscheidungen zu treffen, da man verleitet wird zu schnell zwischen den Varianten zu wechseln.

Wie man sieht resultiert die größte Zurückhaltung der Entwicklungsabteilungen durch eine schlechte Informationspolitik innerhalb des Betriebs. Um diese zu verbessern werden Vorführveranstaltungen abgehalten, um den Mitarbeitern die bestehenden Möglichkeiten und die Infrastruktur näher zu bringen. Dazu wurde eine Demo-VR Szene erstellt, in der die wichtigsten Funktionen, anhand eines bestehenden Modells, vorgezeigt und von den Mitarbeitern selbst interaktiv getestet werden können. Als VR-Arbeitsumgebung dient das Visualisierungs-Template, welches mit dem Fahrzeugmodell des Magna Steyr Concept Car Mila Plus befüllt ist. Die Funktionen und Eigenschaften des Visualisierungs-Template werden in Kapitel 6 genauer beschrieben.

Um eine optimale Unterstützung der Entwicklungsfachabteilung durch das VR-Team zu ermöglichen ist es wichtig, dass jeweils ein Mitarbeiter der Abteilung das Thema VR aktiv vorantreibt und Ideen für Anwendungsfälle und neuen Funktionen miteinbringt. So kann die VR-Szene genau den Anforderungen der Abteilung entsprechend erstellt werden.

5 Der Virtual Reality Integrationsprozess

Mit dem erhaltenen Wissen aus vergangenen Arbeiten der VR-Abteilung und den Besprechungen mit den Mitarbeitern der verschiedenen Entwicklungsabteilungen wurde ein gesamter Prozess zur Erarbeitung einer VR-Szene erstellt. In diesem Kapitel wird dieser Prozess nun näher beschrieben.

5.1 Workflow zur Erstellung einer VR-Szene

Zu Beginn wurde ein Ablauf zur Erstellung einer VR-Szene festgelegt, welcher in Form eines BPMN-Workflows detailliert abgebildet wurde. In diesem Workflow ist abgebildet, welche Aufgaben die einzelnen Mitarbeiter der Abteilungen und des VR-Teams erfüllen müssen, um das bestmögliche Ergebnis für beide Seiten zu erreichen.

Das *Business Process Model and Notation* (abgekürzt BPMN, deutsch Geschäftsprozessmodell und -notation) ist ein grafisches Werkzeug, welches häufig im Prozessmanagement angewendet wird. Es stellt genormte Symbole zur Verfügung, mit denen Geschäftsprozesse und Arbeitsabläufe erstellt und dokumentiert werden können. (Allweyer, 2015)

Mit den sogenannten *Pools* werden die wesentlichen Prozessbeteiligten bzw. die beteiligten Organisationen dargestellt. Ein *Pool* kann zusätzlich in mehrere *Lanes* unterteilt werden, die eine Benutzerrolle oder einen Aufgabenbereich innerhalb der Organisation repräsentieren. In unserem Fall besteht das erste Pool aus der Entwicklungsabteilung, welches sich in eine Bahn für die Mitarbeiter und eine weitere für das Datenmanagement der Abteilung unterteilt. Das zweite Pool repräsentiert das VR-Team. Dieses Pool unterteilt sich zusätzlich in die Projekt-Kommunikation, die Datenaufbereitung, die Visualisierung der VR-Szene und die Anwendung der visualisierten Szene. Des Weiteren repräsentiert ein drittes Pool das Produktdatenmanagement (abgekürzt PDM), über welches die Beschaffung und Verwaltung der benötigten Daten durchgeführt wird.

Da der detaillierte Workflow sehr viel Platz einnimmt, wurde er dieser Arbeit in den Beiblättern angefügt. Zur Beschreibung des Arbeitsablaufes ist in Abbildung 24 eine vereinfachte Form des Workflows dargestellt.

Zu Beginn der Zusammenarbeit an einer VR-Szene steht das erste Treffen. In diesem Meeting wird der Entwicklungsabteilung (im weiteren Verlauf allgemein Kunde genannt) die bereits erwähnte Demo VR-Szene präsentiert, dass sich der Kunde ein Bild über die Funktionen und Darstellungsqualität der VR-Lösungen machen kann. Mit diesem Wissen wird mit dem Kunden ein Anforderungskatalog erarbeitet, welcher den Inhalt der VR-Szene, bezogen

auf den gewünschten Anwendungsfall, festlegt. So werden im Anforderungskatalog unter anderem der Darstellungs- und Detailgrad des Fahrzeugmodells definiert, die Umgebungen der VR-Szene ausgewählt, eine Auswahl der integrierten interaktiven Werkzeuge getroffen und festgelegt, ob es sich um einen einmaligen oder wiederholenden Anwendungsfall handelt. Des Weiteren wird ein Drehbuch ausgearbeitet, um einen reibungslosen Ablauf der anstehenden VR-Untersuchungen zu gewährleisten und bereits festgelegte Beobachtungspunkte (Viewpoints) vorab in die VR-Szene integrieren zu können.

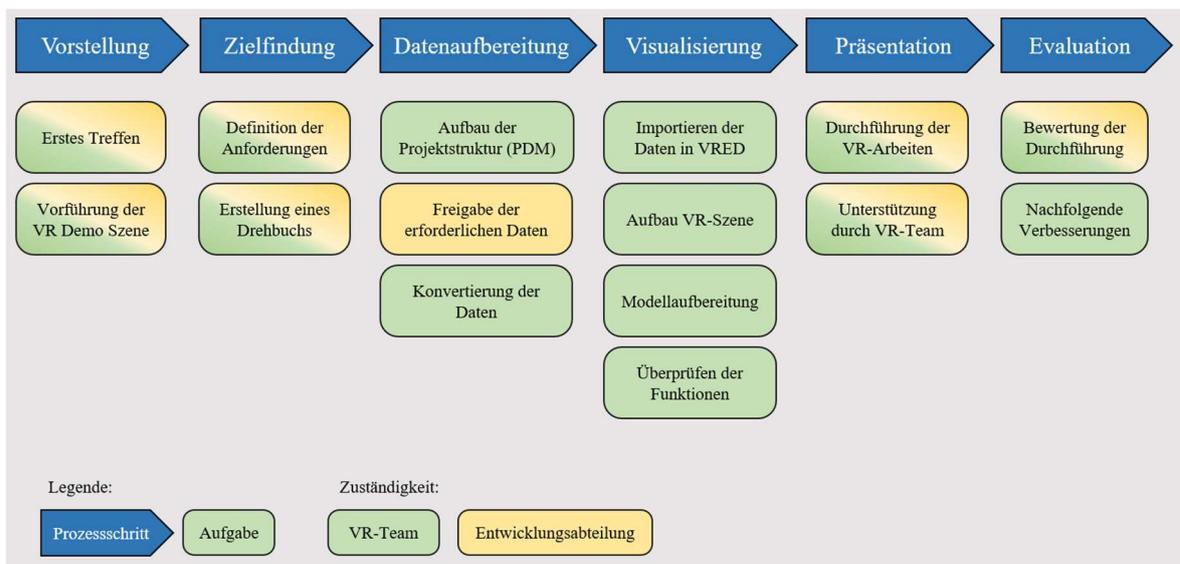


Abbildung 24 Vereinfachte Darstellung des Workflows

Sobald die Festlegung der Anforderungen abgeschlossen ist, wird mit dem Aufbau der Projektstruktur im PDM-System begonnen. Ist bereits eine Struktur vorhanden, welche auch verwendet werden kann, muss keine neue Struktur angelegt werden. Diese Struktur wird mit den Konstruktionsdaten des Fahrzeugmodells befüllt und dient als Vorlage für den späteren Datenexport. Alternativ kann der Umfang der Fahrzeugdaten auch mit dem Siemens Teamcenter Visualization Mockup festgelegt und hieraus exportiert werden. Da hierbei jedoch kein automatischer Daten Exportprozess vorliegt, ist diese Variante nur für einmalige VR-Untersuchungen sinnvoll. Nach dem Export aus dem PDM werden die Daten in das für die Visualisierungssoftware Autodesk VRED konforme Dateiformat konvertiert.

Liegen alle Geometriedaten des Fahrzeugs vor, so können diese in die Visualisierungssoftware importiert und es kann mit der Modellaufbereitung begonnen werden. Die Modellaufbereitung stellt hierbei den größten Arbeitsaufwand dar und wird üblicherweise von einem

Visualisierungsexperten durchgeführt. Dabei werden den Geometrieflächen des Fahrzeugmodells Materialien zugewiesen und bei Bedarf Komponenten, wie zum Beispiel Türen und Klappen, animiert. In Abbildung 25 ist das vollständig visualisierte Modell des Magna Steyr Concept Car Mila Plus dargestellt. Zeitgleich kann mit dem Aufbau bzw. der Anpassung der VR-Szene begonnen werden, in welche das visualisierte Fahrzeugmodell eingefügt wird. Nach der Überprüfung aller Funktionen durch das VR-Team können die Arbeiten des Kunden durchgeführt werden. In einem abschließenden Meeting werden die Zusammenarbeit und das Ergebnis evaluiert und die Erkenntnisse in den Prozess übernommen, um das beste Ergebnis für beide Seiten zu erreichen.



Abbildung 25 Magna Steyr Concept Car Mila Plus

5.2 Bereitstellung und Aufbereitung der Dateien

Bevor mit der Erstellung einer VR-Szene begonnen werden kann, müssen zuerst die CAD-Dateien des Fahrzeugmodells bereitgestellt werden. Im Laufe der Entwicklung durchläuft das Fahrzeug verschiedene Designvarianten, welche sich stark durch ihren Detailgrad unterscheiden. Zu Beginn dieses Kapitels werden die verschiedenen Varianten im Fahrzeugentwicklungsprozess aufgezeigt und genauer erläutert, um einen Weg zu finden die verschiedenen Datensätze nachvollziehbar in den Integrationsprozess einzubinden. Neben der Nachvollziehbarkeit der Dateien spielt die Effektivität des Prozesses eine große Rolle. Durch die, hervorgerufen durch den steigenden Detailgrad des Fahrzeugs, immer größer werdenden Dateimengen kommt es unweigerlich zu langen Import- und Ladezeiten, welche durch einen effektiven Einsatz des vorhandenen Arbeitsumfelds so gering wie möglich gehalten werden sollten. Ein weiteres Ziel ist es, einen automatischen Updateprozess in den Gesamtprozess zu integrieren, auf welchen im Laufe und speziell am Ende des Kapitels näher eingegangen wird.

5.2.1 Modellvarianten im Fahrzeugentwicklungsprozess

Wie bereits erwähnt durchläuft das Fahrzeugmodell im Laufe seiner Entwicklung mehrere Designphasen (Abbildung 26). Die jeweiligen Phasen unterscheiden sich grundsätzlich durch den Detailgrad der Darstellung und somit durch die Anzahl der einzelnen Fahrzeugbauteile. So wird das erste virtuelle Modell zumeist durch einen Scan eines realen Tonmodells oder durch Konstruktion mit Freiformflächen erzeugt, wodurch das Modell nur aus wenigen einzelnen Flächen besteht. Am Ende des Designprozesses ist das virtuelle Fahrzeugmodell vollständig, mit allen Bauteilen und Elementen, konstruiert und besteht somit aus mehreren tausend Einzelteilen. Im folgenden Abschnitt werden die einzelnen Schritte im Designprozess näher beschrieben. Dabei wird näher auf die Meilensteine im Designprozess eingegangen, welche den Prozess an sich und das Zusammenspiel mit der technischen Entwicklung beschreiben.

SB – Styling Briefing

Das *Styling Briefing* stellt den Start des Ideenentstehungsprozesses dar. Inhalt des Briefings sind die Gestaltung eines Lastenhefts mit Fahrzeugdimensionen und -varianten, die Definition der Kundenzielgruppe und die Beschreibung der Styling Philosophie. In dieser Phase entstehen erste zweidimensionale Darstellungen des Fahrzeugs.

In den folgenden Abschnitten (G3D – Go for 3D, GHM – Go for Hard Model, TR – Theme Reduction, TLM – Technical Lead Model) kommt es zu einem Wechsel von der zwei- zur dreidimensionalen Erarbeitung der virtuellen Modelle. Des Weiteren kommt es zu einer Bewertung der Designvorschläge und einer Eingrenzung dieser, bis es letztendlich zu einer ersten Festlegung der Bedingungen und somit zum ersten Engineering Daten Freeze kommt.

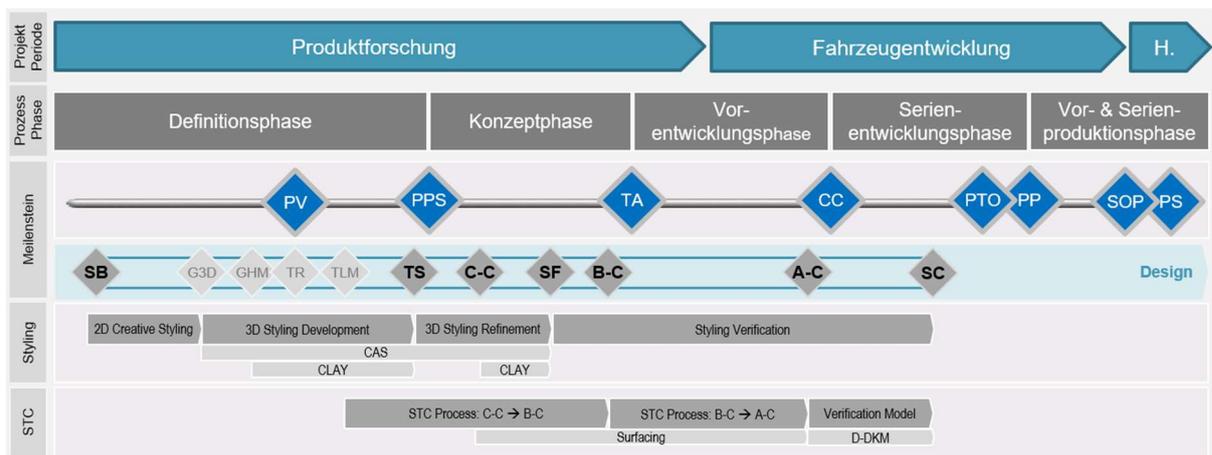


Abbildung 26 Designprozess in der Fahrzeugentwicklung

TS – Theme Selection

Ab diesem Zeitpunkt sind erste digitale Abbildungen (DMU – Digital Mock Up), in Form von Geometrieflächen, vorhanden. Diese Geometrieflächen werden mittels der CAD-Software Autodesk Alias erstellt und eignen sich für die Weiterverwendung in Visualisierungen, was eine Unterstützung der Fachabteilungen durch virtuelle Fahrzeugmodelle ermöglicht. In dieser Entwicklungsphase werden die aus Alias Flächen bestehenden CAD Modelle als 3D-CAS Modelle bezeichnet. Bei diesen Modellen geschieht bereits eine Unterteilung in Exterieur, Interieur und die technischen Komponenten. Eine Unterteilung in einzelne Fahrzeugkomponenten des Exterieurs oder Interieurs findet jedoch noch nicht statt.

C-C – C-Class

Bis zum *C-Class* Meilenstein wird die Konstruktion des virtuellen Fahrzeugmodells rasch vorangetrieben. Es erfolgt eine Aufteilung der Modellstruktur in die Einzelkomponenten des Fahrzeugs und diese werden gekennzeichnet durch eine ID- und Revisionsnummer der Daten Verwaltungsplattform (PDM) hinzugefügt. Dadurch ist ein Export der Daten aus der

verwendeten PDM Software Siemens Teamcenter (Siemens) möglich. Ab diesem Meilenstein unterliegen die Arbeiten dem Design-Technik-Konvergenz Prozess (abgekürzt STC, da in Englisch übersetzt *Styling Technical Convergence*).

SF – Styling Freeze

Zu diesem Zeitpunkt sind alle Designarbeiten abgeschlossen und keine weiteren Änderungen an der Form des Fahrzeugs mehr möglich.

B-C – B-Class

Mit Beendigung der Arbeiten für den *B-Class* Meilenstein wurde der Detailgrad des Fahrzeugmodells weiter verfeinert und Details wie Fugen dem Modell hinzugefügt. Die Anzahl der einzelnen Fahrzeugteile erhöht sich dadurch beträchtlich.

A-C – A-Class

Mit dem Erreichen des *A-Class* Meilensteins ist das virtuelle Fahrzeugmodell vollständig konstruiert. Es sind alle Einzelbauteile in der Fahrzeugstruktur vorhanden und mit all deren Details dargestellt. Des Weiteren sind alle Oberflächen für Exterieur und Interieur im PDM für den Export freigegeben.

SC – Styling Confirmation

In der *Styling Confirmation* wird das erbrachte Designkonzept bestätigt und durch das Management freigegeben.

Die für die Visualisierung relevanten Modellvarianten sind hierbei das sehr früh in der Entwicklung vorhandene CAS-Alias Modell und die im späteren STC Prozess vorliegenden Fahrzeugmodelle der C-, B- und A-Klasse. Wie bereits erwähnt, besteht das CAS-Modell nur aus jeweils einer Datei für das Exterieur und das Interieur des Fahrzeugs, die auch nicht im PDM abgelegt sind. Die STC-Modelle hingegen bestehen aus mehreren Dateien, welche die Einzelteile des Fahrzeugmodells abbilden; diese sind im PDM erfasst. Dies hat einen

erheblichen Einfluss auf die Datenbereitstellung und das spätere Versionsmanagement des Datenaufbereitungsprozesses.

Das folgende Bild (Abbildung 27) veranschaulicht die Unterteilung in die relevanten Modellvarianten und zeigt den stetigen Anstieg des Komponenten- bzw. Dateiumfangs.

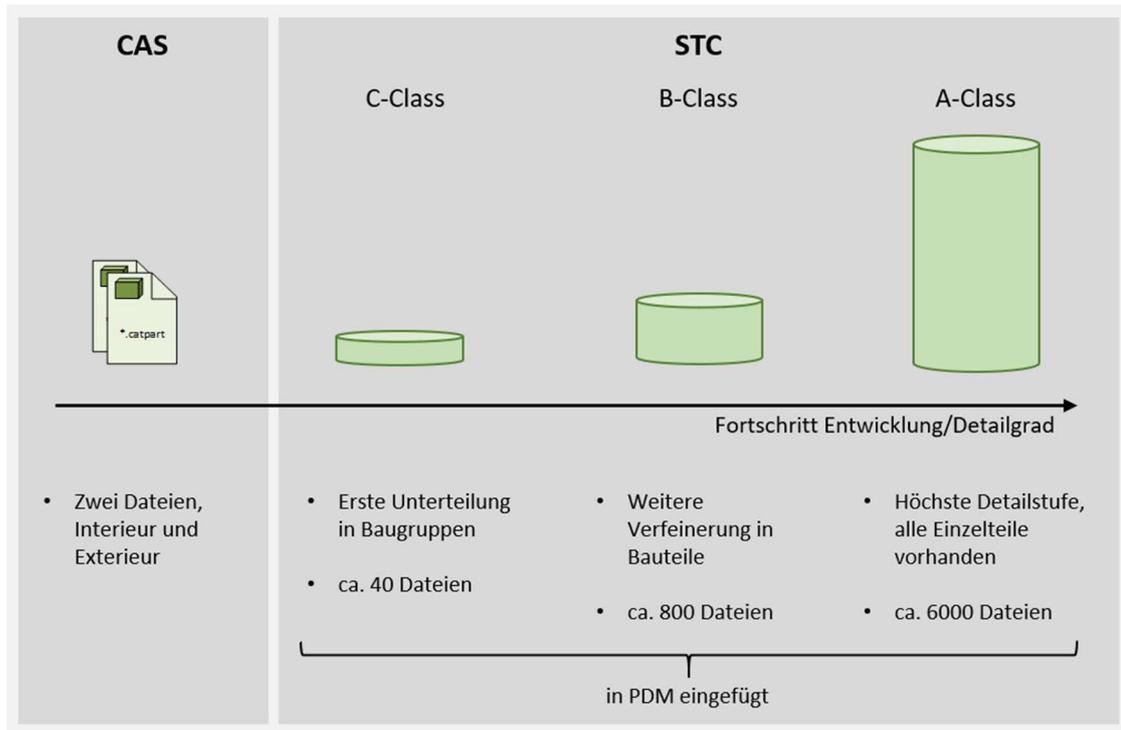


Abbildung 27 Datenentwicklung der Modellvarianten

5.2.2 Erstellung der Modellstruktur

Die PDM Software Siemens Teamcenter (TC) ermöglicht es die CAD-Dateien zu verwalten und für das Fahrzeugmodell eine frei definierbare Bauteilstruktur anzulegen. Dies ermöglicht dem Nutzer die Struktur genau nach seinen Bedürfnissen zu gestalten und diese auch dementsprechend für seine Anwendungen zu exportieren. Der genaue Exportvorgang wird im nächsten Abschnitt näher beschrieben. Wird ein Bauteil ins PDM aufgenommen, bekommt es als Namen eine projektbezogene Identifikationsnummer mit einem Zusatz, welcher die Version und Revision des Bauteils abbildet. Dadurch sind die Bauteile exakt definiert und die verschiedenen Versionsstände über festgelegte Laderegeln explizit abrufbar. Die Erstellung einer Struktur erfolgt über die Definition von Knoten (Items), welche sich automatisch mit der dazugehörigen Datei verknüpfen, sobald diese vorhanden sind. Beim

Ladevorgang wird über diese Knoten die durch die Laderegeln festgelegte Version des Bauteils geladen. Abbildung 28 zeigt schematisch den Aufbau des PDM Systems mit verschiedenen Strukturvarianten.

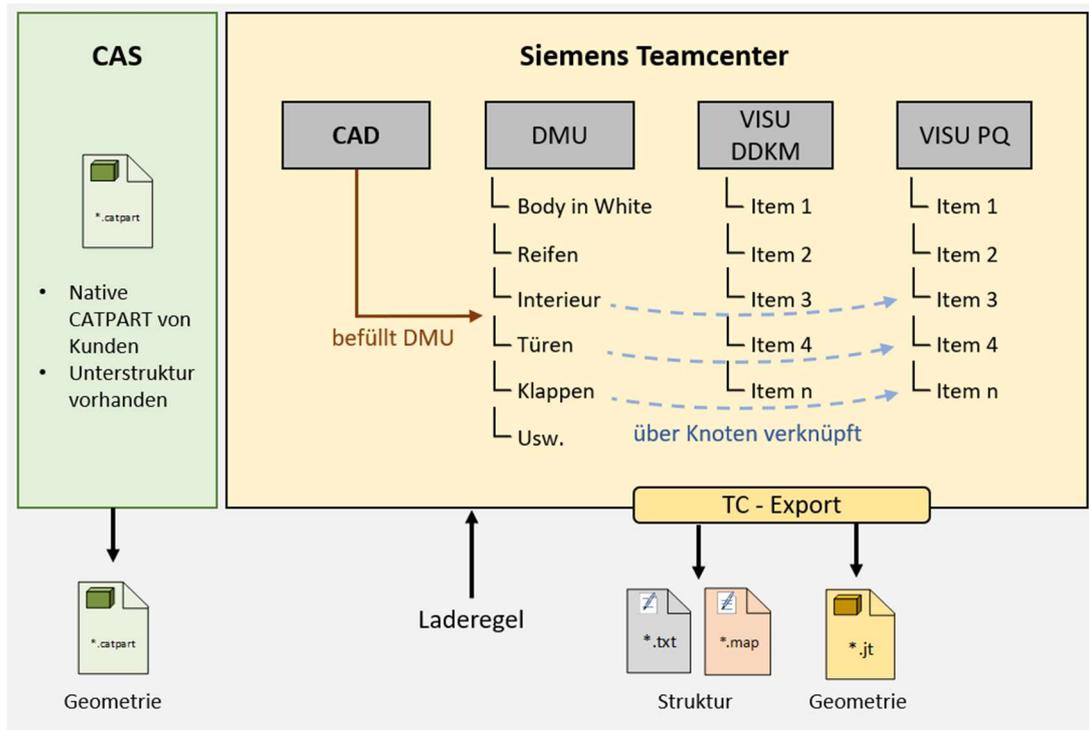


Abbildung 28 Schematischer Aufbau des PDM Systems

Das DMU-Modell stellt hierbei die Gesamtstruktur des virtuellen Fahrzeugmodells dar, welches mit den Dateien aus der Konstruktion (CAD) befüllt wird. Die VISU DDKM (Visualisierung – Digitales Datenkontrollmodell) und VISU PQ (Visualisierung - Perceived Quality) Strukturen sind anwendungsbezogene Fahrzeugstrukturen und sind wie bereits erwähnt über die Item-Knoten mit den Konstruktionsdateien verknüpft.

Bei der Ausarbeitung eines Anwendungsfalls und der damit verbundenen Erstellung eines visualisierten virtuellen Fahrzeugmodells können diese Strukturen nach Wunsch erstellt oder bereits vorhandene Strukturen übernommen werden.

5.2.3 Handhabung der Geometriedateiformate und Dateienexport

Aus prozessstrategischen Gründen wird bei Magna Steyr das JT Dateiformat (Siemens, 2015) favorisiert für den Datenaustausch von Geometriedateien verwendet. Das bedeutet, dass alle Konstruktionsdateien bei der Einbindung ins PDM System automatisch in dieses Format konvertiert und somit auch als dieses Format exportiert werden. Als Konverter dient hierbei der 3D-Evolution JT Konverter der Firma Core Technologie (Core Technologie). Dies führt jedoch zu Problemen bei den Dateien der frühen CAS-Modelle, wodurch der Prozess in die Datenversorgung der CAS-Modelle und der Modelle des STC Prozesses unterteilt werden musste.

CAS-Prozess Modelle

In dieser frühen Designphase besteht das virtuelle Fahrzeugmodell ausschließlich aus Freiformflächen, welche von Designern angefertigt und in den meisten Fällen als einzelne Dassault CATIA Geometriedatei (*.catpart) gespeichert werden (Dassault Systemes). Hierbei werden die einzelnen Baugruppen des Fahrzeugs als Geometrische Sets im Strukturbaum des CATIA Parts angelegt, wodurch diese auch eindeutig definiert sind. Würde man nun diese Datei in das PDM aufnehmen, geht diese Struktur jedoch leider verloren, da der JT-Konverter nicht in der Lage ist, diese Struktur aus einer Vielzahl an Geometrischen Sets in der konvertierten JT Datei abzubilden. Neben der Struktur gehen ebenso die Benennungen der einzelnen Baugruppen und Geometrieflächen verloren und werden durch den Dateinamen ersetzt. Die ursprüngliche Dateistruktur und das Ergebnis nach der Konvertierung sind in Abbildung 29 dargestellt. Die konvertierte Geometriedatei ist somit für unsere Vorhaben nicht verwendbar.

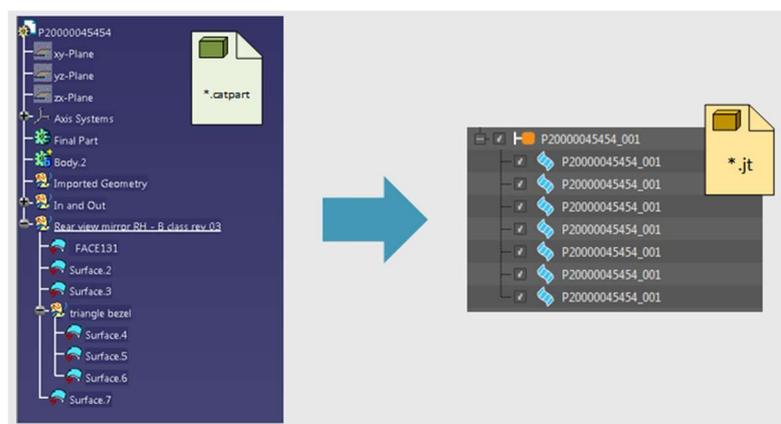


Abbildung 29 Dateistruktur vor und nach der Konvertierung in JT

Unter diesen Voraussetzungen kamen wir zu dem Entschluss, dass wir in dieser frühen Phase die Geometriedaten nicht in das PDM System aufnehmen und für die Datenaufbereitung die nativen Geometriedaten des Kunden bzw. der Designabteilung heranziehen.

Um einen automatischen Updateprozess der einzelnen Komponenten zu ermöglichen, gibt es Pläne ein Tool zu integrieren (Abbildung 30), welches die Geometriedatei in ihre einzelnen Geometrischen Sets aufteilt und deren Benennung übernimmt. Zusätzlich erstellt es eine PLMXML- Textdatei, in welche die Struktur der ursprünglichen Datei abgebildet wird, um diese später mit den aufgeteilten Geometriedateien wiederherstellen zu können. Da es sich jedoch in dieser Designphase noch um sehr einfache Modelle handelt und dadurch ein automatischer Updateprozess zu keiner großen Effizienzsteigerung führen würde, wurde bisweilen auf die Programmierung und Implementierung dieses „Part to Product“ Tools verzichtet.

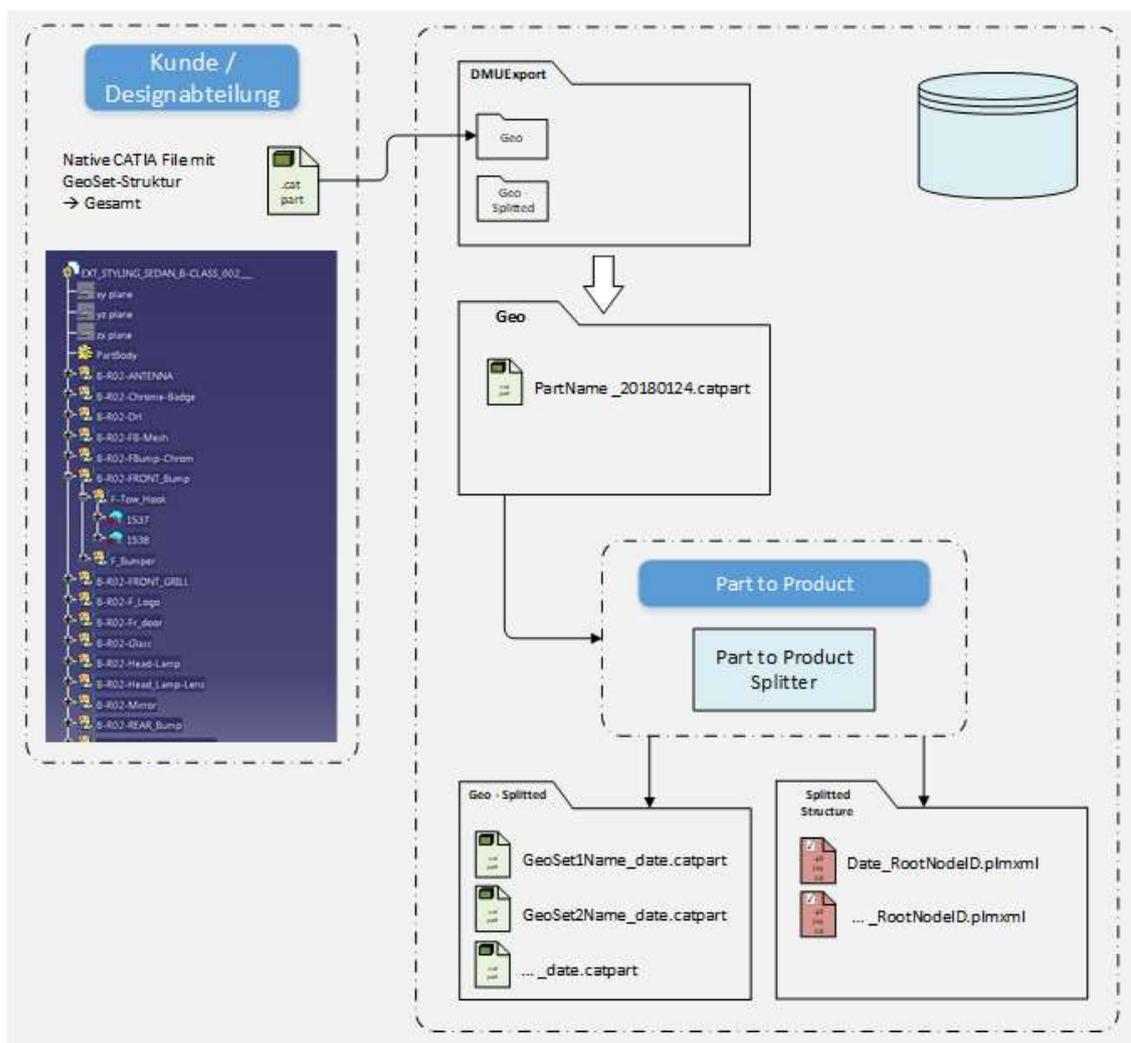


Abbildung 30 Dateien Aufbereitungsprozess CAS

CAD/STC-Prozess Modelle

Die Geometriedaten der späteren Designphasen werden von den Konstrukteuren in das PDM eingefügt und können mittels einer festgelegten Laderegeln geladen und daraus exportiert werden. Bei fortlaufenden Visualisierungen, bei denen das virtuelle Modell immer den aktuellsten Stand aufweisen soll, wird ein automatisierter Export, der täglich am Abend stattfindet, durchgeführt.

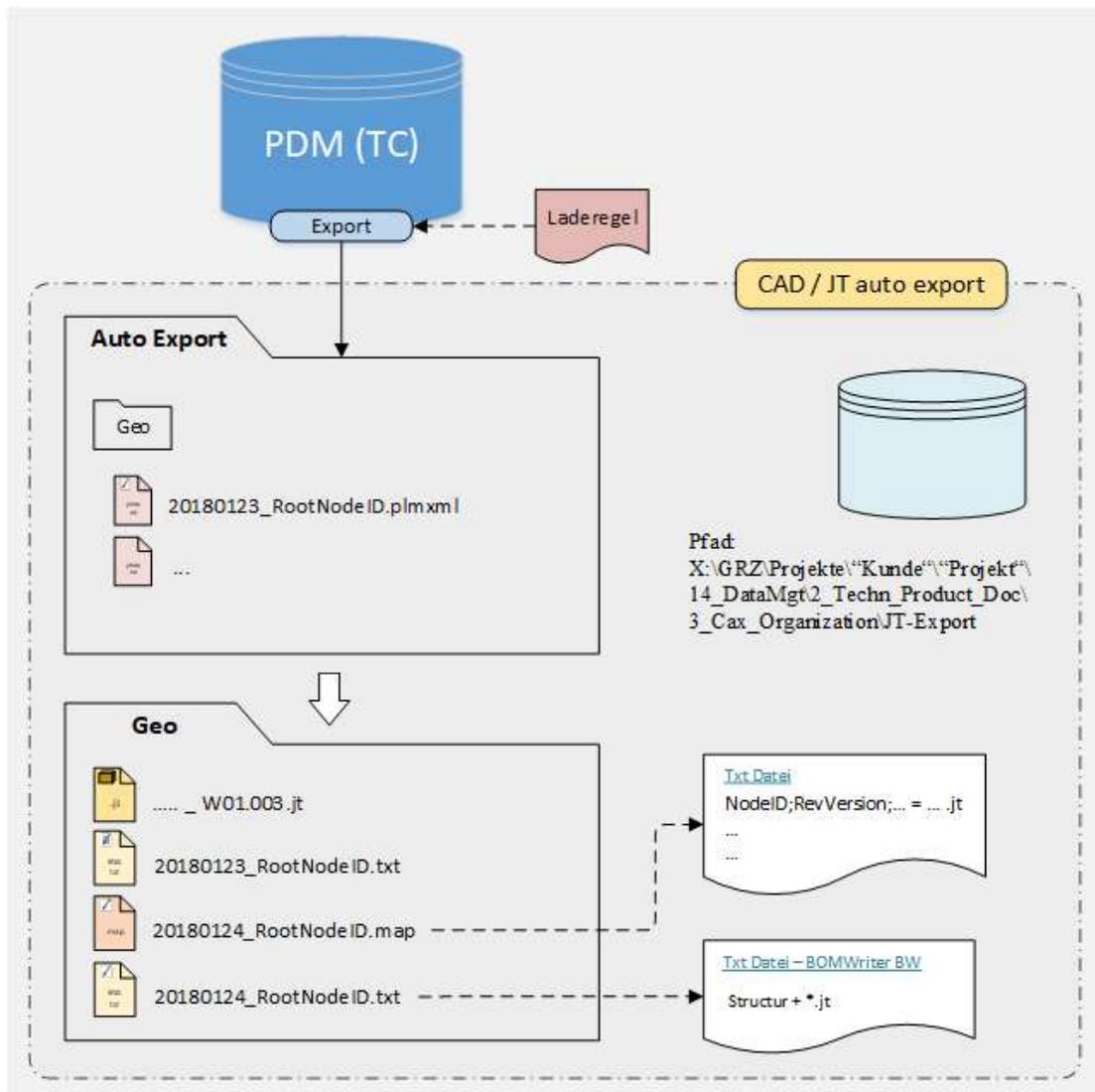


Abbildung 31 Dateien Aufbereitungsprozess CAD

Die Dateien werden in einen, in den Exporteinstellungen definierten, Ordner exportiert. Dafür wird eine festgelegte Ordnerstruktur verwendet, welche in Abbildung 31 dargestellt ist. Neben den JT Geometriedateien werden auch zusätzlich zwei weitere Textdateien exportiert, welche Metadaten der einzelnen Komponenten und Strukturdaten des Fahrzeugmodells aus

dem PDM beinhalten. Zur Identifikation werden die Textdateien im Namen mit dem Datum des Exports gekennzeichnet. Diese Textdateien können jedoch nicht von der verwendeten Visualisierungssoftware gelesen werden, was eine Konvertierung in ein Autodesk VRED unterstütztes Dateiformat nötig macht. Dazu wurde ein Tool geschrieben und in den Prozess implementiert, welches aus den beiden Textdateien eine VRED konforme PLMXML-Datei erstellt (Abbildung 32). Das PLMXML-Format ist ein gängiges Textdateiformat, welches von vielen Softwareanbietern, unter anderem auch von Siemens und Autodesk, als Standardformat zur Abbildung von Baugruppenstrukturen verwendet wird.

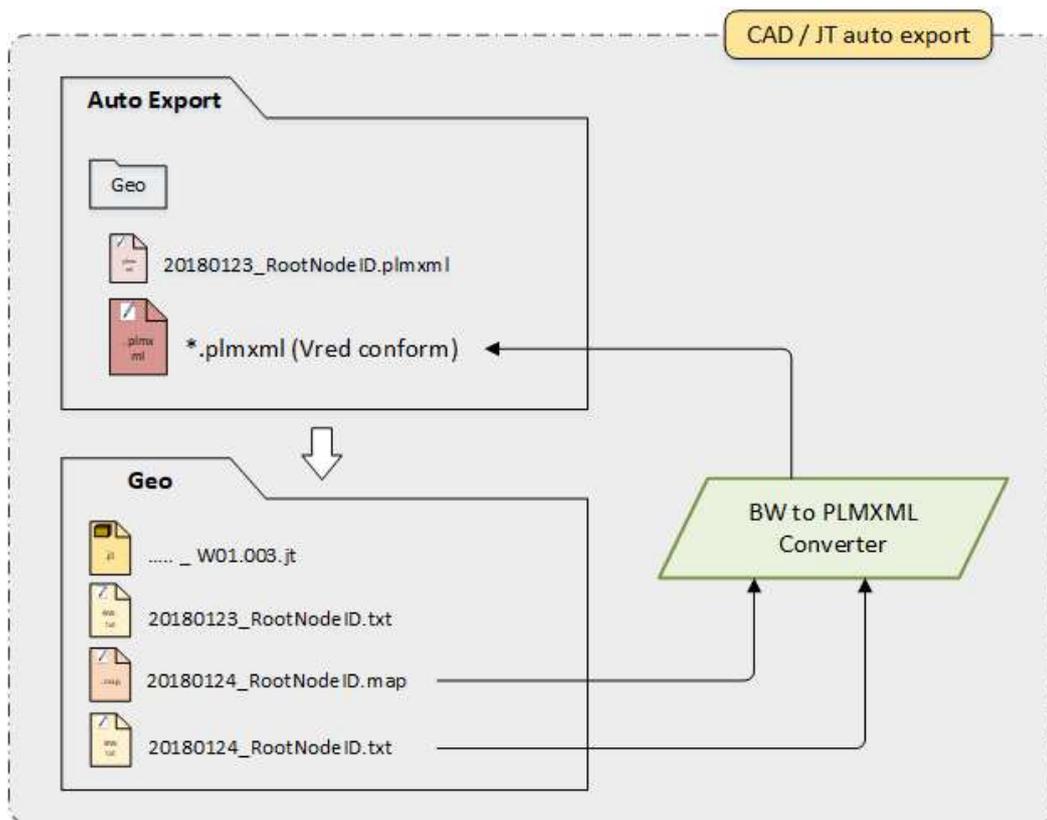


Abbildung 32 PLMXML Konverter

Neben der Möglichkeit eines automatisierten Exports aus dem PDM können die Dateien auch einzeln über den Siemens Teamcenter Visualization Mockup exportiert werden. Dieser erlaubt es in der Modellstruktur Komponenten auszublenden und nur die ausgewählte Modellstruktur im PLMXML-Format, inklusive der beinhaltenden Geometriedateien, zu exportieren.

5.2.4 Einschub: Geometriedateiformat JT (Jupiter Tessellation)

Im Bereich Engineering und besonders in der Automobilindustrie wird intensiv nach alternativen Methoden zum herkömmlichen Austausch von Originaldatensätzen zwischen den einzelnen Kooperationspartnern gesucht. Durch den Einsatz von neutralen 3D-Dateiformaten kann die Zusammenarbeit effektiver gestaltet und dadurch der Zeitaufwand und die Kosten reduziert werden. Das JT-Format ist ein solch neutrales 3D-Format, welches sich auszeichnet für den Datenaustausch und nachfolgende Visualisierungen eignet. Dies ist auf den hohen Kompressionswert, im Gegensatz zu den nativen Geometriedateien, und auf den besonderen Aufbau der Datei zurückzuführen. (Kitsios & Haslauer, 2014)

Das JT Format wurde bereits im Jahr 1998 in einer Zusammenarbeit von Hewlett-Packard und Engineering Animation Inc. (EAI) eingeführt und ist aktuell im Besitz der Siemens AG, welche das Format sukzessive weiterentwickelt und in ihrer PLM Software als Standarddateiformat einsetzt.

Die Besonderheit des JT Formats ist, dass es zwei unterschiedliche Repräsentationen von CAD-Geometrie unterstützt. Neben in mehreren Auflösungen (Levels of Detail – LOD) tesselierten Dreiecksflächen-Geometrien kann das Format auch die mathematisch exakte Brep-Geometrie beinhalten, welche später von der Visualisierungssoftware beliebig tesseliert werden kann. Zusätzlich zu den Geometriedaten beinhaltet das JT-Format auch den Szenengraph (LSG) mit CAD-spezifischen Knoten, die Produktstruktur, Produkt- und Fertigungsinformationen (PMI), visuelle Attribute wie Licht, Textur und Shader sowie bauteilspezifische Metadaten. Der Szenengraph muss hierbei in der Datei enthalten sein, die anderen Eigenschaften können je nach Anwendung optional hinzugefügt oder entfernt werden. (Katzenbach, Handschuh, & Vettermann, 2011) Der prinzipielle Aufbau einer JT Datei ist in Abbildung 33 dargestellt.

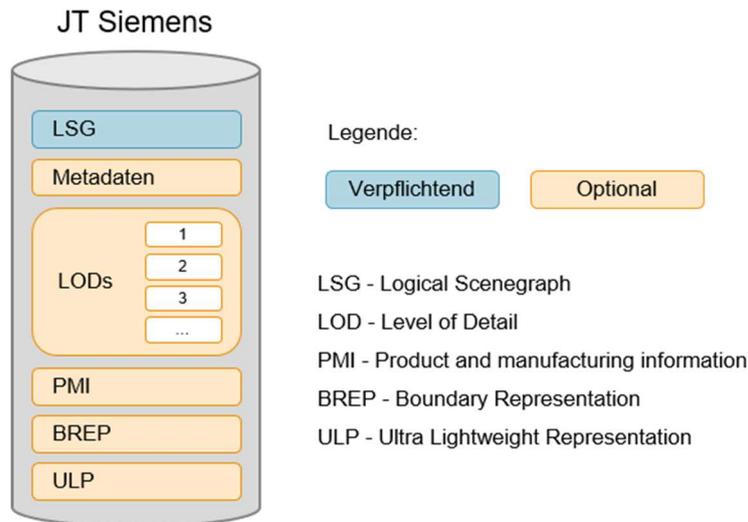


Abbildung 33 JT Geometrieformat nach Siemens

Tessellierung – Levels of Detail

Unter der Tessellierung versteht man die Erzeugung einer vereinfachten Darstellung eines Körpers. Hierbei wird mittels Triangulation ein Dreiecksnetz auf den Flächen des Körpers erzeugt, welches für die spätere Darstellung des Körpers herangezogen wird. Der Detailgrad der Vereinfachung kann im Zuge der Konvertierung über mehrere Parameter eingestellt und die Ergebnisse als verschiedene LODs in der JT-Datei gespeichert werden. Die grundlegenden Parameter sind hierbei der Wert für den maximalen Abstand zwischen der ursprünglichen Kurve und der vereinfachten Linie (chordal distance) und der Wert für den maximalen Winkel zwischen zwei benachbarten Liniensegmenten (angle). Die maximale Länge der generierten Liniensegmente kann ebenfalls definiert werden, nimmt erfahrungsmäßig jedoch wenig bis keinen Einfluss auf das Ergebnis. Die Abbildung 34 zeigt eine grafische Erklärung der Parameter. Je kleiner die Parameter gewählt werden, umso feiner wird das generierte Netz und umso besser wird auch der Originalkörper nachgebildet. Dadurch wird aber auch die Anzahl der generierten Dreiecke signifikant erhöht, was Einfluss auf die Dateigröße und auf die spätere Performance der erstellten VR-Szene nimmt.

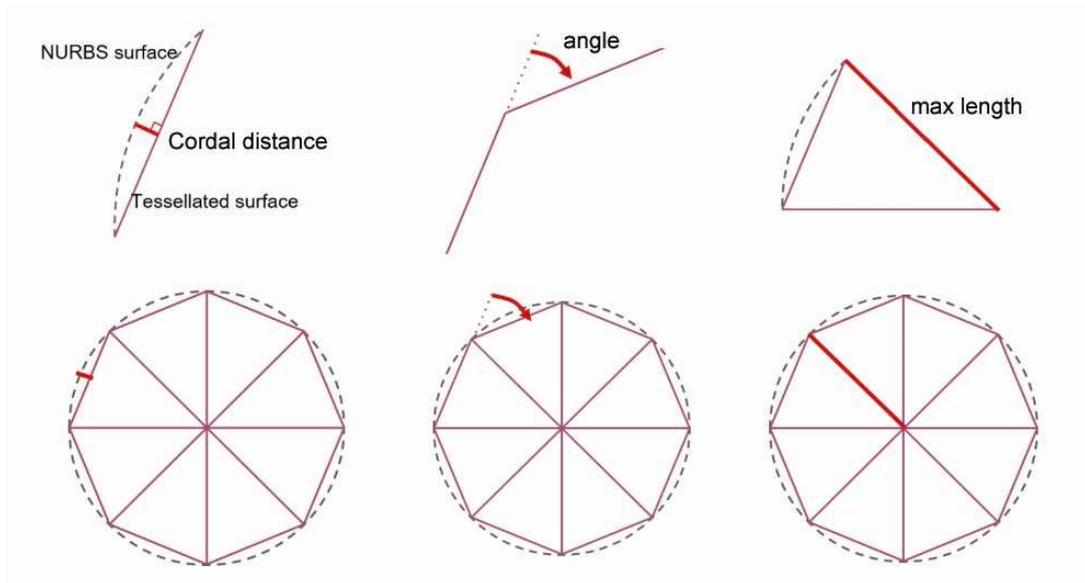


Abbildung 34 Darstellung der Tesselierungsparameter (Unreal engine, 2018)

Die Parameter der Tesselierung können bei der Konvertierung oder beim Import der Dateien in das Visualisierungsprogramm (nur wenn exakte Brep Geometrie vorliegt) gesetzt werden. Bei der Konvertierung der nativen Geometriedateien in JT-Dateien durch den 3D-Evolution Konverter werden drei unterschiedliche LODs erstellt. Die Werte der Parameter sind in Tabelle 2 dargestellt.

Tabelle 2 Parameter der Tesselierung 3D Evolution Konverter

	LOD „1“	LOD „2“	LOD „3“
Level	1	2	3
Cordal distance [mm]	0,2	0,8	1,8
Angular [°]	45	45	45
max length [mm]	0	0	0

Durch den sehr hoch gewählten Wert des Winkels zwischen den einzelnen Liniensegmenten sind die aus dem 3D Evolution Konverter erzeugten Netze sehr wohl für JT-Viewer geeignet, jedoch für VR-Visualisierungsanwendungen nur bedingt verwendbar, da enge Radien bzw. Bohrungen nicht fein genug nachgebildet werden. Aus diesem Grund wird eine weitere Tesselierung angewendet, welche bei einer weiteren Konvertierung oder beim Laden der Dateien in die Visualisierungssoftware durchgeführt wird. Tabelle 3 zeigt die von Autodesk

VRED empfohlenen Tesselierungseinstellungen, wobei Qualitätsstufe „Low“ für die bisherigen Anwendungen das beste Ergebnis, hinsichtlich des Kompromisses zwischen Darstellungsqualität und Performance der VR-Szene, darstellte. Des Weiteren ist in Abbildung 35 das generierte Netz eines Automobilbauteils in Autodesk VRED dargestellt.

Tabelle 3 Empfohlene Tesselierungseinstellungen in Autodesk VRED

Tessellation Quality	Low	Medium	High
Cordal distance [mm]	0,15	0,075	0,0375
Angular [°]	20	10	7,5
max length [mm]	300	200	100

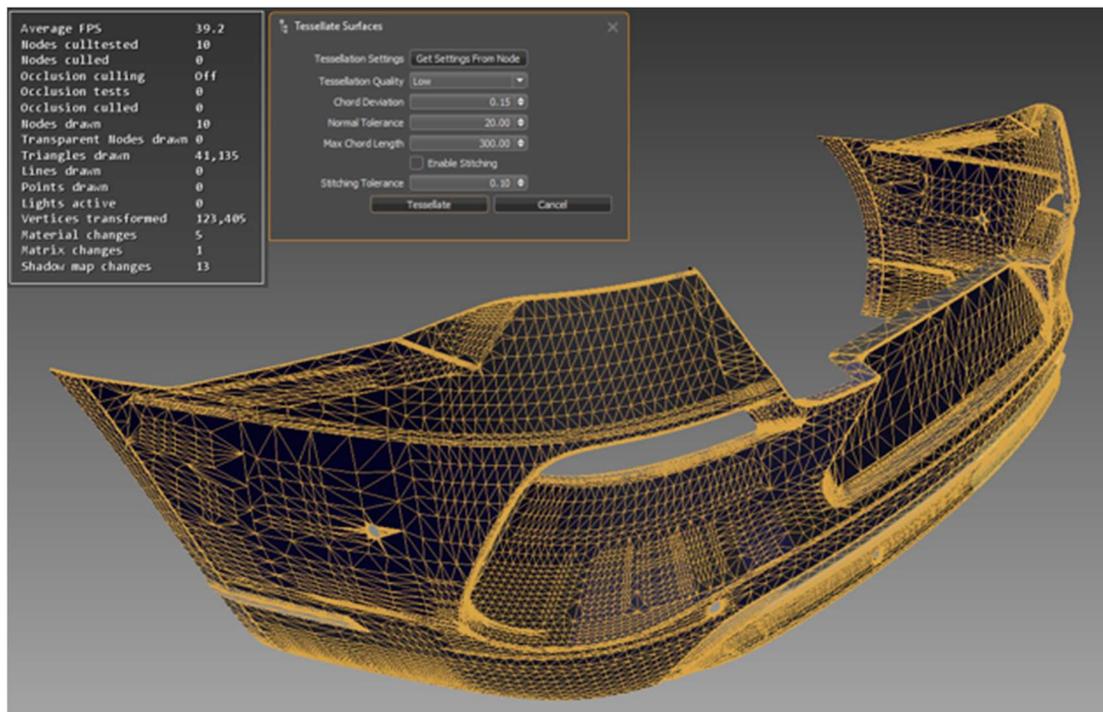


Abbildung 35 Tessellierter Körper in Autodesk VRED

5.2.5 Konvertierung der Daten

Der weitere Prozessablauf sieht eine Konvertierung aller Geometriedateien in das von Autodesk VRED angewendete Geometriedateiformat OSB (OpenSG Binary) vor. Normalerweise werden die Geometriedateien beim Einfügen in Autodesk VRED automatisch in das OSB Format konvertiert, was einiges an Zeit beansprucht. Um die Effizienz dieses Vorganges zu steigern wurde die Softwareanwendung „CAD to OSB“ (vormals „ATF to OSG“) programmiert, welche es erlaubt, den in Autodesk VRED integrierten Konverter separat zu starten und Konvertierungen durchzuführen. Da dieser Konverter in der Lage ist jegliche Geometriedateiformate in das OSB Format zu konvertieren, muss der Prozess ab diesem Schritt nicht mehr unterteilt werden. Des Weiteren wurde der Anwendung eine „File Watcher“-Funktion integriert, welche den PDM Exportordner regelmäßig nach neuen Dateien untersucht und diese automatisch ins Zieldateiformat konvertiert.

In der folgenden Abbildung werden die einzelnen Schritte der Konvertierung aufgezeigt.

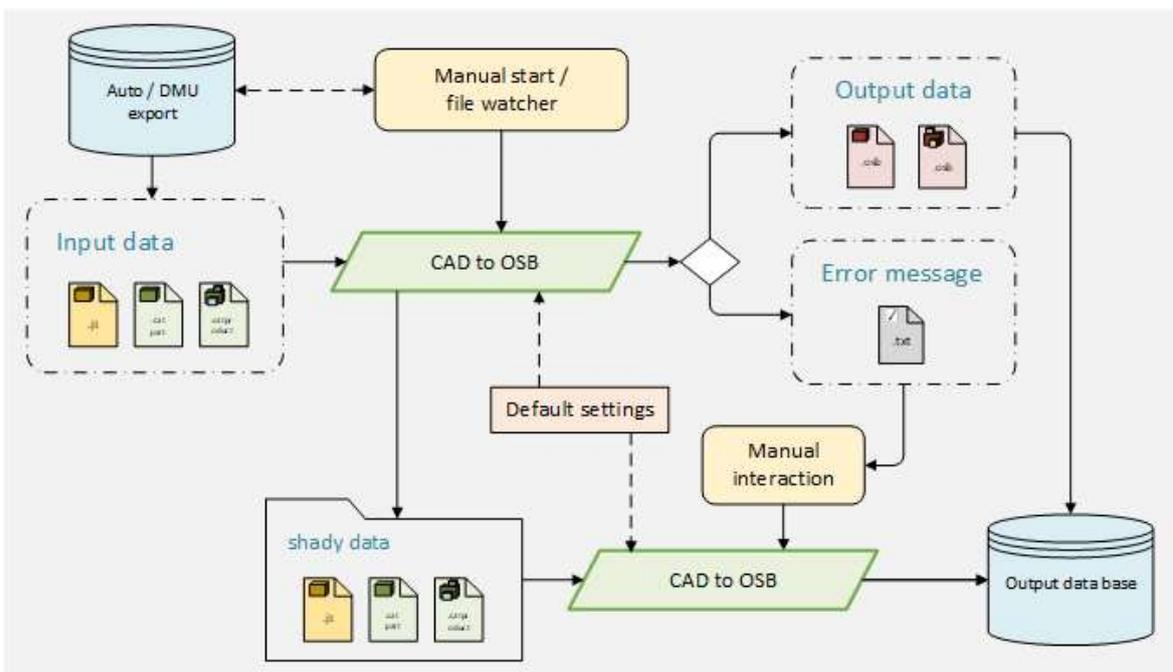


Abbildung 36 CAD to OSB Konverter

Die Konvertierung kann, wie bereits erwähnt, manuell über die Anwendung oder automatisch mittels der „File Watcher“-Funktion erfolgen. Der Eingabe- bzw. Ausgabeordner der Geometriedateien kann im Anwendungsmenü oder über eine Konfigurationsdatei bestimmt werden. Dasselbe gilt für die Konvertierungseinstellungen, welche nachfolgend genauer er-

läutert werden. In seltenen Fällen kommt es zu einem unerwarteten Fehler bei der Konvertierung einer Geometriedatei. Diese Dateien werden in einer Protokolldatei, welche aus einer Textdatei besteht, dokumentiert, um die betroffenen Dateien nachträglich manuell konvertieren zu können. Sollte es zu einem Absturz der Konvertierung kommen, so kann aus der Protokolldatei der Fortschritt ausgelesen werden. Die konvertierten Geometriedateien werden in dem im Vorhinein definierten Ausgabeordner gespeichert.

5.2.5.1 Standardeinstellungen des Konverters

Zur Erstellung der OSB Geometriedateien können beim Konverter mehrere Einstellungen vorgenommen werden. Dazu verfügt die Anwendung über ein eigenes Einstellungs Menü, welches in Abbildung 37 dargestellt ist.



Abbildung 37 Einstellungs Menü „CAD to OSB“ - Konverter

Folgend werden die Einstellungsmöglichkeiten des Konverters genauer erklärt und auf deren Notwendigkeit hingewiesen.

VRED install path

Da die Anwendung im Hintergrund auf den in Autodesk VRED beinhalteten Konverter zurückgreift, muss in diesem Einstellungsschritt der Dateipfad zur Autodesk VRED Startdatei angegeben werden. Zum Zeitpunkt dieser Ausarbeitung wurde hierzu auf Autodesk VRED der Version 2018.4 für die Konvertierung der Daten herangezogen.

Deleting existing OSB files

Ist diese Option aktiv, so werden beim Start der Konvertierung alle bereits im Ausgangsordner befindlichen OSB-Geometriedateien gelöscht. Bei nachträglichen Konvertierungen ist diese Option genau zu überprüfen, um nicht den gesamten Fortschritt zu löschen.

Stitching, stitching tolerance

Diese Optionen sind für die Tessellierung der Geometrie von Bedeutung. Werden die Tessellierungseinstellungen zu hoch gewählt, so kann es bei der Tessellierung zu Netzfehlern kommen, indem es zu Rissen und Lücken im Netz kommt. Bei aktiviertem „stitching“ werden diese Risse automatisch geschlossen. Die „stitching tolerance“ gibt hierbei den Abstand in Millimeter an, wie groß die Lücke maximal sein darf, dass sie geschlossen wird. Bei einer zu hoch gewählten Toleranz kann es jedoch passieren, dass auch gewollte Lücken, wie zum Beispiel Fugen oder angrenzende Bauteile, verschmolzen werden.

Merge geometries

Eine Geometriefläche besteht zumeist aus mehreren Geometrie-einzelstücken, welche auch explizit in der Bauteilstruktur angezeigt werden und Einfluss auf die spätere Performance der VR-Szene haben. Aktiviert man „merge geometries“ so werden die einzelnen Geometrieflächen zu einer Gesamten verschmolzen. Es ist jedoch ratsam die Geometrieflächen in der Visualisierungssoftware selbst zusammenzufügen, da der Konverter keine Rücksicht auf die Ausrichtung der Flächennormalen nimmt.

Remove groups

Wird „remove groups“ aktiviert werden alle Geometrien in der obersten Strukturebene angeordnet und die Unterordner entfernt.

Tessellation

Veranlasst die Tessellierung des Inhalts der Geometriedatei. Diese Option muss aktiviert sein, um zu einem Ergebnis zu kommen.

Keep nurbs

NURBS (Abkürzung für „Nicht uniforme rationale Basis Splines“) sind mathematisch definierte Kurven oder Flächen, welche wie Breps zur Modellierung beliebiger Formen verwendet werden. Wird diese Option aktiviert, so werden diese Geometrieinformationen, falls in der ursprünglichen Datei vorhanden, in die konvertierte Datei übertragen.

Only assemblies

Sollen nur die Baugruppen aus dem Ausgangsordner konvertiert werden, so kann diese Option aktiviert werden. Die einzelnen Bauteile, die einer Baugruppe angehören, sind der Konvertierung miteingeschlossen.

Create layer groups

Diese Option ist für Geometriedateien relevant, welche mit Autodesk Alias erstellt wurden. Hierbei wandelt der Konverter die Layer der ursprünglichen Alias-Datei in Gruppen um, damit die ursprüngliche Bauteilstruktur erhalten bleibt.

Enable curves

Ist diese Option aktiv, werden bei der Konvertierung alle Kurven und Polygonlinien übernommen. Für die späteren Visualisierungsarbeiten sind diese jedoch nicht unbedingt notwendig.

Only visible

Es werden nur Geometrien konvertiert, die in ihren Eigenschaften als sichtbar definiert sind. Unsichtbare Geometrien werden vom Konverter entfernt.

Chord deviation, normal tolerance, max chord length

In diesen drei Feldern können die Parameter für die Tessellierung angegeben werden.

Use max CPU, max CPU, max processes

Zum Abschluss kann noch angegeben werden, welche Ressourcen dem Konverter zur Verfügung stehen. Ist die Option „use max CPU“ aktiviert, kann der Konverter über die ganze Leistung des Computersystems verfügen. Alternativ kann im „max CPU“ Zahlenfeld die maximal zur Verfügung stehende Leistung in Prozent angegeben werden. Mit der Einstellung „max processes“ kann des Weiteren bestimmt werden, wie viele Dateien der Konverter zur selben Zeit maximal konvertieren darf.

5.2.6 Einfügen der Dateien in die Visualisierungssoftware Autodesk VRED

In diesem Abschnitt kommen wir nun zum Einfügen der Dateien in die Visualisierungssoftware. Als Software kommt Autodesk VRED der Version 2018.4 zum Einsatz (Autodesk). Grundsätzlich ist Autodesk VRED in der Lage, jegliche Geometriedateiformate zu laden, da sie vom integrierten Konverter in das OSB Dateiformat umgewandelt werden. Jedoch benötigt dies einiges an Zeit, da die Geometrien neu tesseliert werden. Da im beschriebenen Prozess die Dateien bereits im vorigen Schritt in das von Autodesk VRED verwendete Format konvertiert wurden, ist nun ein Import der Dateien ohne eine weitere Tessellierung möglich. Eine weitere sehr effektive Variante ist der Dateiimport unter Anwendung des ebenfalls in VRED integrierten „Legacy JT Loader“. Ist diese Funktion aktiviert, können JT Dateien ohne zusätzlicher Tessellierung geladen werden, was eine enorme Zeitersparnis mit sich bringt. Die Software lädt hierbei das bereits in der JT Datei befindliche tesselierte Geometrienetz, welches an der Position des vordersten LOD liegt (LOD mit der kleinsten Ziffer). Diese Funktion ist jedoch nur für JT Dateien bis Version 9.5 kompatibel. Für JT Version 10 und höher wurden von Autodesk noch keine Anpassungen vorgenommen.

Neben Geometriedateien ist es mit Autodesk VRED auch möglich, Textdateien zu laden, welche die CAD-Modellstruktur beinhalten. So ist es möglich, die im Prozess erstellte PLMXML Datei zu öffnen. Da die dazugehörigen Geometrien in der PLMXML Datei verlinkt sind, werden diese ebenfalls in das Visualisierungsprogramm geladen. Dies ermöglicht es, einen automatisierten Import zu erstellen, welcher über eine .NET-Anwendung gesteuert wird und im Hintergrund über ein programmiertes Skript mit Autodesk VRED kommuniziert. Über diese Anwendung besteht auch die Möglichkeit, eine Update Funktion zu integrieren, auf welche später noch Bezug genommen wird.

5.2.7 Vergleich der relevanten Aufbereitungsvarianten

Im Laufe der Beschreibung des Datenaufbereitungsprozesses wurden nun schon mehrere verschiedene Varianten erwähnt und auf ihre Vor- und Nachteile hingewiesen. Die folgenden Tabellen in Matrixform fassen diese nochmals zusammen. Auf der Abszisse sind hierbei die einzelnen Schritte des Datenaufbereitungsprozesses aufgetragen. Die Ordinate bezieht sich auf die verschiedenen Varianten bzw. Vorgehensweisen. Des Weiteren sind die verschiedenen Vorgehensweisen bezogen auf deren Ergebnis farblich markiert. Eine grüne Markierung kennzeichnet die empfohlene Vorgehensweise, eine gelbe Markierung eine alternative Vorgehensweise und eine rote Markierung zeigt, dass dieser Aufbereitungsweg für unsere Anforderungen nicht zielführend ist.

CATIA als Ausgangsdatei

Tabelle 4 Aufbereitungsvarianten von CATIA Geometriedateien

		Prozessschritt Datenaufbereitung			
Dateityp	Konvertierung JT v9.5 (CT 3D-Evolution)	Konvertierung CADtoOSB	VRED Import	VRED Bearbeitung	
CATIA	CAS: Struktur und Teilennamen gehen verloren	behält JT Struktur	VRED 18: Surface - Tesselerung CADtoOSB	neue Tesselerung möglich	
			VRED 19: Surface - Tesselerung CADtoOSB; Mesh - LOD 1		
		/	Surface und Mesh wählbar; Surface - Tesselerung VRED; Mesh - LOD wählbar	mit Surface neue Tesselerung möglich	
	CAD: alle Geometrien bekommen Bauteilnamen	behält JT Struktur	behält JT Struktur	VRED 18: Surface - Tesselerung CADtoOSB	neue Tesselerung möglich
				VRED 19: Surface - Tesselerung CADtoOSB; Mesh - LOD 1	
		/	Surface und Mesh wählbar; Surface - Tesselerung VRED; Mesh - LOD wählbar	mit Surface neue Tesselerung möglich	
	/	/	behält ursprüngliche Struktur	Surface - Tesselerung CADtoOSB	neue Tesselerung möglich
			/	Surface und Mesh wählbar; Surface - Tesselerung VRED; Mesh - LOD wählbar	mit Surface neue Tesselerung möglich

Beim ersten Vergleich (Tabelle 4) dient eine CATIA Geometriedatei als Ausgangsdateiformat. Dieses Dateiformat wird, wie bereits bekannt, nur in der frühen Fahrzeugentwicklungsphase zur Datenaufbereitung herangezogen, da dessen CAD-Modellstruktur bei der Integration ins PDM und der daraus resultierenden Konvertierung ins JT Dateiformat verloren gehen würde. Wird die Datei ausschließlich mit dem in Autodesk VRED integrierten Konverter über die „CAD to OSB“ Anwendung umgewandelt, so bleibt die ursprüngliche Struktur erhalten und kann für die Visualisierungen verwendet werden. Die Qualität des Geometrienetzes wird hierbei über die Einstellungen des „CAD to OSB“ Konverters bestimmt. Nach dem Import sind die Geometrien in VRED als „Surfaces“ vorhanden, was eine Nachbearbeitung der Tesselierungsqualität ermöglicht.

JT als Ausgangsdatei

In Tabelle 5 ist der Vergleich der JT Geometriedateien abgebildet. Neben der Unterteilung in die beiden JT Versionen 9.5 und 10 werden die JT Dateien auch in Hinsicht auf ihrem Inhalt unterschieden. So werden die einzelnen Datenaufbereitungsschritte je Version mit drei unterschiedlichen JT Dateien durchgeführt, welche nur drei bereits tesselierte LOD Geometrienetze beinhalten, nur aus der Brep-Geometrie bestehen und beide Arten der Geometriedarstellungsformen enthalten. Durch die unterschiedlichen Inhalte der Geometriedateien ergeben sich auch unterschiedliche Ergebnisse.

Die zu empfehlende Vorgehensweise sieht hierbei eine Verwendung von JT Dateien vor, welche nur die Brep-Geometriedarstellungsform enthalten. Diese beinhalten kein tesseliertes Geometrienetz, wodurch Speicherbedarf gespart werden kann. Die Tesselierung der Geometrie erfolgt bei der Konvertierung ins OSB-Format, bevor das entstehende Geometrienetz als „Surface“ ins Visualisierungsprogramm geladen wird. Somit kann die Tesselierungsqualität des Geometrienetzes jederzeit nachbearbeitet werden.

Alternativ können auch JT-Dateien, welche die Brep-Darstellungsform und drei LODs beinhalten, für die Datenaufbereitung verwendet werden. Dies wird sogar die priorisierte Vorgehensweise darstellen, da wir zum aktuellen Stand die JT-Dateien in dieser Form aus dem PDM Export beziehen. Im Unterschied zur empfohlenen Vorgehensweise kann hier beim Import gewählt werden, ob das Geometrienetz als „Surface“, „Mesh“ oder beide Varianten importiert werden sollen. Als „Mesh“ wird hierbei das Geometrienetz des ersten LOD der ursprünglichen JT Datei herangezogen.

Tabelle 5 Aufbereitungsvarianten von JT Geometriedateien

		Prozessschritt Datenaufbereitung			
Dateityp	Inhalt	Konvertierung CADtoOSB	Legacy JT Loader	VRED Import	VRED Bearbeitung
JT v9.5	3 LOD	behält JT Struktur	/	Mesh - LOD 1	keine neue Tesselerung möglich
		/	Ja	Mesh - LOD 1	
		/	Nein	Mesh - LOD wählbar	
	Breps	behält JT Struktur	/	Surface - Tesselerung CADtoOSB	neue Tesselerung möglich
		/	Ja	kein Import	/
		/	Nein	Surface - Tesselerung VRED	neue Tesselerung möglich
	Breps + 3 LOD	behält JT Struktur	/	Surface und Mesh wählbar; Surface - Tesselerung CADtoOSB; Mesh - LOD 1	mit Surface neue Tesselerung möglich
		/	Ja	Mesh - LOD 1	keine neue Tesselerung möglich
		/	Nein	Surface und Mesh wählbar; Surface - Tesselerung VRED; Mesh - LOD wählbar	mit Surface neue Tesselerung möglich
	JT v10	3 LOD	behält JT Struktur	nicht unterstützt	Mesh - LOD 1
/			/	Mesh - LOD wählbar	
Breps		behält JT Struktur	nicht unterstützt	Surface Tesselerung CADtoOSB	neue Tesselerung möglich
		/	/	Surface - Tesselerung VRED	
Breps + 3 LOD		behält JT Struktur	nicht unterstützt	Surface und Mesh wählbar; Surface - Tesselerung CADtoOSB; Mesh - LOD 1	mit Surface neue Tesselerung möglich
		/	/	Surface und Mesh wählbar; Surface - Tesselerung VRED; Mesh - LOD wählbar	mit Surface neue Tesselerung möglich

Ein großer Vorteil ist, dass die angewandten Vorgehensweisen nicht zwischen den beiden JT Versionen unterscheiden, wodurch die Wahl der JT Version nicht eingeschränkt wird.

5.3 Visualisierung - Modellaufbereitung

Die Visualisierung, in diesem Fall insbesondere die 3D-Visualisierung, bezeichnet die Konvertierung von technischen Konstruktionen aus 3D-Zeichenprogrammen zu dreidimensionalen virtuellen Modellen. Das CAD-Modell wird dabei von eigens geschultem Personal realitätstreu aufbereitet und in eine fotorealistische Umgebung transferiert. Durch die Verwendung von aktueller Visualisierungssoftware mit modernsten Licht- und Schattenberechnungsmethoden können so virtuelle Modelle erstellt werden, welche nur schwer von ihrem realen Ebenbild unterscheidbar sind. Nach dem Setup der Belichtung und der Kamera kann die erstellte Szene dann entweder zu einem Bild gerendert oder mit Hilfe eines VR-Ausgabegerätes interaktiv betrachtet werden.

Wie bereits in den vorherigen Kapiteln ersichtlich, wird bei Magna Steyr als Visualisierungssoftware VRED von Autodesk verwendet. Autodesk VRED bietet die optimale Plattform, um eine, für die Ansprüche der Fachabteilung, ideale Visualisierung herzustellen. Da die Software mit einem Python Skript arbeitet, welches über einen Skript-Editor frei zugänglich und anpassbar ist, ist es sogar möglich eigens erstellte Anwendungen zu implementieren und das Visualisierungsprogramm individuell zu erweitern. So können alle Anforderungen mittels VRED oder eigens erstellten Erweiterungen in die VR-Szene integriert werden. Abbildung 38 zeigt eine Übersicht über das Zusammenspiel zwischen der eigentlichen Visualisierungssoftware Autodesk VRED, den Anwendungsfall-spezifischen Funktionen und den in die VR-Szene implementierten Werkzeugen, die für eine interaktive Präsentation eines Anwendungsfalls benötigt werden.

Autodesk VRED dient hierbei durch dessen offene Programmierstruktur als eine Art Sandkasten für etwaige Visualisierungsarbeiten und bietet zusätzlich jegliche Funktionen für die visuelle Aufbereitung eines CAD-Modells. So bietet es eine Vielzahl an realistischen Materialien bis hin zu modernsten Raytrace Rendermethoden, um Licht- und Schattenverhältnisse realistisch zu berechnen und darzustellen. Die Anwendungsfall-spezifischen Funktionen stellen Möglichkeiten dar, inwieweit das Fahrzeugmodell aufbereitet werden kann. Dies ist vom Anwendungsfall an sich und dem geforderten Detailgrad abhängig. Grundsätzlich ist es möglich, Klappen und Türen die sich öffnen und schließen, veränderbare Interieur- und Exterieur Ausstattung bis hin zu interaktive Bedienelemente in die VR-Szene einzufügen. Diese Funktionen sind direkt auf das virtuelle Fahrzeugmodell bezogen und müssen für jedes Modell gesondert erstellt werden. Nicht so ist es bei den interaktiven Werkzeugen,

welche über das Skript und der nötigen Hilfsgeometrie direkt in das Template der VR-Szene integriert werden können.

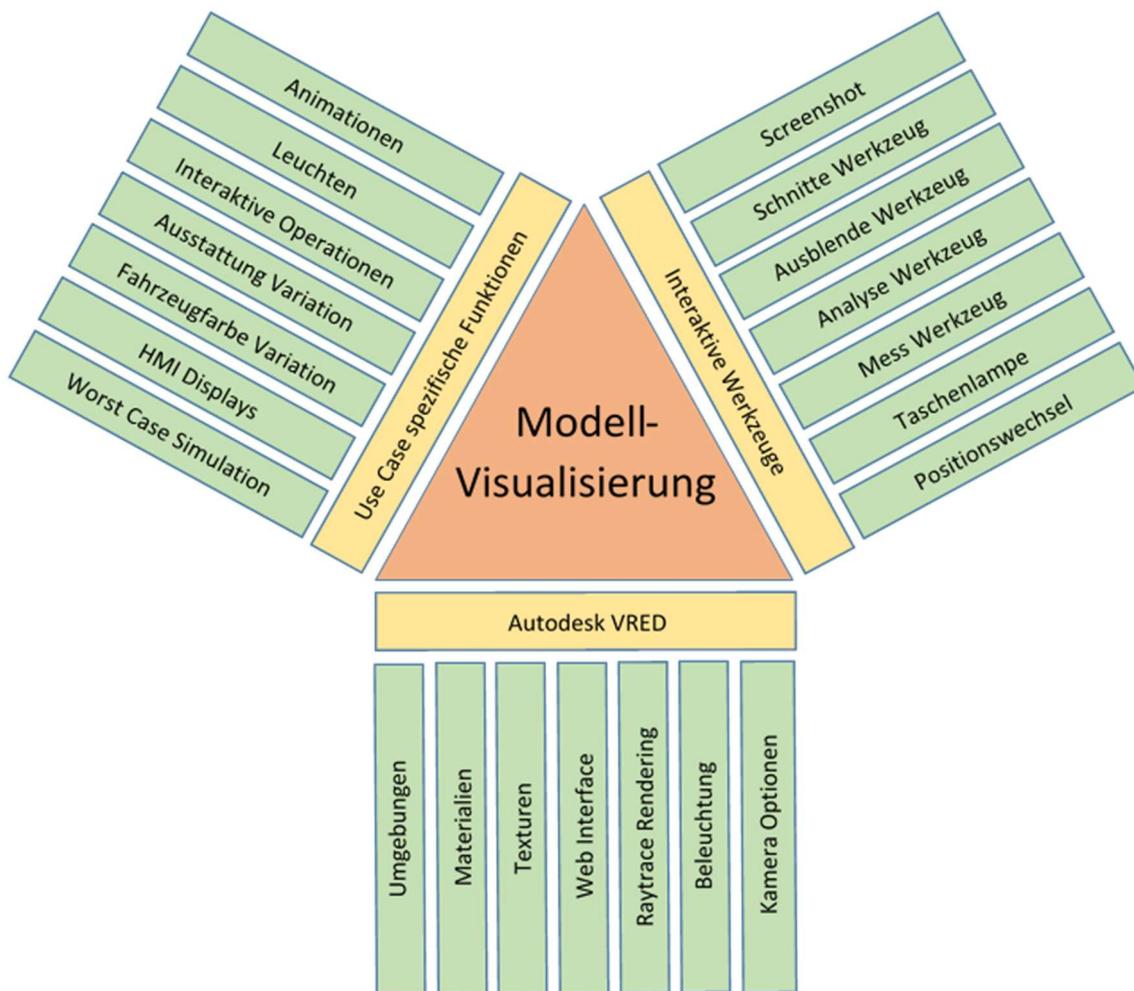


Abbildung 38 Dreieck der Modellvisualisierung

In den folgenden Kapiteln wird näher auf die Autodesk VRED Arbeitsumgebung eingegangen und gezeigt, in welcher Verbindung die einzelnen Editoren zueinanderstehen. Dieses Erkenntnis ist die Grundvoraussetzung für den Aufbau einer optimalen Dateiverwaltungsstruktur. Zusätzlich ist es wichtig, die VRED-internen Abläufe zu verstehen, um den automatischen Updateprozess diesbezüglich abstimmen zu können.

5.3.1 Die Autodesk VRED Arbeitsumgebung

Die Arbeitsumgebung von Autodesk VRED besteht aus mehreren Editor-Modulen, mit deren Hilfe es möglich ist, das importierte CAD-Modell den Anforderungen der Visualisierung anzupassen. In diesem Abschnitt werden nun deren Verwendungszweck und wichtigsten Funktionen näher beschrieben (Autodesk).

Viele der Editoren interagieren miteinander, was in Abbildung 39 grafisch dargestellt ist. Des Weiteren wird abgebildet, welcher Editor einen Knoten im Strukturbaum des VRED Projektes benötigt bzw. ob ein Knoten durch die Interaktion des Editors erstellt wird. Diese Informationen werden benötigt, um in späterer Folge ein anwendungsspezifisches Template mit bereits integrierten Funktionen erstellen zu können.

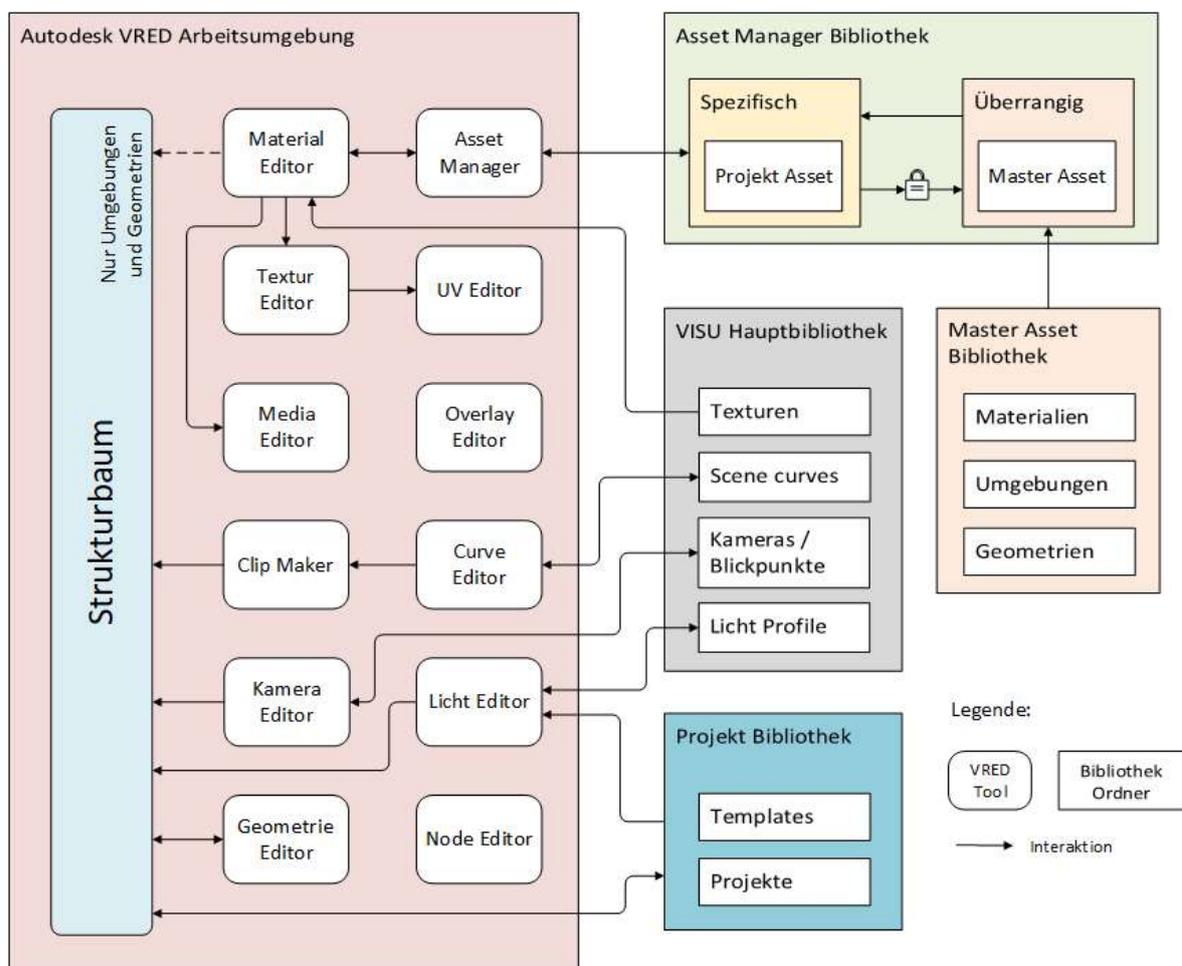


Abbildung 39 Arbeitsumgebung Autodesk VRED

Darüber hinaus sind in der Abbildung die angelegten Bibliotheken dargestellt, um verschiedene Gegenstände, wie zum Beispiel Materialien, Umgebungen und Muster - Geometrien, Animationen von bewegten Teilen bis hin zu fahrenden Gesamtfahrzeugen, Texturen zur

Materialerstellung, spezielle Kamerapunkte bzw. animierte Kamerafahrten für Renderings und verschiedenste Lichtprofile, an einem definierten Speicherort abzulegen und so für verschiedene Projekte abrufbar zu machen.

5.3.1.1 Asset Manager

Der Asset Manager ist das Hauptbindeglied zwischen VRED und der Asset Bibliothek. Er ist in VRED direkt als Modul integriert und hilft bei der Verwaltung von Gegenständen wie Materialien, Umgebungen und Geometrien. Der Asset Manager greift hierbei auf ein oder mehrere Bibliothekverzeichnisse zu, welche vom Anwender frei verwaltbar sind. Diese Bibliotheken können in einem lokalen Ordner liegen oder auch über eine Netzwerkverbindung aufgerufen werden.

Der Asset Manager ist aber nicht nur als integriertes Modul, sondern auch als externe Anwendung verfügbar. Hierzu liegt im VRED-Installationsverzeichnis die Anwendung namens „Autodesk Automotive Assets“, welche auch ohne Verbindung mit dem Lizenzserver ausführbar ist und neben den üblichen Funktionen des Asset Manager auch die Funktionen des Material Editors beinhaltet (Abbildung 40). Hier arbeitet der Material Editor jedoch nicht auf eine VR-Szene bezogen und dient daher nur zur Verwaltung des Asset Managers.

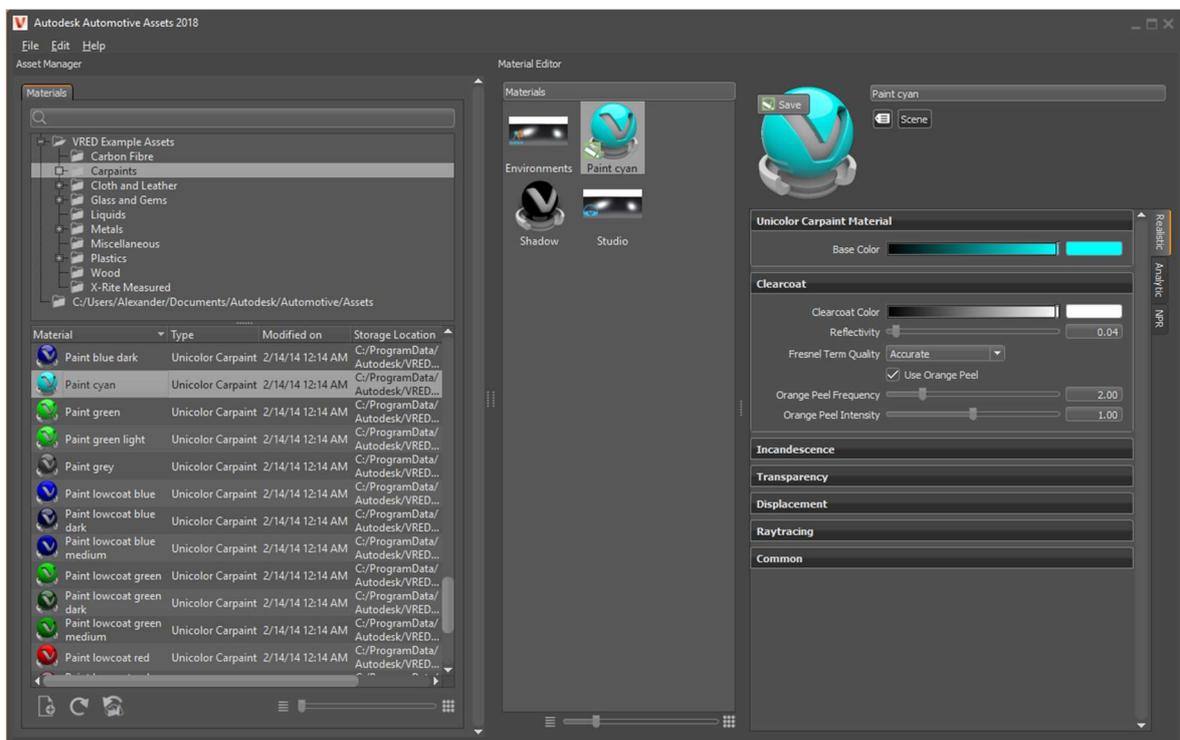


Abbildung 40 Autodesk Automotive Assets

5.3.1.2 Material Editor

Der Material Editor dient zur projektbezogenen Verwaltung von Materialien und Umgebungen. Des Weiteren können mit dem Material Editor Materialien erstellt, modifiziert und gruppiert werden. Die Oberfläche des Material Editor (Abbildung 41) ist prinzipiell in drei Bereiche unterteilt. Im ersten Bereich wird eine Übersicht aller Materialien und Umgebungen, welche im Projekt vorhanden sind, dargestellt. Hier können projektbezogene Materialgruppierungen erstellt werden. Der zweite Bereich dient als Voransicht der vorhandenen Materialien und Umgebungen. In diesem Abschnitt können die Gegenstände kopiert, eingefügt, dupliziert oder gelöscht werden. Der dritte Bereich ermöglicht die Bearbeitung aller Material- bzw. Umgebungseigenschaften. In einer weiteren Voransicht werden die Änderungen dementsprechend dargestellt.

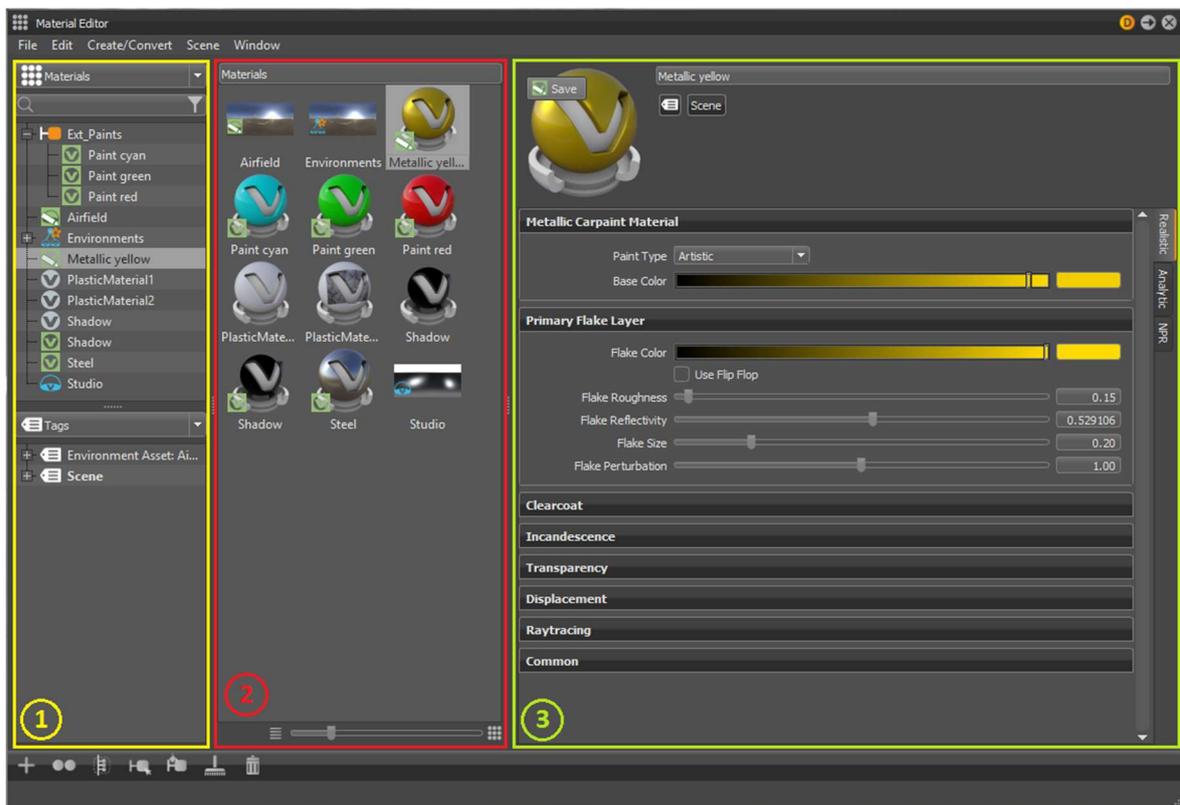


Abbildung 41 VRED Material Editor

Bei Verwendung einer Asset Bibliothek ist der Asset Manager eng mit dem Material Editor verbunden. Im Projekt gewünschte Materialien können per Drag and Drop direkt vom Asset Manager in den Material Editor gezogen und dort weiterverarbeitet werden. Jedoch ist das Material im Projekt zu jederzeit mit dem Material in der Asset Bibliothek verknüpft, was bei

gewissen Vorhaben große Sorgfalt voraussetzt, um nicht aus Versehen das ursprüngliche Material zu verändern. Dasselbe gilt für Umgebungen, welche von der Asset Bibliothek ins Projekt übernommen werden. In Abbildung 42 werden die Interaktionen zwischen Asset Manager und Material Editor für die wichtigsten Funktionen aufgezeigt und nachfolgend genauer beschrieben.

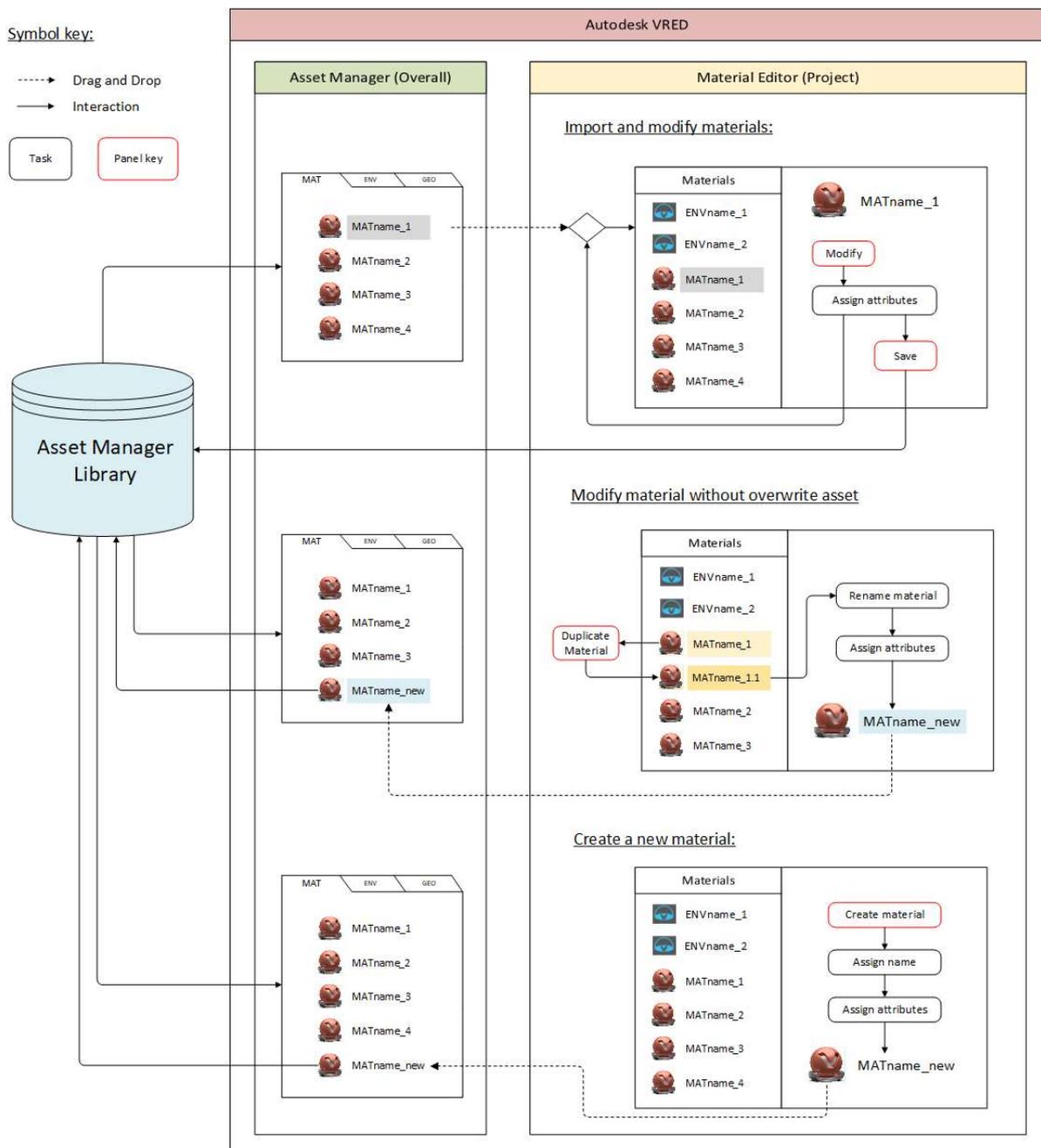


Abbildung 42 Interaktionen im zwischen VRED Asset Manager und Material Editor

Importieren und modifizieren von Materialien

Wie bereits erwähnt, kann ein Gegenstand aus dem Asset Manager durch einfaches hinüberziehen dem Material Editor, und somit dem Projekt, hinzugefügt werden. Dem Material-Vorschaubild wird in der unteren linken Ecke ein grünes Symbol (grünes V) hinzugefügt, welches symbolisiert, dass dieses Material mit der Asset Bibliothek verbunden ist.

Um ein Material, welches aus der Asset Bibliothek bezogen wurde, modifizieren zu können, muss dies zuerst über die „Modify“ Schaltfläche bestätigt werden. Wird diese gedrückt, werden die verschiedenen Einstellmöglichkeiten aktiv. Die vorgenommenen Anpassungen werden sofort übernommen und müssen projektbezogen nicht gespeichert werden.

Vorsicht: Durch das Drücken der „Save“ Schaltfläche wird das ursprüngliche Material in der Asset Bibliothek überschrieben. Hier ist besondere Vorsicht geboten, um nicht die bereits abgelegten Materialien der Asset Bibliothek zu verändern.

Modifikation eines Materials ohne die Gefahr ein Bestehendes zu überschreiben

Um die Gefahr des Überschreibens eines Assets zu umgehen empfiehlt es sich, das Material, welches man anpassen möchte, zu duplizieren. Das Duplikat ist nur im Projekt vorhanden und nicht mehr mit der Asset Bibliothek verbunden. Nun kann das Material gefahrlos angepasst und bei Bedarf als neues Material der Asset Bibliothek hinzugefügt werden.

Erstellen eines neuen Materials

Die Erstellung eines neuen Materials ist auf verschiedene Wege möglich. Die Option, ein bestehendes Material zu duplizieren und anzupassen, wurde bereits angeführt. Eine weitere Möglichkeit ist es, eine Autodesk Vorlage zu verwenden und anzupassen. Es ist natürlich auch möglich, ein Material von Grund auf neu zu erstellen, was jedoch einiges an Visualisierungserfahrung benötigt.

Durch das Hinüberziehen (Drag and Drop) des erstellten Materials in ein Ordnerverzeichnis des Asset Managers wird es der Asset Bibliothek hinzugefügt.

5.3.1.3 Kamera Editor

Der Kamera Editor macht alle mit der Kamera verbundenen Funktionen für den Benutzer zugänglich. Dazu gehören realitätsnahe Objekteigenschaften, benutzerdefinierte Projektionen und weitere spezifische Kameraeinstellungen. Aufrufbar ist der Editor über das Scene-Bedienfeld oder aber auch über das Kamerasymbol in der unteren Schnellstartleiste der VRED Benutzeroberfläche. In Abbildung 43 ist die Bedienoberfläche des Kamera Editors dargestellt.

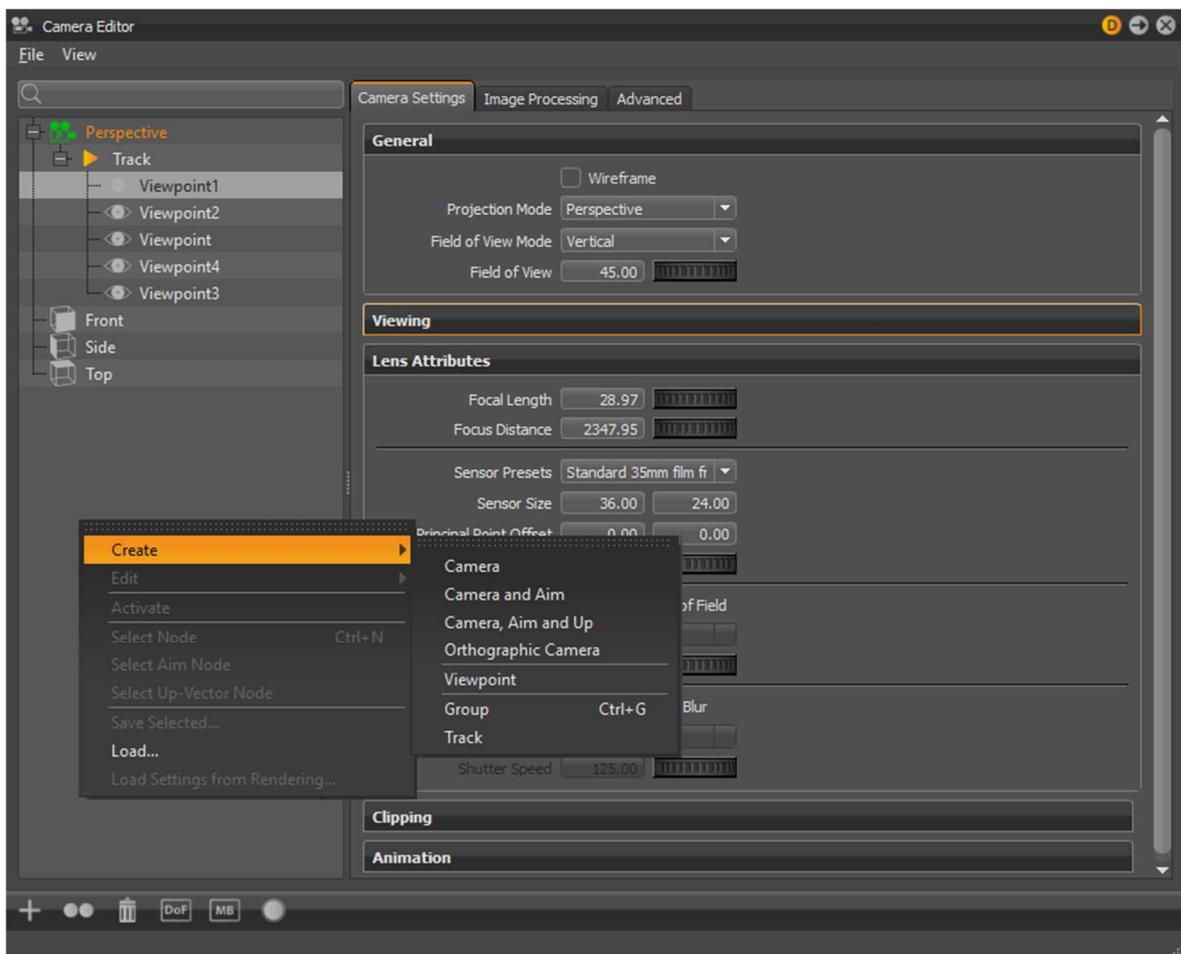


Abbildung 43 VRED Kamera Editor

Mit dem Kamera Editor sind folgende Kamera-Moden in die VRED Szene implementierbar:

- Perspective camera
Selbe Perspektive wie mit dem menschlichen Auge: Weiter entfernte Gegenstände werden kleiner dargestellt als nahe Gegenstände.

- Orthographic camera
Die Darstellung unterscheidet sich nicht bezugnehmend auf die Entfernung der Gegenstände. Alle Gegenstände werden in ihrer Originalgröße dargestellt.
- Animation camera with aim
Kamerapunkt, von dem ein Transformationsknoten erzeugt wird. Position und Ausrichtung von diesem Knoten sind über den Szenengraphen jederzeit veränderbar.
- Animation camera with aim and up
Dies ist ein spezieller Kameratyp, der die Position und Ausrichtung der Kamera aus definierten Transformationsknoten berechnet. Eine interaktive Navigation ist hier nicht möglich. Dieser Typ wird derzeit hauptsächlich für den Import von Animationen verwendet.
- Camera track
Ermöglicht die Verwendung der Kameraanimation im Clip Maker. Dazu muss im Kurveneditor ein Animationsblock erstellt werden. Dieser Kamerablock kann per Drag & Drop im Clip Maker verwendet werden.
- Viewpoint
Bei der Erstellung eines Viewpoints wird die aktuelle Kameraposition gespeichert. So ist es möglich, vor einer Präsentation wichtige Ansichten vorzudefinieren, welche dann schnell anwählbar sind. Die einzelnen Viewpoints können auch für eine animierte Kamerafahrt herangezogen werden. Da diese Viewpoints jedoch nicht mit der Verwendung einer VR-Brille funktionieren, weichen wir für solche Anwendungen auf eine eigens erstellte Erweiterung aus.

Die im Projekt erstellten Kameras und Viewpoints können als OSB-Datei gespeichert und in jedem beliebigen Projekt wiederverwendet werden. So ist es möglich, anwendungsbezogene Kameraansichten zu erstellen und diese über die Bibliothek für alle folgenden Projekte zugänglich zu machen.

5.3.1.4 Licht Editor

Der Licht Editor (Abbildung 44) ermöglicht das Erstellen und Bearbeiten von Lichtquellen in der VRED Szene und enthält eine Auflistung aller beinhalteten Lichtquellen. Der „Headlight“ Scheinwerfer ist in jeder Szene bereits vordefiniert und deaktiviert sich automatisch, wenn eine neue Lichtquelle zur Szene hinzugefügt wird. Das „Headlight“ kann wieder aktiviert werden, indem es innerhalb des Licht-Editors oder mit der Schaltfläche „Headlight“ in der Hauptsymbolleiste gesteuert wird.

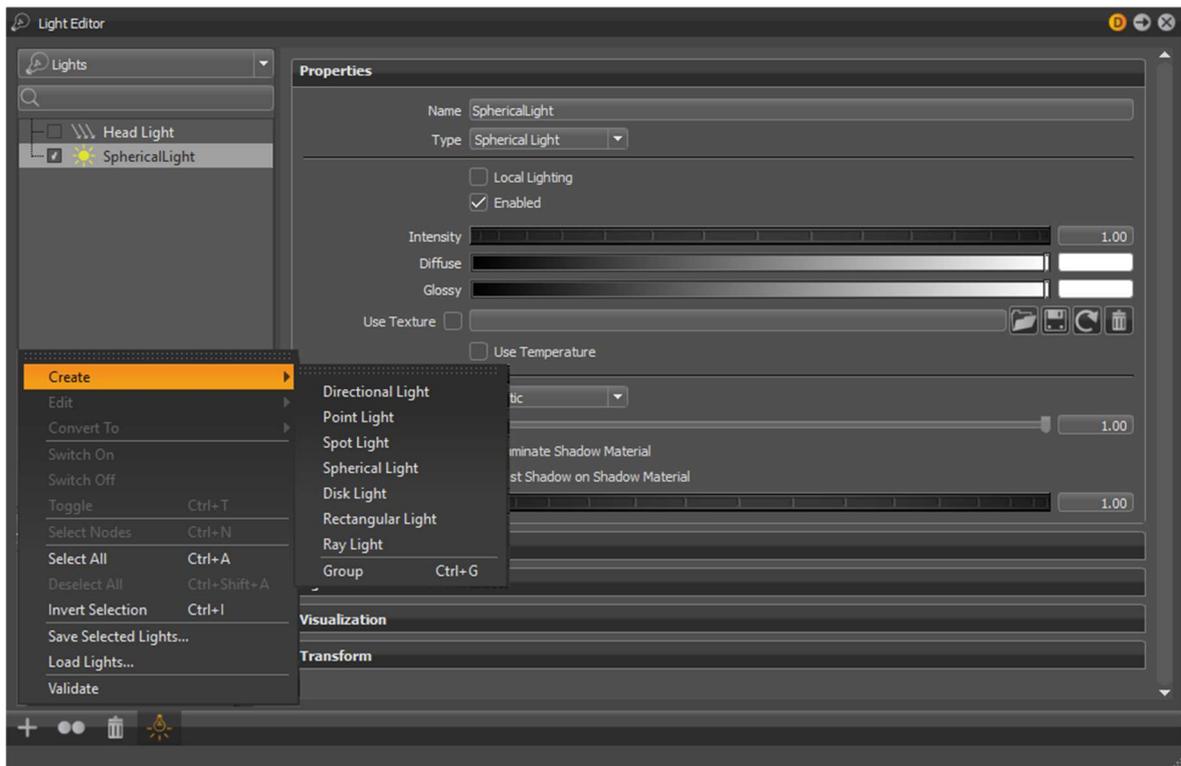


Abbildung 44 VRED Licht Editor

Folgende Lichtquellen sind mit dem Licht Editor in die VRED Szene integrierbar:

- Gerichtetes Licht (directional light)

Die erstellte Lichtquelle emittiert parallele Strahlen wie Sonnenlicht. Ein gerichtetes Licht stellt eine entfernte Lichtquelle dar und hat keine Position, sondern nur eine Richtung. Richtungslichter erzeugen immer harte Schatten in der VRED Szene.

- Punktlicht (point light)
Punktlichtstrahlen beginnen an einem einzelnen Punkt und emittieren die Lichtstrahlen sternförmig. Im Raytracing-Modus erzeugt es einen harten Schatten, im OpenGL-Modus wird kein Schatten dargestellt.
- Spot Licht (spot light)
Spotlichtstrahlen beginnen wie Punktlichtstrahlen an einem einzelnen Punkt, wobei sich die Strahlen kegelförmig ausbreiten. Hierbei kann der Öffnungswinkel des Lichtkegels und über die Penumbra-Eigenschaft der Lichtabfallsbereich rund um den Lichtkegel definiert werden.
- Sphärisches Licht (spherical light)
Mit diesem Licht Tool kann durch Hilfe einer leuchtenden Kugel Sonnenlicht visualisiert werden. Es erzeugt diffuses Licht und weiche Schatten. Es bietet eine indirekte Beleuchtung und erscheint natürlicher als ein Punktlicht. Schatten werden nur im Raytracing-Modus dargestellt.
- Disk or rectangular light
Lichtstrahlen von Scheiben oder rechteckigen Lichtquellen erscheinen wie aus einem Loch in der Szene. Je nach Typ in Form eines Kreises oder einer Scheibe beziehungsweise in einer quadratischen oder rechteckigen Form.
- Ray light
Durch laden einer Raylight Datei wird die Lichtquelle der VRED Szene hinzugefügt. Diese Dateien enthalten die simulierte räumliche Winkelverteilung sowie die photometrische oder spektrale Information des emittierten Lichts einer komplexen Lichtquelle.

Bei der Erstellung einer Lichtquelle wird ein Knoten im Strukturbaum des Projektes angelegt. Auch Lichtquellen können gespeichert und somit in der Bibliothek abgelegt werden. Als Dateiformat für die Speicherung dient das VRED Standardformat OSB. In ein neues Projekt geladen können aber nicht nur in OSB Dateien gesicherte Lichtquellen, sondern es können auch die von VRED zur Sicherung von ganzen Projekten verwendeten Dateiformate

VPE und VPB ausgewählt werden, aus denen automatisch die darin vorhandenen Lichtquellen ausgelesen und übernommen werden.

5.3.1.5 Clip Maker

Der Clip Maker dient zur Erstellung und Verwaltung von Animationssequenzen in der VRED Szene. In der linken Liste der Benutzeroberfläche (Abbildung 45) werden alle Clips der Szene angezeigt, bei deren Auswahl sich rechts eine Timeline-Ansicht öffnet. In dieser Timeline-Ansicht können Reihenfolge, Startzeit und Dauer der verschiedenen Animationssequenzen angepasst und so zu einem Clip vereint werden.

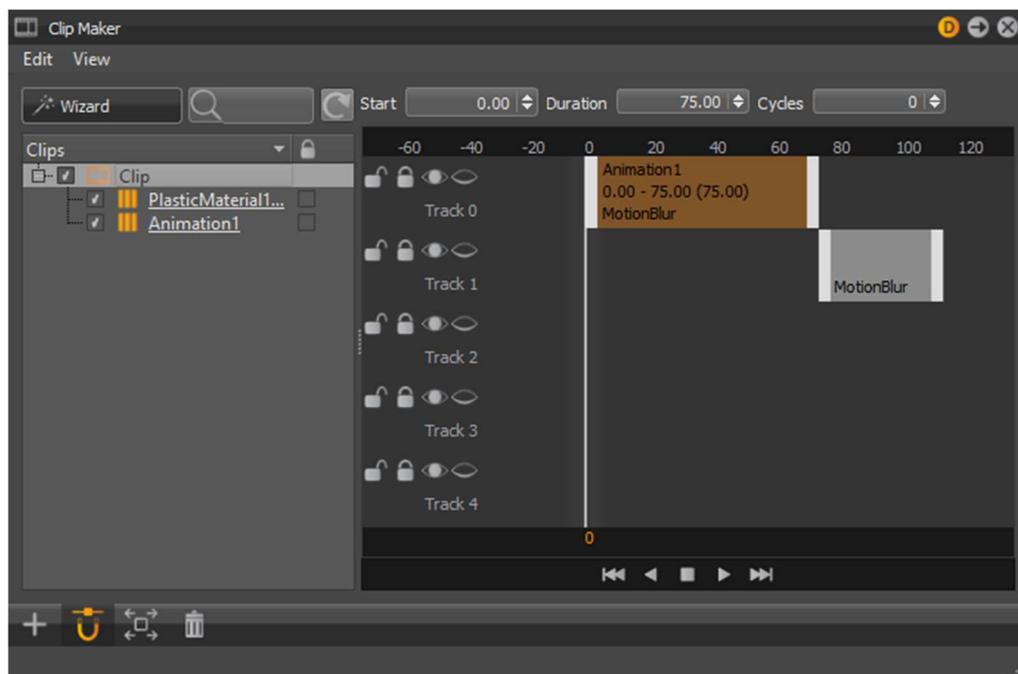


Abbildung 45 VRED Clip Maker

Um eine Animationssequenz zu erzeugen kann der anwendungseigene „Clip Wizard“ verwendet bzw. vorab erstellte Bewegungskurven integriert werden. Diese Bewegungskurven werden mit dem Curve Editor erstellt.

Des Weiteren können mit dem Clip Maker weitere Animationen, Kameratracks, Ansichtspunkte, Variantenwechsel und Sequenzen hinzugefügt werden. So ist es möglich, gesamte Präsentationsvorführungen zu erstellen.

5.3.1.6 Curve Editor

Der Curve Editor dient zur Erstellung einzelner Animationssequenzen, welche im Clip Marker zu einem Clip zusammengefügt werden können. Die Erstellung erfolgt über das Anlegen verschiedener Bewegungspunkte des zu animierenden Objekts. Durch die Verbindungslinie dieser Punkte kann die Geschwindigkeit und der zeitliche Ablauf der Animation bestimmt werden. Die Anpassung der Bewegungspunkte erfolgt über Transformation von Translation und Rotation des Objekts. Ebenso können die Größe und die Sichtbarkeit des Objekts variiert werden. In Abbildung 46 ist die Benutzeroberfläche des Curve Editor mit allen Einstellungsmöglichkeiten dargestellt.

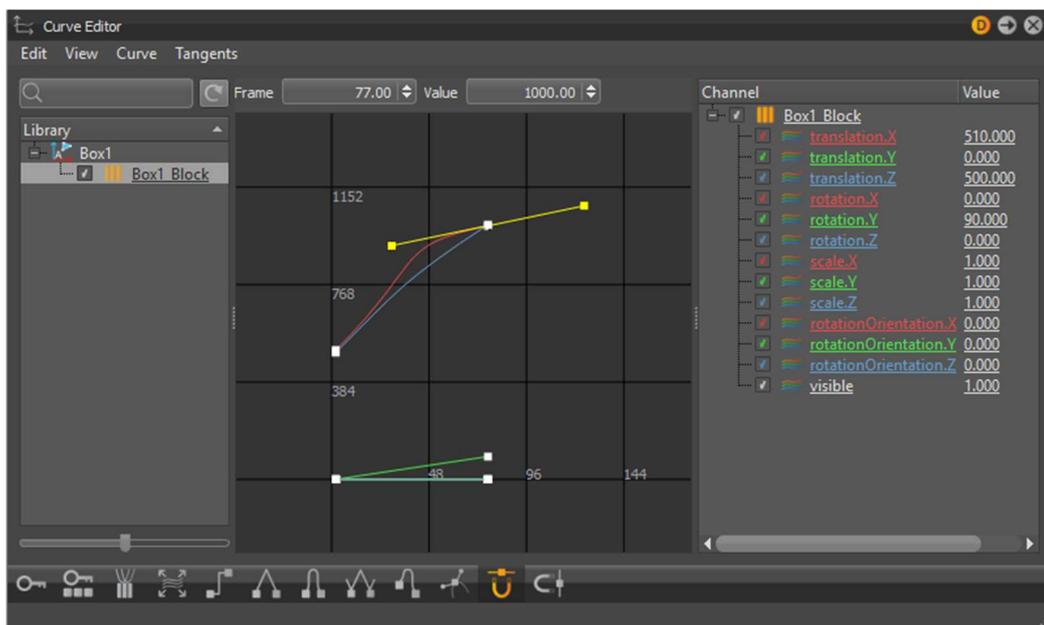


Abbildung 46 VRED Curve Editor

Die einzelnen Animationssequenzen können im OSB Dateiformat gespeichert, sowie bereits vorhandene in das Projekt geladen werden. So ist es auch hier möglich, die Animationssequenzen in einer Bibliothek zu verwalten und für mehrere Projekte zu verwenden.

5.3.1.7 Textur Editor

Der Textur Editor hilft dabei, Texturen auf Objekten zu positionieren. Mit dem Editor können der VR-Szene jedoch keine Texturen hinzugefügt werden. Das Hinzufügen hat über den Material Editor zu erfolgen, da Texturen immer einem Material zugewiesen werden müssen.

Der Textur Editor (Abbildung 47) listet alle Texturen mit dessen zugewiesenen Attributen eines ausgewählten Materials auf und erlaubt es diese zu bearbeiten. Standardmäßig verwendet VRED die UV-Koordinaten eines importierten Objekts, sofern solche vorhanden sind. Die Änderungen an der Textur werden sofort auf das Objekt angewandt und bedürfen keiner zusätzlichen Speicherung. Die Platzierung der Textur kann durch numerisch Eingabe als auch mithilfe eines Platzierungssteuerkreuzes im Ansichtsfenster erfolgen. Die bessere Lösung eine Textur zu platzieren bietet der UV-Editor, welcher im nächsten Punkt näher beschrieben wird.

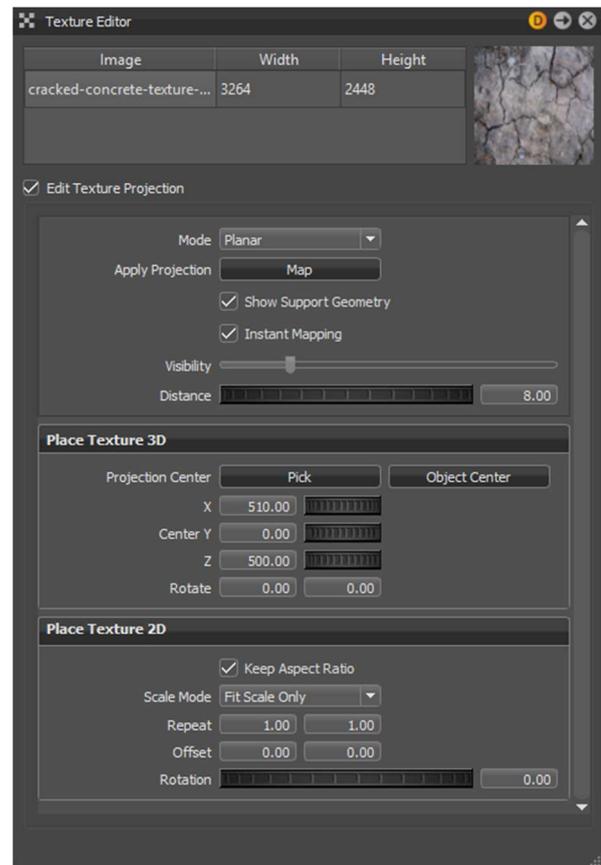


Abbildung 47 VRED Textur Editor

5.3.1.8 UV-Editor

Der UV Editor dient zur Anpassung von Texturen aller Art auf eine beliebige Geometrie. Es ist darauf zu achten, dass im Material Editor die richtige Methode ausgewählt ist, dass die Änderungen auch wirksam sind. Die Anpassung der Textur erfolgt direkt über die NURBS (nicht-uniforme rationale B-Splines) der Geometrie des ausgewählten Objektes, welche durch das tesselierte Gitternetz im Editor dargestellt werden (Abbildung 48). Somit wird eigentlich nicht die Textur auf die Geometrie, sondern die Geometrie auf die Textur angepasst. Daraus schließend funktioniert der UV-Editor nur mit Geometrien, welche mit VRED tesseliert wurden. Vorsicht: Wird die Geometrie erneut tesseliert, bleiben ihre UV-Koordinaten nicht erhalten und Änderungen an den UVs müssen erneut vorgenommen werden. Durch die verschiedenen Auswahlfunktionen kann das gesamte Objekt, aber auch einzelne Inseln oder Knoten, ausgewählt und an die Textur angepasst werden. Die Änderungen werden sofort am Objekt dargestellt und auch übernommen.

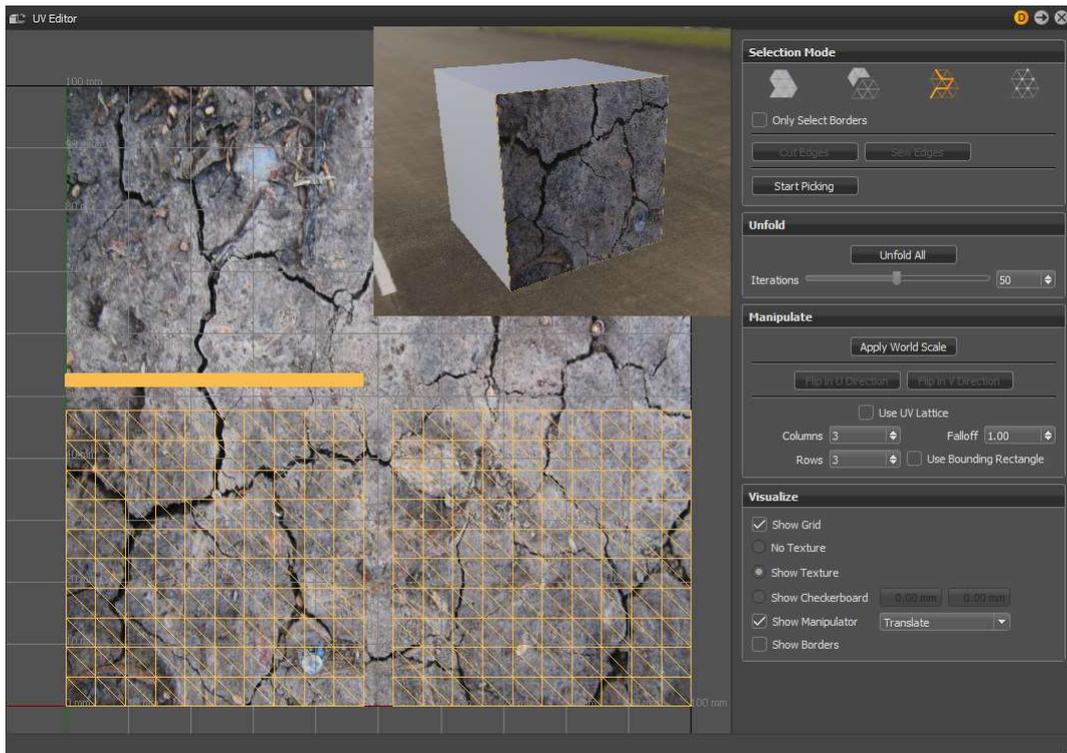


Abbildung 48 VRED UV-Editor

5.3.1.9 Overlay Editor

Mit dem Overlay Editor kann der Benutzer eine Einblendung über der aktuellen Szene erstellen. Die Einblendung kann aus einem Bild oder einem Text bestehen und die eigentliche Szene vollständig oder teilweise überdecken.

Einblendungen können aktiv oder inaktiv gesetzt werden. Ebenso kann im Editor ein Switch erzeugt werden, welcher das schnelle Umschalten zwischen verschiedenen Einblendungen und der VR-Szene ermöglicht.

5.3.1.10 Node Editor

Der Node Editor bietet alle Informationen zu einem ausgewählten Knoten in der aktuellen Szene. Es bietet die Möglichkeit, detaillierte Informationen zu ausgewählten Knoten anzuzeigen, dessen Werte zu ändern und benutzerdefinierte Attribute an den ausgewählten Knoten anzuhängen.

5.3.2 Verwaltung der Projektdateien

Wie aus dem Prozess und aus dem Autodesk VRED Arbeitsumfeld ersichtlich, fällt während der Aufbereitung einer VR-Szene eine sehr große Datenmenge an. Um die Übersicht darüber zu bewahren, ist es ein logischer Schritt, eine einheitliche Verwaltungsstruktur festzulegen, welche in weiterer Folge bei allen Projekten berücksichtigt wird. In Abbildung 39 ist bereits ein erster Teil der Verwaltungsstruktur, bezogen auf das VRED Umfeld, dargestellt, auf dessen Komponenten auch gleich näher eingegangen wird. Abschließend wird noch auf die vordefinierte projektbezogene Ordnerstruktur zur Datenverwaltung Bezug genommen.

5.3.2.1 *Asset Manager Bibliothek*

Wie bereits in Kapitel 5.3.1.1 erwähnt, bildet der Asset Manager das Bindeglied zwischen der Visualisierungssoftware VRED und vielen Objekten, speziell den Materialien, Umgebungen und gesicherten Geometrien. Diese Objekte liegen in der Asset Bibliothek, die aus einer einfachen Ordnerstruktur besteht (Abbildung 49), deren Abbild jedoch vom Asset Manager vorgegeben wird. Es können beliebig viele Asset Bibliotheken erstellt und im VRED Einstellungsmenü über ihren Verzeichnispfad geladen werden. So ist es möglich, projekt-spezifische Asset Bibliotheken anzulegen.

Durch den einfachen Aufbau der Asset Bibliothek als frei zugängliches Ordnerverzeichnis ist es auch möglich, die Objekte ohne die Verwendung von Autodesk VRED zu verwalten. Es ist jedoch nicht empfehlenswert ein Objekt, wie zum Beispiel ein Material, in den Unterverzeichnis eines anderen Projektes zu kopieren, da die Kopie dieselbe Identifikationsnummer wie das Original behält. Dadurch ist im Asset Manager das vervielfältigte Material zwar in beiden Projekten vorhanden, jedoch wird es nur im aktuell geladenen Verzeichnis dargestellt. Aus diesem Grund sollten, für die Verwaltung der Asset Bibliotheken, nur die von Autodesk bereitgestellten Lösungen verwendet werden.

Die Option mehrere Asset Bibliotheken erstellen zu können verschafft der Verwaltung einen erheblichen Vorteil. Dadurch kann eine Master Asset Bibliothek erstellt werden, die allen Projekten übergeordnet ist und alle relevanten Objekte beinhaltet. Diese Master Asset Bibliothek ist, um der Gefahr das Objekte überschrieben werden zu entgehen, schreibgeschützt und wird von einer einzelnen Person aus dem VR-Team verwaltet. Soll ein neu erstelltes Objekt der Master Asset Bibliothek hinzugefügt werden, ist dies nur von diesem einen Mitarbeiter möglich.

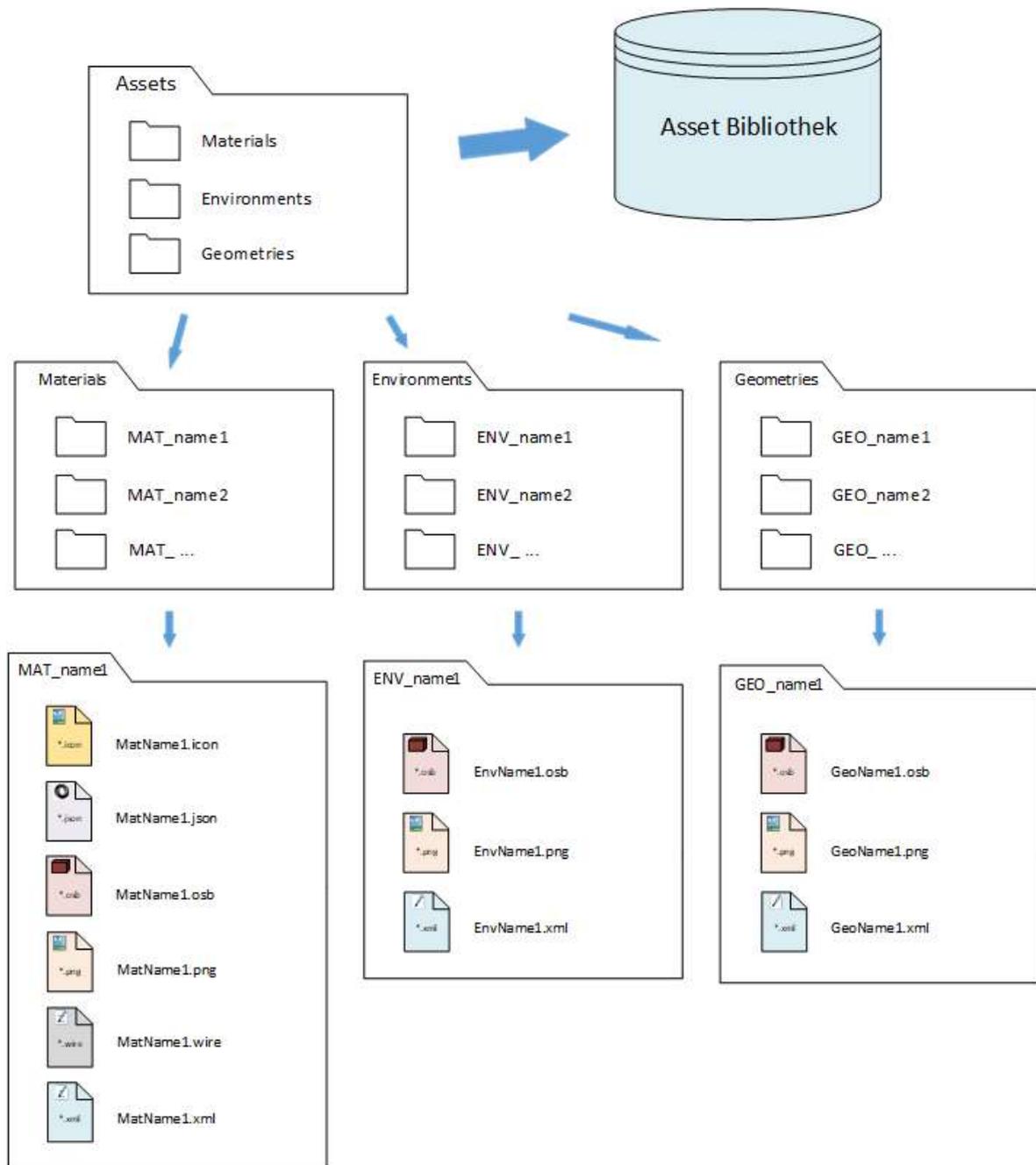


Abbildung 49 Ordnerstruktur einer Asset Bibliothek

Master Asset Bibliothek

Wie bereits erwähnt dient die Master Asset Bibliothek als schreibgeschützte Asset Hauptbibliothek. Bei der Aufbereitung einer VR-Szene können aus ihr die benötigten Objekte gezogen und in der projektbezogenen Asset Bibliothek abgelegt werden. Die Bibliothek wird von einer Person aus dem VR-Team betreut, welche bei Bedarf neue Objekte hinzufügen kann.

Projekt Asset Bibliothek

Die projektbezogene Asset Bibliothek wird am Beginn der Erstellung einer neuen VR-Szene im Projektordner angelegt. Diese Bibliothek wird mit den benötigten Objekten aus der Master Asset Bibliothek befüllt und dient für die Aufbereitung der Szene als priorisierte Bibliothek. Alle Materialien, Umgebungen und Objekte, die während der Arbeiten an der Szene, aus der Master Asset Bibliothek oder aus anderen Projekten herangezogen werden, müssen in dieser projektbezogenen Asset Bibliothek abgelegt werden. Nur so kann eine vollkommene Nachvollziehbarkeit der einzelnen Projekte erzielt werden.

5.3.2.2 VISU Hauptbibliothek

In der VISU Hauptbibliothek werden alle weiteren Objekte abgelegt und verwaltet, die nicht durch die Asset Bibliothek berücksichtigt werden (Abbildung 50). Dazu zählen unter anderem Texturen, Szenenkurven, Kameraeinstellungen, Blickpunkte und Lichtprofile. Die VISU Hauptbibliothek wird für jedes Projekt separat im Projektordner angelegt.

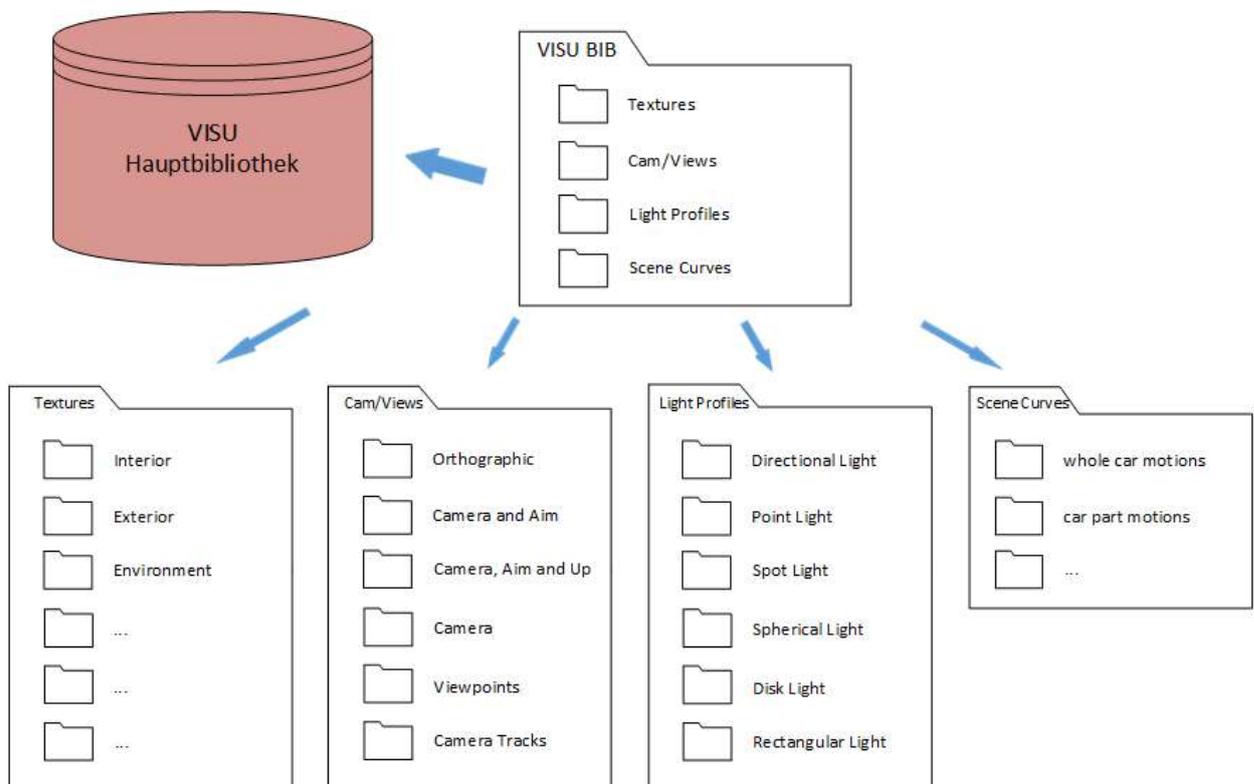


Abbildung 50 Ordnerstruktur der VISU Hauptbibliothek

5.3.2.3 VISU Projekt Bibliothek

Die VISU Projekt Bibliothek beinhaltet abgeschlossenen Projekte, anwendungsbezogene Templates und bereits fertig visualisierte Fahrzeugmodelle. Sie ist nicht projektbezogen und wird vom VR-Team Experten betreut. Durch das Anfertigen einer Excel-Datenbank, welche alle Eigenschaften und Funktionen der einzelnen Projekte und Templates beinhaltet, besteht ständig ein Überblick über die bereits angefertigten Arbeiten.

5.3.2.4 Projektbezogene Ordnerstruktur

Die projektbezogene Ordnerstruktur umfasst alle Objekte, die zur Erstellung einer VR-Szene benötigt werden (Abbildung 51). Neben den verschiedenen Bibliotheken beinhaltet sie auch ein Geometrieverzeichnis, in welches die Geometriedateien des Fahrzeugmodells im OSB-Format abgelegt werden. Auf diesen Geometrie-Ordner greift in späterer Folge der automatisierte Update Prozess zu. Werden einer Szene für Vergleichsanwendungen ein bereits visualisiertes Fahrzeugmodell einer älteren Version hinzugefügt, so wird dieses Modell im „Fahrzeugmodelle“-Ordner abgelegt. Der „VR-Szene“-Ordner dient als Speicherort für die im Aufbau befindlichen Szene.

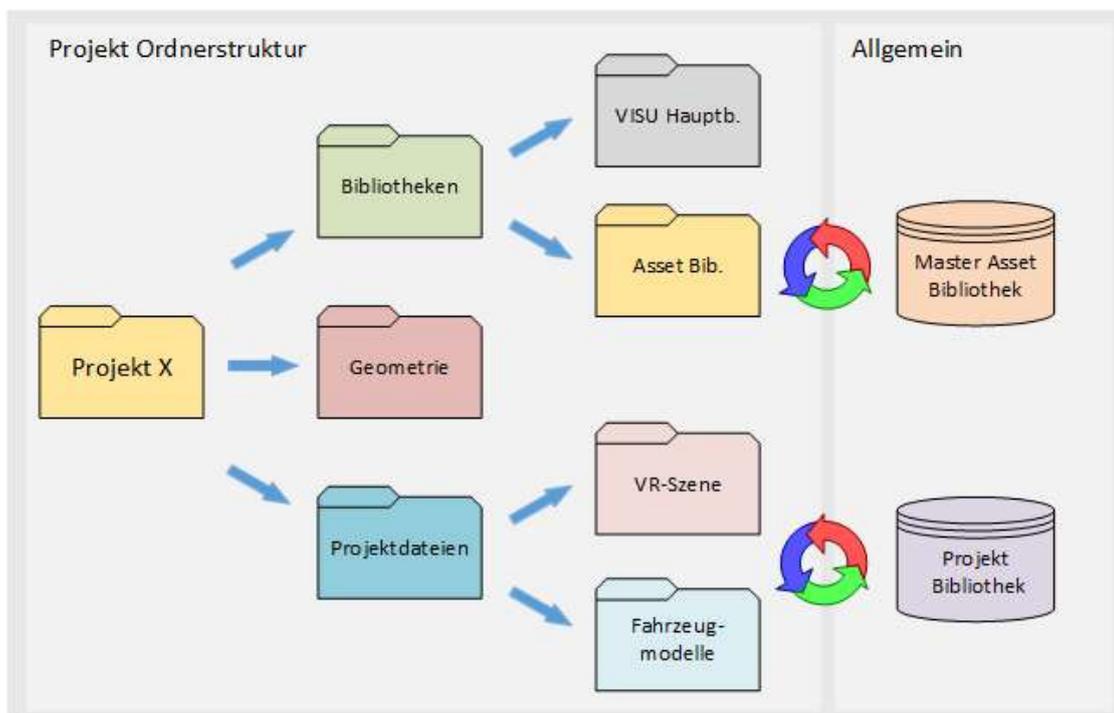


Abbildung 51 Vordefinierte projektbezogene Ordnerstruktur

5.4 Der automatisierte Update Prozess

Während der Entwicklungsphase eines Fahrzeugs kommt es täglich zu Veränderungen und Überarbeitungen der Geometrien und durch den steigenden Detailgrad werden dem Modell sukzessiv Bauteile hinzugefügt. Um eine Übersicht darüber zu erhalten, wird, wie bereits erwähnt, jedem Bauteil als Identifikation ein projektbezogener Name vergeben (Abbildung 52). Dieser Name besteht aus drei Teilen, dem Bauteilnamen, der Versionsnummer und der Revisionsnummer. Diese drei Teile sind üblicherweise durch einen Unterstrich und einen Punkt voneinander getrennt. In einzelnen Projekten werden auch zwei Unterstriche dazu verwendet. Der Bauteilname besteht aus zwölf Zeichen, wobei die ersten Zeichen aus Buchstaben und die restlichen Zeichen aus Ziffern bestehen. Der Bauteilname verändert sich im Laufe der gesamten Entwicklung nicht. Die Versions- und Revisionsnummer bestehen aus jeweils drei weiteren Ziffern. Wird ein Bauteil durch einen Konstrukteur innerhalb einer Entwicklungsphase verändert und neu gesichert, so wird das Bauteil mit fortlaufender Revisionsnummer neu im PDM abgelegt. Wird eine Entwicklungsphase beendet, werden die Bauteile für diese Phase gefroren und für den nächsten Entwicklungsschritt mit neuer Versionsnummer bereitgestellt.

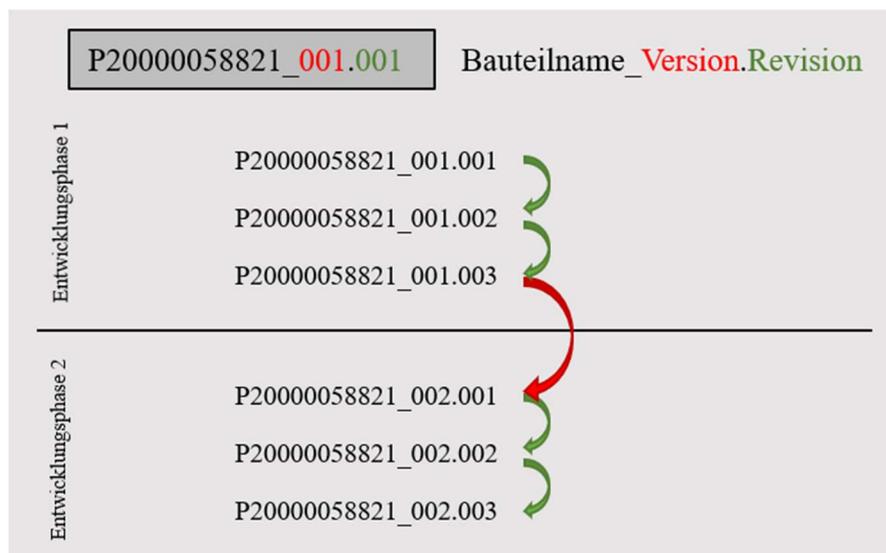


Abbildung 52 PDM Versionsmanagement

In manchen VR-Anwendungsfällen wird nun verlangt, dass die Untersuchungen immer mit dem aktuellsten Entwicklungsstand des Fahrzeugs durchgeführt werden. Die Anzahl der Bauteile, die sich täglich verändern oder hinzugefügt werden, ist vor allem in den späteren Entwicklungsphasen enorm und so benötigt es einige Stunden an Vorbereitungszeit, bis das

Fahrzeugmodell auf den neuesten Stand aktualisiert ist. Um diesen Vorgang so effektiv wie möglich zu gestalten, ist der Integrationsprozess schon dahin konzipiert, dass in späterer Folge ein automatischer Update Prozess integriert werden kann. Der Plan ist es, eine Anwendung in den Prozess zu implementieren, welche die PLMXML-Datei des aktuellen PDM Exports mit einem vorhergegangenen Fahrzeugentwicklungsstand vergleicht und eine PLMXML-Datei erstellt, welche die Unterschiede beinhaltet. Mit Hilfe einer weiteren Anwendung sollen so nur die neuen und veränderten Bauteilgeometrien der bestehenden VR-Szene hinzugefügt werden.

5.4.1 Definition der Vergleichsparameter

Um den Inhalt der relevanten PLMXML-Dateien zielführend vergleichen zu können, müssen zuerst Vergleichsparameter festgelegt werden. Als erste Parameter werden hierzu der Bauteilname, die Versions- und die Revisionsnummer herangezogen. Zusätzlich zu diesen Parametern wird auch die Lagebestimmung der einzelnen Bauteile als Kenner berücksichtigt. Ohne die Lagebestimmung ist eine eindeutige Identifizierung der Bauteile nicht möglich, da wir sonst keine Information darüber erhalten, wie oft ein identes Teil im Fahrzeugmodell verbaut und somit in der Modellstruktur vorhanden ist. Des Weiteren könnte an einem Bauteil nur die Position, aber nicht die Version oder Revision, verändert worden sein, was ohne die Berücksichtigung der Lagebestimmung unentdeckt bliebe. Die einzelnen Parameter sind in separaten Zeilen in der PLMXML-Datei gelistet und können somit einfach ausgelesen werden.

5.4.2 Erhaltung der Arbeitsstruktur

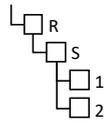
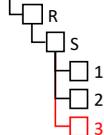
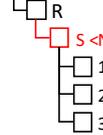
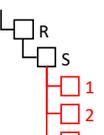
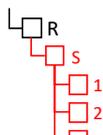
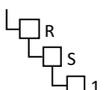
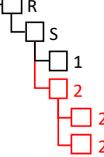
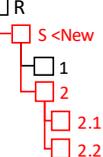
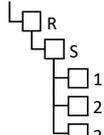
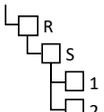
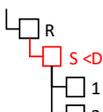
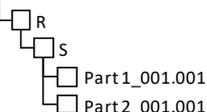
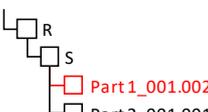
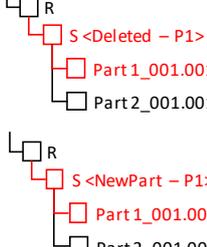
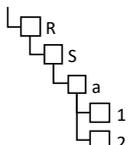
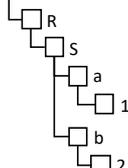
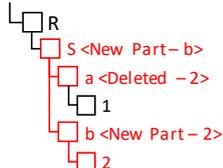
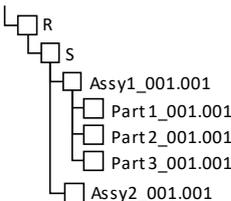
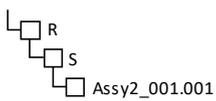
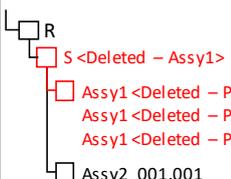
Eine der größten Herausforderungen, die einem vollautomatisierten Update Prozess im Wege steht, ist die verwendete Modellstruktur. Diese wird im PDM angelegt, beim Export mit den CAD-Geometriedateien befüllt ausgegeben und in späterer Folge in das Visualisierungsprogramm übernommen. Während des Prozesses verändert sich die Modellstruktur bzw. wird sie während der Visualisierung angepasst. Dadurch kommt es zu Unterschieden zwischen der exportierten PDM Struktur und der späteren Zielstruktur im Visualisierungsprogramm. Dies macht einen Vergleich der beinhaltenden Objekte sehr schwierig bzw. eine Zuordnung von neuen oder geänderten Objekten in die veränderte Modellstruktur beinahe unmöglich. Bezogen auf die Datenbereitstellung ist es jedoch möglich, zwei verschiedene

PDM-Exporte zu vergleichen, da sich deren Struktur im Laufe des Prozesses nicht allzu drastisch verändert. Die üblichen Veränderungen in dieser Phase sind Bauteilaktualisierungen oder das Hinzufügen/Entfernen von einzelnen Bauteilen und Baugruppen, hervorgerufen durch den steigenden Detailgrad des Fahrzeugmodells. Es werden selten grundlegende Veränderungen an der Modellstruktur vorgenommen. In der späteren Visualisierung der CAD Dateien in Autodesk VRED wird die Struktur jedoch unweigerlich verändert. So werden zum Beispiel für Animationen Bauteile verschiedener Baugruppen in eine neue Gruppe zusammengefasst. Daher ist man zu dem Entschluss gekommen, den Update Prozess in zwei Teile, die Datenbereitstellung und das Einfügen der Dateien in Autodesk VRED, zu unterteilen.

5.4.2.1 Dateienbereitstellung

Wie bereits erwähnt werden in der Phase der Datenbereitstellung zwei PLMXML-Dateien verschiedener Arbeitsstände miteinander verglichen. Für diesen Vergleich wird eine zusätzliche Anwendung im Integrationsprozess, zwischen der Datenkonvertierung und dem Einfügen ins Visualisierungsprogramm, integriert. Der gesamte Updateprozess ist in Abbildung 53 dargestellt. Diese Anwendung (Hinweis 1 in Abbildung 53) vergleicht die relevanten Zeilen im Dateicode und gibt die Unterschiede mit einer zusätzlichen Kennung in einer neuen PLMXML-Datei aus, die für den nachfolgenden Import der Geometriedateien verwendet wird. Bei der zusätzlichen Kennung handelt es sich um einen Zusatz, um welche Veränderung es sich handelt. Hierbei wird unterschieden, ob ein Bauteil bzw. eine Baugruppe neu hinzugekommen ist, entfernt wurde oder die Position bzw. die Transformation verändert wurde. Bei einer Verschiebung eines Objektes wird es zuerst an der alten Stelle im Strukturbaum als gelöscht und an der neuen Stelle als neues Objekt markiert. Um alle möglichen Fälle von Veränderungen zu erkennen und sie somit in die Programmierung der Anwendung aufzunehmen, wurden sie mit der obligaten Strukturänderung in einer Tabelle zusammengefasst. In Tabelle 6 ist ein Auszug der für die Dateienbereitstellung relevanten Fälle dargestellt. Die Spalten „Struktur Alt“ und „Struktur Neu“ zeigen hierbei die Modellstruktur der alten bzw. neuen PLMXML-Datei. Die Spalte „Delta PLMXML“ stellt den Inhalt der daraus resultierenden PLMXML-Datei dar. Neben der neuen Struktur ist auch die Kennung für den Import angegeben. Nur bei Fall 2, dem ersten Import, wird der relevante Knoten mit keinem Kenner versehen, da der Erstimport über eine eigene Funktion und nicht über ein Update ausgeführt wird.

Tabelle 6 Änderungsfälle bezüglich der Dateienbereitstellung

Nr.	Szenario	Struktur Alt	Struktur Neu	Delta PLMXML
1	Neues Part über Teamcenter			
2	Alles neu - erster Import	/		
3	Neues Assembly über Teamcenter			
4	Ein Part wird aus Teamcenter gelöscht			
5	Update Part Revisionsnummer in Teamcenter			
6	Veränderte Teamcenter Export Struktur			
7	Assembly aus Teamcenter gelöscht			

5.4.2.2 Einfügen der Dateien in VRED

Die Kenner sind nun wichtig, um die neuen bzw. veränderten Objekte dem Import-Knoten (3) und in späterer Folge dem Modell-Knoten (4) im VRED Strukturbaum (2) zuordnen zu können. Dazu ist der Import-Knoten in mehrere Unterknoten unterteilt, auf welche die Objekte bezüglich ihrer Kennung zugeteilt werden. Bis zu diesem Arbeitsschritt ist der automatisierte Update-Prozess zum Ende der Masterarbeit bereits einsatzbereit. Die weitere Zuordnung der Objekte muss im Visualisierungsprogramm jedoch weiterhin händisch erfolgen.

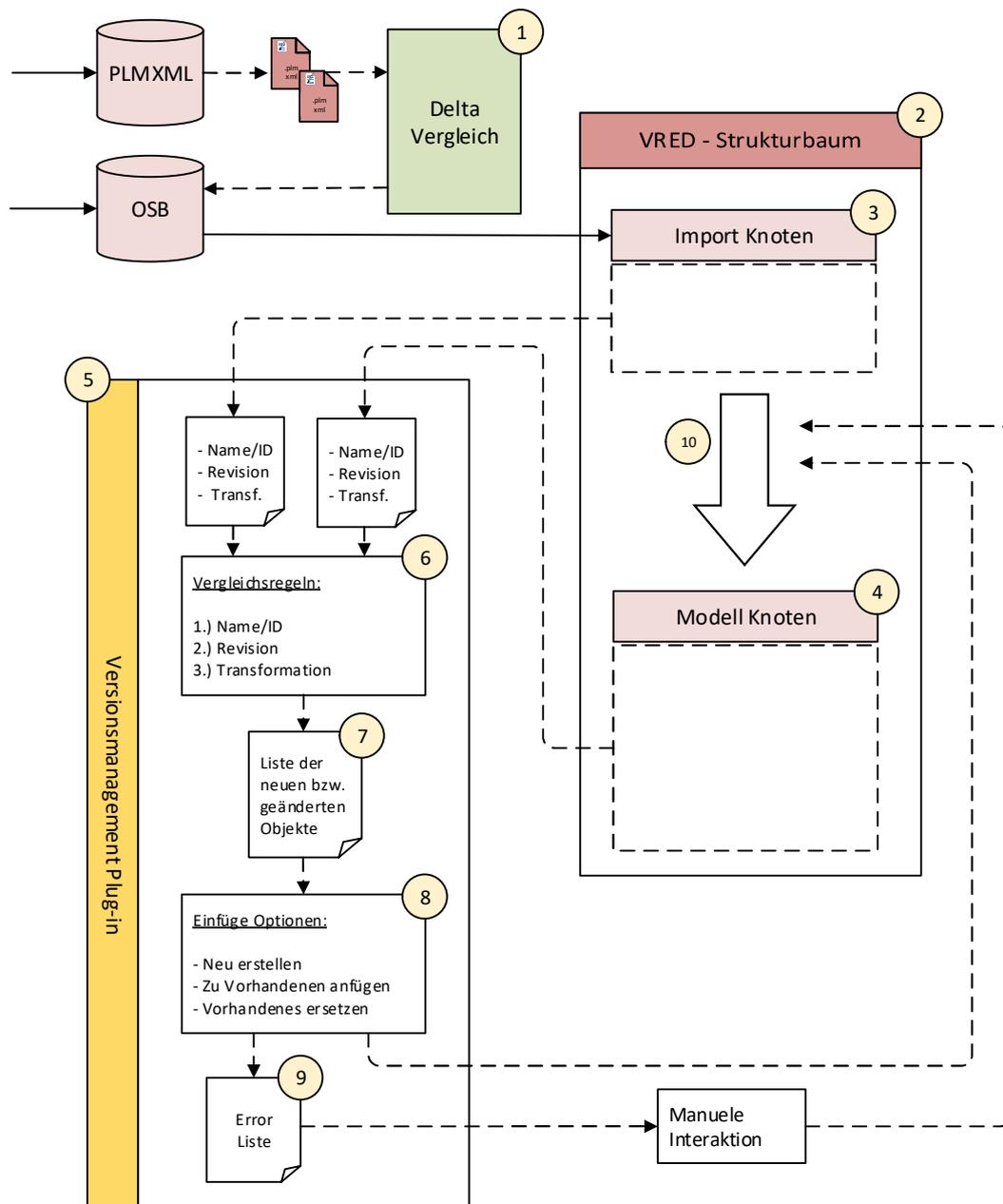


Abbildung 53 Entwurf des Update-Prozesses

Geplant ist es, in die VRED Umgebung ein Plug-in (5) zu integrieren, welches die im Import-Knoten abgelegten Objekte selbständig dem Modell-Knoten zuordnet (10). Diese Anwendung soll nun den Inhalt des Import-Knotens mit jenen des Modell-Knotens vergleichen (6) und übereinstimmenden Objekte in einem Ausgabefenster auflisten (7). Bei übereinstimmenden Bauteilnamen wird als nächstes die Revision verglichen und so eine Versionsänderung lokalisiert. Nach der Lokalisierung der Objekte soll es für nicht eindeutig zuweisbare Objekte möglich sein, über einen Eingabe-Assistenten (8) das weitere Vorgehen auszuwählen. Die Objekte, bei denen eine Zuweisung zum Modell-Knoten scheitert, werden in einer Fehlerliste (9) ausgewiesen und können vom Visualisierer händisch in die Modellstruktur des Modell-Knoten eingefügt werden.

Die Schwierigkeit besteht nun darin, die an die Objekte während der Modellaufbereitung vergebenen Zusatzinformationen, wie zum Beispiel Materialzuweisungen oder die Rotationsachse für Animationen, an die neuen Objekte zu übergeben. Dazu müssten diese Informationen beim Vergleich zwischen dem Import- und dem Modell-Knoten zusätzlich ausgelesen und in die Metadaten der eingefügten Objekte geschrieben werden. Es wurden dahingehend mehrere Ansätze theoretisch ausgearbeitet, jedoch bis dato noch kein Lösungsansatz praktisch umgesetzt. Als zu komplex erweist sich der dafür notwendige Vergleich der neuen Modellstruktur aus dem Import mit der veränderten Modellstruktur des aufbereiteten Fahrzeugmodells. Ein Beibehalten der Modellstruktur ist durch den ständigen Anstieg der Bauteilanzahl und den Visualisierungsarbeiten am Fahrzeugmodell leider nicht möglich. Ein weiterer Ansatz wäre es, eine direkte Schnittstelle zwischen dem Visualisierungsprogramm Autodesk VRED und der Produktdatenmanagement-Software Siemens Teamcenter zu bilden, um das Versionsmanagement direkt im PDM durchzuführen und die für die Visualisierung benötigten Informationen direkt im PDM den Objekten zuzuweisen. Eine solche Schnittstelle wird jedoch ohne die Mithilfe der softwarebereitstellenden Firmen nur schwer bis unmöglich umzusetzen sein.

5.4.2.3 Unterteilung des Import-Knotens

Um dem Visualisierer eine gute Übersicht über die neu hinzugefügten Objekte zu bieten und eine geordnete Importstruktur für den späteren automatisierten Update-Prozess zu haben, ist der Import-Knoten in mehrere Unterknoten unterteilt. Das in Kapitel 6 beschriebene Master Template verfügt bereits über diese spezielle Struktur, wobei der Haupt-Import-Knoten als

„T_Data_In“ bezeichnet wird. Das T am Anfang der Bezeichnung dient als Hinweis, dass dieser Strukturknoten unter keinen Umständen gelöscht werden darf. Die weitere Unterteilung erfolgt in einen „T_Data_In_Deleted“, „T_Data_In_Changed“ und „T_Data_In_New“-Knoten. Diesen Knoten werden die Objekte bezogen auf ihre Kenner, welche sie über den PLMXML-Datei Vergleich erhalten, zugeordnet. Neben der Übersicht dient dieses Vorgehen auch der Prozesssicherheit, falls es während dem Importvorgang zu einem Fehler oder Abbruch kommt. In Abbildung 54 ist die Importstruktur in Autodesk VRED dargestellt.

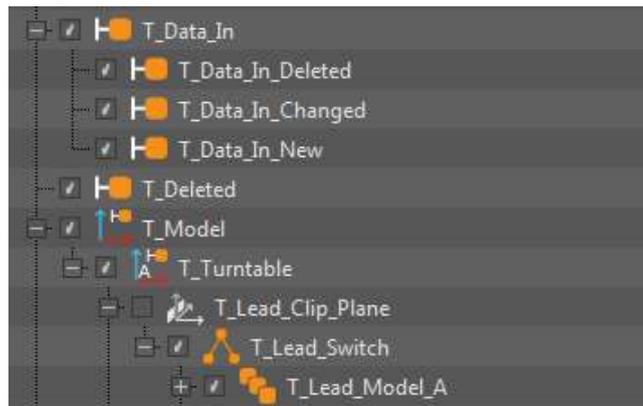


Abbildung 54 Importstruktur des VRED Master Template

Neben den „T_Data_In“ Import-Knoten gibt es noch einen weiteren Knoten im Strukturbaum, der eine besondere Rolle für den Updateprozess übernimmt. Dabei handelt es sich um den „T_Deleted“-Knoten. Dieser Knoten wird jedoch nicht über den Updateprozess mit neuen Objekten befüllt, sondern dient als Container für Objekte, welche aus dem Fahrzeugmodell entfernt wurden und wird vom Versionsmanagement Plug-in ausgelesen. Wird ein gelöscht Objekt nicht in diesen Knoten verschoben, so würde es beim nächsten Updatevorgang als neues Objekt in die VR-Szene geladen werden. Die Objekte müssen dazu jedoch nicht vollständig mit ihrer Geometrie in diesem Knoten abgelegt werden. Es ist vollkommen ausreichend, wenn der Hauptknoten des Objektes mit Namen, Versions- und Revisionsnummer im Knoten vorhanden ist.

6 Erstellung eines Visualisierungs-Template

Der Aufbau der VR Szene nimmt den größten Arbeitsaufwand und somit auch den größten Zeitaufwand im Visualisierungsprozess ein. Dazu gehört nicht nur die Aufbereitung des Fahrzeugmodells, sondern auch die Bereitstellung der virtuellen Umgebung und die Implementierung spezieller Werkzeuge und Funktionen. Um diesen Zeitaufwand zu verkürzen und dadurch die Effektivität des Prozesses zu erhöhen, wurde ein Visualisierungs-Template aufgebaut. Dieses Template beinhaltet alle zurzeit verfügbaren Funktionen und Werkzeuge und kann beliebig erweitert werden. Eine vordefinierte Arbeitsstruktur ermöglicht es, ein virtuelles Fahrzeugmodell in das Template zu laden und sofort alle Funktionen darauf anzuwenden. Die Arbeitsstruktur des Template ist auch bereits auf den automatisierten Updateprozess vorbereitet, wodurch sich das Template auch gut zur Aufbereitung eines virtuellen Fahrzeugmodells eignet. In den nächsten Punkten werden der Aufbau und die Funktionen des Template näher beschrieben. Des Weiteren wurde für das Template eine Benutzungsanleitung erstellt, welche in den Beiblättern dieser Arbeit einsehbar ist.

6.1 Aufbau des VRED Master Template

Zu Beginn des Aufbaus wurde eine Datenstruktur festgelegt. Dieser sogenannte Szenengraph muss ein geordnetes Arbeiten in der VR Szene ermöglichen und alle Knoten und Geometrien, welche für die Funktionalität der Funktionen und Werkzeuge benötigt werden, enthalten. Diesen Elementen wird der Benennung als Kennung ein „T_“ vorgesetzt, sodass sie nicht versehentlich aus der VR Szene entfernt werden. Ebenso dürfen die Benennungen der Knoten nicht verändert werden, da diese direkt mit dem Skript der Szene verknüpft sind.

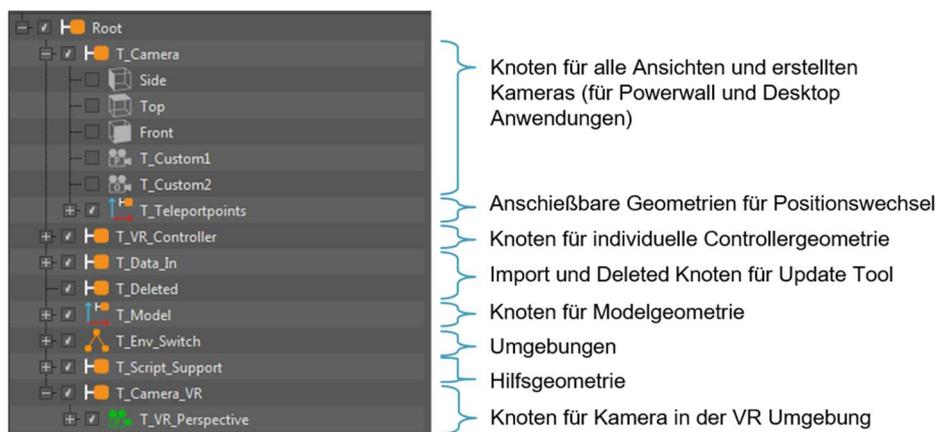


Abbildung 55 Szenengraph des VRED Master Template

T_Camera

Der „T_Camera“ Knoten (Abbildung 55) dient als Container für alle Kameras, welche mit dem Kamera-Editor erstellt werden. In diesen Kameras können bei Bedarf definierte Ansichtspunkte und animierte Kamerafahrten erzeugt werden. Sie sind jedoch nur für Durchführungen an der Powerwall oder direkt am Arbeitsbildschirm anwendbar. Für interaktive Anwendungen mittels einer VR Brille dient zur Erstellung von vordefinierten Ansichtspunkten das Teleport Plug-in.

Die Szene verfügt im Vorhinein bereits über einige definierte Positionen, welche um das Fahrzeugmodell angeordnet sind. Während den Arbeiten in der virtuellen Szene können diese Positionen, die über eine Geometrie gekennzeichnet sind, interaktiv mit dem Controller angewählt und dadurch Positionswechsel vollzogen werden. Die dazu benötigten Geometrien liegen im „T_Teleportpoints“ Knoten.

T_VR_Controller

In diesem Knoten sind die benutzerdefinierten Geometrien für die Eingabegeräte der VR Szene, wie zum Beispiel die beiden Handcontroller, hinterlegt. Die einzelnen Geometrien sind mit dem Skript der Szene verknüpft und so passt sich zum Beispiel das Aussehen des Controllers in der VR Szene dem verwendeten Werkzeug an.

T_Data_In, T_Deleted

Diese beiden Knoten dienen dem automatisierten Updateprozess. Deren Funktion wird in Kapitel 5.4.2.3 genauer erläutert.

T_Model

Im „T_Model“-Knoten (Abbildung 56) werden die Geometrien des virtuellen Fahrzeugmodells abgelegt. Dazu ist er des Weiteren in einen „T_Lead_Switch“ und „T_Comparison“-Knoten unterteilt. Der „T_Lead_Switch“-Knoten dient hierbei als Container für das Hauptmodell der Szene. In diesem Knoten liegen weitere fünf Unterknoten, welche bei Bedarf mit verschiedenen Fahrzeugmodellen befüllt und über eine Interaktion durchgewechselt werden können.

Der „T_Comparison“-Knoten ist für die „Comparison“ Funktion vorgesehen. Er kann ebenfalls fünf Fahrzeugmodelle aufnehmen, die ebenso interaktiv gewechselt werden können.

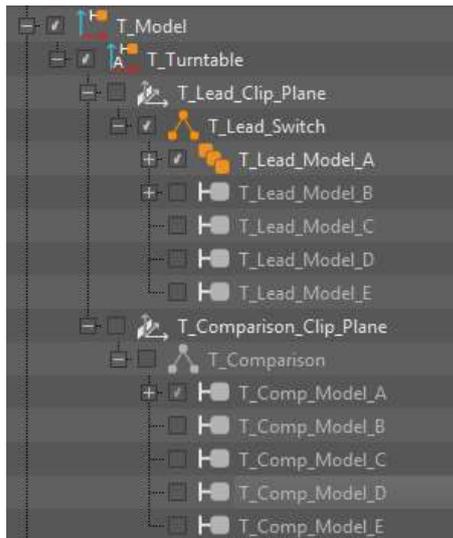


Abbildung 56 „T_Model“ Knoten

Wird die „Comparison“ Funktion aktiviert, so schalten sich der „T_Comparison“-Knoten und die beiden Clip Planes aktiv und die ausgewählten Fahrzeugmodelle können direkt in der virtuellen Szene verglichen werden.

Der „T_Turntable“-Knoten enthält die Einstellungen für die „Turntable“ Funktion der VR-Szene, welche es ermöglicht, alle Geometrien die in diesem Knoten liegen, um eine vordefinierte Achse zu rotieren.

T_Env_Switch

Im „T_Env_Switch“ Knoten sind die verschiedenen Umgebungen abgelegt (Abbildung 57). Diese können, wie die Fahrzeugmodelle, über einen Befehl interaktiv gewechselt werden. In die Master-Template-Szene sind vier Umgebungen (Red Bull Ring Spielberg, Potsdamer Platz Tag, Potsdamer Platz Nacht und Feld im Sonnenuntergang), die mittels einer Fotoaufnahme auf eine Kuppel projiziert werden, und das Design Haus, welches vollständig aus Geometrien aufgebaut ist, integriert.

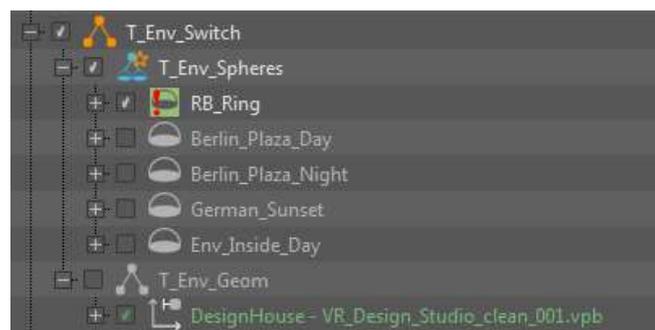


Abbildung 57 „T_Env_Switch“ Knoten

T_Script_Support

Der „T_Script_Support“ Knoten (Abbildung 55) dient als Container für Hilfsgeometrie, welche vom Skript der VR Szene benötigt wird, jedoch nicht in den anderen Knoten der Szene zuordbar ist.

T_Camera_VR

Im „T_Camera_VR“ Knoten (Abbildung 55) wird die vordefinierte Kamera, welche für die Ausgabe der Szene durch eine VR Brille angewendet wird, abgelegt. Diese Kamera mit der Bezeichnung „T_VR_Perspective“ wird beim Start der Szene selbständig über das Skript der Szene erstellt. Die Einstellungen der Kamera sind ebenfalls im Skript hinterlegt und werden der Kamera zugewiesen.

6.2 Funktionen des Master Template, Werkzeuge und Plug-ins

Wie bereits erwähnt, ist das Master-Template mit mehreren Funktionen ausgestattet. In den folgenden Punkten wird die Bedienung dieser Funktionen und deren Eigenschaften näher beschrieben.

6.2.1 Auswahl der Funktionen

Die Funktionen können interaktiv vom Nutzer in der VR Szene oder durch eine weitere Person, welche die VR Untersuchung unterstützt, über Tastenkombinationen angewählt werden. So ist es für einen erfahrenen Nutzer möglich, die VR Untersuchungen selbständig, ohne die VR Brille abzunehmen, durchzuführen und ebenso möglich neue Nutzer bei ihren Untersuchungen zu unterstützen.

6.2.1.1 Bedienung mittels Tastenkombinationen

Die Steuerung der Funktionen über die Tastenkombinationen kann zu jederzeit während einer VR Untersuchung durchgeführt werden. Es ist ratsam den Nutzer, welcher die VR Brille trägt, vor einem Wechsel dahingehend zu informieren, da es sonst, hervorgerufen durch die schnellen unvorhersehbaren Wechsel, zu Übelkeit beim Nutzer führen könnte. In Abbildung 58 sind die möglichen Tastenkombinationen mit deren Funktion dargestellt.

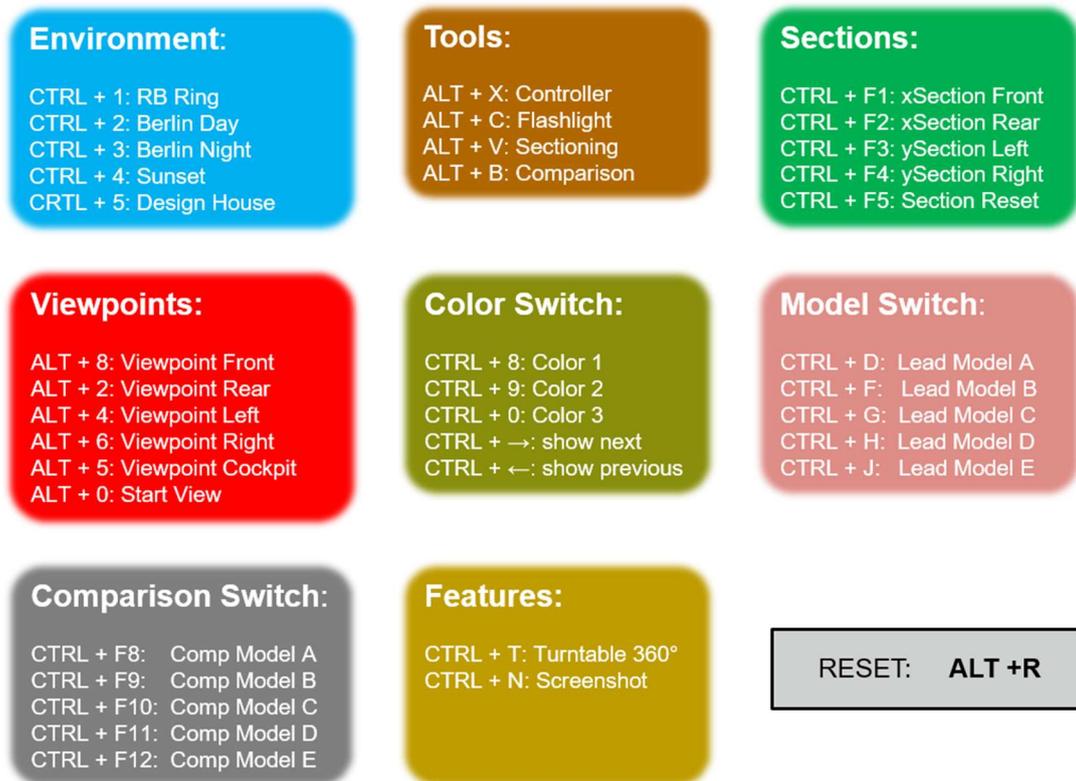


Abbildung 58 Tastenkombinationen zur Steuerung des Master-Template

6.2.1.2 Interaktive Bedienung des Master Template

Die interaktive Steuerung ermöglicht es die Funktionen direkt als Nutzer in der VR Szene auszuwählen und zu bedienen. Dazu wurde in der VR Template Szene ein Auswahlmenü (Abbildung 60) integriert, welches durch das Drücken der Griff Taste am linken Controller aufgerufen werden kann. In Abbildung 59 ist eine Beschreibung zu den Tasten des Controllers der HTC Vive VR Brille dargestellt.

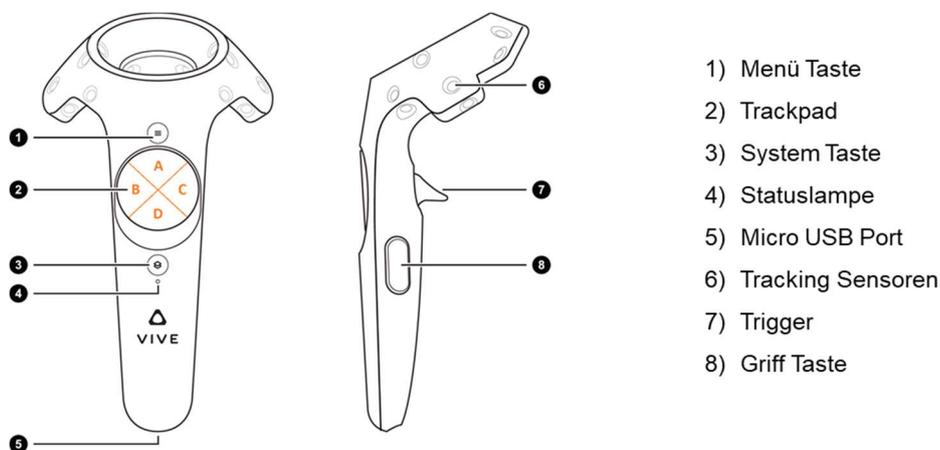


Abbildung 59 Controller der HTC Vive VR Brille (HTC, 2018)

Durch leichtes Ziehen des Triggers erscheint ein Lichtstrahl am oberen Ende des Controllers, mit welchem auf die Auswahlfläche der gewünschten Funktion im Menü gezielt werden kann. Das vollständige Drücken des Triggers bestätigt die Auswahl.

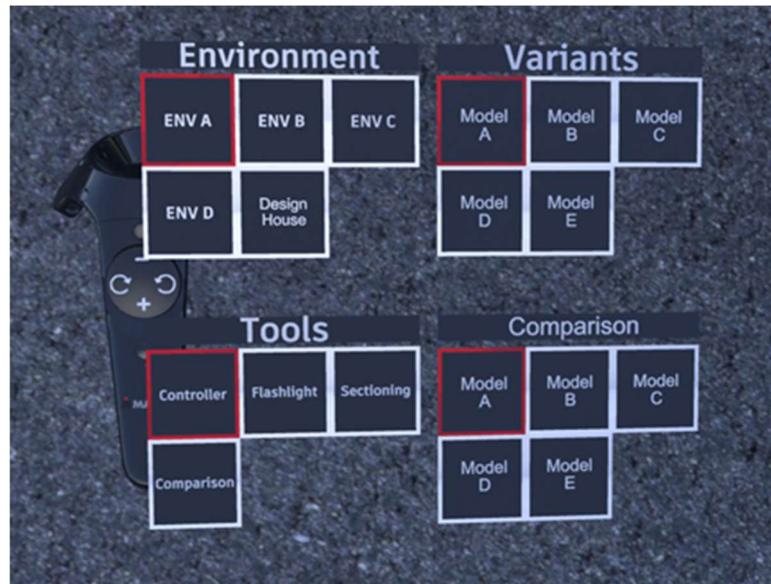


Abbildung 60 Interaktives Menü des Master Template

Nach der Auswahl einer Funktion kann das Menü mit einem weiteren Druck auf die Griff Taste wieder ausgeblendet werden, um die Sicht auf das Fahrzeugmodell während den Untersuchungen nicht einzuschränken.

6.2.2 Taschenlampe – „Flashlight“



Abbildung 61 VR Taschenlampe

Durch die Auswahl der Taschenlampenfunktion erhält der rechte Controller die Kontur einer Taschenlampe (Abbildung 61) und kann auch als solche in der VR Szene verwendet werden. Das Licht der Taschenlampe wird durch das Drücken des Trackpad ein- bzw. ausgeschaltet. Mit der Taschenlampe kann die Geometrie des virtuellen Fahrzeugmodells angeleuchtet werden, um Details besser erkennen zu können.

6.2.3 Schneiden des Fahrzeugmodells – „Sectioning“

Die Schneidefunktion ermöglicht es, das virtuelle Fahrzeugmodell in einer beliebigen Ebene zu schneiden. Die Ebene wird dabei durch das Führen des rechten Controllers definiert. Position und Winkel der Schnittebene sind dabei frei wählbar. Die Schnittkanten des Fahrzeugmodells werden durch eine gelbe Kontur hervorgehoben (Abbildung 62).

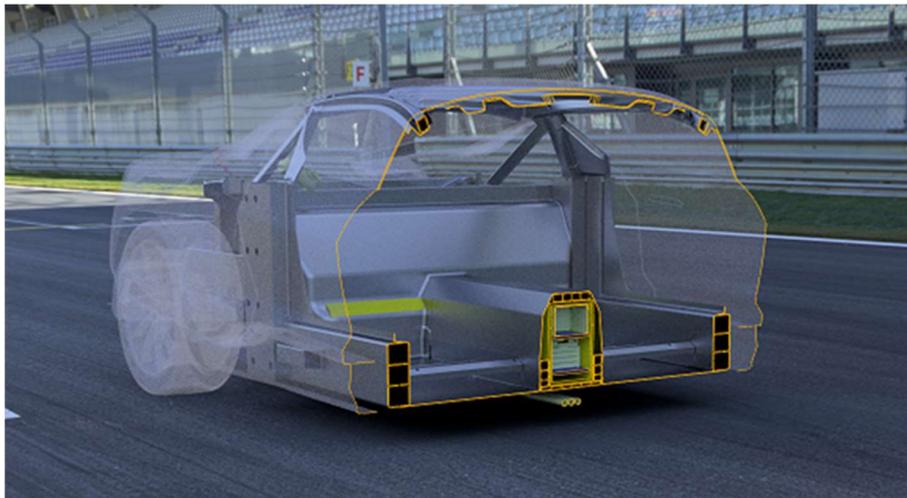


Abbildung 62 Schnitt des Fahrzeugmodells in der VR Szene

6.2.4 Vergleich von Fahrzeugmodellvarianten – „Comparison“

Mit dieser Funktion können zwei verschiedene Modellvarianten des Fahrzeugmodells direkt miteinander verglichen werden (Abbildung 63). So ist es zum Beispiel möglich, Änderungen am Fahrzeugmodell direkt in der Szene zu betrachten und zu bewerten.

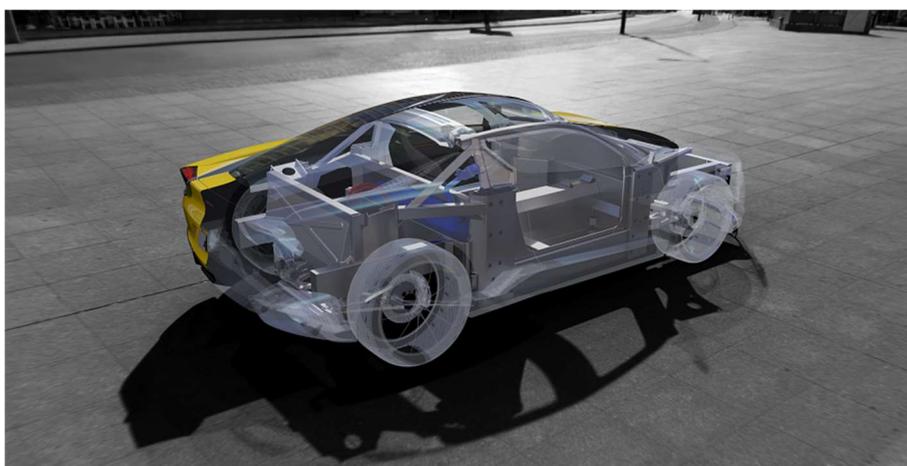


Abbildung 63 Vergleich zweier Modelle in der VR Szene

Es wird hierbei immer ein Modell aus dem „T_Lead_Model“-Knoten mit einem weiteren Modell aus dem „T_Comparison“-Knoten verglichen, welche auch dahingehend befüllt werden müssen. Welche Modelle miteinander verglichen werden sollen, kann über das interaktive Menü oder die Tastenkombinationen bestimmt werden. Die Vergleichsebene ist hierbei an das globale Koordinatensystem der VR Szene gebunden und kann mit dem Trackpad des rechten Controllers bestimmt werden, an dem die Koordinatenrichtungen angezeigt werden.

6.2.5 Positionswechsel - Viewpoints

Die Hauptbewegungsart in der VR Szene ist das normale Gehen durch den Raum. Durch ein Gitter werden die Grenzen des realen Raums in der VR Szene angezeigt, dass es zu keinen Kollisionen mit Wänden oder Gegenständen kommen kann. Zusätzlich sind in der VR Szene des Master Template Kacheln angebracht (Abbildung 64), mit deren Hilfe man sich an einen



Abbildung 64 Kacheln in VR Szene

vordefinierten Punkt der Szene teleportieren kann. Mittels leichten Ziehens des Triggers können die Kacheln mit dem erscheinenden Lichtstrahl anvisiert und mit dem vollständigen Drücken des Triggers der Positionswechsel durchgeführt werden. Die Positionen der Kacheln und die Koordinaten für das Teleportieren können nach Belieben angepasst werden. Eine Anleitung dazu findet sich in der Bedienungsanleitung des Master-Template, die in den Beiblättern dieser Arbeit vorliegt.

6.2.6 Umgebungswechsel

Mit dieser Funktion kann zu jeder Zeit die Umgebung der VR Szene geändert werden. Diese Funktion kann nicht nur für den Wechsel von Umgebungen, sondern auch für den Wechsel von Beleuchtungsszenarien verwendet werden. Eine genaue Beschreibung dazu befindet sich ebenfalls in der Bedienungsanleitung des Master-Template.

6.2.7 Modellwechsel

Die Modellwechsel-Funktion erlaubt es, das in der Szene dargestellte Fahrzeugmodell zu wechseln. Grundvoraussetzung ist, dass der Szenengraph dahingehend mit verschiedenen Modellen befüllt ist.

6.2.8 Wechsel der Farbe des Fahrzeugmodells – Color Switch

Der Materialeditor der Master-Template Szene beinhaltet einen vordefinierten Material-Switch mit der Benennung „T_Ext_Switch_Lack“. Dieser ermöglicht es, das Material bzw. die Farbe des Fahrzeugmodells mittels der festgelegten Tastenkombination zu wechseln. Dazu muss der Material-Switch jedoch zuerst allen Geometrieflächen, die in ihrer Erscheinung variabel sein sollen, zugewiesen werden. Der Inhalt des Material-Switch kann im Material-Editor beliebig angepasst und erweitert werden. Zusätzlich zum Material-Editor müssen die Veränderungen jedoch auch in den Variantensätzen und dem Skript der Szene vorgenommen werden.

6.2.9 Vordefinierte Schnitte - Sections

Um mit Hilfe der Funktion Schnitte am Fahrzeugmodell interaktiv in der VR Szene durchzuführen, sind zusätzlich mehrere vordefinierte Schnittebenen in das Master Template integriert. Diese Schnittebenen orientieren sich am globalen Koordinatensystem, ihre Position ist jedoch anpassbar. Mittels der festgelegten Tastenkombinationen können die Schnitte ein- bzw. ausgeblendet werden.

6.2.10 Zurücksetzen der Szene - Reset

Durch die Auswahl der Reset-Funktion kann die Szene auf ihre ursprünglichen Einstellungen zurückgesetzt werden. Im Skript sind folgenden Einstellungen als Anfangszustand festgelegt:

- Modell: Lead Model A
- Umgebung: Red Bull Ring
- Ansicht: Start View
- Farbe: Erste Farbe der Reihenfolge
- Funktion: keine, Controller wird dargestellt

6.2.11 Teleporting Plug-in

Das Teleporting Plug-in ist eine in Autodesk VRED integrierte Anwendung (Abbildung 65). Sie ermöglicht es, während der Ausgabe der Szene über ein VR Gerät, Ansichtspunkte zu erstellen und anzusteuern. So können während einer Untersuchung Ansichtspunkte von wichtigen Stellen am Fahrzeugmodell aufgezeichnet werden, um diese in einer späteren Untersuchung selbständig anzusteuern, oder Renderings von diesen Stellen erstellen zu lassen. Durch die Erstellung von Ansichtspunkten im Kamera-Editor Menü ist dies nicht möglich, da nur die Position in der Szene, aber nicht die Orientierung der VR Brille, aufgezeichnet wird.

Die Bedienung der Anwendung ist jedoch nur über den Desktop PC möglich, wodurch der Nutzer bei seinen Untersuchungen am virtuellen Fahrzeugmodell auf eine unterstützende Person angewiesen ist. Eine ausführliche Bedienungsanleitung der Anwendung befindet sich in der Anleitung des Master-Template.

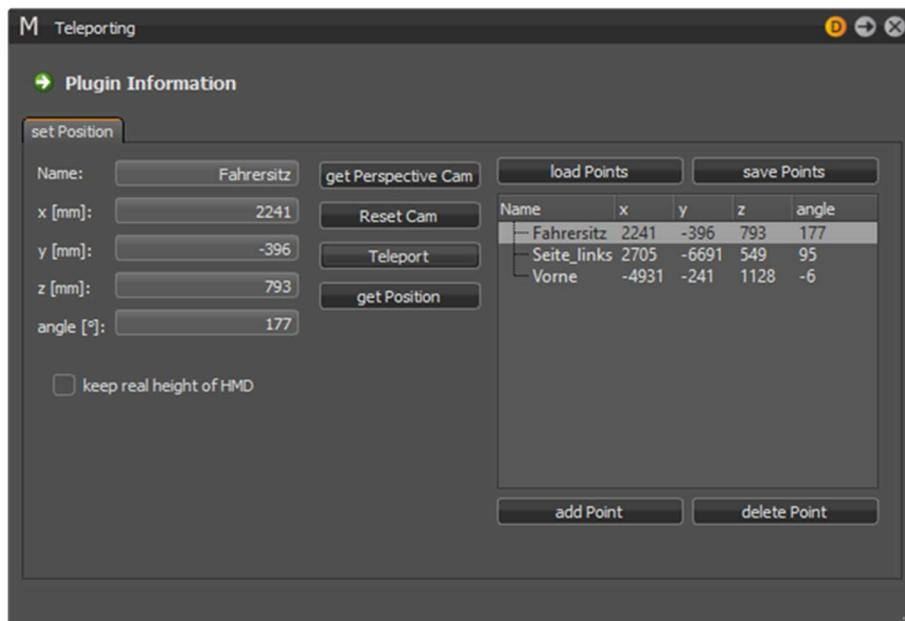


Abbildung 65 Benutzeroberfläche des Teleporting Plug-in

6.3 Performance des Master-Template

Ein sehr wichtiges Attribut einer VR-Szene ist deren Performance. Ist die Performance zu schlecht, so kommt es bereits bei kleinen und einfachen Modellen zu Leistungseinbrüchen und somit zu Verzögerungen in der Darstellung und im Tracking der Eingabegeräte. Wird diese Verzögerung zu groß bzw. fällt die Bildwiederholrate der Szene zu weit ab, so kann dies beim Anwender schnell zu einem Unwohlsein und Übelkeit führen. Ein längeres Arbeiten am virtuellen Fahrzeugmodell ist unter diesen Umständen nicht möglich. (Danneberg, 2017)

Um einem Leistungsabfall entgegenzuwirken gilt es herauszufinden, welche Einflüsse die Leistung der VR Szene am meisten beeinflussen. Durch Untersuchungen an verschiedenen bereits verfügbaren Projekten wurde ersichtlich, dass Parameter wie die Anzahl der dargestellten Dreiecke des tesselierten Geometrienetzes, die Anzahl der Knoten im Strukturbaum und die Tiefe des Strukturbaums den größten Einfluss auf die Performance der VR Szene in Autodesk VRED ausüben. Zum Beispiel kann bereits ein unsorgfältig aufbereitetes Geometrienetz eines einzigen Bauteils eines Fahrzeugmodells dazu führen, dass die Performance der gesamten VR Szene abfällt. Für solche Fälle bietet Autodesk VRED selbst ein in die Software integriertes Optimierungswerkzeug. Mit diesem Werkzeug ist es möglich, die Performance der Szene zu optimieren, indem unter anderem alle Geometrien desselben Materials zu einer Geometrie verschmolzen werden oder die Strukturtiefe der Szene auf die minimal mögliche Anzahl an Ebenen vereinfacht wird. Diese Optimierungen sind jedoch im Master-Template nicht möglich, da sie unsere vorgegebene Struktur, welche für die Ausführung der verschiedenen Funktionen notwendig ist, zerstören würden. Aus diesem Grund kann nur durch Befüllen des Master-Templates herausgefunden werden, ob die Leistungsfähigkeit des Template für zukünftige Anwendungen ausreichend ist.

6.3.1 Beschreibung des Performance Tests

Um nun die Leistungsfähigkeit des Master-Template zu bestimmen, werden zwei Versuche daran durchgeführt. Im ersten Versuch wird im Strukturbaum des Master-Template ein Knoten erstellt, welcher ein aus annähernd fünf Millionen Dreiecken bestehendes Geometrienetz enthält. Dieser Knoten bzw. diese Geometrie wird nun sukzessive vervielfältigt, bis sich die Bildwiederholrate der VR Brille verschlechtert. Der zweite Versuch besteht aus demselben Prinzip, jedoch wird das Hauptaugenmerk nicht auf die dargestellten Dreiecke, sondern auf

die Anzahl der Geometrien an sich gelegt. Als Testobjekt dient hierbei ein Knoten, der knapp über 1500 einzelne Geometrien enthält. Die Geometrien selbst sind hierbei vereinfacht, um die Anzahl der Dreiecke gering zu halten. Als Vergleichswerte werden die Anzahl der dargestellten Dreiecke, die Anzahl der erstellten Knoten, die Bildwiederholrate der VR Brille und die Auslastung der CPU und GPU der Testumgebung aufgezeichnet. Neben der Bildwiederholrate der VR Brille wurde auch die Bildwiederholrate bei einer Wiedergabe der Szene auf einem Desktop Bildschirm in Full HD Auflösung (1920 x 1080 Pixel) aufgezeichnet.

6.3.2 Testumgebung

Die Testumgebung besteht aus einem PC-System, welches vom VR Team auch für die Durchführung von VR-Untersuchungen verwendet wird. Die zum Zeitpunkt der Testdurchführung im PC System verbauten Komponenten sind in Tabelle 7 angeführt. Als Ausgabegerät wurde eine HTC Vive Pro VR Brille verwendet (Tabelle 8).

Tabelle 7 Komponenten des VR PC-Systems

Mainboard	ASUS ROG Strix Z270E Gaming
Prozessor (CPU)	INTEL Core i7 - 7700K
Arbeitsspeicher	32 GB (2x 16 GB) – 3200MHZ
Grafikkarte (GPU)	2x NVIDIA Titan XP im SLI Verbund
Festplatte	500 GB SSD
Betriebssystem	Windows 10 64-bit
Software	Autodesk VRED Professional 2019.1

Tabelle 8 Verwendete VR Brille

Typ	HTC Vive Pro
Auflösung	2880 x 1600 Pixel (1440 x 1600 Pixel pro Auge)
Framerate (maximal)	90 Hz

6.3.3 Ergebnis und Auswertung

6.3.3.1 Versuch 1: Erhöhung der Anzahl der dargestellten Dreiecke

Im ersten Versuch wird, wie bereits beschrieben, die Anzahl der in der Szene dargestellten Dreiecke des Geometrienetzes erhöht. In VRED wird als Darstellungsmethode OpenGL bei mittlerer Antialiasingstufe gewählt. Die aufgezeichneten Werte sind in Tabelle 9 dargestellt.

Tabelle 9 Ergebnisse aus Versuch 1

VR Szene		Desktop Bildschirm			HTC Vive Pro		
Dreiecke (in Mio.)	Knoten	Bildwiederholrate [Hz]	Systemauslastung [%]		Bildwiederholrate [Hz]	Systemauslastung [%]	
			CPU	GPU		CPU	GPU
5	1	60	18	14	90	19	55
10	2	60	18	19	90	18	63
15	3	60	18	25	90	19	71
20	4	60	18	31	90	19	79
25	5	60	17	35	90	19	86
30	6	60	18	40	82	18	88
35	7	60	17	42	63	19	74
40	8	60	17	55	60	19	76
45	9	60	17	56	56	19	80
50	10	60	17	64	55	19	84
55	11	60	18	74	52	19	85
60	12	60	18	82	51	19	88
65	13	60	18	96	49	19	90
70	14	60	29	98	47	19	92
75	15	52	30	100	47	19	96
80	16	49	30	100	45	19	98
85	17	46	30	100	43	19	100
90	18	44	30	100	40	20	100
95	19	42	30	100	38	19	100
100	20	40	30	100	35	20	100

Es ist zu erkennen, dass sich die Performance der VR Brille bereits ab einer Anzahl von 30 Millionen dargestellten Dreiecken zu verschlechtern beginnt. Der abrupte Abfall ist darauf zurückzuführen, dass die Software die Bildwiederholrate der VR Brille verringert, bevor die Auslastung der Grafikkarte (GPU) ihr Maximum erreicht. Daher ist eine leichte Erholung der Grafikkartenauslastung erkennbar. Mit diesem Ergebnis kann man durchaus zufrieden

sein. Zum Vergleich besteht ein virtuelles Fahrzeugmodell zu Beginn der Entwicklungsphase bei mittlerer Tessellierungsqualität aus weniger als 5 Millionen dargestellten Dreiecken. Ein vollständig detailliertes Fahrzeug kann am Ende seiner Entwicklung bis zu 75 Millionen Dreiecke erreichen. Bei dieser Menge liegt nach den Ergebnissen immer noch eine Bildwiederholrate von 45 Hz vor, was für ein Arbeiten am virtuellen Fahrzeugmodell ohne erkennbare Verzögerungen ausreichend sein sollte. Bei der Verwendung eines Desktop-Bildschirms ist ein Abfall der Bildwiederholrate, aufgrund der geringeren Darstellungsauflösung, erst ab 70 Millionen Dreiecken zu erkennen, was ein reibungsloses Arbeiten an jeglichen Modellen ermöglicht. In Abbildung 66 und 67 ist das Ergebnis des ersten Versuchs vervollständigend visuell dargestellt.

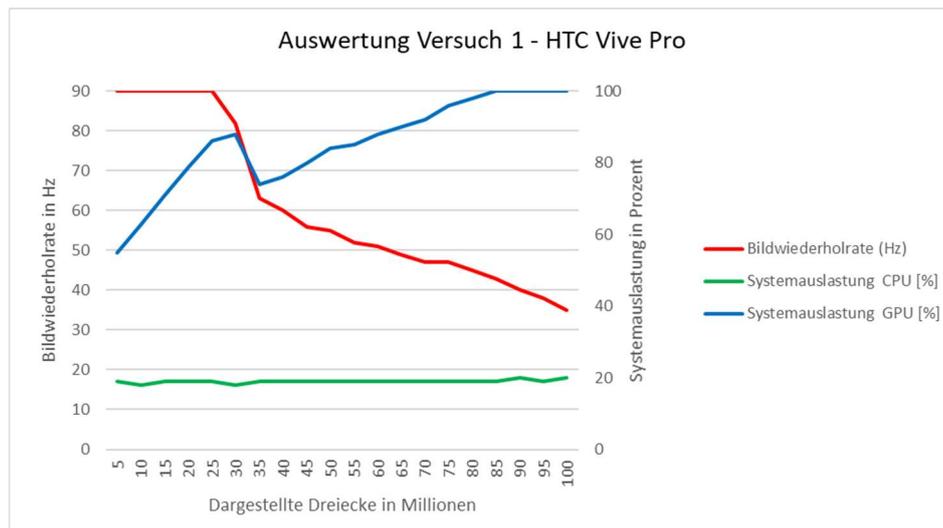


Abbildung 66 HTC Vive Pro Versuch 1

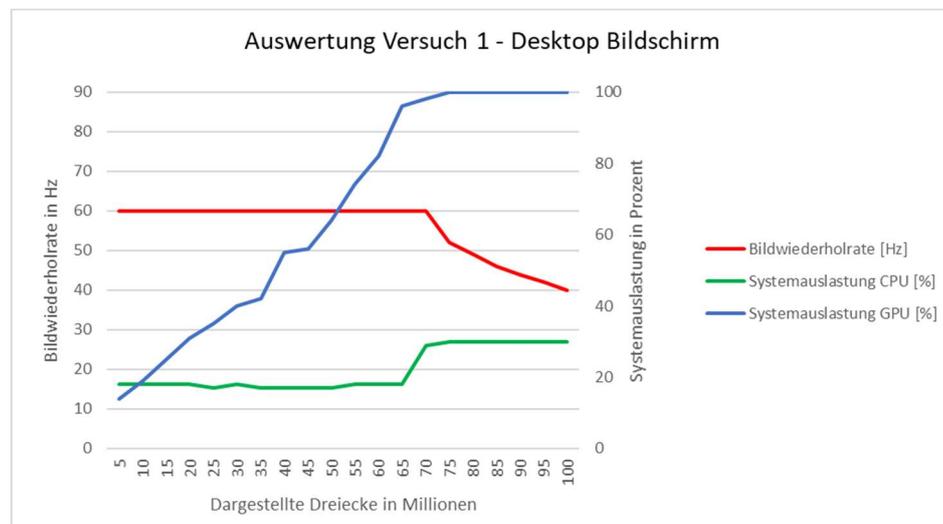


Abbildung 67 Desktop Bildschirm Versuch 1

6.3.3.2 Versuch 2: Erhöhung der Anzahl der Struktur-/Geometrieknoten

Im zweiten Versuch wird nun nicht die Anzahl der dargestellten Dreiecke, sondern die Anzahl der Geometrieknoten im Strukturbaum der Szene erhöht und beobachtet, welchen Einfluss dies auf die Performance nimmt. Als Darstellungsmethode dient ebenso OpenGL bei mittlerer Antialiasingstufe.

Tabelle 10 Ergebnisse aus Versuch 2

VR Szene		Desktop Bildschirm			HTC Vive Pro		
Dreiecke (in Mio.)	Knoten	Bildwiederholrate [Hz]	Systemauslastung [%]		Bildwiederholrate [Hz]	Systemauslastung [%]	
			CPU	GPU		CPU	GPU
0,28	1524	60	23	6	90	21	49
0,57	3048	60	25	6	90	24	50
0,85	4572	60	26	7	90	26	50
1,13	6096	60	27	8	90	28	52
1,42	7620	60	28	8	86	29	51
1,70	9144	60	28	8	66	27	38
1,98	10668	60	29	9	45	26	28
2,26	12192	60	30	9	45	30	28
2,55	13716	60	31	10	45	27	28
2,83	15240	57	31	10	45	28	28
3,11	16764	54	33	9	45	29	30
3,40	18288	46	30	8	42	30	30
3,68	19812	43	31	8	38	30	29
3,96	21336	42	30	8	32	28	24
4,25	22860	39	30	7	30	27	20
4,53	24384	37	32	8	30	28	20

In Tabelle 10 ist zu erkennen, dass sich in diesem Fall die Auslastung des Systems nicht wie im ersten Versuch verhält. Während im ersten Versuch die Auslastung der GPU der Auslöser für den Performanceeinbruch ist, scheint es im zweiten Versuch andere Ursachen zu haben. Vor allem bei der Darstellung am Desktop Bildschirm ist die GPU kaum gefordert und bei der Verwendung der HTC Vive Pro liegt die Auslastung jederzeit im unteren bis mittleren Bereich. Trotzdem fällt die Bildwiederholrate der HTC Vive Pro ziemlich früh abrupt ab, bevor sie sich bei der Hälfte der maximal möglichen Bildwiederholrate einpendelt. Diese Regulierung der Darstellungsfrequenz scheint wieder auf einen in der Software verankerten Algorithmus hinzudeuten. Genaue Belege dazu konnten jedoch nicht gefunden werden. Dieser Versuch zeigt aber wie wichtig es ist, das virtuelle Fahrzeugmodell, bevor es in das

Template eingefügt wird, so sorgfältig wie möglich aufzubereiten. Große Anhäufungen an Geometrieknoten sollten so gut wie möglich vermieden und zu einzelnen Geometrieknoten mit selbem Material verschmolzen werden, um die Anzahl der Knoten in der Szene gering zu halten. In den Abbildungen 68 und 69 ist das Ergebnis aus Versuch 2 wieder visuell dargestellt.

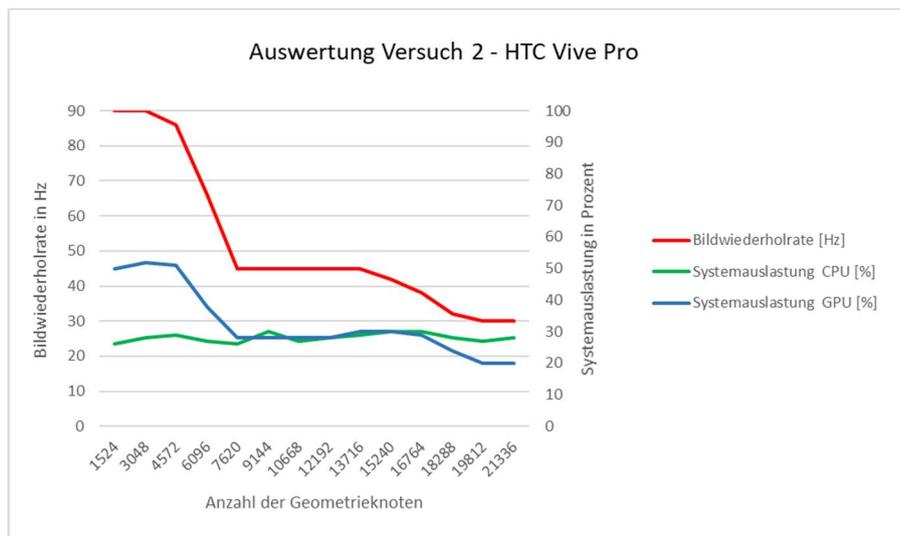


Abbildung 68 HTC Vive Pro Versuch 2

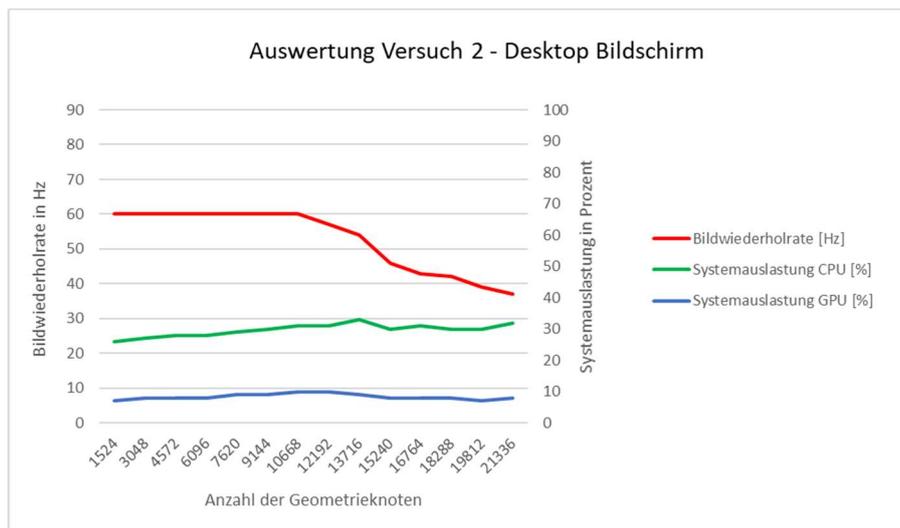


Abbildung 69 Desktop Bildschirm Versuch 2

7 Zusammenfassung und Ausblick

Die Automobilbranche erlebt zurzeit einen regelrechten Umschwung. Neue Antriebskonzepte erobern große Marktanteile und auch die Bedürfnisse der Endkunden verändern sich ständig. So sind alle Automobilhersteller und deren Zulieferer bemüht, ihre Produkte den neuen Begebenheiten anzupassen und weiterzuentwickeln. Neben den Produkten selbst ist hierzu jedoch auch eine Weiterentwicklung und Anpassung der inneren Strukturen, wie zum Beispiel des Fahrzeugentwicklungsprozesses, notwendig. Vor allem die frühen Phasen des Fahrzeugentwicklungsprozesses sind von großer Bedeutung, da in diesen Arbeitsabschnitten bereits die wesentlichen Produkthanforderungen definiert werden, auf deren Grundlage technische Lösungskonzepte ausgearbeitet werden. In diesen Entwicklungsphasen bestehen aber noch keine physischen Prototypen und Versuchsträger. Jedoch kann in diesen Phasen bereits auf moderne CAD-Programme zurückgegriffen werden, welche es ermöglichen, an virtuellen Modellen Auslegungs- und Absicherungsarbeiten durchzuführen. Mit diesen Werkzeugen können in diesen Phasen, trotz der unvollständigen und sich schnell ändernden Informationen, bereits weitreichende Entscheidungen in Bezug auf die Entwicklung des Fahrzeugs getroffen werden. Umso besser und qualitativ hochwertiger hierbei die Informationsbereitstellung und Aufbereitung funktioniert, desto leichter haben es die Mitarbeiter bei ihren Entscheidungen und Fehlentscheidungen, welche zumeist zu hohen Folgekosten führen können, werden vermieden. Um nun die Qualität der Informationsbereitstellung und deren Aufbereitung bzw. visuellen Darstellung zu verbessern, sind die Automobilhersteller darum bemüht, Virtual Reality Methoden in den Fahrzeugentwicklungsprozess zu integrieren. Durch den Einsatz von VR-Geräten ist es möglich, an virtuellen Fahrzeugmodellen zu arbeiten, als wären die Modelle physisch vorhanden. Dadurch kann auf die Erstellung von physischen Prototypen zum Großteil verzichtet und bereits in den frühen Entwicklungsphasen können Arbeiten durchgeführt werden, welche sonst nur im späteren Verlauf der Entwicklung an realen Fahrzeugmodellen möglich wären. Die Verwendung von virtuellen Fahrzeugmodellen ermöglicht es außerdem, Modelle mit einer großen Variantenvielfalt zu erstellen, was Entscheidungsprozesse sehr erleichtert. Damit die Integration der VR-Technologie funktionieren kann, sind jedoch viele Faktoren im Vorhinein festzulegen, um eine reibungslose und effiziente Unterstützung der verschiedenen Entwicklungsabteilungen zu gewährleisten. Ein sehr wichtiger Faktor ist hierbei die enge Zusammenarbeit zwischen den Entwicklungsabteilungen, welche die Ideen für benötigte Anwendungsfälle einbringen, und des VR-Teams,

welches die gewünschten Anwendungsfälle so gut wie möglich umzusetzen versuchen. Dabei ist der Umfang des Vorhabens konsequent festzulegen, um die höchstmögliche Effizienz der Zusammenarbeit zu erreichen, da die Erstellung einer VR- Szene einiges an Zeit beansprucht. Neben der Umsetzung der Anwendungsfälle ist das VR- Team auch für die Bereitstellung der VR-Infrastruktur und der Instandhaltung und Weiterentwicklung des VR-Integrationsprozesses verantwortlich, der die Dateibereitstellung, die Dateiaufbereitung, die Visualisierung des Fahrzeugmodells und die Erstellung der VR-Szene, umfasst.

Eines der Hauptziele ist es, einen automatisierten Update-Prozess in den VR-Integrationsprozess zu integrieren, welcher es ermöglicht, das virtuelle und bereits vollständig visualisierte Fahrzeugmodell per Knopfdruck auf den aktuellsten Entwicklungsstand zu aktualisieren. Dies erweist sich jedoch als äußerst komplex, da während der Modellaufbereitung viele Informationen der einzelnen Bauteile des Modells verloren gehen bzw. die Modellstruktur verändert wird und dadurch eine Zuweisung der neuen Bauteile erschwert. Hier gilt es eine Lösung zu finden, wie die benötigten Informationen erhalten bleiben und wie die während der Visualisierung vergebenen zusätzlichen Eigenschaften auf die neu eingefügten Bauteile übertragen werden können.

Ein weiteres Ziel, welches sukzessive verfolgt wird, ist das kooperative Arbeiten von mehreren Nutzern in derselben VR-Szene. Grundsätzlich wäre es bereits möglich, jedoch bilden sich neue Probleme sobald sich die verschiedenen Nutzer nicht mehr am selben Standort befinden. Mit der aktuellen Lösung muss an allen Standorten ein komplettes VR-System inklusive der VR-Szene vorhanden sein, damit alle Nutzer interaktiv in derselben VR-Szene, am selben virtuellen Fahrzeugmodell, arbeiten können. Dies schafft ganz neue Herausforderungen bezüglich der Datensicherung.

Man sieht, dass für den optimalen Einsatz von Virtual Reality Anwendungen im Fahrzeugentwicklungsprozess noch viele Fragen offen sind. Außer Frage steht jedoch das riesige Potential, welches sich dahinter verbirgt.

Literaturverzeichnis

- 3Dims. (kein Datum). *3Dims*. Abgerufen am 20. Mai 2018 von <http://www.3dims.de/StationaereSysteme.html>
- Albrand, C. (12. April 2016). *Vrodo*. Abgerufen am 16. Mai 2018 von <https://vrodo.de/htc-vive-so-zieht-ihr-die-vr-brille-richtig-auf/>
- Allweyer, T. (2015). *BPMN 2.0 Business Process Model and Notation. Einführung in den Standard für die Geschäftsprozessmodellierung*. Norderstedt: Books On Demand.
- Audi AG. (30. August 2017). Von <https://www.audi-mediacyber.com/de/pressemitteilungen/audi-startet-virtual-reality-im-autohaus-9270> abgerufen
- Autodesk. (2016). *VRED Professional 2016 Documentation*. Von http://download.autodesk.com/us/VRED/2016HELP/DOCS_AUTODESK_VRED_PROFESSIONAL_2016_EN.PDF abgerufen
- Autodesk. (kein Datum). *Autodesk VRED*. Von <https://www.autodesk.com/products/vred/overview> abgerufen
- BETA CAE Systems. (2018). *BETA CAE Systems*. Von <https://www.beta-cae.com/ansa.htm> abgerufen
- Brooks, F. (Dezember 1999). What's real about virtual reality? *IEEE Computer Graphics and Applications*.
- Buhr, M. (2013). *Virtual und Augmented Reality (VR/AR)*. Berlin Heidelberg: Springer Verlag.
- Burmester, R., & Vahs, D. (2005). *Innovationsmanagement: Von der Produktidee zur erfolgreichen Vermarktung*. Stuttgart: Schäffer-Poeschel.
- Canalys. (27. November 2017). Von <https://www.canalys.com/newsroom/media-alert-virtual-reality-headset-shipments-top-1-million-first-time> abgerufen
- Charara, S. (26. Dezember 2017). *Wearable*. Abgerufen am 15. Mai 2018 von <https://www.wearable.com/vr/how-does-vr-work-explained>
- Computer History Museum. (kein Datum). Abgerufen am 12. Mai 2018 von <http://www.computerhistory.org/revolution/input-output/14/356/1830>
- Core Technologie. (kein Datum). *3D Evolution CAD Converter*. Von <https://www.coretechnologie.de/produkte/3d-evolution.html> abgerufen

- Danneberg, B. (20. Februar 2017). *Motion Sickness in VR: Wie es entsteht und wie man es bekämpft*. Von VR World: <https://vr-world.com/was-tun-bei-motion-sickness-in-vr/> abgerufen
- Dassault Systemes. (kein Datum). *CATIA*. Von <https://www.3ds.com/de/produkte-und-services/catia/> abgerufen
- Deuschl, M. (2006). *Dissertation: Gestaltung eines Prüffelds für die Fahrwerksentwicklung unter Berücksichtigung der virtuellen Produktentwicklung*. Technische Universität München.
- Dörner, R. (2013). *Virtual und Augmented Reality (VR/AR)*. Berlin Heidelberg: Springer Verlag.
- Etienne, S. (23. Mai 2018). *The Verge*. Von <https://www.theverge.com/circuitbreaker/2018/5/23/17383990/google-lg-vr-display-high-res-headsets> abgerufen
- Gitlin, J. (10. Januar 2016). *Ars Technica*. Von <https://arstechnica.com/cars/2016/01/hyundais-augmented-reality-manual-a-simple-but-extremely-good-idea/> abgerufen
- Gusig, L.-O., & Kurse, A. (2010). *Fahrzeugentwicklung im Automobilbau*. München: Hanser Fachbuch.
- Harrich, A. (2014). *Dissertation: CAD-basierte Methoden zur Unterstützung der Karosseriekonstruktion in der Konzeptphase*. Technische Universität Graz.
- Hartmann, S. (3. August 2017). *IT Zoom*. Von <https://www.it-zoom.de/dv-dialog/e/dhl-mit-augmented-reality-schneller-im-lager-17425/> abgerufen
- Hirz, M. (2013). *Integrated Computer-Aided Design in Automotive Development*. Berlin Heidelberg: Springer Verlag.
- HTC. (16. Mai 2018). *VIVE Support*. Von https://www.vive.com/de/support/vive/category_howto/about-the-controllers.html abgerufen
- Jeckle, M., Rupp, C., Hahn, J., & Zengler, B. (2003). *UML 2 glasklar*. München: Carl Hanser Verlag.
- Katzenbach, A., Handschuh, S., & Vettermann, S. (2011). *JT Format (ISO14306) and AP 242 (ISO 10303): The Step to the Next Generation Collaborative Product Creation*. Heidelberg: Springer Verlag.
- Kitsios, V., & Haslauer, R. (2014). *3D-Master: zeichnungslose Produktbeschreibung mit CATIA V5*. Wiesbaden: Springer Verlag.

- Kleinman, J. (28. August 2016). *Technobuffalo*. Von <https://www.technobuffalo.com/2016/08/28/eyeforce-kickstarter-virtual-reality-headset/> abgerufen
- Körner, A. (September 2017). Mixed Reality, Virtual Reality. *Bertrandt Magazin*.
- La Rocco, N. (6. Januar 2016). *Computer Base*. Von <https://www.computerbase.de/2016-01/virtual-reality-audi-nutzt-rift-und-vive-fuer-die-vr-autokonfiguration/> abgerufen
- Lang, B. (10. Oktober 2017). *Road to VR*. Von <https://www.roadtovr.com/oculus-open-sources-rift-dk2-mechanical-designs-firmware-freely-available/> abgerufen
- Lang, B. (5. September 2018). *Road to VR*. Von <https://www.roadtovr.com/htc-vive-wireless-adapter-release-date-price/> abgerufen
- Länger, K. (06. Oktober 2017). *IT-Business*. Von <https://www.it-business.de/was-ist-virtual-augmented-und-mixed-reality-a-650442/> abgerufen
- Mattgey, R. (1. Januar 2017). *Chip*. Abgerufen am 15. Mai 2018 von https://www.chip.de/news/VR-ist-tot-Sony-begraebt-die-Spielezukunft_104478360.html
- microTOOL. (18. April 2018). Von <https://www.microtool.de/wissen-online/was-sind-use-cases/> abgerufen
- Milgram, P. (1995). *Augmented Reality: A class of displays on the reality-virtuality continuum*. Kyoto, Japan: ATR Communication Systems Research Laboratories.
- Nunez, M. (10. März 2015). *Popular Science*. Von <https://www.popsci.com/oculus-rift-how-it-works> abgerufen
- nViz. (2018). *nViz Motion*. Von http://www.nviz.de/nVIZ_110_DE_nVIZ_home.php abgerufen
- Rademacher, M. (2014). *Virtual Reality in der Produktentwicklung*. Wiesbaden: Springer Verlag.
- Robertson, A., & Zelenko, M. (kein Datum). *The Verge*. Abgerufen am 12. Mai 2018 von https://www.theverge.com/a/virtual-reality/oral_history
- Schäppi, B. (2005). *Handbuch Produktentwicklung*. München: Hanser Fachbuchverlag.
- Schulte-Frankenfeld, N., Brass, M., & Pieck, A. (2007). Methoden und Prozesse zur Kostensenkung - Ein Status der Wandlungen im Fahrzeugentwicklungsprozess durch CAE-Methoden. Frankenthal: LS-DYNA Anwenderforum.
- Siemens. (2015). *JT File Format Reference - Version 10.0*. Von https://www.plm.automation.siemens.com/en_us/Images/JT-v10-file-format-reference-rev-B_tcm1023-233786.pdf abgerufen

- Siemens. (kein Datum). *Siemens PLM Teamcenter*. Von <https://www.plm.automation.siemens.com/global/de/products/teamcenter/> abgerufen
- Slater M., L. B.-V. (2009). *How we experience immersive virtual environments: the concept of presence and its measurement*. Facultat de Psicologia, Universitat de Barcelona.
- Sparx Systems. (18. April 2018). *SparxSystems Software GmbH*. Von <https://www.sparxsystems.de/ressourcen/literatur/leseprobe-zu-projektentwicklung-mit-uml-und-enterprise-architect/anwendungsfalldiagramm-use-case-diagram/> abgerufen
- Unreal engine. (2018). *Using Datasmith with CAD File Formats*. Von <https://docs.unrealengine.com/en-us/Studio/Datasmith/SoftwareInteropGuides/CAD> abgerufen
- William R. Sherman, A. B. (2003). *Understanding Virtual Reality: Interface, Application, and Design*. San Francisco: Morgan Kaufmann.