

# **Ein Richdata-Konzept für Virtuelle Realität dargestellt am Beispiel der Fahrzeugkonfiguration**

Von der Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik  
der Universität Stuttgart  
zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.)  
genehmigte Abhandlung

Vorgelegt von Dipl.-Inform. Cheng Ni  
aus Hubei V.R. China

Hauptberichter:	Prof. Dr.-Ing. Dr.-Ing. E.h. Dieter Spath
Mitberichter:	Prof. Dr.-Ing. Prof. e. h. Dr.-Ing. e. h. Dr. h. c. mult. Engelbert Westkämper

Tag der Einreichung:	17. März 2011
Tag der Mündlichen Prüfung:	22. Juli 2011

Institut für Arbeitswissenschaft und Technologiemanagement der Universität Stuttgart  
2011

# **IPA-IAO Forschung und Praxis**

Berichte aus dem  
Fraunhofer-Institut für Produktionstechnik und  
Automatisierung (IPA), Stuttgart,  
Fraunhofer-Institut für Arbeitswirtschaft und  
Organisation (IAO), Stuttgart,  
Institut für Industrielle Fertigung und  
Fabrikbetrieb (IFF), Universität Stuttgart  
und Institut für Arbeitswissenschaft und  
Technologiemanagement (IAT), Universität Stuttgart

Herausgeber:

Univ.-Prof. Dr.-Ing. Prof. e.h. Dr.-Ing. e.h. Dr. h.c. mult. Engelbert Westkämper  
und

Univ.-Prof. Dr.-Ing. habil. Prof. E.h. mult. Dr. h.c. mult. Hans-Jörg Bullinger  
und

Univ.-Prof. Dr.-Ing. Dr.-Ing. E.h. Dieter Spath



**Universität Stuttgart**

Institut für Arbeitswissenschaft und  
Technologiemanagement IAT



**Fraunhofer**

**IAO**

Cheng Ni

# Ein Richdata-Konzept für Virtuelle Realität dargestellt am Beispiel der Fahrzeug- konfiguration

Nr. 509

**JOST-JETTER VERLAG**

Fachverlag · 71296 Heimsheim

Dr.-Ing. Dipl.-Inform. Cheng Ni

Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO), Stuttgart

Univ.-Prof. Dr.-Ing. Prof. e.h. Dr.-Ing. e.h. Dr. h.c. mult. Engelbert Westkämper

ord. Professor an der Universität Stuttgart

Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA), Stuttgart

Univ.-Prof. Dr.-Ing. habil. Prof. E.h. mult. Dr. h.c. mult. Hans-Jörg Bullinger

ord. Professor an der Universität Stuttgart

Präsident der Fraunhofer-Gesellschaft, München

Univ.-Prof. Dr.-Ing. Dr.-Ing. E.h. Dieter Spath

ord. Professor an der Universität Stuttgart

Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO), Stuttgart

D 93

ISBN 978-3-939890-80-4

Jost Jetter Verlag, Heimsheim

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils gültigen Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

© Jost-Jetter Verlag, Heimsheim 2011.

Printed in Germany.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Sollte in diesem Werk direkt oder indirekt auf Gesetze, Vorschriften oder Richtlinien (z. B. DIN, VDI, VDE) Bezug genommen oder aus ihnen zitiert worden sein, so kann der Verlag keine Gewähr für die Richtigkeit, Vollständigkeit oder Aktualität übernehmen. Es empfiehlt sich, gegebenenfalls für die eigenen Arbeiten die vollständigen Vorschriften oder Richtlinien in der jeweils gültigen Fassung hinzuzuziehen.

Druck: printsystem GmbH, Heimsheim

## Geleitwort der Herausgeber

Über den Erfolg und das Bestehen von Unternehmen in einer marktwirtschaftlichen Ordnung entscheidet letztendlich der Absatzmarkt. Das bedeutet, möglichst frühzeitig absatzmarktorientierte Anforderungen sowie deren Veränderungen zu erkennen und darauf zu reagieren.

Neue Technologien und Werkstoffe ermöglichen neue Produkte und eröffnen neue Märkte. Die neuen Produktions- und Informationstechnologien verwandeln signifikant und nachhaltig unsere industrielle Arbeitswelt. Politische und gesellschaftliche Veränderungen signalisieren und begleiten dabei einen Wertewandel, der auch in unseren Industriebetrieben deutlichen Niederschlag findet.

Die Aufgaben des Produktionsmanagements sind vielfältiger und anspruchsvoller geworden. Die Integration des europäischen Marktes, die Globalisierung vieler Industrien, die zunehmende Innovationsgeschwindigkeit, die Entwicklung zur Freizeitgesellschaft und die übergreifenden ökologischen und sozialen Probleme, zu deren Lösung die Wirtschaft ihren Beitrag leisten muss, erfordern von den Führungskräften erweiterte Perspektiven und Antworten, die über den Fokus traditionellen Produktionsmanagements deutlich hinausgehen.

Neue Formen der Arbeitsorganisation im indirekten und direkten Bereich sind heute schon feste Bestandteile innovativer Unternehmen. Die Entkopplung der Arbeitszeit von der Betriebszeit, integrierte Planungsansätze sowie der Aufbau dezentraler Strukturen sind nur einige der Konzepte, welche die aktuellen Entwicklungsrichtungen kennzeichnen. Erfreulich ist der Trend, immer mehr den Menschen in den Mittelpunkt der Arbeitsgestaltung zu stellen - die traditionell eher technokratisch akzentuierten Ansätze weichen einer stärkeren Human- und Organisationsorientierung. Qualifizierungsprogramme, Training und andere Formen der Mitarbeiterentwicklung gewinnen als Differenzierungsmerkmal und als Zukunftsinvestition in *Human Resources* an strategischer Bedeutung.

Von wissenschaftlicher Seite muss dieses Bemühen durch die Entwicklung von Methoden und Vorgehensweisen zur systematischen Analyse und Verbesserung des Systems Produktionsbetrieb einschließlich der erforderlichen Dienstleistungsfunktionen unterstützt werden. Die Ingenieure sind hier gefordert, in enger Zusammenarbeit mit anderen Disziplinen, z. B. der Informatik, der Wirtschaftswissenschaften und der Arbeitswissenschaft, Lösungen zu erarbeiten, die den veränderten Randbedingungen Rechnung tragen.

Die von den Herausgebern langjährig geleiteten Institute, das

- Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA),
- Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO),
- Institut für Industrielle Fertigung und Fabrikbetrieb (IFF), Universität Stuttgart,
- Institut für Arbeitswissenschaft und Technologiemanagement (IAT), Universität Stuttgart

arbeiten in grundlegender und angewandter Forschung intensiv an den oben aufgezeigten Entwicklungen mit. Die Ausstattung der Labors und die Qualifikation der Mitarbeiter haben bereits in der Vergangenheit zu Forschungsergebnissen geführt, die für die Praxis von großem Wert waren. Zur Umsetzung gewonnener Erkenntnisse wird die Schriftenreihe „IPA-IAO - Forschung und Praxis“ herausgegeben. Der vorliegende Band setzt diese Reihe fort. Eine Übersicht über bisher erschienene Titel wird am Schluss dieses Buches gegeben.

Dem Verfasser sei für die geleistete Arbeit gedankt, dem Jost Jetter Verlag für die Aufnahme dieser Schriftenreihe in seine Angebotspalette und der Druckerei für saubere und zügige Ausführung. Möge das Buch von der Fachwelt gut aufgenommen werden.

Engelbert Westkämper    Hans-Jörg Bullinger    Dieter Spath

## **Vorwort des Autors**

Die vorliegende Arbeit entstand während meiner Tätigkeit als Softwareentwickler und Projektmanager bei der Firma ICIDO GmbH Stuttgart. Sie basiert teilweise auf Arbeiten im Rahmen des Projekts „Computerised Automotive Technology Reconfiguration System for Mass Customization“ (CATER), das im sechsten Rahmenprogramm von der Europäischen Union gefördert und vom Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO) koordiniert wurde.

Herrn Prof. Dr.-Ing. Dr.-Ing. E.h. Dieter Spath, Leiter des Instituts für Arbeitswissenschaft und Technologiemanagement (IAT) der Universität Stuttgart und Leiter des Fraunhofer-Instituts für Arbeitswirtschaft und Organisation (IAO), Stuttgart gilt für die wissenschaftliche Unterstützung und die wohlwollende Förderung mein erster Dank.

Herrn Prof. Dr.-Ing. Prof. e. h. Dr.-Ing. e. h. Dr. h. c. mult. Engelbert Westkämper, Leiter des Instituts für industrielle Fertigung und Fabrikbetrieb der Universität Stuttgart (IFF) und Leiter des Stuttgarter Fraunhofer-Instituts für Produktionstechnik und Automatisierung (IPA), danke ich für die Übernahme des Mitberichts, die eingehende Durchsicht der Arbeit und die sich daraus ergebenden konstruktiven Hinweise.

Bei Herrn Dr.-Ing. Manfred Dangelmaier, Direktor des Geschäftsfelds Engineering-Systeme am Fraunhofer IAO, bedanke ich mich herzlich für die langjährige Betreuung und Unterstützung dieser Arbeit. Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Andreas Rößler der Hochschule Esslingen und Mitgründer der Firma ICIDO sowie Herrn Dipl.-Inform. (FH) Jan Wurster für alle Ermunterung, die Bereitschaft zur Diskussion und die wertvollen Anregungen. An Frau Ph.D. Laura Bernhart-Wong und Herr Dr. sc. math. Christian Hidber bedanke ich mich herzlich für die sprachliche Korrekturlesung.

Meiner Frau Min YU und meine Tochter Yinghe-Annika bin ich großen Dank schuldig, da sie während der Entstehung der Arbeit häufig auf die Anwesenheit des Ehemanns und Vaters verzichten mussten. Ohne meine Eltern Chonghua Yin und Maozhi Ni und ihre vielfältige Unterstützung und Förderung wäre diese Arbeit niemals zustande gekommen. Das Buch widme ich meiner Familie.

Jona, im September 2011

Cheng Ni





## Inhalt

Abkürzungen .....	13
Verzeichnis der Abbildungen .....	15
Verzeichnis der Tabellen .....	18
1 Einleitung .....	21
2 Zielsetzung, Aufgabenstellung und Vorgehensweise .....	23
3 Stand der Forschung und Technik .....	27
3.1 Strukturierung des Themenfeldes .....	27
3.1.1 3D und Immersion .....	27
3.1.2 Mensch-Maschine-Schnittstelle und Benutzungsoberfläche für die VR .....	29
3.1.3 Richdata .....	30
3.2 Grundlegende Richdata-Typen in der Literatur .....	31
3.2.1 Symbolische Information .....	31
3.2.2 Sprachaufzeichnung und Audio .....	36
3.2.3 Bilder .....	38
3.2.4 Bewegte Bilder .....	40
3.2.5 Stereo-Bilder und –Video .....	44
3.3 Datenquelle der Richdata .....	45
3.3.1 Metadaten aus Geometriedatensätzen .....	45
3.3.2 Externe unstrukturierte Datenquellen .....	47
3.3.3 Strukturierte Datenquellen und VR-Integration .....	48
3.4 Eingabe- und Trackinggeräte .....	49
3.5 Fazit .....	51
4 Analyse und Konzeption zu Richdata in VR .....	53
4.1 Ansatz .....	53
4.2 Analyse des Integrationsprozesses von VR-Anwendungen .....	55
4.2.1 Gedanken zum Nutzungsszenario .....	55
4.2.2 VR-Plattform und Rahmenbedingungen .....	56
4.2.3 Einfluss der Entwicklungsstrategie auf den Erfolg des Richdata-Konzepts .....	57

4.3	Bewertung der Repräsentationsformen hinsichtlich der VR-Integration.....	60
4.3.1	Text.....	60
4.3.2	Audio.....	62
4.3.3	Bilder.....	63
4.3.4	Bewegte Bilder.....	64
4.3.5	Stereo-Bilder und -Videos.....	65
4.3.6	Vergleich und Fazit.....	65
4.4	Modularisierung des Richdata-Konzepts.....	66
4.4.1	Basis-Architekturmuster.....	67
4.4.2	Struktur der Richdata-Datensätze.....	68
4.4.3	Verschachtelung.....	72
4.5	Erzeugung der Richdata.....	74
4.5.1	Automatisierte Erzeugung.....	74
4.5.2	Erzeugung aus Anforderungen der Anwendung.....	74
4.6	Richdata-Repräsentation.....	76
4.6.1	Analyse der WIMP.....	76
4.6.2	Immersives Repräsentationsmuster.....	81
4.7	Modellierungsprozess.....	83
4.8	Interaktion.....	85
4.8.1	Interaktion mit Standardmethode.....	85
4.8.2	Anwendungsspezifische Interaktion.....	86
4.8.3	Animation.....	90
4.9	Kommunikation mit externen Informationsquellen.....	91
4.9.1	Strukturierte Datenquellen.....	91
4.9.2	Unstrukturierte Datenquellen.....	93
5	Umsetzung des Richdata-Konzepts auf einer VR-Plattform.....	97
5.1	Vorgehensweise.....	97
5.2	Auswahl der Zielplattform.....	97
5.3	Bedingungen bzw. Voraussetzungen des Richdata-Konzepts.....	99
5.4	Technische Umsetzung.....	101

5.4.1	Anpassung der Zielplattform .....	101
5.4.2	Basisstruktur der Richdata-Datensätze .....	102
5.4.3	Richdata-Datensätze auf VR-Plattformen.....	103
5.4.4	Hilfsmethoden .....	105
5.4.5	Programmierschnittstelle für VR-Anwendungen .....	106
6	Anwendungsentwicklung am Beispiel der Fahrzeugkonfiguration .....	107
6.1	Rahmenbedingungen der Fahrzeugkonfiguration .....	107
6.1.1	Überblick und Komponenten des Projektkontexts.....	107
6.1.2	Anwendungsfall VR-Konfiguration.....	108
6.2	Konzeption einer Anwendung für die Fahrzeugkonfiguration .....	110
6.3	Richdata-Datensätze aus Abstraktion der Anwendungslogik.....	112
6.4	Immersives Layout und Richdata-Repräsentation des Konfigurators .....	113
6.4.1	Layout des Fahrzeugkonfigurators .....	113
6.4.2	Verteilung der Richdata-Repräsentationen im Szenegraphen .....	116
6.5	Interaktion.....	117
6.5.1	Werkzeuggürtel-Metapher .....	117
6.5.2	Katalog-Metapher .....	120
6.6	Verbindung mit Datenquellen.....	125
7	Evaluierung .....	127
7.1	Vorgehensweise .....	127
7.2	Evaluierungsaktivitäten in der Anforderungsphase.....	130
7.2.1	Kriterien und Hypothesen.....	130
7.2.2	Setup .....	130
7.2.3	Durchführung.....	130
7.2.4	Ergebnisse .....	133
7.3	Evaluierungsaktivitäten in der Grobentwurfsphase .....	134
7.3.1	Kriterien und Hypothesen.....	134
7.3.2	Setup .....	135
7.3.3	Durchführung.....	136
7.3.4	Ergebnisse .....	140

7.4	Evaluierungsaktivitäten mit betriebsfähigem Prototyp .....	141
7.4.1	Kriterien und Hypothesen .....	141
7.4.2	Setup .....	142
7.4.3	Durchführung .....	143
7.4.4	Ergebnisse .....	145
7.4.5	Reliabilität und Analyse der Ergebnisse .....	149
8	Zusammenfassung und Ausblick .....	152
	Abstract .....	156
	Literaturverzeichnis .....	160
	Anhang .....	172
	Anhang A: Fragebogen der Benutzbarkeitsstudie .....	172
	Anhang B: Wilcoxon-Test über Ergebnisse der Benutzbarkeitsstudie .....	185

## Abkürzungen

AR	Augmented Reality
CAD	Computer Aided Design
CATER	EU-Project: Computerized Automotive Technology Reconfiguration System for Mass Customization
CAVE	Cave Automatic Virtual Environment
CLI	Command Line Interface
GPU	Graphics Processing Uni
GUI	Graphical User Interface
HHI	Heinrich-Hertz-Institut
HMD	Head-Mounted Display
IAO	Fraunhofer-Institut für Arbeitswirtschaft und Organisation
IBP	Fraunhofer-Institut für Bauphysik
IGD	Fraunhofer-Institut für Graphische Datenverarbeitung
IPA	Fraunhofer-Institut für Produktionstechnik und Automatisierung
IPT	Immersive Projection Technology
ISO	International Organization for Standardization
IUI	Immersive User Interface
LCD	Liquid Crystal Display
LED	Leuchtdiode
NP	Nondeterministic polynomial time
PDM	Produktdatenmanagement
PLM	Produktlebenszyklusmanagement
RD	Richdata
RGB	Die Grundfarben rot, grün und blau

SOA	Service-Oriented Architecture
UI	User Interface
URI	Uniform Resource Identifier
VE	Virtuelle Umgebung, Virtual Environment
VMC	Video-Mediated Communication
VNC	Virtual Network Computing
VR	Virtuelle Realität, Virtual Reality
WIMP	Windows, icons, menus and pointing devices
WSDL	Web Services Description Language

## Verzeichnis der Abbildungen

Abbildung 1-1:	Herausforderungen der globalen Kommunikation .....	21
Abbildung 2-1:	Vorgehensweise .....	26
Abbildung 3-1:	Bewegungs- und Interaktionsraum von VR-Systemen mit IPT .....	28
Abbildung 3-2:	Überlagern von 2D-Elementen und 3D-Szene .....	32
Abbildung 3-3:	2D-Elemente in IRVE .....	34
Abbildung 3-4:	Annotierte Miniatur von Bell et al .....	35
Abbildung 3-5:	Erkennbarkeit eines 3D-Textes .....	36
Abbildung 3-6:	Projekt VAnno mit Sprachaufzeichnung .....	37
Abbildung 3-7:	Texturen in Commodity-Software.....	39
Abbildung 3-8:	Bilder und Texturen mit HDRI und Tone-Reproduktion .....	40
Abbildung 3-9:	IBM Sametime 3D und Alice Street.....	41
Abbildung 3-10:	Illustration der vier Interaktionsmodi .....	42
Abbildung 3-11:	VirtualPad und FrictionSurface .....	44
Abbildung 3-12:	Wii™, MotionPlus und der Umbau zum Tracking-System .....	51
Abbildung 4-1:	Komponenten des Richdata-Konzepts .....	53
Abbildung 4-2:	Drei Konfigurationen von Anzeigeanlagen im Vergleich .....	62
Abbildung 4-3:	Vergleich der Informationsformen hinsichtlich Reifegrad und Notwendigkeit für VR-Integration.....	66
Abbildung 4-4:	MVC-Architekturmuster.....	67
Abbildung 4-5:	Modularisierung des Richdata-Konzepts .....	68
Abbildung 4-6:	Klassendiagramm der Verschachtelung von Richdata-Datensätzen ..	73
Abbildung 4-7:	Ablauf des Erstellungsprozesses .....	75
Abbildung 4-8:	Vergleich typischer Benutzungsoberflächen .....	77
Abbildung 4-9:	Sequenzdiagramm für die gekoppelte Erzeugung von Richdata-Inhalt und -Repräsentation .....	84
Abbildung 4-10:	Sequenzdiagramm für die separate Erzeugung von Richdata-Inhalt und -Repräsentation .....	85
Abbildung 4-11:	Technischer Ablauf bei der Interaktionsmethode .....	88

Abbildung 4-12:	Klassendiagramm der Transformations- und zeitlichen Strategie im Richdata-Konzept .....	91
Abbildung 4-13:	Klassendiagramm der Programmierschnittstelle für Webservices.....	93
Abbildung 4-14:	Struktur und Klassendiagramm für die Methode IVRPanel .....	94
Abbildung 4-15:	Metapher des IVR-Panels als eigenständige Anwendung und integriert .....	95
Abbildung 5-1:	VR-Plattform VDP der Firma ICIDO .....	98
Abbildung 5-2:	Klassendiagramm für die Basisklasse des Richdata-Datensatzes .....	103
Abbildung 5-3:	3D-Text-Tabelle für die Fahrzeugkonfiguration.....	103
Abbildung 5-4:	Klassendiagramm für Beispieldatensätze auf VR-Plattformen.....	104
Abbildung 6-1:	Anspruchsgruppen und Benutzungsoberfläche im Projekt CATER...	107
Abbildung 6-2:	VR-Konfigurator und im Kontext des Projekts CATER.....	109
Abbildung 6-3:	VR-Benutzungsoberfläche für die Fahrzeugkonfiguration als VDP-Modul .....	110
Abbildung 6-4:	Komponenten des Moduls für die VR-Konfiguration.....	111
Abbildung 6-5:	Klassendiagramm für RD-Datensätze zur Fahrzeugkonfiguration ....	113
Abbildung 6-6:	Gesamtlayout und Komponenten der Fahrzeugkonfigurator.....	114
Abbildung 6-7:	Layout des Fahrzeug-Konfigurators aus Benutzersicht.....	114
Abbildung 6-8:	Klassendiagramm der Werkzeuggürtel-Metapher .....	118
Abbildung 6-9:	Zustandsmaschine der Werkzeuggürtel-Metapher .....	119
Abbildung 6-10:	Layout der Katalog-Metapher .....	120
Abbildung 6-11:	Helix-Layout für Katalog-Metapher und deren Parameter.....	122
Abbildung 6-12:	Zustandsdiagramm der Katalog-Metapher .....	123
Abbildung 6-13:	Zustandsmaschine der Animation zum Öffnen der Kataloge .....	124
Abbildung 7-1:	Spiral-Vorgehensmodell nach Boehm.....	128
Abbildung 7-2:	Kriterien in der Evaluierung der Anwendung und des RD-Konzepts	129
Abbildung 7-3:	Ablauf zum Bestimmen und zur Bewertung des Funktionsumfangs	131
Abbildung 7-4:	Ablauf der Evaluierungsaktivitäten in Grob- bis Feinentwurfsphase	137
Abbildung 7-5:	Layout der ersten Version der immersiven Benutzungsschnittstelle .....	138
Abbildung 7-6:	Layout des ersten lauffähigen Prototyps .....	139



Abbildung 7-7:	Verbesserung in der Entwurfsphase .....	139
Abbildung 7-8:	Testsystem für die finale Evaluierung .....	142

## Verzeichnis der Tabellen

Tabelle 3-1:	Drei Generationen computergestützter Benutzerschnittstellen .....	30
Tabelle 3-2:	Überblick über kommerzielle Lösungen für 3D-Konferenzen.....	41
Tabelle 3-3:	Überblick der VR-relevanten Datenformate .....	46
Tabelle 3-4:	Umfang der Untersuchung vom Wright und Gracanin .....	47
Tabelle 3-5:	VR-Eingabegeräte .....	50
Tabelle 4-1:	Vergleich der Entwicklungsstrategien für VR-Standardsoftware und projektbasierte Softwareentwicklung.....	59
Tabelle 4-2:	Reifegradskala und Notwendigkeitsskala .....	66
Tabelle 4-3:	Analyse des Designmusters WIMP und Übertragung auf die immersive Benutzungsoberfläche.....	80
Tabelle 4-4:	Immersives Designmuster im Vergleich mit WIMP .....	83
Tabelle 5-1:	Checkliste zur Erfüllung der Voraussetzungen des RD-Konzepts auf VDP 2007.....	100
Tabelle 5-2:	Checkliste zur Erfüllung der Anforderungen zur Modularität des Richdata-Konzepts auf VDP 2007 .....	100
Tabelle 5-3:	Checkliste zur Erfüllung der Anforderungen zum Interaktionsablauf auf VDP 2007 .....	100
Tabelle 5-4:	Schwachstellen der VDP 2007 und deren Behandlung in der technischen Konzeption .....	102
Tabelle 5-5:	Kategorien von Hilfsmethoden .....	105
Tabelle 6-1:	Inhalte in den Informationstafeln des Konfigurators.....	116
Tabelle 6-2:	Parameter der Helix-Form des Katalogs .....	121
Tabelle 6-3:	Spezifikation des Webservices für Konfiguration und Statistik .....	125
Tabelle 6-4:	Methode des Webservices für Konfiguration und Statistik .....	126
Tabelle 7-1:	Analyse des Funktionsumfangs zur ersten Erprobung.....	133
Tabelle 7-2:	Ausstattung der Test-Systeme .....	135
Tabelle 7-3:	Test-Datensätze.....	136
Tabelle 7-4:	Checkliste der Anforderungen an Funktionalitäten .....	140

Tabelle 7-5: Verbesserungsvorschläge für Funktionalitäten am Ende der Entwurfsphase.....	141
Tabelle 7-6: Konfiguration des Test-Systems mit stationärer VR-Anlage .....	143
Tabelle 7-7: Probanden für die Benutzbarkeitsstudie .....	144
Tabelle 7-8: Absolute Häufigkeiten von Einstufungen der VR-Benutzungsoberfläche im Vergleich mit einer 2D-GUI .....	145
Tabelle 7-9: Absolute Häufigkeiten von Einstufungen der VR-Benutzungsoberfläche im Vergleich mit dem Desktop-Computer .....	146
Tabelle 7-10: Bewertungen der im VR-Konfigurator implementierten Funktionalitäten .....	147
Tabelle 7-11: Bewertungen der unterschiedlichen Wege zu einer Funktionalität .....	147
Tabelle 7-12: Bewertungen der unterschiedlichen Möglichkeiten zur Positionierung von Metadaten.....	148
Tabelle 7-13: Bewertungen der Funktionalitäten im VR-Konfigurator.....	148



## 1 Einleitung

In der globalisierten Wirtschaft verteilen sich die Entwicklungs-, Produktions- und Vertriebsprozesse immer stärker über die Grenzen der einzelnen Unternehmen und über die Grenzen von Ländern und Kontinenten hinaus [Davis 1997; Piller 2002; Piller 2006; Bender 2009]. Die einzelnen Akteure wie Kunden, Vertriebsorganisationen, Hersteller und Lieferanten können sich also in unterschiedlichen Ländern befinden. Und das bedeutet: Sie haben unterschiedliche Hintergründe im Hinblick auf Sprache, Kultur und technischen Wissensstand. Zuverlässige und effektive Kommunikation unter den Akteuren ist daher eine der aktuellen Herausforderungen unserer Zeit. Das Medium *Virtuelle Realität* (VR) ist in der Lage, durch seine integrative Funktion fachliche Kommunikationsbarrieren zu überwinden und Missverständnisse zu vermeiden [Spath 2008; Spath 2009].



Abbildung 1-1: Herausforderungen der globalen Kommunikation

Schon heute wird die VR-Technologie sowohl für spezielle Anwendungsfälle wie ergonomische Untersuchungen oder Inspektionen als auch zur Unterstützung von Entwicklungs- und Qualitätsmanagementprozessen eingesetzt [Dangelmaier 2001; Westkämper 2007; Dangelmaier 2008]. Sie ist aber nicht ein isoliertes Werkzeug, das die Integration von Arbeitsprozessen verbessert, sondern muss auch selbst im Prozessumfeld integriert sein [Spath 2009]. Dieses Ziel betreffend stellen sich zwei große Herausforderungen für die derzeitigen VR-Anwendungen bzw. -Plattformen.

Zum Ersten werden in der immersiven Umgebung zusätzlich vielfältige Informationsformen benötigt. Die Entwicklung der VR-Technologie war allerdings bisher stark auf die stereoskopische Darstellung von reinen 3D-Modellen fokussiert. Die realitätsnahe und intuitiv verständliche Darstellung von Modellen hat den großen Vorteil, dass mögliche Missverständnisse und Informationsverluste im Kommunikationsprozess zwischen den

Akteuren vermieden werden. Andererseits werden auch bei der Arbeit an immersiv dargestellten Modellen weitere Informationsformen, wie zum Beispiel Text, Bilder, Audio und Video benötigt. Wie sich diese ergänzenden Daten darstellen und nutzen lassen, wurde bei der Forschung und Entwicklung virtueller Umgebungen bisher nicht ausreichend berücksichtigt [Bowman 2004; Lentz 2007]. Diese Tatsache erschwert derzeit die Weiterentwicklung der VR-Anwendungen vom Forschungsprototyp zum marktreifen Software-Produkt. In der Praxis müssen die Benutzer meist zur herkömmlichen, zweidimensionalen Benutzerschnittstelle zurückkehren, wenn komplexe Arbeitsprozesse die Einbindung unterschiedlicher Informationsformen erfordern oder der notwendige Funktionsumfang für die Prozessunterstützung zu groß wird.

Die zweite Herausforderung bei der Integration von VR-Anwendungen in die Arbeitsprozesse ist die erforderliche komplexe Kommunikation zwischen VR-Systemen und den betrieblichen Informationssystemen. In der heutigen industriellen Praxis gehört die 3D-Modellierung bzw. das 3D-Design bereits als integrierter Teil zum Standard-Entwicklungsprozess. Jedoch bleibt dabei die VR meist eine Lösung, die nur isoliert für spezifische Anwendungsfälle und für eine eingeschränkte Benutzergruppe einsetzbar ist. Zwar gibt es Bemühungen, VR-Systeme so zu gestalten, dass eine automatisierte Datenerzeugung bzw. -konvertierung für die Erzeugung von 3D-Modellen möglich wird. Diese setzen allerdings selten auf der Ebene der Anwendungslogik an. Auch dieser Ansatz ermöglicht es also nicht, die Eingabe und das Arbeitsergebnis der VR-Anwendung in einen übergeordneten Prozess einzubetten [Eberle 2007; Schirra 2008].

Um die oben genannten Herausforderungen zu überwinden, wird in dieser Arbeit ein Richdata-Konzept erforscht. Das Konzept erweitert das traditionelle VR-System um immersive Darstellungen und Interaktionen in vielfältigen Ausprägungsformen wie Texten, Bildern oder Animationen.

Das Richdata-Konzept ist eine plattformunabhängige Erweiterung der konventionellen VR-Systeme. Es erlaubt VR-Nutzern einerseits, sowohl 3D-Modelle als auch formale und technische Daten in immersiven Umgebungen zu erleben und ohne Schwierigkeiten mit ihnen zu interagieren. Andererseits bietet das Konzept den Softwareentwicklern die Möglichkeit, die VR-Plattform um Richdata zu erweitern, um integrierte, benutzerfreundliche und effiziente VR-Benutzungsoberflächen zu konstruieren.

## 2 Zielsetzung, Aufgabenstellung und Vorgehensweise

Diese Arbeit verfolgt die Zielsetzung, ein Richdata-Konzept (RD-Konzept) zu entwickeln, um die VR-Technologie in industrielle Arbeitsprozesse integrieren zu können. Während die traditionelle VR auf die Repräsentation von Geometrie-Datensätzen fokussiert ist, erweitert das RD-Konzept VR-Systeme um integrierte Datenstrukturen und vielfältige Informationsformen aus heterogenen Quellen. Das Konzept wird zur Verifizierung auf einer VR-Plattform umgesetzt. Auf der Grundlage dieser Plattform wird eine Beispielanwendung – die Fahrzeugkonfiguration – implementiert.

Zu den Aufgaben der vorliegenden Arbeit gehört im ersten Schritt die Recherche der fachlichen Grundlagen. Dabei wird erörtert und bestimmt, welche Repräsentationsformen in immersiven Anwendungen möglich und sinnvoll sind. Des Weiteren wird analysiert, welche Repräsentationen in der industriellen Praxis bereits eingesetzt werden bzw. technisch ausgereift sind und für welche dies noch nicht zutrifft. Im Richdata-Konzept werden die als grundlegend festgestellten Repräsentationsformen schließlich zusammengeführt. Ferner soll das Konzept offen und erweiterbar gestaltet werden.

Die innere Konstruktion der Richdata ist ein zentraler Bestandteil der Konzeptentwicklung. Sie soll eine flexible Verbindung zwischen Dateninhalt und deren Repräsentation in der immersiven Umgebung erlauben. Ein Richdata-Datensatz soll sowohl als eigenständiger Informationsträger, als auch als Input zur Ergänzung eines anderen Dateninhalts dienen. Solche Dateninhalte beziehen sich nicht nur auf die Grundformen, wie z. B. Texte, Bilder oder Audio, sondern auch auf kombinierte und strukturierte Komponenten wie GUI-Elemente, Animationen oder Beschreibungen des Anwendungszustands. Hier liegt eine grundlegende Schwachstelle vieler heutiger VR-Anwendungen, da die Programmlogik und die Informationsdarstellungen bisher zwei unterschiedlichen Welten angehören.

Ein Richdata-Datensatz soll also nicht nur eine Verbindung zwischen der Anwendungslogik und den Objekten in den virtuellen Szenen ermöglichen, sondern auch das VR-System mit externen Datenquellen verzahnen. Zustände der Richdata sollen unabhängig vom Szenegraphen persistent gespeichert werden. Es soll möglich sein, sie aus der VR-Anwendung zu exportieren und so auch unabhängig vom VR-System zu modifizieren. Deshalb sieht das Richdata-Konzept neben der Schnittstelle für den internen Zugriff des VR-Systems eine zweite Schnittstelle vor, welche eine bidirektionale Verbindung zur Außenwelt ermöglicht.

Die Interaktionen mit Richdata lassen sich unter zwei Gesichtspunkten betrachten: Zum einen der methodische Aspekt, welcher den Ablauf von Benutzeraktionen über Eingabegeräte, Systemereignisse und Zustandsänderungen der Richdata bis hin zu wahrnehmbaren Änderungen auf den Anzeigesystemen oder anderen Ausgabegeräten betrifft. Zum anderen geht es um die Entwicklung von Interaktionsmetaphern für die konkrete Anwendung.

Als Teil des RD-Konzepts soll auch ein impliziter Leitfaden für die Entwicklung integrierter VR-Anwendungen entstehen. Das Konzept beinhaltet nicht nur statische Komponenten und Vorschriften, sondern auch Beschreibungen von Prozessen, die Entwickler dabei unterstützen, Schritt für Schritt VR-Anwendungen mit RD-Konzept zu designen und zu implementieren.

Das RD-Konzept ist unabhängig vom Design der VR-Plattformen. Eine Umsetzung des Konzepts muss aber dennoch den technischen Zustand der jeweiligen VR-Plattform berücksichtigen. Dazu werden im RD-Konzept Richtlinien erarbeitet, die aufzeigen, welche programmiertechnische Infrastruktur die VR-Plattformen jeweils erfüllen müssen, damit eine Erweiterung nach dem RD-Konzept möglich ist.

In der Beispielanwendung „Fahrzeugkonfiguration“ werden verschiedene Formen von Richdata einbezogen. Neben den Beschreibungen einzelner technischer Fahrzeugbauteile zählen dazu auch die Beschreibungen von Kombinations- und Substitutionsmöglichkeiten. Die in der Konzeption beschriebenen Formen und Darstellungen von Richdata finden so im Fahrzeugkonfigurator beispielhaft Anwendung. Die Dateninputs einer Konfiguration werden aus einer betrieblichen Datenbank übernommen und ändern sich während der Laufzeit durch die Benutzeraktion in der VR-Anwendung. Die Ergebnisse können in Form von Richdata wieder in betriebliche Datenbanken exportiert und dort gespeichert werden.

Die Erzeugung neuer Richdata während der Laufzeit ist ein eigenständiger Forschungsgegenstand. Schwerpunkt ist dabei die Eingabemethode bzw. -metapher, die während der Arbeit in der immersiven Umgebung verwendet wird [Verlinden 1993; Stefani 2003; Tsang 2003]. Die dafür geeigneten Verfahren werden in dieser Ausarbeitung nur am Rande erwähnt. Umgesetzt wird in diesem Bereich aber die Möglichkeit, Richdata während der Laufzeit von externen Quellen, z. B. Remote-Datenbanken, zu laden bzw. zu aktualisieren.

Das Richdata-Konzept beschränkt sich im Rahmen dieser Arbeit auf Anwendungen im Virtual-Prototyping-Bereich. Darüber hinaus sind Anwendungen auch für Aufgaben des Data-Mining und der Daten-Visualisierung möglich. Der Vorteil von entsprechenden Anwendung von VR-Anwendungen auf der Basis des RD-Konzepts liegt darin, große Mengen an Informationen zu erzeugen und in der immersiven Umgebung anzuzeigen. Die hierfür relevanten Daten unterscheiden sich von Daten im Bereich *Virtual Prototyping* allerdings grundlegend. Wie diese in ein RD-Konzept integriert werden können, ist daher ebenfalls ein eigenständiger Forschungsgegenstand und damit nicht Bestandteil der vorliegenden Arbeit.

Die Vorgehensweise bei der Bearbeitung oben genannter Aufgaben ist in Abbildung 2-1 dargestellt. Nach Recherchen zum Stand der Forschung und Technik werden die bestehenden Defizite ermittelt. Diese sollen durch die Ausarbeitungen der vorliegenden Arbeit beseitigt werden (Kapitel 2). In Kapitel 1 wird eine grundlegende Konzeption entwi-



ckelt, die unabhängig von konkreten VR-Systemen für alle VR-Plattformen anwendbar ist. Nach Vorstellung des Konzeptansatzes (Kapitel 4.1) wird der VR-Integrationsprozess analysiert und spezifiziert (Kapitel 4.2). Die verschiedenen Repräsentationsformen von Daten werden hinsichtlich ihrer Integration in eine VR-Umgebung bewertet (Kapitel 4.3). Die einzelnen Komponenten und die Struktur des RD-Konzepts werden schließlich in Kapitel 4.4 (Modularisierung) vorgestellt. Anschließend wird der Erstellungsprozess der Richdata-Inhalte (Kapitel 4.5) und deren Repräsentation (Kapitel 4.6) erläutert. Im Rahmen des Modellierungsprozesses werden die RD-Inhalte und die Repräsentationsformen dann miteinander verknüpft (Kapitel 4.6). Danach wird das Interaktionskonzept für die Arbeit in der VR-Umgebung (Kapitel 4.8) und die Verbindung mit externen Ressourcen während der Laufzeit (Kapitel 4.9) vorgestellt.

Das RD-Konzept erweitert konventionelle VR-Plattformen. Nach Erläuterung der Vorgehensweise im Erweiterungsprozess in Kapitel 5.1 wird daher eine Zielplattform für die Beispielanwendung ausgewählt und vorgestellt (Kapitel 5.2). Anschließend wird analysiert und spezifiziert, welche softwaretechnischen Voraussetzungen die Zielplattform erfüllt bzw. nicht erfüllt (Kapitel 5.3). Für festgestellte Defizite wird eine Lösungsmöglichkeit ausgearbeitet. Danach werden die Funktionalitäten des RD-Konzepts in der Zielplattform implementiert (Kapitel 5.4). Mit der angepassten und erweiterten VR-Plattform ist es nun möglich, Anwendungen zu entwickeln, die lückenlos in einen übergeordneten Prozess integriert werden können.

Als Beispielanwendung wird ein Fahrzeugkonfigurator entwickelt (Kapitel 1). Nach Analyse und Vorstellung der Konzeption der Anwendung (Kapitel 6.1- 6.2) wird die Programmlogik abstrahiert (Kapitel 6.3) und das immersive Layout bestimmt (Kapitel 6.4). Anschließend werden Interaktionsmetaphern für die Bedienung des Konfigurators ausgearbeitet (Kapitel 6.5). Schließlich werden die Verbindungen des Konfigurators mit Datenbanken und einem Remote-Computer realisiert (Kapitel 6.6). Zur Überprüfung der Praxistauglichkeit der Beispielanwendung wird eine erste Evaluierung durchgeführt. Das Vorgehensmodell der Evaluierungsaktivitäten wird in Kapitel 7.1 vorgestellt. Die Evaluierung findet begleitend in jeder Phase der Anwendungsentwicklung statt (Anforderungsphase Kapitel 7.2, Grobentwurfsphase Kapitel 7.3 und Feinentwurfsphase mit betriebsfähigem Prototyp Kapitel 7.4). Die gesamte Arbeit und die erzielten Ergebnisse werden abschließend in Kapitel 1 zusammengefasst.

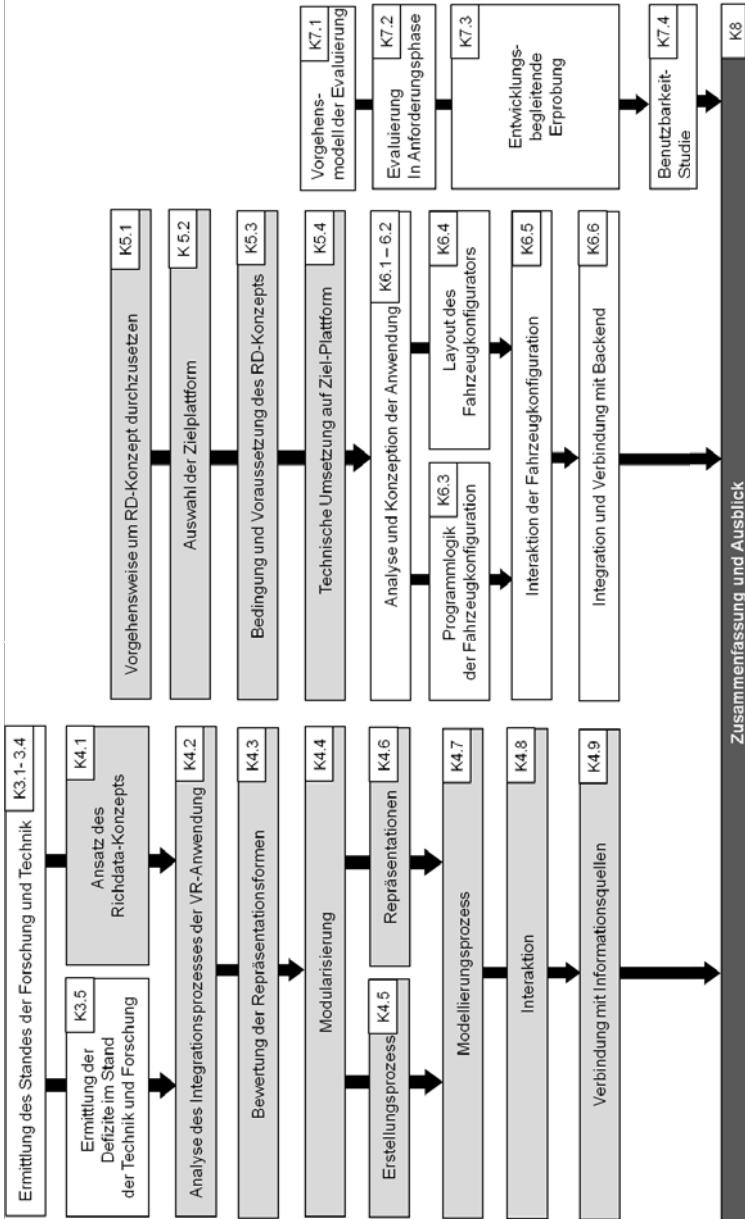


Abbildung 2-1: Vorgehensweise

## 3 Stand der Forschung und Technik

### 3.1 Strukturierung des Themenfeldes

#### 3.1.1 3D und Immersion

Die Begriffe drei-dimensional (3D) und virtuelle Realität (VR) werden in der Literatur nicht scharf abgegrenzt. Zahlreiche Anwendungen können 3D-Inhalte repräsentieren. Die Inhalte können vom Benutzer dennoch meistens nur mit zweidimensionalen Anzeigesystemen (2D-Anzeigesysteme) betrachtet und mit einer 2D-Benutzungsoberfläche bedient werden. Zum Teil lassen sich diese Anwendungen dennoch als VR bezeichnen, obwohl die Immersion – im Sinne von „Eintauchen in die virtuelle Welt“ – dabei nicht möglich ist.

Die stereoskopische Anzeige ist ein Hauptmerkmal der Immersion. Sie ist für ein immersives Wahrnehmen deshalb so entscheidend, weil ca. 80 % aller menschlichen Eindrücke über das Sehen vermittelt werden [Kreimer 2004]. In der vorliegenden Ausarbeitung werden nur solche Plattformen bzw. Anwendungen mit VR bezeichnet, die auf stereoskopischer Immersion basieren. Wenn es darum geht, das Layout und die Interaktion für die immersive Umgebung zu entwerfen, steht der Entwickler vor grundlegend anderen Rahmenbedingungen als bei einer zweidimensionalen *Grafischen Benutzeroberfläche* (GUI).

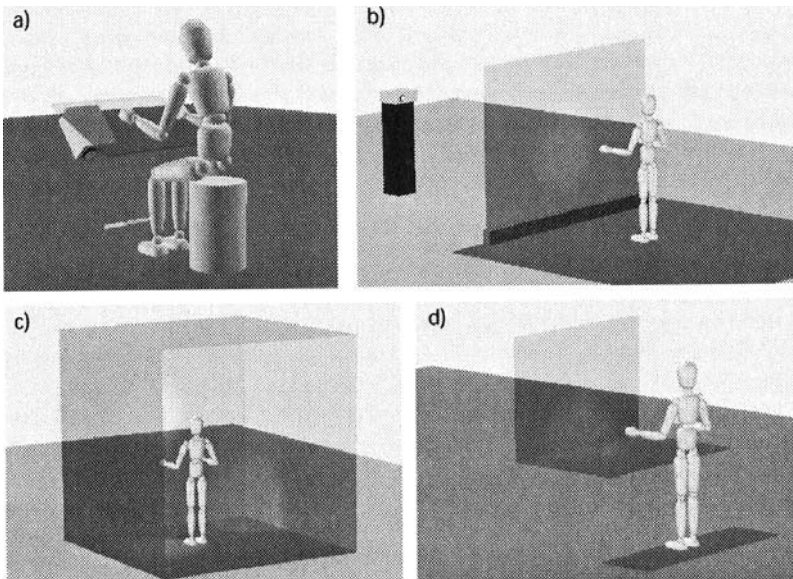
Es gibt aktuell verschiedene Techniken, um Stereoskopie zu realisieren. Das noch vor einigen Jahren in der Anwendung häufig eingesetzte *Head-Mounted Display* (HMD) ist heute in der industriellen Praxis kaum verbreitet. Im Vergleich zum HMD besitzen die großflächigen Stereoprojektionen (*immersive projection technology*, IPT) weit größere Bedeutung [Bullinger 1999]. Die Systeme, die mit Stereoprojektoren arbeiten, lassen sich hinsichtlich des Bewegungs- und Interaktionsraums des Benutzers weiter unterteilen. In der Dissertation von Häfner [Häfner 2002] werden sie in vier Kategorien eingeordnet (siehe Abbildung 3-1).

Beim einfachsten IPT-System sitzt der Benutzer vor einer waagrechten oder geneigten Projektionsscheibe (siehe Abbildung 3-1 a). Der Bewegungs- und Interaktionsraum wird durch die sitzende Haltung des Benutzers eingeschränkt. Die Inhalte werden zwar stereoskopisch angezeigt, und teilweise auch die Kopfposition des Benutzers vom Anzeigesystem verfolgt. Dennoch kann diese Art von VR dem Benutzer nur eingeschränkt ein Eintauchen in die Szenerie im Sinne einer immersiven Wahrnehmung vermitteln.

Typische IPT-Systeme, die in der Industrie verbreitet eingesetzt werden, erlauben dem Benutzer in stehender Haltung bzw. im Gehen zu interagieren. Die Abbildung 3-1 b zeigt einen Benutzer vor einer Stereo-Powerwall. Der Bewegungs- und Interaktionsraum solcher Systeme kann zusätzlich erweitert werden, indem mehrere Powerwalls miteinander verbunden werden.

Wenn die Wände dabei einen kompletten, abgeschlossenen Raum bilden, taucht der Benutzer physikalisch ganz in die virtuelle Umgebung ein (siehe Abbildung 3-1 c). Ein solches System wird *CAVE Automatic Virtual Environment (CAVE™)* genannt [Cruz-Neira 1993]. Mit dieser Art von virtueller Umgebung liegen die Repräsentationen nicht nur vor dem Benutzer, sondern sind wie in der realen Welt um ihn herum angeordnet. Im Gegensatz zum HMD hat der Benutzer in einer CAVE™ nicht nur ein uneingeschränktes Sichtfeld, sondern zudem auch uneingeschränkte Sichtwinkel.

Die Abbildung 3-1 d zeigt eine Zwischenform von Desktop-VR und Powerwall. Trotz stehender Haltung bietet ein solches System dem Benutzer nur einen kleinen Aktionsraum am Anzeigesystem.



**Abbildung 3-1: Bewegungs- und Interaktionsraum von VR-Systemen mit IPT [Häfner 2002]**

Auf dem Markt für Konsumenten gewinnen stereofähige Fernsehgeräte immer mehr an Bedeutung. Mehrere Hersteller sind seit dem Jahr 2009 mit entsprechenden Geräten am Markt vertreten. Die Fernsehgeräte werden mit immer größeren Bildschirmdiagonalen angeboten und sind auch für Konsumenten erhältlich (Stand Mai 2010 maximale Diagonale von 165 cm). Für die Bildschirmanzeige werden dabei Technologien wie LCD, LED-Backlighting und Plasma eingesetzt. Da die Stereofernseher für das Betrachten von Stereo-Filmen konzipiert sind, verfügen sie im Gegensatz zu den oben genannten Systemen nicht über ein Tracking-System.

Aufgrund des fehlenden Trackingsystems und der relativ kleinen Displayfläche sind die Geräte für Konsumenten nicht für Interaktion, insbesondere im Stehen und in Bewegung geeignet. Dennoch ist die derzeitige Entwicklung am Konsumentenmarkt für die weitere Verbreitung der VR-Technologie von Interesse, da die beiden genannten Defizite auch kurzfristig beseitigt werden könnten. Entsprechende Ansätze, z. B. das Trackingsystem der Wii™-Spielekonsole, werden im Kapitel 3.4 vorgestellt.

Eine weitere Technologie, die für die Erzeugung stereoskopischer Szenen eingesetzt wird, ist die Autostereoskopie. Im Vergleich zur IPT-Technik muss der Benutzer bei autostereoskopischen Systemen keine Brille tragen. Der stereoskopische Effekt wird entweder durch Folien mit kleinen, zylinderförmigen Strukturen oder durch ein volumetrisches Display realisiert [Bowman 2004]. Das Fraunhofer Heinrich-Hertz-Institut (HHI) bietet zum Beispiel entsprechende Lösungen, die mit einem feinen Raster über der Anzeige aus Flüssigkeitskristall arbeiten. Das Raster ermöglicht jedem Auge des Betrachters nur den Blick auf einen Teil der Pixel. Weil beide Augen unterschiedliche Pixel zu sehen bekommt, lassen sich gleichzeitig verschiedene Bilder an die Augen liefern [Börner 2000; HHI 2010]. Die Lösung mit Folien auf der Bildschirmoberfläche reduziert allerdings die physikalische Auflösung des Anzeigesystems. Ferner kann sie dem Benutzer nur dann einen korrekten Stereoeffekt bieten, wenn er in einer bestimmten oder zumindest in einer von wenigen möglichen Positionen vor dem Gerät steht. Volumetrische Displays dagegen erzeugen ein „reales“ 3D-Objekt innerhalb ihres Anzeigeraums. Allerdings kann sich der Benutzer selbst auch bei dieser Anzeigetechnik nicht in die Szene hinein bewegen.

Das Konzept in der vorliegenden Arbeit orientiert sich an den virtuellen Umgebungen mit IPT, bei denen ein die Person zumindest teilweise umgehender Interaktionsraum gegeben ist. Die Recherchen zum Stand der Technik in diesem Kapitel berücksichtigen Literatur über VR-Systeme mit verschiedenen Displaytechnologien. In einigen Punkten werden auch Arbeiten über 3D-Anwendungen mit 2D-GUI angeführt.

### 3.1.2 Mensch-Maschine-Schnittstelle und Benutzungsoberfläche für die VR

Die Mensch-Maschine-Schnittstelle erlaubt dem Benutzer das *Bedienen* der Maschine und das *Beobachten* der Anlagenzustände bzw. das *Eingreifen* in den Prozess [Wikipedia 2010]. Im Computerbereich definiert die internationale Organisation für Normung (*International Organization for Standardization*, ISO) die Benutzerschnittstelle als „alle Hardware- oder Softwarebestandteile eines interaktiven Systems, die die Informationen und Steuerelemente zur Verfügung stellen, die für den Benutzer notwendig sind, um eine bestimmte Arbeitsaufgabe mit dem interaktiven System zu erledigen“ [ISO-9241-110]. Im Softwarebereich sind bis heute drei Generationen von Computer-Benutzerschnittstellen zu unterscheiden. Tabelle 3-1 listet diese in einer Übersicht auf.

Ferner gibt es auch computergestützte Benutzerschnittstellen, die auf Sprachsteuerung oder anfassbaren Eingabegeräten basieren. Allerdings sind diese bis jetzt nicht so weit verbreitet, wie die oben genannten Arten mit generischen Eingabegeräten.

Die Mensch-Maschine-Schnittstelle für virtuelle Umgebungen unterscheidet sich sowohl in den Hardware- als auch in den Software-Komponenten von den Benutzerschnittstellen für 2D-Systeme. In diesem relativ jungen Entwicklungsbereich gibt es bis heute keinen vergleichbaren Standard, wie er bei Desktop-Anwendungen etwa mit dem WIMP-Konzept (WIMP steht für *Windows*, *Icon*, *Menü* und *Pointing device*) gegeben ist. In der vorliegenden Arbeit steht der Begriff immersive Benutzungsoberfläche oder VR-Benutzungsoberfläche (IUI) für den softwaretechnischen Teil der Mensch-Maschine-Schnittstelle in den immersiven virtuellen Umgebungen.

Generation	Merkmale	Eingabegerät	Beispiele
Kommandozeilen (Command Line Interface, CLI)	Eintippen von Befehlen	Tastatur	Command (DOS) bash (Unix)
Zeichenorientierte Benutzungsschnittstellen (Text User Interface, TUI)	Menüs aus textbasierenden Zeichen	Tastatur, eventuell Maus	Norton Commander für MS DOS
Grafische Benutzungsoberfläche (Graphical User Interface GUI)	Fenster, Icons, Menüs und Zeiger (WIMP)	Maus Tastatur	MS Windows KDE GNOME

**Tabelle 3-1: Drei Generationen computergestützter Benutzerschnittstellen**

### 3.1.3 Richdata

Eine wesentliche Herausforderung für heutige VR-Plattformen besteht darin, neben den Geometriedaten auch weitere vielfältige Informationsformen in den immersiven Umgebungen repräsentieren zu können, diese mit diversen internen und externen Quellen verknüpfen und intuitiv mit der VR-Benutzungsoberfläche bedienen zu können.

Die Entwicklung der VR-Technologie war von Anfang an eng mit der stereoskopischen Darstellung der 3D-Modelle verbunden. Inzwischen werden entsprechende stereoskopische Systeme immer öfter auch in der Industrie eingesetzt. Damit erhöhen sich auch die Anforderungen an die Prozessintegration und Benutzbarkeit der VR-Systeme [Spath 2009]. Insbesondere werden für die Arbeit in immersiven Umgebungen vielfältige Informationsformen benötigt. Dazu zählen beispielsweise Beschriftungen, Bilder, Sprachaufzeichnungen oder Webseiten. In der VR-Literatur werden die Informationsformen unter verschiedenen Bezeichnungen behandelt, ohne eine konkrete Definition zu geben:

- Metadaten oder Meta-Information  
Mit dem Begriff *Metadaten* werden allgemein Daten bezeichnet, welche Informationen über andere Daten enthalten [Wikipedia 2010]. Zu den typischen Metadaten in 3D-Anwendungen gehören Beschreibungen in Form von Texten, Skizzen, Diagrammen oder Maßen. Solche Beschreibungen stehen meistens mit ei-

nem Bezugsobjekt in Verbindung, das in Form eines 3D-Modells in der VR-Anwendung dargestellt ist.

- **Abstrakte Informationen**  
Welche Informationen als konkret oder abstrakt bezeichnet werden können, hängt vom Einsatzbereich und Ausgangsaspekt des Beobachters ab. Unter dem Begriff abstrakte Informationen können zum Beispiel die symbolischen Daten eingeordnet werden, die keine konkrete Geometrie-Repräsentation besitzen [Bowman 1999; Card 1999; Bowman 2003; Polys 2005]. Viele Benutzer würden aber beispielsweise Geometriedaten, die eine komplexe Strömung visualisieren, eher als abstrakt bezeichnen, als symbolische Informationsformen wie Text oder Bilder.
- **Attributierung**  
Eine Zuordnung von Attributen kann generell als Attributierung bezeichnet werden. In der Computergraphik wird üblicherweise von der Attributierung eines Graphen gesprochen, wenn Knoten oder Kanten im Graphen mit Daten versehen werden [Dahm 2003].
- **Annotation**  
Annotationen sind Informationen, die der Benutzer während des Arbeitsprozesses erstellt, um die gewonnenen Erkenntnisse zu dokumentieren. In 3D-Anwendungen liegen Annotationen meist in Form von Text oder Sprachaufzeichnung vor. Der Begriff der Annotation ist damit mehr mit dem Erstellungsprozess als mit den Informationsformen verbunden.

Der Begriff **Richdata** (RD), der in dieser Arbeit verwendet wird, bezeichnet eine Anreicherung der virtuellen Umgebungen, die bisher von statischen Geometriedaten dominiert werden, durch eine Vielfalt von Datenquellen, -strukturen und Repräsentationsformen. Richdata ist dadurch von den konventionellen 3D-Modelldaten abgegrenzt, dass sie nicht wie die Modelldaten selbst zur Nachbildung von realen Szenen in einer virtuellen Umgebung dienen.

## 3.2 Grundlegende Richdata-Typen in der Literatur

### 3.2.1 Symbolische Information

Symbolische Informationen wie Zahlen oder Text können kurz und präzise Inhalt übermitteln. Die virtuelle Umgebung ohne symbolische Informationen ist wie die reale Welt ohne Sprache oder Mathematik. Trotzdem sind sie heute in 3D-Benutzerschnittstellen beziehungsweise in den VR-Benutzerschnittstellen wenig präsent [Bowman 2004].

Eine verbreitete Technik, um eine Text-Beschriftung des 3D-Modells in der Szene anzuzeigen, ist die Überlagerung von 3D-Objekt und den Metadaten auf der Bildelebene. Positionen einzelner Elemente der 3D-Inhalte werden dazu auf eine 2D-Oberfläche projiziert.

ziert und dienen hier als Referenzpunkte für die Beschriftung. Beispiele für diese Vorgehensweise finden sich sowohl in vielen 3D-Spielen oder in Anwendungen wie Google-Earth™ (siehe Abbildung 3-2 links).

Schwierig zu lösen ist bei dieser Technik das Überdeckungsproblem. Die 2D-Informationen müssen möglichst so positioniert werden, dass sie sich nicht gegenseitig und nicht die zu beschreibende Geometrie verdecken. Da dieses Überdeckungsproblem im Sinne der Komplexitätstheorie NP-vollständig ist [Marks 1991], wird es kaum möglich sein, eine allgemeine, effiziente Lösung zu finden. Stein und Decoret entwickeln ein GPU-basiertes Verfahren, das unter Verwendung einer Graphikarte *GeForce® 7800* bis zu 20 Beschriftungen positionieren kann. Die Anzeige arbeitet dabei mit 30 Hz und 512 x 512 Pixel Auflösung [Stein 2008], siehe Abbildung 3-2 rechts.



Abbildung 3-2: Überlagern von 2D-Elementen und 3D-Szene

Die Technik der Überlagerung kann in immersiven Umgebungen nur eingeschränkt eingesetzt werden. Bei Systemen mit 2D-GUI gibt es eine physikalische Projektionsebene im Bildschirm, der gleichzeitig als Fenster zur virtuellen Welt fungiert. Bei VR-Systemen mit HMD ist es ähnlich. Auch die IPT-Systeme haben zwar die physikalischen Projektionswände, sie sind aber durch die Stereoskopie nicht mehr vom Benutzer wahrnehmbar. IPT-Systeme allerdings bieten eine weitere Möglichkeit, die in der Arbeit von Polys und Kollegen vorgestellt wird: Sie verzichten dabei an einer Projektionswand auf die Stereoskopie [Polys 2004a].

Die eingeschränkte Anwendbarkeit der Überlagerungstechnik liegt grundsätzlich darin, dass zweidimensionale Information wie Text oder Bild nicht innerhalb der 3D-Szene existiert, sondern nachträglich zum gerenderten Bild hinzugefügt wird. Die Forschung an der technischen Universität Virginia unterscheidet verschiedene Repräsentationstechniken für 2D-Objekte in 3D- und immersiven Systemen. Zum Projekt *Information-rich virtual environments* (IRVE) wurden mehrere Artikel veröffentlicht [Bowman 1999; Bowman 2003; Polys 2004b], in denen eine Reihe von Begriffen eingeführt wurden, um die Techniken sowie ihre Vor- und Nachteile kennzeichnen und analysieren zu können.



- Objekt-Kontext (object space)

Beschriftungen werden direkt an dem zu beschreibenden Objekt oder zumindest in dessen unmittelbarer Nähe angezeigt [Polys 2006b]. Die Position der Beschriftung lässt sich mit dem jeweiligen Referenz-Objekt verbinden. Ein Beispiel zeigt die Abbildung 3-3 a. Die Texte selbst haben zwar keine Tiefe, sie existieren aber in der 3D-Szene. Damit ist diese Technik auch geeignet für immersive Benutzungsoberflächen. Ein Nachteil ist, dass die Texte andere Beschriftungen oder einen Teil des 3D-Objekts verdecken können beziehungsweise selbst von anderen Elementen verdeckt werden. Von Vorteil ist, dass die Beschreibungen direkt neben dem Bezugsobjekt angezeigt werden und der Benutzer so alle Information ohne Umstände findet.

- Display-Kontext (display space)

Mit Display-Kontext wird eine Methode bezeichnet, bei der die Beschriftung eines Objekts im 3D-Raum selbst minimiert wird. Zusätzlich wird jedoch ein weiteres Anwendungsfenster geöffnet, um detaillierte Informationen anzeigen zu können [Polys 2006b]. Verschiedene Visualisierungen der Daten lassen sich so ebenfalls in weiteren Fenstern problemlos darstellen. Die Methode gleicht der in 2D-Anwendungen üblichen Bildschirmanzeige (siehe Abbildung 3-3 b). Diese Technik ist demnach auch nur für solche 3D-Anwendungen geeignet, die mittels traditioneller 2D-Benutzungsoberfläche bedient werden.

- Viewpoint-Kontext (viewpoint space)

Der Viewpoint-Kontext wird durch Überlappen realisiert und zeigt einen möglichen Mittelweg zwischen dem Objekt- und dem Display-Kontext auf (siehe Abbildung 3-3 c). Die Beschriftungen werden im selben Fenster wie das 3D-Objekt an einer berechneten Position angezeigt [Polys 2005]. Die Verbindung zum Referenzobjekt wird durch eine Linie im 3D-Raum dargestellt. Ein Manager wird implementiert, der die Positionen der Beschriftungen auf der 2D-Benutzungsoberfläche automatisch an die 3D-Szene anpasst. In der Abbildung 3-3 d werden Objekt- und Viewpoint-Kontext gleichzeitig verwendet [Polys 2007].

- Snap-Sicht (snap view)

Die Snap-Sicht ist eine Kombination aus einer immersiven und einer 2D-Benutzungsoberfläche. Eine Wand der CAVE™ wird getrennt von den übrigen Projektionsflächen zum Anzeigen der Metadaten benutzt. So wird die abstrakte Information auf einer 2D-Benutzungsoberfläche angezeigt, die neben der immersiven Umgebung liegt [Polys 2004a].

Polys und Kollegen vergleichen die unterschiedlichen Techniken mit dem Fokus auf den Objekt- und den Display-Kontext [Polys 2006b]. In der Arbeit werden insgesamt fünf Repräsentationstechniken für abstrakte Informationen erwähnt. Neben Objekt- und Display-Kontext zählen dazu noch Viewpoint-Kontext, Welt-Kontext (world space) und Be-

nutzer-Kontext (user space). Im Rahmen der vergleichenden Analyse werden zwei Aktionen auf ihre Benutzbarkeit untersucht: Suchen und Vergleichen. Der Vergleich zwischen den beiden Techniken ist auch deshalb von besonderer Bedeutung, da sie für jeweils andere VR-Systeme besonders geeignet sind: Der Display-Kontext überwiegend für 2D-GUI und der Objekt-Kontext für immersive UI mit IPT.

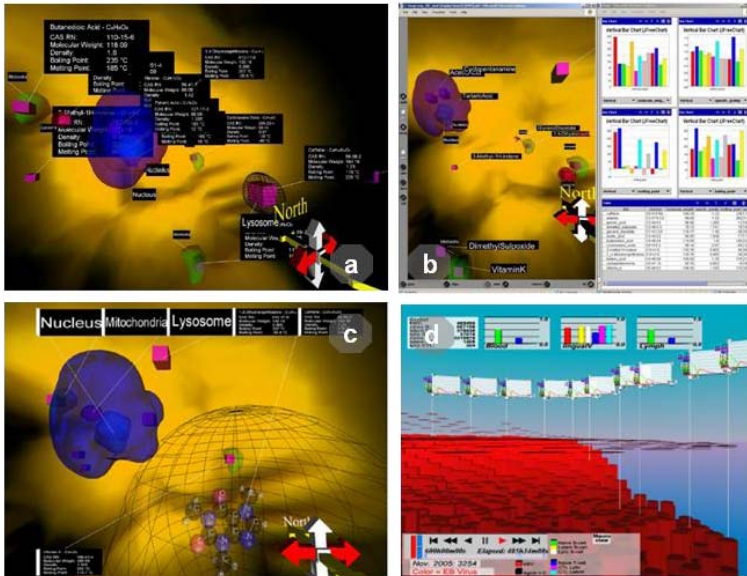


Abbildung 3-3: 2D-Elemente in IRVE

[Polys 2005; Polys 2006b; Polys 2007]

- a) Objekt-Kontext      b) Display-Kontext  
c) Viewpoint-Kontext      d) Objekt- und Viewpoint-Kontext

Laut obiger Vergleichsstudie zeigten die Benutzer eine starke Vorliebe zum Display-Kontext. Der Autor folgert daraus, dass die Sichtbarkeit der Information wichtiger ist als die Wahrnehmung ihrer Position im 3D-Kontext. Wird Display-Kontext eingesetzt, lassen sich abstrakte Informationen schneller vergleichen. Allerdings zeigen die Screenshots der verwendeten Anwendungsbeispiele (Abbildung 3-3 a und b), dass im Display-Kontext bereits visualisierte Informationen wie Tabellen und Diagramme eingesetzt wurden, während sich beim Objekt-Kontext die Informationen auf reinen Text beschränken.

In einer weiteren Arbeit untersucht Polys den Effekt eines Multi-Display-Systems im Vergleich zu Systemen mit einem einzigen Display. Ein Resultat davon ist, dass bei Einsatz einer größeren Anzeigefläche die Benutzer den Objekt-Kontext bevorzugen [Polys 2005].

Bell und Kollegen entwickelten eine annotierte Miniatur mit *Augmented Reality* Technologie (AR), um den Benutzer bei der Orientierung in einer fremden Umgebung zu unterstützen, wie in der Abbildung 3-4 angezeigt [Bell 2001; Bell 2002]. Als Hardware wird ein HMD eingesetzt. Der Schwerpunkt der Arbeit liegt auf der Koordination der notierten Informationen. Es wird eine statische Lösung für das Überdeckungsproblem in der AR-Anwendung gegeben. Jeder Annotation wird eine Priorität zugeordnet. Falls die Annotation mit niedriger Priorität in der Anzeige keinen Platz mehr findet, wird sie am Rand des Viewports angezeigt. Ferner realisieren Bell und Kollegen eine automatische Abstimmung des Metadaten-Bezugs zwischen der realen Umgebung und der mit AR generierten Miniatur.



Abbildung 3-4: Annotierte Miniatur von Bell et al  
[Bell 2001; Bell 2002]

Ebenfalls im Forschungsbereich AR-Technologie stellen Reitmayr und Schmalstieg von der Universität Wien eine Navigationsanwendung vor [Reitmayr 2001; Reitmayr 2003]. Besucher der Stadt Wien werden mit Hilfe eines mobilen HMD durch die Stadt geführt. Informationen wie Standort, Markierungen und Zeiger, die den weiteren Wegverlauf anzeigen, werden am mobilen HMD zu der realen Szene hinzugefügt. In der AR-Benutzungsoberfläche werden sowohl Objekt- als auch Viewpoint-Kontext verwendet.

Die VR-Gruppe am Rechen- und Kommunikationszentrum der RWTH-Aachen hat ein Dokumentationssystem entwickelt, mit dem 3D-Texte in das immersive VR-System VISTA [Reimersdahl 2000] integriert werden können. Aus Sprachaufzeichnungen werden Annotationen in 3D-Text umgewandelt und im 3D-Raum positioniert [Assenmacher 2006].

Texte, die in eine virtuelle Umgebung eingebracht werden, können nicht immer senkrecht zum Benutzer und parallel zu der Projektionsebene stehen. Larson und Kollegen untersuchten die Beziehung zwischen der Erkennbarkeit der Texte, der Orientierung des Benutzers und der Schriftgröße [Larson 2000]. Die Ergebnisse werden in Abbildung 3-5 illustriert. Eine kleine Schriftgröße wirkt sich demnach ungünstig auf die Orientierung aus (siehe Diagramm in der Abbildung 3-5 links). Das Diagramm in Abbildung 3-5 rechts zeigt, dass derselbe Text auf einer linken Wand besser als auf einer rechten Wand er-

kannt wird. Dies könnte damit erklärt werden, dass ein deutlich wahrnehmbarer erster Buchstabe die Worterkennung stärker erleichtert, als wenn der letzte Buchstabe im Vordergrund ist.

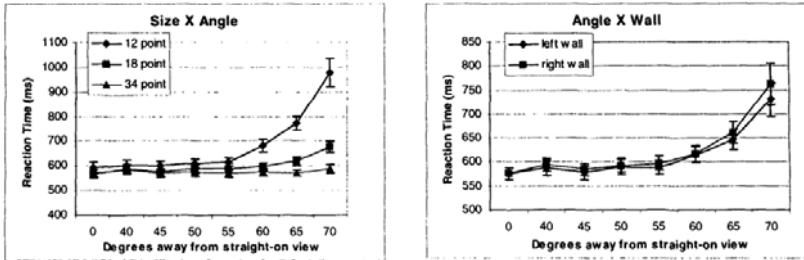


Abbildung 3-5: Erkennbarkeit eines 3D-Textes

### 3.2.2 Sprachaufzeichnung und Audio

In immersiven Umgebungen hat die Verwendung von Sprachaufzeichnung als Eingabeform große Vorteile. Vor allem, da in der VR-Umgebung Standard-Eingabegeräte wie Tastatur und Maus schwierig einsetzbar sind. Bei der Sprachaufzeichnung sind beide Hände des Benutzers frei. Somit ist die Arbeit am 3D-Objekt in natürlichen Bewegungsabläufen möglich und es lassen sich auch lange Texte problemlos eingeben [Bowman 2004]. Für Anwendungsdesigner bedeutet dies beispielsweise, dass sie mehr Spielraum in der Interaktion, insbesondere für Zwei-Hand-Interaktionsmetaphern, erhalten.

Im VR-System wird die Augenposition und -Orientierung des Benutzers ständig verfolgt. Über den Augenfokus bestimmt und steuert der Benutzer die aktuell angezeigte stereoskopische Szene. Daher ist die Sprachaufzeichnung für die Arbeit des Interaktionsdesigners in immersiven Umgebungen ein besonders hilfreiches Eingabemedium. Denn sie ist unabhängig vom Augenfokus einsetzbar, da sie keinen Blickkontakt zu einem Eingabegerät benötigt.

Verlinden und Kollegen entwickelten ein Annotationssystem mit Sprachaufzeichnung [Verlinden 1993]. Diese Arbeit basiert auf dem immersiven System *VAnno*. In der Arbeit von Harmon und Kollegen wurde das Annotationssystem weiterentwickelt [Harmon 1996]. *VAnno* wird bei mehreren VR-Projekten eingesetzt.



nen Kopfhörer tragen, da das VirKopf-System die Stimme auf beiden Ohren mit der *Crosstalk Cancellation* Technik simulieren kann. Voraussetzung bei der Sprachaufzeichnung ist, dass zusätzlich ein optisches Tracking-System verwendet wird. Dies ist in immersiven Umgebungen jedoch in der Regel der Fall, da die Augenposition und – Orientierung auch für die stereoskopische Bilderzeugung in der VR benötigt wird. Die korrekte Simulation des räumlichen Effekts bei der Tonwiedergabe gilt allerdings nur für die Person, die vom Tracking verfolgt wird. Der Stereo-Effekt kann sowohl für einen einzelnen Raum als auch für mehrere aufeinander folgende Räume errechnet werden. Laut dem Autor braucht das *VirKopf*-System nur 35 Millisekunden, um den korrekten Raumklang wiedergeben zu können.

Sodnik und Kollegen untersuchten mit ihrem AR-System, ob der Benutzer die Tonquelle anhand einer simulierten Stimme in der virtuellen Umgebung lokalisieren kann [Sodnik 2006]. Die Ergebnisse zeigen große Schwankung in der Lokalisierungszeit in Abhängigkeit von fünf zusätzlichen Nebenbedingungen. Aber auch unter denselben Bedingungen können von Fall zu Fall teils große Schwankungen beobachtet werden. Dennoch lässt sich als Ergebnis festhalten, dass die Versuchspersonen bezüglich der Seitenwinkel besser differenzieren können, als hinsichtlich der Raumtiefe oder der Distanz.

Als weiteres Forschungsgebiet beschäftigt sich die Auralisation mit Möglichkeiten, einen Raum hörbar zu machen. Dabei werden Verfahren wie die Simulation von Spiegelschallquellen, Raytracing und die Errechnung des Diffusschalls eingesetzt. Die Rechenzeiten raumakustischer Simulationen sind bedingt durch prinzipielle Unterschiede zwischen Licht- und Schallausbreitung um Größenordnungen höher als bei Visualisierungsproblemen. Die bei der VR geforderte Echtzeitfähigkeit ist daher ohne Vereinfachung ein kaum lösbares Problem [Schupp 1994].

Eine Kooperation der Fraunhofer-Institute IAO, IBP, IPA und IGD setzte die Echtzeit-Auralisation mittels einer Näherungslösung um [Schupp 1994]. Die Lösung basiert auf einer vorherigen Berechnung der Nachhallzeit und berücksichtigt nur die grobe Raumform. Unter der Nachhallzeit versteht man das Zeitintervall, innerhalb dessen der Schalldruck in einem Raum bei plötzlichem Verstummen der Schallquelle auf den tausendsten Teil seines Anfangswerts abfällt. Dies entspricht einer Pegelabnahme von 60 dB (Wikipedia). Die Übertragung der Schallfelddaten dauert je nach Anzahl der diskreten berücksichtigten Schallquellen zwischen 0,2 und 1 Sekunde. Hardwareseitig setzt das Projekt HMD und Kopfhörer ein.

### 3.2.3 Bilder

Bilder und bewegte Bilder sind zweidimensionale Informationen. Sie können ebenso wie symbolische Informationen auf einer physikalischen Anzeigeebene überlappend angezeigt werden. Wie im Kapitel 3.2.1 erläutert, ist diese Technik für VR-Systeme mit IPT allerdings ungeeignet. Eine häufig verwendete Möglichkeit, Bilder in eine immersive

Umgebung bringen zu können, ist es, sie als Textur einer Geometrie in der 3D-Szene zu verwenden.

Softwareanwendungen für Konsumenten wie *2nd Life*® versuchen, mit einfacher Geometrie eine möglichst akzeptable Szene zu erstellen. Deshalb werden Texturen sowohl als Hintergrundstruktur, zur Darstellung eines Textes oder zur Präsentation von Bildern eingesetzt. Eine Beispielszene zeigt Abbildung 3-7. Ein wichtiges Ziel des Einsatzes von Texturen ist es, die Dateigröße bzw. die Komplexität der 3D-Modelle zu reduzieren. Auch die anzuzeigenden Bilder selbst dürfen daher nicht zu groß sein. Häufig werden sie daher komprimiert [ITU 1992] oder lediglich im Mono-Format verwendet.



Abbildung 3-7: Texturen in Commodity-Software [LindenLab 2009]

In technischen Anwendungsfällen werden 3D-Modelle direkt von CAD-Systemen generiert bzw. aus der CAD-Anwendung heraus in ein View-Format exportiert. So erzeugte Fahrzeugmodelle beinhalten erfahrungsgemäß zwischen fünf und fünfzehn Millionen Polygone. Spezifische Renderer können über eine Billion Polygone bearbeiten [Brüderlin 2007]. Zudem sind die Modelle komplex strukturiert, damit auch Baugruppen und das Zusammenspiel von Komponenten nachgebildet werden können. Die Details des Modells werden daher nicht durch vereinfachende Texturen, sondern durch Polygone und Materialien bestimmt und dargestellt.

Dennoch werden Bilder und Texturen in der VR gebraucht: Bilder und Texturen in hoher Qualität sind eine Voraussetzung, um die virtuelle Szene realitätsnah abzubilden. Ein Beispiel zeigt Abbildung 3-8 [Grimm 2009]. Ein Hochkontrastbild (*High Dynamic Range Image*, HDRI) mit Tone-Reproduktion (Tone-Mapping) wird verwendet, um die Umgebung und den Shader für reflektierende Oberflächen zu erzeugen. Mit HDRI lassen sich auch große Helligkeitsunterschiede, wie sie in natürlichen Umgebungen vorzufinden sind, detailgetreu speichern.



Abbildung 3-8: Bilder und Texturen mit HDR1 und Tone-Reproduktion [Grimm 2009]

In der Natur entspricht der Unterschied zwischen dem Sonnenlicht und dem Sternenlicht einem Dynamikumfang von über  $10^9:1$  [Reinhard 2006]. Die für eine Kamera oder einen Betrachter sichtbare Umgebung weist typischerweise einen Dynamikumfang in der Größenordnung von  $10^4:1$  auf; maximal wird innerhalb einer Szene ein Dynamikumfang bis zu ungefähr  $10^5:1$  erreicht [Ferwerda 2001]. Beim Tone Mapping wird der Kontrastumfang eines HDR-Bildes verringert, um es auf herkömmlichen Ausgabegeräten darstellen zu können.

Die Grenze zwischen zweidimensionalen Bildern und dreidimensionaler Geometrie kann in speziellen Fällen verwischen. Moderne Graphikkarten erlauben Abänderungen bzw. das Erstellen von Geometrie mittels besonderer Textur-Technologie. Z. B. können mit *Displacement-Mapping* flache Oberflächen durch komplexe geometrische Strukturen ersetzt werden.

Außerdem kann Geometrie mit Hilfe unterschiedlicher Algorithmen in Form von quadratischen Bildern gespeichert werden [Gu 2002; Losasso 2003]. Dazu werden die Positionen der Vertices als RGB-Werte des Bildes dargestellt. Diese Technik ermöglicht es, irreguläre Mesh-Strukturen in einem regulären Format zu speichern. Zudem ist es möglich, das Geometrie-Bild durch Wavelet-Transformation oder Midmap zu komprimieren.

### 3.2.4 Bewegte Bilder

Manche Videokonferenz-Lösungen (*Video-Mediated Communication*, VMC) bieten den Gesprächsteilnehmern neben der konventionellen 2D-GUI auch die Möglichkeit, sich in einem virtuellen 3D-Raum zu treffen.





**Abbildung 3-9: IBM Sametime 3D (links) und Alice Street (rechts)**  
[AliceStreet 2009; Fishkind 2009]

IBM Sametime 3D stellt den Gesprächspartner durch einen Avatar in einem virtuellen Sitzungsraum dar [Fishkind 2009]. Einen Screenshot der Anwendung zeigt Abbildung 3-9 links. Am Anfang eines Gesprächs kann der Benutzer den virtuellen Raum aus einer Liste auswählen. Die Gesprächspartner können auch Informationen in Form von Texten oder Präsentationen austauschen. Ähnliche Funktionalität bietet ebenfalls das Konferenzsystem smartMeeting [smartMeeting 2006].

Andere VMC-Anwendungen, wie z. B. Alice Street, stellen die Gesprächspartner in einem Video-Rahmen in der virtuellen Umgebung dar. Dadurch wird eine Gesprächssituation geschaffen, die auch Blickkontakt und Körpersprache ermöglicht (siehe Abbildung 3-9 rechts). Einige VMC-Programme erlauben dem Benutzer eine freie Navigation im 3D-Raum [Regenbrecht 2003; Regenbrecht 2004; TixeoSoft 2008], während andere dem Benutzer eine festgelegte Position vorgeben [AliceStreet 2009].

Softwarelösung	Freie Navigation	Immersion	Bemerkungen
IBM Sametime 3D	Ja	Nein	Text, Präsentationen, Board [Fishkind 2009]
Smart Meeting	Ja	Nein	[smartMeeting 2006]
WorkSpace 3D	Ja	Nein	Bilder, Web-Browser, PDF, 3D-Text, Desktop-Sharing [TixeoSoft 2008]
Alice Street	Nein	Nein	MS PowerPoint, Outlook IBM Domino [AliceStreet 2009]
cAR/PE!	Ja	AR	Bilder, Remote-Desktop (VNC) [Regenbrecht 2004]

**Tabelle 3-2: Überblick über kommerzielle Lösungen für 3D-Konferenzen**



Abbildung 3-10: Illustration der vier Interaktionsmodi  
[Hills 2005; Hauber 2006]

Da die VMC-Lösungen speziell für Konferenzsituationen entwickelt werden, unterstützen sie in der Regel vielfältige Formen zum Informationsaustausch wie z. B. Texte, Bilder und PowerPoint. Workspace 3D und cAR/PE! erlauben den Gesprächspartnern zudem, im virtuellen Raum einen Computerdesktop miteinander zu teilen. Immersive Darstellung wird dabei nur bei cAR/PE! in Kombination mit einem HMD realisiert. Einen Überblick über die verschiedenen VMC-Lösungen und deren Funktionalitäten gibt Tabelle 3-2.

Hills, Hauber und Kollegen haben eine Benutzbarkeitsuntersuchung [Hills 2005; Hauber 2006] durchgeführt, um die Vor- und Nachteile verschiedener Kommunikationsschnittstellen für VMC-Anwendungen zu vergleichen. Berücksichtigt werden dabei vier Interaktionsmodi: die konventionelle 2D-Schnittstelle, die Spatial-Lokal-, die Spatial-Remote- und die Face-to-Face-Schnittstelle. Eine Illustration der jeweiligen Funktionsprinzipien zeigt Abbildung 3-10.

Das Ergebnis der Untersuchung zeigt, dass die Face-to-Face-Kommunikation wie erwartet in allen Bereichen die besten Bewertungen erhielt. Bei den computerunterstützten Modi erreichte die Spatial-Remote-Schnittstelle die besten Bewertungen in Bezug auf die soziale Präsenz. Die konventionelle 2D-Schnittstelle schnitt in diesem Punkt zwar am schlechtesten ab, erhielt aber die besten Bewertungen bei der Arbeitseffizienz.

Dieses Ergebnis widerspricht der Erwartung des Autors. Es wurde generell erwartet, dass durch die Nutzung einer zusätzlichen Dimension und den damit verbundenen zusätzlichen Freiheitsgraden sich die Arbeitseffizienz erhöhen lässt. Allerdings lassen sich die 3D-Benutzerschnittstellen künftig noch in vielen Punkten verbessern. Im Folgenden werden einige Vorschläge dazu vorgestellt:

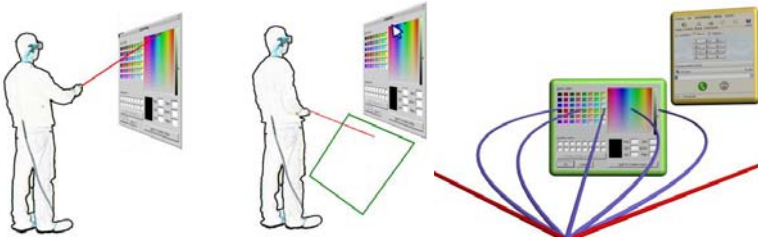
- Fokussieren auf Funktionen, die dem Benutzer wirklich helfen. Der Augenkontakt führt laut den Untersuchungsergebnissen zwar nicht zur Effizienzsteigerung, aktiviert aber die Gestik der Hand und die Körpersprache der Benutzer. Es kann damit erwartet werden, dass sich die Arbeitseffizienz dadurch verbessern lässt, indem die Körpersprache gezielt durch weitere Interaktionsmetaphern unterstützt wird.

- Bevorzugtes Fördern des Arbeitsprozesses anstatt des Kontexts. Die Interaktionschnittstelle muss das spezielle Ziel der Arbeitsperiode explizit unterstützen können. Wird dies nicht erreicht, können auch die Nebenbedingungen, wie mehr Navigationsfreiheit, keine positive Wirkung entfalten.
- Eine spezielle Interaktionsmethode für die 3D-Umgebung kann zielführend sein. Für die Arbeit mit Spatial-Lokal-Schnittstellen wird im Test ein Touch-Table eingesetzt. Im Vergleich zu Interaktionen mittels Mouseeingabe zeigte der Einsatz eines Touch-Table eine deutliche Überlegenheit.
- Eine freie Navigation bietet zwar bessere immersive Effekte, verursacht aber auch Benutzeraktionen, die irrelevant für das eigentliche Arbeitsziel sind. Um zielorientiertes Arbeiten zu unterstützen kann es daher hilfreich sein, bestimmte Freiheitsgrade zu deaktivieren oder einzuschränken.
- Quantität der Information: Es wird ein Zusammenhang vermutet zwischen der Menge der für eine Arbeit benötigten Information und der Benutzbarkeit der verschiedenen Schnittstellen. Ist die Menge der benötigten Informationen so groß, dass sie auf einem konventionellen Display nicht oder nur unzulänglich angezeigt werden können, werden die Benutzer die dreidimensionale Schnittstelle bevorzugen.

Eine ähnliche Ansicht äußert auch Polys in seiner Arbeit [Polys 2006b]. Er stellt fest, dass die Arbeitseffizienz von vielen Faktoren, z. B. der Menge der Informationen, abhängig ist. Nielsen erläutert im Internet seine Auffassung [Nielsen 1998], dass 3D-Benutzungsoberflächen in Kombination mit konventioneller Hardware bzw. existierenden Anwendungen weniger effektiv sind als 2D-Benutzungsoberflächen. Insbesondere ist die Darstellung von abstrakten Informationen nach der Meinung Niensens in 3D schlechter als in 2D.

Zu den bewegten Bildern gehören nicht nur Video, sondern auch weitere 2D-Anwendungen, die durch allgemeine oder spezielle Methoden in eine 3D-Umgebung eingebracht werden können. Das Projekt *WebVRI* hat z. B. den Microsoft® Internet Explorer® durch ActiveX erweitert, um ihn in 3D-Umgebungen verwenden zu können [Barsoum 2008].

Beim Konferenzsystem *cAR/PE!* (siehe auch Tabelle 3-2) kann ein Remote-Bildschirm mit einer 3D-Szene ein abgeändertes UltraVNC-Programm [UltraVNC 2009] anzeigen bzw. darauf zugreifen. Diese allgemeine Lösung eignet sich für jede beliebige Anwendung mit 2D-GUI, die auf einem VNC-fähigen Rechner laufen kann. Ähnliche Ansätze, die VNC dazu nutzen, 2D-GUI bzw. GUI-Elemente in VR zu bringen, gibt es auch für die VR-Plattformen Lightning [Bues 2003] und *3Dwm* [Elmqvist 2003].



**Abbildung 3-11: VirtualPad und FrictionSurface**  
[Andujar 2006; Andujar 2007]

Andujar und Kollegen verbesserten die Bedienungsmethode für die Verwendung eines Remote-Computers in immersiven Umgebungen. Um in stehender Haltung problemlos auf den Remote-Computer zugreifen zu können, verwenden sie verfolgte 6-DoF-Eingabegeräte [Andujar 2006; Andujar 2007]. Ihre Bedienungsmethode ermöglicht es, die Maus-Simulation in vergrößertem Bewegungswinkel oder auf einer virtuellen separaten Fläche durchzuführen (siehe Abbildung 3-11).

### 3.2.5 Stereo-Bilder und -Video

Die Geschichte der Stereoskopie beginnt mit Charles Wheatstone, der im Jahr 1833 das Spiegelstereoskop erfand [Wikipedia 2010]. Es gibt zahlreiche Literatur und Veröffentlichungen, welche die Technik zur Aufnahme und Anzeige von Stereo-Bildern und -Videos vorstellen. Fritz G. Waack gibt in seinem Buch eine grundlegende Einführung in diese Technik und praktische Hinweise [Waack 1985], wie sich mit einfachen Mitteln eine Stereokamera einrichten lässt. Gerhard P. Herbig gibt auf seiner privaten Website einen Überblick über die Thematik in deutscher Sprache [Herbig 2005]. Die Website vermittelt auch praxisnahes Wissen z. B. bezüglich der Wahl der Stereo-Basis und der Tiefenwahrnehmung.

Die gegebenen Informationen helfen dabei, physikalisch korrekte Stereo-Bilder bzw. -videos aufzunehmen. VR bietet weitere Möglichkeiten, Stereo-Bilder und -Videos durch Rendering für zwei Augenpositionen zu erzeugen. In der virtuellen Welt spielen die auf der Linsen- und Filmebene zu beachtenden Zusammenhänge keine Rolle. Daher sind auch die Berechnungen wesentlich einfacher.

Wimmer stellt in seiner Diplomarbeit die Geschichte und die technische Realisierung der Aufnahme von Stereo-Bildern bzw. -Videos vor [Wimmer 2004]. Im Rahmen seiner Arbeit hat er das Programm *StereoscopicPlayer* entwickelt. Diese Software bietet Unterstützung bei Übertragung, Einstellung sowie Wiedergabe und ist in der Praxis heute weit verbreitet. Außerdem wird ein Treiber, *Stereoscopic Multiplexer*, implementiert, der die Aufzeichnung von Life-Videos ermöglicht.

### 3.3 Datenquelle der Richdata

Richdata können sowohl von der Anwendung nach Bedarf erzeugt als auch aus einer Quelle geladen bzw. Quelldaten in Richdata umgewandelt werden. Eine wichtige Quelle sind die Geometriedatensätze selbst (siehe Kapitel 3.3.1), z. B. die Metadaten des 3D-Modells. Außer den Geometriedatensätzen können vielfältige Datenquellen mit einbezogen werden. In der vorliegenden Arbeit werden diese in zwei Kategorien eingeteilt: unstrukturierte und strukturierte Informationsquellen (Kapitel 3.3.2 und 3.3.3).

#### 3.3.1 Metadaten aus Geometriedatensätzen

CAD-Datensätze können Metadaten enthalten. Wenn sie zu Austauschformaten oder View-Formaten exportiert werden, lassen sich in der Regel auch die Metadaten mit exportieren. Die in den Geometriedatensätzen verfügbaren Metadaten sind meistens technische Beschreibungen in Form von Text und Zahlen. Eine Auswahl solcher Metadaten, die grundsätzlich als Quelle für Richdata verwendet werden können, wird in Tabelle 3-3 dargestellt.

Kategorie	Formate	Organisation	Einsatzschwerpunkte und Bemerkungen
Native CAD-Formate	CATIA	Dassault Systems	Body-Engineering Interieur
	Pro/ENGINEER	Parametric Technology Corporation	Powertrain
	NX	Siemens/Unigraphics Solutions	Body-Engineering Interieur
	Tribon	Kockum Computer Systems	Schiffsbau
	MicroStation	Bentley Systems	Anlagenbau, Schiffsbau
	Inventor	Autodesk	Maschinenbau
	SolidWorks	Dassault Systems	Maschinenbau
	Solid Edge	Siemens/Unigraphics Solutions	Maschinenbau
	AutoCAD	Autodesk	Architektur
	ArchiCAD	Graphisoft	Architektur
	Nemetschek	Nemetschek Gruppe	Architektur, Stadtplan
Austausch-	IGES	ASME/ANSI	[ASME-Y14.26M ; Schupp

Kategorie	Formate	Organisation	Einsatzschwerpunkte und Bemerkungen
Formate			1994]
	STEP	ISO	[ISO-10303] Nachfolge von IGES
	FBX	Alias/Autodesk	[Besner 2005]
Viewing-Formate	JT	Siemens/Unigraphics Solutions	[JT-Open]
	cgr	Dassault Systems	Digital Mock-Up
	Open Inventor	Silicon Graphics	[Wernecke 1994]
	VRML	Silicon Graphic /ISO/IEC	[ISO-14772]
	X3D	ISO/IEC	[ISO-19775-1.2] Nachfolge von VRML
	Pdf (3D-pdf)	Adobe	[Adobe 2006]
<b>Reine Geometrie</b>	STL	3D-Systems Corporation	3D-Drucker

**Tabelle 3-3: Überblick der VR-relevanten Datenformate**

Wie aus der Tabelle ersichtlich, sind die nativen CAD-Formate vielfältig. Zudem muss beachtet werden, dass es von jedem Format nochmals mehrere Versionen gibt, die nicht immer miteinander kompatibel sind. Zudem sind die technischen Details das Eigentum des CAD-Softwarehauses. Deshalb können VR-Plattformen meistens keine nativen CAD-Formate lesen. Soweit dies in Einzelfällen möglich ist, beschränkt sich die Lesefähigkeit meist auf eine bestimmte Version des einzelnen ausgewählten Formats.

Austausch- und Viewing-Formate werden in der Regel von Standardisierungsorganisationen erarbeitet. Sie werden von vielen CAD-Programmen unterstützt. Der Unterschied zwischen nativen und Viewing-Formaten liegt darin, dass im Viewing-Format vorliegende Datensätze nicht ohne Weiteres wieder zu CAD-Datensätzen konvertiert werden können. Datensätze im Austauschformat können auch mathematische Beschreibungen beinhalten, sodass sie in andere CAD-Programme importiert und weiterbearbeitet werden können.

Neben Geometriedaten können Datensätze im Austausch- oder Viewing-Format weitere Informationen enthalten. Diese können sowohl aus den ursprünglichen CAD-Datensätzen stammen oder nachträglich hinzugefügt werden. Letzteres ist möglich, weil der Standard oft eine Spezifikation für den Container-Typ vorsieht, damit externe Pro-

gramme, z. B. die Produktdaten- bzw. Lebenszyklusmanagement-Systeme (PDM/PLM), die Datensätze um zusätzliche Informationen ergänzen können. Solche Daten existieren meistens in Form von Text. Weitere Informationsformen sind möglich, da der Container-Typ erweiterbar ist.

### 3.3.2 Externe unstrukturierte Datenquellen

Ein typisches Beispiel einer unstrukturierten Informationsquelle ist das Internet. Jeder kann jede Art von Information in das System einfügen. Im Web sind sowohl abstrakte als auch in unterschiedlichen Wahrnehmungsformen vorliegende (*perceptual*) Informationen unter einer gesamten Struktur organisiert.

Barsoum und Kuester haben im Projekt *WebVRI* [Barsoum 2008] eine duale Lösung für Web- und 3D-Benutzungsoberflächen realisiert. Das WebVRI besteht aus zwei Teilen. Zum einen wird ein Webbrowser auf Basis des *Internet Explorers*® mit *ActiveX* erweitert. Dieser Webbrowser generiert statt einem sichtbaren Fenster ein Bitmap. Die Bitmaps werden in eine Textur im OpenGL-Format umgewandelt und einem I/O-Kontext zugeordnet. Der I/O-Kontext dient dazu, die Textur durch Änderungen der besuchten Webseite zu aktualisieren und die Benutzeraktionen, z. B. das Anklicken eines Links in der Texturdarstellung zum Webbrowser zu schicken. Falls die gewählte Webseite 3D-Modelle im OBJ-Format mit *python script* beinhaltet, können diese auf der 3D-Benutzungsoberfläche automatisch als Geometrie geladen werden. Die 3D-Benutzungsoberfläche ist der zweite Teil von WebVRI – ein Plugin von *VR-Engine*. Die Geometrie schwebt oberhalb des Objekts, das die Textur der Webseite zeigt. In der Veröffentlichung wird nicht erklärt, ob die erwähnte 3D-Benutzungsoberfläche bzw. die *VR-Engine* eine immersive Oberfläche bietet oder sich auf eine 3D-Darstellung beschränkt.

Faktor	Umfang
Anzeigesysteme	1 Bildschirm, 12 Bildschirme, 50 Bildschirme
Immersiones Niveau	None-, Semi, Full-Immersiones
Eingabegeräte	Maus, 3D-Maus, Wand, Daten-Handschuhe, Stift, Gurt-Tastatur, Berührungsbildschirm
Interaktionsmetapher	WIMP, interaktives Menü, Gesten-Erkennung, Sprachsteuerung, virtuelle Tastatur, Stift und Notizblock
Daten- und Link-Visualisierung	2D-Fenster, Hierarchie, Workspace, Cityscape

Tabelle 3-4: Umfang der Untersuchung vom Wright und Gracani  
[Wright 2008]

Wright und Gracanin versuchen in einem laufenden Projekt [Wright 2008], einen Vergleich des Leistungsverhaltens zwischen Desktop- und immersivem System bei der Behandlung von Web-Inhalten durchzuführen. Die umfangreiche Zusammenstellung von Faktoren, die in der Untersuchung berücksichtigt werden, werden in Tabelle 3-4 zusammen gefasst. Die Struktur und weitere logische Abhängigkeiten, wie z. B. Programme oder Hyperlinks, sollen in dem beschriebenen Projekt in einem 3D-Raum visualisiert werden. Die Ergebnisse der Projektarbeiten wären zwar für die vorliegende Arbeit interessant. In dem Artikel werden allerdings weder Layouts noch Screenshots veröffentlicht.

### 3.3.3 Strukturierte Datenquellen und VR-Integration

Datenbanken sind typische Beispiele für strukturierte Informationsquellen. Der Zugriff auf die Datenbanken ist nur über die vorgegebene Schnittstelle, die vom jeweiligen System bzw. von der jeweiligen Anwendung abhängig ist, möglich. Für die VR-Integration sind die Datenbanken die wichtigste Datenquelle.

Spath und Kollegen halten die VR für ein wichtiges Werkzeug in der digitalen Produktion [Spath 2009]. Sie darf dabei aber nicht nur als Werkzeug fungieren, sondern muss auch im Prozessumfeld der Digitalen Produktion integriert sein. Neun Faktoren werden als Bedingung des erfolgreichen Einsatzes von VR im Unternehmen genannt:

- Nutzen
- Kosten
- Personalkapazität
- Strukturintegration
- Wissensintegration
- Bauliche/räumliche Integration
- Prozessintegration
- Systemintegration
- Datenintegration

Wichtig ist festzuhalten, dass die Mehrzahl der Erfolgsfaktoren mit verschiedenen Integrationsaspekten verknüpft sind.

Stelzer und Kollegen erläutern, dass für die integrierte Verwaltung produktbeschreibender Informationen im engeren Sinn das Produktlebenszyklusmanagement (PLM) weitgehend etabliert und als Mittel zur Unterstützung eines durchgängigen Entwicklungsprozesses anerkannt ist. Insofern ist es naheliegend, auch die Integration von VR-Technologie in den Entwicklungsprozess als Teil der PLM-Strategie zu verstehen [Stelzer 2009].

Schilling und Kollegen verbinden im Projekt *Memphis* [Schilling 2006] die VR-Systeme mit dem betrieblichen Produktdatenmanagement (PDM). Alle Daten werden im Rahmen des Projekts auf zwei Servern organisiert, einem File-Server und einem Metadaten-



Server. Auf dem Metadaten-Server erfolgt eine weitere Aufteilung der Daten in drei Kategorien:

- CAD-Parameter
- VR-relevante Metadaten wie Shader, Events oder *Level of Detail* (LoD)
- PDM-Daten z. B. mit Bezug zu Personal, Organisation, *Bills of Material* (BOM).

Die PDM-Daten werden weiter einer sechsstufigen Hierarchie zugeordnet: Personal, Organisation, Projekt, Item, Dokument und Property.

Eberle hat in seiner Diplomarbeit ein VR-System in das PLM-System der Daimler AG eingebunden [Eberle 2007]. Die Integration erfolgt auf der Ebene des Dateimanagements. Über Nacht werden alle modifizierten CAD-Daten in VR-Formate umgewandelt, sodass am nächsten Arbeitstag alle aktuellen Änderungen, die auf dem CAD-System vorgenommen wurden, auch in VR betrachtet werden können.

Seit einigen Jahren wird das Prinzip der Service-Orientierten Architektur (SOA) verstärkt eingesetzt. Der Vorteil dabei: Obwohl jedes Unternehmen eigene Informationssysteme hat, kann mit SOA über Firmengrenzen und die unterschiedlichen Softwareplattformen hinweg ein Datenaustausch stattfinden [He 2003; Selvage 2005].

Eine typische Form der SOA ist der Webservice. Ein Webservice oder Webdienst ist eine Software-Anwendung, die mit einem *Uniform Resource Identifier* (URI) eindeutig identifizierbar ist und deren Schnittstelle als XML-Artefakt definiert, beschrieben und gefunden werden kann. Ein Webservice unterstützt die direkte Interaktion mit anderen Software-Agenten unter Verwendung XML-basierter Nachrichten durch den Austausch mit internetbasierten Protokollen [Vasudevan 2001; Wikipedia 2010]. Die technologische Basis für Webdienste ist als Web Service Description Language (WSDL) standardisiert.

WSDL ist eine plattform-, programmiersprachen- und protokollunabhängige Beschreibungssprache für Netzwerkdienste zum Austausch von Nachrichten auf Basis von XML [Christensen 2001; Wikipedia 2010]. Sowohl SOA, als auch Web-Service und WSDL werden in der Prototypimplementierung der vorliegenden Arbeit zum Zweck der VR-Integration eingesetzt.

### 3.4 Eingabe- und Trackinggeräte

Für die Eingabe in VR gibt es zahlreiche technisch ausgereifte bzw. etablierte Produkte (siehe Tabelle 3-5). Design und Funktionsweise der Geräte werden in vorherigen Arbeiten grundlegend vorgestellt, z. B. in der Dissertation von Häfner [Häfner 2002]. Deshalb werden in der vorliegenden Arbeit nicht alle Eingabegeräte einzeln vorgestellt.

Gerät	Institut/Firma	Bemerkung
DataGlove	Sun Microsystems	[Zimmerman 1987; Sturman 1989]
CyberGlove	Immersion/CyberGlove Systems	[CyberGlove 2009]
CyberTouch	Immersion/CyberGlove Systems	[CyberGlove 2009]
Pinch Glove	Mechdyne Corporate	[Mechdyne 2008]
6-DoF Maus	Ascension Technology Corporation	[Ascension 2009]
Wanda	Ascension Technology Corporation/Murray Consulting	[Wanda 2009]
Cubic Mouse	Fraunhofer IZB	[Fröhlich 2000]

**Tabelle 3-5: VR-Eingabegeräte**

Ein spezielles 3D-System ist die Anlage *Boom Chameleon*. Sie besteht aus einem Touch-Screen, einem Mikrofon, einem Rechner und einem Maschinenhalter. In der Arbeit von Tsang und Kollegen sind vielfältige Daten wie Text, Bilder, Sprachaufzeichnung und sogar Video integriert [Tsang 2002; Tsang 2003]. Allerdings werden die Funktionalitäten eher auf traditionelle Art und Weise durch die auf dem Touch-Screen montierte Hardware realisiert. Die Technik ist insofern als immersiv einzuordnen, als der Touch-Screen in Reichweite des Touch-Screen-Halters frei beweglich ist und somit der Viewpoint des Beobachters mit dem Monitor variiert werden kann. Sie ist andererseits aber nicht immersiv, da der Benutzer nur durch ein kleines Mono-Fenster die dreidimensionale Umgebung betrachtet.

Die Nintendo® Videospielkonsole *Wii™* kam Ende 2006 auf den Markt [Nintendo 2006; Wikipedia 2010]. Ihr wesentliches Merkmal ist ein neuartiger Controller, der über eingebaute Bewegungssensoren verfügt. Diese registrieren die Positionen und die Bewegungen des Controllers im Raum und setzen sie in entsprechende Bewegungen von Spielfiguren oder -elementen auf dem Bildschirm um. Das Prinzip ähnelt den optischen Tracking-Systemen in VR-Umgebungen [A.R.T. 2007]. Allerdings sind die Komponenten anders angebracht als in den meisten VR-Systemen. Die Infrarot-Kamera ist im Handgriff (Wii-Remote) eingebaut und eine LED-Leiste wird unterhalb oder oberhalb des Fernsehgeräts positioniert.

Lee verwendet den Spielhandgriff der Videospielkonsole, um daraus ein Tracking-System zu bauen (siehe Abbildung 3-12). Veröffentlicht ist seine Arbeit als Video in YouTube [Lee 2007]. Er setzt dabei ein Wii™-Remote als Tracking-Kamera und das andere als Handgriff ein. Die LEDs werden auf eine Brille umgebaut. Die gesamte Anlage ähnelt damit wieder den in VR-Umgebungen üblichen Tracking-Systemen.



**Abbildung 3-12: Wii™, MotionPlus und der Umbau zum Tracking-System**  
[Nintendo 2006; Lee 2007]

Herrmann und Kollegen von der Technischen Universität Dresden nutzen zwei Wii-Remotes für ein Tracking-System, das sowohl die Position als auch die Orientierung des Benutzers verfolgt [Herrmann 2009]. Diese Arbeit ist ebenfalls als Video in YouTube veröffentlicht. Die Autoren geben an, dass die Kamera in Wii™-Remote einen 41-Grad-Blickwinkel auf horizontaler und 31 Grad auf vertikaler Ebene erfasst. Da aber die LEDs nur bis auf ungefähr zwei Meter Entfernung gut erkennbar sind, ist der effektive Bereich des Wii™-Remote-Tracking stark eingeschränkt.

Nintendo® bietet mit *MotionPlus* seit Mitte 2009 eine Erweiterung von Wii™-Remote an. Diese Erweiterung entwickelte Nintendo in Kooperation mit der Firma AiLive [AiLive 2008]. In dem MotionPlus Aufsatz für den Spielehandgriff wird ein Bewegungssensor von InvenSense [InvenSense 2009] eingesetzt. Laut Produktinformation von AiLive bietet MotionPlus ein 1:1-Tracking in Position und Orientierung.

Im Vergleich zu den professionellen Tracking-Systemen kann sich der Wii™-Remote heute noch nicht wirklich behaupten. Vor allem ist das System der Videospielkonsole für Anlagen mit großem Bewegungsraum, wie sie im Bereich IPT benötigt werden, nicht geeignet. Zum Vergleich: Das Tracking-System von A.R.T. z. B. verfügt über eine Reichweite bis zu 12 Meter [A.R.T. 2007].

### 3.5 Fazit

In diesem Kapitel wurde für die grundlegenden Informationsformen der Stand der Forschung und Technik recherchiert und zusammengefasst:

- Text bzw. 3D-Text in eine immersive Umgebung zu integrieren ist technisch problemlos möglich. Jedoch werden textintensive UI-Elemente in der Praxis bisher mit Zurückhaltung verwendet.
- Audio kann ebenfalls ohne technische Schwierigkeiten in einem 3D-Raum wiedergegeben werden. Dies gilt allerdings nur dann, wenn die Audiowiedergabe die

raumakustische Situation der dargestellten 3D-Szene nicht berücksichtigen muss. Laut Schupp [Schupp 1994] ist die Echtzeitfähigkeit einer solchen Simulation ohne Vereinfachung vermutlich ein unlösbares Problem.

- Bilder werden in VR schon heute häufig verwendet. Sie dienen allerdings meistens zur Erhöhung des realen Eindrucks und nicht als Unterstützung eines Arbeitsprozesses.
- Bewegte Bilder anzuzeigen ist ebenfalls technisch kein Problem. Sie werden bisher meistens im Rahmen von 3D-Konferenzsoftware eingesetzt. Nur wenige dieser Systeme unterstützen Stereoskopie.
- Stereo-Bilder und –Videos sind technisch analog zu Bildern und bewegten Bildern umsetzbar. In diesem Zusammenhang gilt es jedoch noch zu klären, ob es Anwendungsfälle gibt, für die der Einsatz von „Stereo in Stereo“ sinnvoll ist.

Die Verbindungen zu Datenbanken und Webseiten werden bislang in einzelnen Projekten realisiert. Die Integration der VR wird dabei allerdings meist auf der Ebene von Datei-Management durchgeführt. Als neue Entwicklung in der Spielindustrie zeichnet sich ab, dass die VR-Technik weiter in den privaten Konsum vordringt.

Die Recherchen zeigen, dass die verschiedenen Repräsentationsformen meist schon in früheren Projekten einzeln untersucht bzw. realisiert wurden. Jedoch haben einige der Umsetzungen deutliche Einschränkungen bei den Randbedingungen. Außerdem muss festgestellt werden, dass die Aufgabe, eine immersive Benutzungsoberfläche für spezielle Anwendungen auf Basis einer VR-Plattform zu konzipieren, für einen Entwickler heute noch aufwendig ist. Nicht nur das Design einer immersiven Benutzerschnittstelle muss von Anfang an neu programmiert werden, sondern in der Regel müssen auch die Grundstrukturen dafür erst „erfunden“ werden. Die fehlende Softwareinfrastruktur führte dazu, dass integrierte VR-Benutzerschnittstellen mit vielfältigen Informationsformen bisher nicht weiter verbreitet werden konnten, obwohl dies technisch problemlos machbar ist und obwohl ein großer Bedarf an VR-Integration gegeben ist.

## 4 Analyse und Konzeption zu Richdata in VR

### 4.1 Ansatz

Um die in Abschnitt 3.5 zusammengefassten Defizite zu beseitigen, wird im Rahmen dieser Arbeit ein Richdata(RD)-Konzept entwickelt. Dieses RD-Konzept versteht sich als eine softwaretechnische Infrastruktur. Es ermöglicht, Informationen verschiedener Form in einer VR-Plattform zu integrieren. Somit können Anwendungen, welche auf dieser Plattform basieren, auf alle Formen von Daten zugreifen und sie kombiniert repräsentieren. Geometrie-Daten können neben ihrer förmlichen Nutzung auch semantische Bedeutung haben, die mit der Anwendungslogik verbunden und durch Interaktion geändert werden kann.

#### Datenbanken und Web

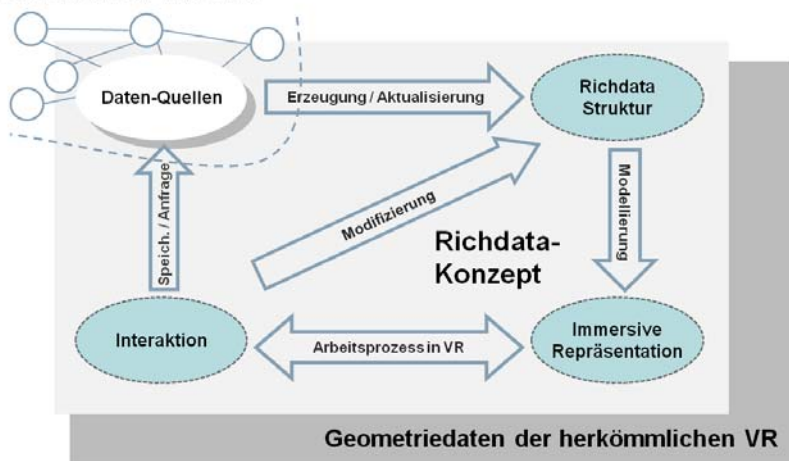


Abbildung 4-1: Komponenten des Richdata-Konzepts

Im Unterschied zu Projekten, bei denen VR-Lösungen entwickelt werden, die zusätzlich zur Darstellung von Geometriedaten einzelne weitere Datenformen verwenden, ist das RD-Konzept universell angelegt. Das Konzept ist nicht auf die Unterstützung einzelner oder einer Auswahl von Datenformen beschränkt, sondern offen für Erweiterungen um weitere Datenformen. Auch logische Zustände bzw. Arbeitsergebnisse einer Anwendung können als zusammengesetzte RD-Datensätze erfasst werden. Damit wird es auch möglich, die Arbeitsroutine in der VR auf Basis von Richdata zu formulieren, sodass sie in einen übergeordneten Prozess eingebettet werden kann.

Jeder Datensatz hat eine Ansicht, die während der Laufzeit den Zustand des Datensatzes widerspiegelt, aber auch unabhängig von Laufzeitbedingungen existiert. Somit stehen

die Zustände der Datensätze bzw. des VR-Programms transparent für eine mögliche Kommunikation mit der Außenwelt zur Verfügung. Die Zustände basieren auf den einzelnen Datenstrukturen und der jeweiligen Anwendungslogik. Damit unterscheidet sich das Konzept deutlich von bestehenden VR-Konzepten, die sich lediglich auf die Nachbildung des *Szenengraphs* fokussieren.

Die Vision des vorgestellten Konzeptes lässt sich mit einigen Szenarien veranschaulichen:

- Funktionalitäten wie 3D-Menü, Übergangsanimation oder Aufzählungslayout sollen mit dem Konzept leicht erzeugt und kombiniert werden können.
- Zudem sollen 2D-Ressourcen, wie Dokumente oder andere 2D-GUIs, durch Vernetzung mit anderen Datenwelten für das VR-System zugänglich sein. Benutzer müssen also nicht mehr zwischen Desktop und immersiven Umgebungen pendeln, um mit den unterschiedlichen Datenformen arbeiten zu können.
- Immersive Anwendungen können durch Web-Services mit betrieblichen Informationssystemen kommunizieren. Damit ist es auch möglich, die Richdata während der Laufzeit zu aktualisieren.
- Anfangs- und Endzustand der Anwendung werden als Richdata implementiert. Damit wird erreicht, dass die VR-Anwendung als integrierter Teil der betrieblichen Informationssysteme in die Arbeitsprozesse eingebunden ist.

Abbildung 4-1 illustriert, wie das RD-Konzept das traditionelle VR-System um drei Komponenten erweitert und wie diese durch vier Teilprozesse miteinander verbunden werden.

Der RD-Datensatz ist die grundlegende Einheit des Konzeptes. Ein solcher Datensatz besteht aus zwei Teilen: dem RD-Inhalt, in dem die Zustände des Datensatzes gespeichert werden, und einer bzw. mehreren Ansichten, die den Inhalt in immersiven Umgebungen repräsentieren. Der RD-Inhalt enthält den Bezug zu den Datenquellen, bündelt sie mit einer logischen Struktur und definiert die Zustände. Ein RD-Inhalt kann mehrere Ansichten haben, z. B. können 3D-Texte, eine Sprachaufzeichnung und eine Farbzuoordnung den gleichen RD-Inhalt repräsentieren. Die *Kontenterzeugung* ist der Prozess, bei dem der RD-Inhalt unter Anwendung einer festgelegten logischen Struktur aus den jeweiligen Datenquellen konstruiert wird.

Die RD-Datensätze können aus drei Quellen stammen. Als erste Quelle der Richdata kann der Geometrie-Datensatz Beschreibungen beinhalten, die mit der CAD-Software oder betrieblichen Informationssystemen erzeugt wurden. Solche Daten sind meistens in Form von Text angelegt. Sie ändern sich während der Laufzeit nicht und haben in der VR ein Referenzobjekt, nämlich den zu beschreibenden Geometrie-Datensatz. Zweitens können Richdata als eigenständige Datensätze aus externen Quellen, die auch zur Laufzeit aktualisiert und synchronisiert werden, erzeugt werden. Als dritte Möglichkeit werden Richdata gemäß konkreter Anwendungen generiert, um einen logischen Zustand bzw. die Arbeitsergebnisse darzustellen.

Nachdem die Richdata erzeugt wurden, werden gemäß dem RD-Inhalt immersive Repräsentationen erstellt oder zugeordnet. Dieser Prozess läuft in drei Stufen ab: In der ersten Stufe erfolgt die Repräsentation einzelner Datenformen, z. B. die Anzeige eines 3D-Textes oder das Abspielen einer Audiodatei. In der zweiten Stufe wird das Zusammenspiel mehrerer Repräsentationen mit der Logik des RD-Inhalts verbunden. Die dritte Stufe stellt Verbindungen zwischen der Benutzerinteraktion und der Modifikation des RD-Inhalts her. Während dieses dreistufigen Modellierungsprozesses muss für jeden Datensatz mindestens eine immersive Repräsentation erstellt werden, damit alle Daten auch tatsächlich durch die immersive Benutzungsoberfläche zugänglich sind.

Alle Einzelkomponenten des Konzeptes sind über die Interaktion des Benutzers miteinander verbunden. Je nach Anforderung der konkreten Anwendung werden Interaktionsmethoden entworfen und implementiert, mit denen der Benutzer während der Laufzeit mit der VR-Szene einschließlich der RD-Repräsentationen agieren kann. Dabei führen Änderungen in der Szene auch zu Zustandsänderungen der RD-Inhalte. Am Ende einer Arbeitsperiode wird es damit möglich, die Arbeitsergebnisse in Form von RD-Datensätzen mit einem externen Prozess rückzukoppeln. Die Arbeit in der VR-Umgebung wird dadurch nahtlos in übergeordnete Prozesse eingebunden.

Das RD-Konzept ist unabhängig von konkreten VR-Plattformen. Um die Realisierungsmöglichkeiten des Konzepts zu prüfen, wird es sowohl in einer konkreten VR-Plattform umgesetzt, als auch bei einer auf der Plattform basierenden Anwendung eingesetzt.

## 4.2 Analyse des Integrationsprozesses von VR-Anwendungen

### 4.2.1 Gedanken zum Nutzungsszenario

Das Ziel ist es, industrielle Prozesse zu verbessern, indem ein konventioneller Teilprozess durch VR-Technologie ersetzt wird. Da die VR eigene Anzeigesysteme, Eingabegeräte und spezielle Räumlichkeiten voraussetzt, bleibt sie oft ein eigenständiges Untersystem innerhalb eines Prozesses. Typisch hierfür ist z. B., dass ein Programm auf der VR-Plattform implementiert wird, das gleichzeitig der Anwendungslogik des übergeordneten Prozesses entspricht. Somit wird zwar das VR-System mit dem Prozess verbunden. Die Anfangsdaten und die Arbeitsergebnisse können jedoch meist nur durch manuellen Eingriff zwischen der VR-Anwendung und den betrieblichen Informationssystemen übertragen werden.

Um eine schlüssige Integration zu ermöglichen, ist es notwendig, dass die Backend-IT-Systeme das Verhalten des VR-Programms, als Eingabe, Ausgabe und Zustände verstehen können. Es folgt daraus als Anforderung, dass die Datenstrukturen des VR-Programms eigene Zustände formulieren, um sich mit dem externen Prozess verständigen zu können.

Der Szenengraph ist die zentrale Struktur der dreidimensionalen Repräsentationen. Ein VR-Programm muss deren Logik bzw. Ereignisse durch Änderung des baumartigen Sze-

nengraphen dem Benutzer wahrnehmbar machen. Ereignisse aus Benutzeraktionen werden in der Regel automatisch im Szenengraphen reflektiert, da das Verhalten des Benutzers vom Tracking-System in hoher Frequenz verfolgt wird. Danach wirken die Ereignisse erst auf die Anwendungslogik, deren Datenstruktur im Prinzip eine freie Topologie haben kann. Für Entwickler, die noch nicht mit VR-Systemen vertraut sind, ist der Weg von der Anwendungslogik zum Szenengraphen und zurück eine Herausforderung.

Eine weitere Herausforderung der VR-Integration liegt für die Entwickler darin, dass beim Design der Benutzerschnittstelle in der immersiven Umgebung andere Kriterien entscheidend sind, als bei traditionellen Desktopanwendungen. Bei letzterer ist mit WIMP ein in der Praxis dominantes Muster vorhanden. In der 3D-Umgebung sind die für 2D-Entwickler verfügbaren UI-Komponenten aber nicht einsetzbar. Bis heute gibt es auch keine softwaretechnischen Strukturen, die ähnliche Effekte bzw. Funktionalität wie in 2D-GUI erzeugen.

Nach der Festlegung der Datenstruktur, müssen auch entsprechende Schnittstellen konzipiert werden, um eine bidirektionale Kommunikation nach außen zu ermöglichen. Verbindungen sind dabei nicht nur lokal notwendig, sondern auch über Netzwerkanbindungen etwa zu Remote-Datenquellen, damit sich die Arbeitsanforderungen, die Informationen zur Prozesskontrolle und die Arbeitsergebnisse mit dem Backend zur Laufzeit synchronisieren lassen.

#### 4.2.2 VR-Plattform und Rahmenbedingungen

Design-, Entwicklungs- und Arbeitsprozesse unterscheiden sich von Unternehmen zu Unternehmen. Für die VR-Integration kann es daher in der Regel auch keine allgemeine Lösung geben. Projekte mit dem Ziel, VR-Plattformen zu integrieren, werden in den Unternehmen häufig von Entwicklern durchgeführt, deren fachliche Schwerpunkte nicht direkt im VR-Bereich liegen. In erster Linie dafür ursächlich sind nicht-technische Gründe wie die Verfügbarkeit von entsprechendem Personal im Unternehmen und der Datenschutz, der eine Inhouse-Bearbeitung der Projekte erforderlich macht. Die Verknüpfung von Unternehmens-IT und VR wird durch das RD-Konzept erleichtert, da es die benötigten Datenstrukturen und einen Leitfaden für Integrationsmaßnahmen zur Verfügung stellt. Die Analyse im letzten Unterkapitel zeigt die software-technisch notwendigen Rahmenbedingungen auf, die für eine VR-Integration nötig sind. Insbesondere sind folgende Punkte wichtig:

- Zugriff auf den Szenengraphen – Erstellen, Löschen und Umstrukturieren von Knoten und Lichtquellen
- Manipulieren des Szenengraphen – Transformation, Material, Shader
- Identifizieren einzelner Knoten im Szenengraphen durch externen, eindeutigen und laufzeitunabhängigen Schlüssel
- Eventübertragung aus Benutzer-Aktion zur Anwendungslogik
- Verbinden logischer Strukturen mit Szenengraph-Knoten



- Identifizieren logischer Strukturen durch Szenengraph-Knoten

Die Bedingungen sind unabhängig von der konkreten VR-Plattformen. Falls jedoch eine VR-Plattform eine oder mehrere der oben genannten Anforderungen nicht direkt erfüllen kann, muss dies im Rahmen der Integrationsmaßnahmen nachgeholt werden. Bei ungünstiger Softwarearchitektur kann dies mit großem Aufwand verbunden sein.

#### 4.2.3 Einfluss der Entwicklungsstrategie auf den Erfolg des Richdata-Konzepts

Wie im Kapitel zum Stand der Forschung und Technik zusammenfassend festgestellt wurde, gibt es schon eine Reihe von Projekten, die sich bemüht haben, einzelne oder mehrere Informationsformen neben den Modelldaten in die virtuelle Welt zu bringen. Soll allerdings ein immersives UI für eine Anwendung erstellt werden, stellt sich dennoch in der Regel erneut die Herausforderung, dass die Arbeit von einem niedrigen Niveau aus angegangen werden muss. Im Folgenden wird mit Blick auf die VR-Entwicklungsstrategie diskutiert, wie der Weg für die Entwicklung des RD-Konzepts oder nachfolgender Arbeiten aussehen könnte.

Die VR-Entwicklung erfolgt in den meisten Fällen auf Projektbasis. Im Vergleich zur konventionellen Softwareentwicklung hat eine Projektarbeit Vor- und Nachteile, die in Tabelle 4-1 kurz zusammengefasst werden. Ein gewichtiger Nachteil der Projektarbeit liegt darin, dass ein Projekt von Natur aus zeitlich begrenzt ist. Das erarbeitete Know-how geht daher zu einem großen Teil mit dem Ende des Projekts bzw. spätestens mit dem Weggang des ausführenden Mitarbeiters verloren.

Zudem werden die Implementierungen bzw. auch die Konzepte in der Regel nur für eine bestimmte VR-Plattform oder eine spezielle Hardwareausstattung entwickelt. Dies hat zur Folge, dass die Ergebnisse meist für nachkommende Projekte nicht weiter nützlich sind.

	Entwicklung von Standardsoftware	Entwicklung auf Projektbasis
Charakter	Ein robustes, skalierbares Basissystem mit Anwendungen, die für einen großen Kundenkreis akzeptable Lösungen bieten	Ein einfaches Basissystem mit speziell entwickelten Anwendungen
Beispiel	ICIDO GmbH, RTT AG	Ehemalige VRCom, IAO, VRCA und viele weitere Forschungsinstitute
Software	Die Kunden können Software kaufen und bedienen. Mit der Software können sie selbstständig Probleme	Die Kunden können ein Kern-System kaufen und darauf aufbauend eigene Anwendungen entwickeln. Die

	Entwicklung von Standardsoftware	Entwicklung auf Projektbasis
	lösen, wenn sie ausreichend Erfahrung haben.	Entwicklung setzt große Erfahrung im Basissystem voraus.
Services	Die Software wird eingeführt, um beim Kunden eine Lösung anzubieten. Die Entlohnung richtet sich nach dem Marktpreis.	Die Anwendung wird je nach Anforderung entwickelt. Die Entlohnung deckt die Personalkosten.
Vorteile	<ul style="list-style-type: none"> <li>• Erworbene Kenntnisse werden im Softwareprodukt gespeichert.</li> <li>• Es kann eine stabile Software erstellt werden, da dasselbe Programm kontinuierlich gewartet und von vielen Benutzern verwendet wird.</li> <li>• Je mehr Kunden die Software einsetzen, desto profitabler ist die Programmentwicklung. Die einzelne Softwarelizenz wird damit zumindest rechnerisch günstiger.</li> <li>• Die Software arbeitet in stabilen Arbeitsroutinen, die kontinuierlich verbessert werden können.</li> </ul>	<ul style="list-style-type: none"> <li>• Großer Einsatzbereich, da die VR-Software individuell für den jeweiligen Anwendungsfall entwickelt wird.</li> <li>• Geringe Kosten bei Vertrieb und Schulung, da die Software nach spezifischen Anforderungen entwickelt wird.</li> <li>• Übersichtliche Personalkosten, da alle Kosten innerhalb des Projekts gedeckt werden.</li> </ul>
Nachteile	<ul style="list-style-type: none"> <li>• Großes Risiko beim Produktmanagement, da neue Module mit hohen, langfristigen Kosten verbunden sind</li> <li>• Der Kundenkreis für eine spezielle Anwendung ist durch die Marktkapazität eingeschränkt.</li> <li>• Hohe laufende Kosten, da sowohl das Basissystem als auch die Anwendungen laufend gewartet werden müssen</li> </ul>	<ul style="list-style-type: none"> <li>• Erworbene Kenntnisse gehen am Ende des Projekts oder mit dem Weggehen des ausführenden Mitarbeiters zum Großteil verloren.</li> <li>• Die Software erreicht keinen hohen Reifegrad, da jedes Projekt nur bei einem oder wenigen Kunden eingesetzt wird.</li> <li>• Entwicklung und Wartung des Basissystems werden oft nicht ausreichend berücksichtigt</li> </ul>

	Entwicklung von Standardsoftware	Entwicklung auf Projektbasis
	<p>sen</p> <ul style="list-style-type: none"> <li>• Hohe Vertriebskosten</li> </ul>	<p>sichtigt. Notwendige Arbeiten am Basissystem werden erst erkannt und durchgeführt, wenn sie für das Funktionieren des konkreten Projekts unumgänglich sind.</p>

**Tabelle 4-1: Vergleich der Entwicklungsstrategien  
für VR-Standardsoftware und projektbasierte Softwareentwicklung**

Bei der Entwicklung von Standardsoftware wird ein Produkt auf einer stabilen Plattform in einem langfristigen Prozess entwickelt und gepflegt. Auch wenn der Lebenszyklus eines Softwareprodukts zu Ende ist, werden die Kenntnisse meistens für die Entwicklung der Nachfolgersoftware weiter verwendet. Allerdings berücksichtigen die Anbieter von Standardsoftware bislang die VR-Integration in ihrem Produktangebot kaum, da dies als nicht profitabel eingestuft wird. Denn ein Softwareprodukt zu entwickeln und zu pflegen ist mit hohen Kosten bzw. Risiken verbunden. Deshalb lohnt sich dieser Aufwand nur für Anwendungsfälle, die für einen großen potenziellen Kundenkreis von Interesse sind. Die bei der VR-Integration notwendigen Maßnahmen werden jedoch von spezifischen Prozessen beim Kunden bestimmt. Auch wenn die Industrie insgesamt einen großen Bedarf bei der VR-Integration zeigt, bringt daher jede einzelne Integrationsaktivität kaum Mehrwert für das Softwareprodukt als Ganzes. Soweit keine Synergie-Effekte zwischen den verschiedenen individuellen Integrationsprojekten zu erkennen sind, bleibt die VR-Integration für die Anbieter von Standardsoftware daher nur ein Tätigkeitsfeld am Rand ihres Geschäftsfeldes.

Die Entwicklung des RD-Konzepts legt besonders Wert darauf, eine nachhaltige Weiterentwicklung der Ergebnisse zu ermöglichen. Mit folgenden Maßnahmen verfolgt das RD-Konzept das Ziel, die Vorteile der Entwicklung von Standardsoftware mit denen der Softwareentwicklung auf Projektbasis miteinander zu kombinieren:

- Das Konzept wird unabhängig von einer bestimmten VR-Plattform entworfen.
- Eine zum Konzeptumfang gehörende Liste von Kriterien umfasst die Voraussetzungen und Bedingungen, die bei der Umsetzung auf der jeweiligen Ziel-Plattform eingehalten werden müssen bzw. für die, soweit sie nicht im Basissystem verfügbar sind, nachträglich eine Lösung erarbeitet werden muss.
- Das Konzept ist offen für die Erweiterung um weitere Informationsformen.
- Für Layout und Interaktion wird ein Designmuster vorgeschlagen. Die Bausteine, die für das Konstruieren eines immersiven UI erzeugt werden, können in einer Bibliothek hinterlegt und für nachfolgende Projekte wiederverwendet werden.

- Bei der Entwicklung von Mechanismen für die Kommunikation mit VR-externen Datenquellen werden nach dem Modell der service-orientierter Architektur allgemeine Schnittstellen für Datenbanken und einen Remote-Bildschirm für den Zugriff auf unstrukturierte Datenquellen spezifiziert. Dadurch ist gewährleistet, dass diese Schnittstellen mit kleinen Anpassungen auch plattformunabhängig eingesetzt werden können.

### 4.3 Bewertung der Repräsentationsformen hinsichtlich der VR-Integration

Um ein realistisches und wirksames Konzept entwickeln zu können, wird ein Überblick über die Wirkungsfaktoren und die technische Machbarkeit einzelner grundlegender Repräsentationstypen benötigt. In diesem Kapitel werden daher folgende Gesichtspunkte analysiert:

- wie notwendig bzw. wie wichtig die einzelnen Basistypen für die Integration der VR in die Entwicklungsprozesse sind,
- ob und inwieweit die heutige Technik reif ist für die Darstellungen der verschiedenen Datenformen unter Berücksichtigung der in immersiven Umgebungen gegebenen Randbedingungen.

Die in der Praxis erzeugten RD-Repräsentationen sind nicht auf die Basistypen beschränkt. Die RD-Datensätze können auch verschachtelte Inhalte enthalten, deren Repräsentationen durch Kombination der Basistypen mit logischen Zusammenhängen komplexe Szenen bilden.

Die Ausführungen in diesem Kapitel konzentrieren sich auf die beiden oben genannten Fragestellungen. Die technische Fragen, wie die Repräsentationsformen in der VR realisiert werden bzw. wie der Stand der Technik dabei ist, wurde bereits in Kapitel 3.2 *Grundlegende Richdata-Typen in der Literatur* recherchiert und dargelegt.

#### 4.3.1 Text

Text ist einerseits ein grundlegendes Element der Benutzerschnittstelle, z. B. für Menüs, Hilfsfelder oder allgemeine Beschreibungen, andererseits aber auch die dominante Informationsform zum Austausch und zur Kommunikation zwischen Informationssystemen. Da die Förderung der VR-Integration eine zentrale Zielsetzung der Arbeit ist, ist eine zuverlässige und flexible Repräsentation des Texts mit hoher Priorität gefordert.

Für eine ausgereifte Software-Lösung ist es notwendig, die Kontrollierbarkeit des Arbeitsprozesses in der Laufzeit zu erhöhen [Blau 2007]. Dazu ist nicht nur die Verknüpfung der VR-Anwendungen mit externen Systemen zur Eingabe und Ausgabe von Daten erforderlich. Zudem muss ebenso ermöglicht werden, dass die Anwendungszustände auch innerhalb der VR-Anwendung als Text repräsentierbar sind. Denn damit erhält der

Benutzer die Möglichkeit, den laufenden Arbeitsprozess in der immersiven Umgebung zu überwachen.

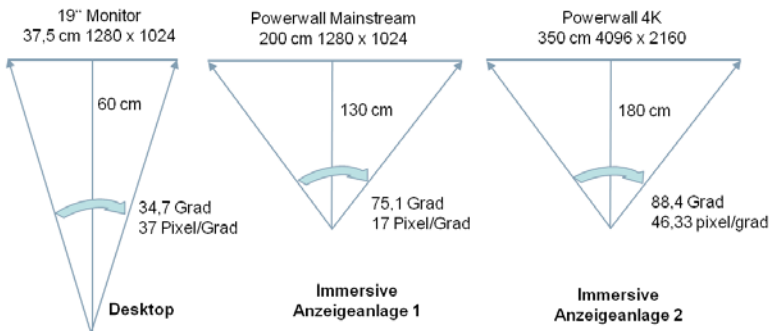
Auch als reines UI-Element spielt Text eine grundlegende Rolle. Viele VR-Plattformen bieten ein 3D-Menü-System, mit dem vordefinierte Ereignisse aufgerufen werden können. Die Menüs bestehen analog zu solchen in Desktop-Anwendungen meistens aus Text. Allerdings werden am Desktop teilweise auch längere Texte eingesetzt. In immersiven Umgebungen werden längere Menütex te ebenso wie textintensive UI-Elemente bisher aus folgenden Gründen vermieden:

- Schwierigkeiten bei der Eingabe von Symbolinformationen  
Ein traditionelles Eingabegerät für Symbole ist die Tastatur. Diese lässt sich in immersiver Umgebung jedoch nur schwer einsetzen. Spezielle Eingabegeräte werden zwar unter verschiedensten Aspekten erforscht, es gibt aber bis jetzt keine verbreitete Lösung. Ein Programm wie ein Text-Editor kommt daher für einen Einsatz in VR-Anwendungen derzeit nicht in Frage.
- Niedriger Reifegrad der VR-Anwendungen  
VR-Anwendungen haben meist noch Forschungscharakter. Daher konzentrieren sich die Entwickler meist auf die computergrafischen Effekte und die Funktionalitäten. Kriterien der Benutzbarkeit wie etwa ergänzende Erklärungen und Hintergrundinformationen bei der Arbeit in der VR werden nicht berücksichtigt.
- Eingeschränkte physikalische Auflösung der bisherigen Anzeigeanlagen  
Die am meisten genutzten Anzeigeanlagen für VR sind derzeit zum einen die Head Mounted Displays (HMD) und zum anderen die Großflächen-Projektoren. Die Auflösung der auf dem Markt etablierten HMD liegt bei SVGA (800 x 600 Pixel). Dies reicht jedoch nicht aus, um textintensive Elemente anzeigen zu können. Systeme mit einem Großflächen-Projektor haben zwar eine höhere Auflösung (üblich sind 1280 x 1024 Pixel), der Benutzer aber auch ein größeres Sichtfeld. Außerdem stehen die angezeigten Texte nicht immer senkrecht zum Benutzer, daher wird nicht die vollständige physikalische Auflösung wahrgenommen.

Die genannten Hindernisse können durch technische Fortschritte großteils überwunden werden. Zudem lässt sich feststellen, dass die VR-Technologie innerhalb des letzten Jahrzehnts von einem immer größer werdenden Kundenkreis nachgefragt wird. Als Folge davon bieten insbesondere die VR-Plattformen kommerzieller Anbieter einen zunehmenden Reifegrad.

Für die Eingabe von Symbolinformationen gibt es in der immersiven Umgebung noch keine überzeugende Methode. Aktuelle Forschungsbemühungen könnten allerdings in naher Zukunft zu praktikablen Lösungen führen. Vielversprechende Beispiele sind Fingertracking mit Berührungsfeedback [Streubel 2008] und etablierte Eingabemethoden für Touch-Screens.

Neu auf dem Markt sind 4K-Projektoren erhältlich und das Highend-HMD von Sensics™ erreicht eine horizontale Auflösung bis zu 2400 Pixel. In Abbildung 4-2 werden zwei typische Szenarien von Powerwall-Anlagen mit einem Desktop-Monitor verglichen. Es zeigt sich, dass die wahrnehmbare Auflösung des 4K-Projektors die des Desktop-Systems weit in den Schatten stellt. Dennoch gibt es immer noch ein Optimierungspotenzial. Denn die menschlichen Augen sind in der Lage, eine Auflösung bis zu ca. einer Bogenminute (1/60 Grad) zu erkennen [Wandell 1995; Bues 2009].



**Abbildung 4-2: Drei Konfigurationen von Anzeigeanlagen im Vergleich**

Kann ein Benutzer am Monitor einen Text nicht gut erkennen, geht er normalerweise näher an den Monitor heran, um den Text besser zu sehen. In der immersiven Umgebung gibt es unterschiedliche Möglichkeiten, um dasselbe Ziel zu erreichen:

- das Objekt zu sich heranziehen,
- sich virtuell näher an das Objekt heran zu navigieren,
- sich real dem Objekt zu nähern (z. B. näher an die Powerwall herantreten).

Zu bedenken gilt es dabei allerdings, dass die dritte Möglichkeit zu einem Umkehrereffekt führt, falls die physikalische Auflösung nicht ausreichend hoch ist. Wenn der Benutzer näher an die Projektionsfläche herantritt, vergrößert sich zwar sein Sichtwinkel in Bezug auf das Objekt. Gleichzeitig stehen aber weniger physikalische Pixel für die Darstellung des Objekts zur Verfügung.

#### 4.3.2 Audio

Die Wiedergabe von Audio inklusive Sprachwiedergabe ist auf Desktop-Computersystemen eine bereits ausgereifte Technik. Von der technischen Seite her entsteht kein Mehraufwand, wenn Audio im Sinne von Richdata (anstatt Auralisation) in immersiver Umgebung eingesetzt werden soll. Analog zum Desktop-System lassen sich Audiodateien mit einem Event verbinden oder als Hintergrund wiedergeben. Ein spezielles, eng mit Richdata verbundenes Gebiet ist die Sprachaufzeichnung, also die Aufnahme bzw. Wiedergabe der menschlichen Stimme.

Die Sprachaufzeichnung ist ein Unterbereich des gesamten Bereichs Audio. Als Element der Benutzerschnittstelle kann sie klar von sonstigem Audio, z. B. Musik, differenziert werden. Die Sprachaufzeichnung kann einen Befehl oder eine Beschreibung darstellen. In diesen Fällen übernehmen sie ähnliche Funktionen wie ein Text. Allerdings werden textintensive Benutzungsoberflächen in immersiven Bereichen bis jetzt kaum benutzt oder sogar bewusst vermieden. Im Unterschied zu Textelementen wurden die Möglichkeiten der Sprachaufzeichnung schon frühzeitig in der Entwicklung der VR erforscht [Verlinden 1993; Harmon 1996]. Insbesondere in immersiven Umgebungen hat die Sprachaufzeichnung den Vorteil, dass ihre Aufnahme und Wiedergabe zu jeder Zeit und in jedem Zustand der Anwendung einsetzbar ist. Der Benutzer-Fokus und beide Hände bleiben frei für weitere Interaktionen.

Bei der konventionellen 2D-Benutzerschnittstelle werden zwar auch Sprachbefehl bzw. -beschreibung realisiert. In dieser Anwendungsumgebung sind sie jedoch nicht so weit verbreitet wie das Mittel der Textinformation. Sprache als Kommunikationsform hat einen synchronen Charakter. Die Gesprächspartner müssen sich während des Gesprächs oder der Wiedergabe auf das Geschehen konzentrieren. Verpasst der Benutzer den Zeitpunkt der Wiedergabe einer Information, ist der Aufwand, der nötig ist, um die Information erneut zu erhalten, höher, als bei einer asynchronen Kommunikationsform, z. B. einer Textinformation.

Ein weiterer Nachteil der Sprachaufzeichnung als Informationsträger liegt darin, dass die sprachlichen Informationen schwierig zu bearbeiten sind. Außerdem dauert die Wiedergabe einer sprachlichen Information genau so lange wie die bei der Aufnahme verstrichene Zeit. In vielen Fällen benötigt ein Benutzer aber nur einen schnellen Einblick in eine Information. Das selektive Lesen eines Textstücks wäre hier deutlich effizienter, als die vollständige Wiedergabe einer Sprachaufzeichnung. Zudem ist die Such-Funktion für Informationssysteme von besonderer Bedeutung. Entsprechende Suchen durchzuführen und deren Ergebnisse anzuzeigen ist bei der Verwendung von Texten einfacher als bei Sprachaufzeichnungen.

Der Einsatz sonstiger Audio-Wiedergaben in der VR unterscheidet sich nicht von der Verwendung in konventionellen Multimediazusammenstellungen.

#### 4.3.3 Bilder

Ein Bild ist ein Datentyp, der schon heute verbreitet eingesetzt wird, um dreidimensionale bzw. immersive Anwendungen zu unterstützen. Als sogenannte Textur geben Bilder einfachen Modellen mehr Detail. Dazu werden ein oder mehrere Bilder auf die Geometrie-Oberflächen aufgezogen. Länge und Breite der Textur sollten immer als Zweierpotenz vorgegeben sein. Denn solche Formate lassen sich im OpenGL-Standard effizient rendern.

Ein Bild als eigenständiger Informationsträger, wie im Richdata-Konzept vorgesehen, kommt nur in wenigen Anwendungsfällen vor. Da dabei die Bilder selbst die darzustellenden Objekte sind, sollte gegebenenfalls der Träger (das Modell) bzw. dessen Textur-Koordinaten an die Bildvorgabe angepasst werden. Generell gibt es keine technischen Schwierigkeiten, Bilder im Rahmen von Richdata zu verarbeiten und anzuzeigen.

#### 4.3.4 Bewegte Bilder

Zu bewegten Bildern zählen Video, zweidimensionale Animation und Remote-Computerdesktop. Technisch kann die Wiedergabe bewegter Bilder in der VR als dynamische Textur realisiert werden. Die Darstellung ähnelt der eines Bildes. Allerdings muss die Textur permanent aktualisiert werden können. Aufgrund der stereoskopischen Anzeige in der immersiven Umgebung muss die Wiedergabe hochpräzise synchronisiert werden. Dies wird erschwert, wenn das Stereo-Anzeigesystem mehr als einen Kanal nutzt.

Für bewegte Bilder aus einer statischen Quelle können die Dateien vor der Laufzeit vom Master auf den Cluster-Rechner kopiert werden. Bewegte Bilder aus einer Streaming-Quelle müssen zur Laufzeit auf den Cluster-Rechner übertragen werden. Eine sequenzielle Übertragung ist nur bei geringer Datenmenge effizient einsetzbar. Daher empfiehlt sich der Einsatz von Multicast. Wenn ein Stream komprimiert ist, kann er vom Master-Rechner entweder komprimiert auf den Cluster-Rechner weitergegeben oder bereits vor der Übertragung dekomprimiert werden. Beide Verfahren haben Vor- und Nachteile. Beim ersten muss jeder Cluster-Rechner für sich denselben Code dekomprimieren. Im zweiten Fall muss zwar nur einmal dekodiert werden, aber dafür ist die zu übertragende Datenmenge erheblich größer. Angesichts der großen Datenmenge hätte hier ein inkrementelles Verfahren einen signifikanten Vorteil. Allerdings kann es auch bei Verwendung eines inkrementellen Verfahrens bei der Datenbereitstellung für die Wiedergabe zu einem Engpass kommen, und zwar dann, wenn sich die Bildinformationen von einem Frame zum anderen komplett ändern.

Der Remote-Computerdesktop ist eine interaktive Datenquelle. Im Vergleich zu den sequenziellen Quellen muss sie zwar einerseits zusätzlich den Benutzerbefehl an den Remote-Computer übertragen, andererseits hat sie in der Regel eine niedrigere Bildwiederholfrequenz. Häufig muss der Benutzer nur kurz eine Information recherchieren oder einzelne Befehle ausführen, die nicht direkt in der immersiven Umgebung erreichbar sind. In diesen Fällen bietet ein Remote-Computerdesktop eine effektive Lösung.

Eine weitere Anwendung bewegter Bilder ist die Video-Aufzeichnung von den Arbeitsabläufen innerhalb der virtuellen Umgebung. Das Video der VR-Arbeitsszene kann z. B. dazu dienen, den Arbeitsverlauf auch ohne VR-Anlage nachprüfen zu können.

Insgesamt ist festzustellen, dass es immer noch technisch schwierig ist, bewegte Bilder in VR-Umgebungen in voller Auflösung und mit hoher Frequenz (wie z. B. HD-Video) syn-



chronisiert wiederzugeben. Allerdings ist fraglich, ob eine solche Video-Wiedergabe in der VR überhaupt notwendig ist. Im Gegensatz dazu ist die Anzeige von einem Remote-Computerdesktop eine praktikable Methode, um die immersive Umgebung mit der konventionellen Benutzerschnittstelle zu vereinen.

#### 4.3.5 Stereo-Bilder und -Videos

Stereo-Bilder und Stereo-Videos sind Informationsformen, die in der VR leichter zu realisieren sind als in einer Desktop-Anwendung. Denn während der Stereo-Effekt für eine Desktop-Anwendung eine große Herausforderung darstellt, ist die räumliche Darstellung in der VR-Umgebung trivial. Eine technische Einschränkung besteht allerdings darin, dass die Stereo-Bilder und -Videos nur in einer bestimmten Position innerhalb der immersiven Szene optimal betrachtet werden können. Von Nachteil ist also, dass die optimale Benutzerposition auf einen Punkt oder zumindest einen engen Raum beschränkt ist. Vorteil des Einsatzes von Stereo-Bildern und -videos ist, dass sich so mit relativ geringen Ressourcen komplexe Szenen darstellen lassen.

Trotz des ausgereiften technischen Stands haben die beiden Informationsformen bisher kaum Bedeutung für VR-Anwendungen. Dies liegt vor allem daran, dass bisher nur wenige Inhalte bzw. Anwendungsfälle für die beiden Formen vorliegen. Ein Beispiel sind ein Fahrsimulatoren, bei dem der Fahrer während der Nutzung an einer fixen Position sitzt. Wenn ein Stereo-Video statt einer modellierten 3D-Szene eingesetzt werden soll, muss dafür auf freie Interaktion in der VR verzichtet werden.

#### 4.3.6 Vergleich und Fazit

Die Analysen in 4.3.3 bis 4.3.5 werden im Folgenden um eine synoptische Betrachtung des Autors ergänzt. Die Bewertung der unterschiedlichen Datenformen wird gemäß zwei Kriterien, dem Reifegrad der Technik und der Notwendigkeit für die VR-Integration, durchgeführt. Die beiden Kriterien entsprechen auch den Fragestellungen zu Beginn von Kapitel 4.3 auf Seite 60. Jedes Kriterium wird wie in Tabelle 4-2 dargestellt auf fünf Stufen skaliert.

Da die Sprachaufzeichnung in der VR ihren eigenen Anwendungsfall hat, wird sie separat von Audio bewertet. Ähnlich verhält es sich mit Videos und Remote-Desktops, die beide zu den bewegten Bildern gehören. Die Ergebnisse werden in Abbildung 4-3 illustriert.

Es lässt sich erkennen, dass Text und Remote-Desktop im Vergleich zu anderen Informationsformen beim Reifegrad ein deutliches Defizit haben, jedoch wichtig für die VR-Integration sind. Dieses Ergebnis wird in der Ausarbeitung, insbesondere bei der Implementierung, berücksichtigt.

Reifegrad	1 - 5	Notwendigkeit für VR-Integration	1 - 5
Verbreitet als Produkt	5	Zwingend notwendig	5
Einsetzbar, aber selten	4	Benötigtes Element	4
Prototyp realisiert	3	Optionales Element	3
Realisierbar mit Hindernissen	2	Optional für einzelne Anwendungsfälle	2
Unmöglich	1	Kaum gebraucht	1

Tabelle 4-2: Reifegradkala und Notwendigkeitsskala

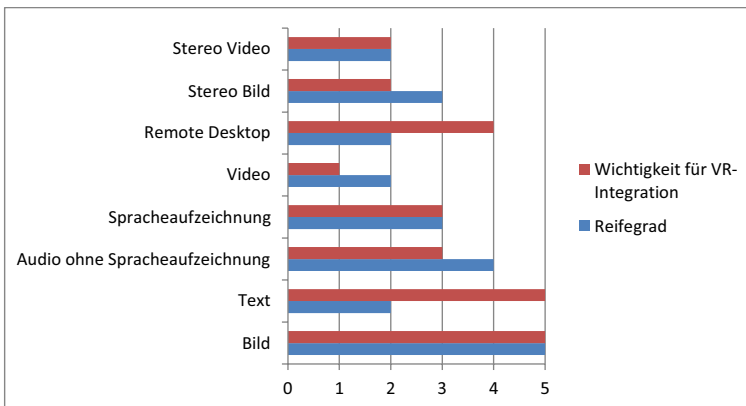


Abbildung 4-3: Vergleich der Informationsformen hinsichtlich Reifegrad und Notwendigkeit für VR-Integration

#### 4.4 Modularisierung des Richdata-Konzepts

Das RD-Konzept wird nach dem Prinzip der Modularisierung entworfen. Durch Abstraktionen werden einzelne Programmeinheiten als Modul gekapselt, sodass die Übersichtlichkeit, Erweiterbarkeit und Wartungsfreundlichkeit der Softwarelösung gesteigert werden können [Nagl 1998]. Die Modularisierung ist unabhängig von einer bestimmten Programmiersprache oder einem VR-System, jedoch wird beim Entwurf des RD-Konzepts davon ausgegangen, dass das Konzept mit einer objektorientierten Programmiersprache implementiert und ein modularisiertes VR-System als Basissystem verwendet wird.

Falls ein VR-System die Modularität nicht unterstützt, ist es erforderlich, eine Abstraktionsschicht zwischen dem Basissystem und der RD-Schicht einzuziehen. Falls die zugrun-

de liegende Programmiersprache nicht objektorientiert ist, muss im Einzelfall geklärt werden, ob eine nachträgliche Erweiterung möglich ist.

#### 4.4.1 Basis-Architekturmuster

Das Software-Architekturmuster mit den Komponenten *Modell*, *Präsentation* und *Steuerung* (MVC) ist im Bereich der Modellierung interaktiver Softwareanwendungen weit verbreitet [Reenskaug 1979]. Auch die Richdata im VR-System werden nach diesem Muster grob konzipiert (siehe Abbildung 4-4).

Das *Modell* enthält die darzustellenden Daten – sowohl Geometriedaten als auch die Richdata. Auf Basis des Modells werden zwei Gruppen von *Präsentationen* erzeugt. Zum einen dienen die Darstellungen in der immersiven Umgebung zur Wahrnehmung des Benutzers. Im Kapitel 4.6 werden sie ausführlich erklärt. Zum anderen bewerkstelligt die Präsentation den Austausch der Informationen mit der Programmlogik. Dazu zählt insbesondere auch die Logik des zu integrierenden Arbeitsprozesses. Ebenso betrifft die *Steuerung* einerseits die Benutzermethode für die VR-Interaktion und andererseits auch die Programmlogik.

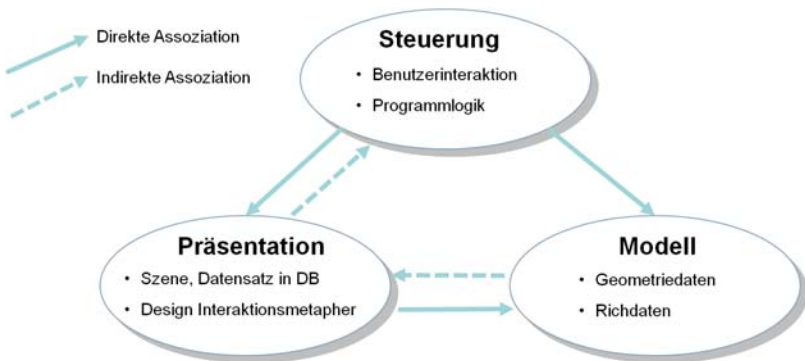


Abbildung 4-4: MVC-Architekturmuster

Die MVC gibt nur ein grobes Architekturmuster vor. Zur Konzipierung wird hinsichtlich Funktionen und Datenformen eine weitere Abstraktion über verschiedene Stufen definiert und in Abbildung 4-5 illustriert:

- Bei der Entwicklung einer VR-Anwendung gemäß dem RD-Konzept kommen die RD-Datentypen aus zwei Quellen: der VR-Plattform und den Anforderungen der Anwendung. In Kapitel 4.5 wird die Erzeugung der Richdata erklärt.
- Funktionalitäten werden nach Bedarf der Anwendung festgelegt und implementiert. Sie entsprechen der Anwendungslogik.

- Die Steuerung der VR-Anwendung kann man unter dem Aspekt der VR-Technik als Interaktion verstehen. In Bezug auf den Arbeitsprozess muss die Steuerung es ermöglichen, die vorgesehenen Aufgaben zu erledigen.

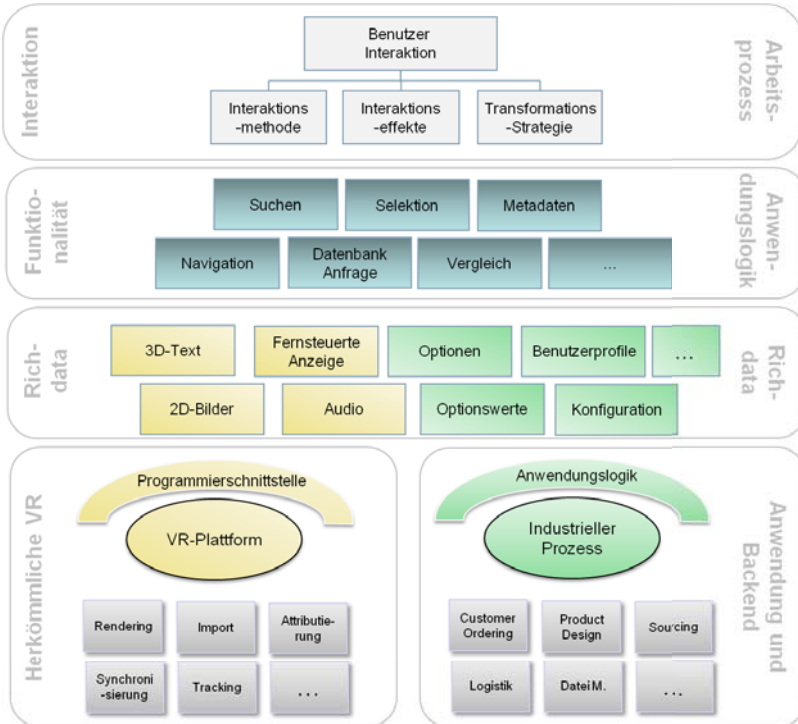


Abbildung 4-5: Modularisierung des Richdata-Konzepts

Dem MVC-Muster entsprechend gehören Abstraktionen der Richdata zum *Modell*; die Funktionalität, die Programmlogik sowie die Logik der Benutzerinteraktion zur *Steuerung*; Design und Modellierung der Interaktion gehören zur *Präsentation*.

#### 4.4.2 Struktur der Richdata-Datensätze

In diesem Kapitel werden die inneren Bestandteile des RD-Datensatzes sowie deren Beziehungen untereinander spezifiziert. RD-Datensätze können sowohl von der VR-Plattform als auch von der Anwendung definiert werden, sofern die Schnittstelle der Abstraktionsschicht mit folgenden Anforderungen implementiert wird:

1. Ein RD-Datensatz muss alle Informationen enthalten, die für die Instanziierung in der VR-Umgebung notwendig sind.

In herkömmlichen VR-Systemen sind Nicht-Geometrie-Datensätze oft zwar physikalisch vorhanden, können aber nicht direkt vom VR-Programm erreicht werden. Im RD-Konzept ist es ein grundlegendes Gebot, dass jeder Datensatz ohne Schwierigkeiten instanziiert werden kann. In objektorientierten Programmiersprachen lässt sich diese Anforderung leicht realisieren.

2. In jedem RD-Datensatz muss mindestens eine Repräsentationsform bestimmt sein, die mit dem Szenegraphen verbunden und vom Benutzer in der immersiven Umgebung wahrgenommen werden kann.

Wenn ein Datensatz nach der Anforderung 1 instanziiert wird, ist er im Arbeitsspeicher für das Programm verfügbar. Gemäß der zweiten Anforderung kann der Datensatz für Benutzer in der virtuellen Umgebung wahrnehmbar sein, sobald der Datensatz erzeugt wird. Der wahrnehmbare Bezug wird im RD-Konzept als **RD-Repräsentation** oder **Repräsentation** bezeichnet. Diese kann als Geometrie, z.B. 3D-Text, oder anderen Formen wie z. B. als Sprachaufzeichnung existieren. Im Gegensatz dazu werden die Instanzen des Datensatzes **RD-Inhalt** oder **Inhalt** genannt.

Eine RD-Repräsentation kann unterschiedlich komplex sein. In vielen Fällen stellt die Repräsentation nicht nur den Inhalt dar, sondern erzeugt eine Verbindung mit der Benutzerinteraktion, wodurch es möglich wird, den Inhalt in der virtuellen Umgebung zu ändern.

3. Ein RD-Datensatz muss Methoden zur Verfügung stellen, um auf die Zustände von außen zugreifen zu können und sie persistent zu verändern.

Diese Anforderung betrifft den RD-Inhalt. Es wird vorgesehen, die Methoden in zwei Fällen aufzurufen: Erstens kann eine Benutzerinteraktion dazu führen, den Zustand eines RD-Inhalts zu ändern. Funktionale Grundlage hierfür ist die in der 2. Anforderung beschriebene Verbindung zwischen RD-Repräsentation und einer Interaktion in der virtuellen Umgebung. Zum Zweiten kann der RD-Inhalt auch direkt manipuliert werden. Ein Beispiel hierfür ist die Synchronisation mit einer externen Datenquelle.

4. Der RD-Datensatz muss die Möglichkeit anbieten, sich selbst aus einem textbasierten Format zu laden.

Diese Muss-Klausel wird in erster Linie eingeführt, um die VR-Integration zu fördern. Diese Anforderung bedingt, dass ein RD-Datensatz komplett außerhalb des VR-Programms, z. B. von einem betrieblichen Informationssystem, erzeugt werden kann.

5. Der RD-Datensatz soll die Möglichkeit anbieten, sich selbst inklusive des eigenen Zustands, der sich während der Laufzeit durch Benutzerinteraktion ändern kann, in einem textbasierten Format zu speichern.

Die Textform eines RD-Datensatzes beschreibt den Zustand des RD-Inhalts bzw. der zugehörigen Repräsentationsformen, sodass die Datensätze per Netzwerk zwischen unterschiedlichen VR-Programmen bzw. anderen Informationssystemen ohne zusätzliche Aufwände übertragen werden können. Textbasierte Formen wie z. B. *Extensible Markup Language* (XML) sind zur Übertragung des Zustands eines Objekts bzw. eines Programms weit verbreitet. Insbesondere arbeiten viele betriebliche Informationssysteme, die über die Grenzen des einzelnen Unternehmens hinausgehen, mit Web-Services auf der Basis von Service-Orientierter Architektur (SOA). In diesem Bereich ist XML bereits als Standard etabliert.

Überwachungs- und Berichtsfunktionen, die über den aktuellen Zustand eines Arbeitsprozess informieren, sind wichtige Aspekte für reife Softwarelösungen. Sie werden in herkömmlichen VR-Anwendungen, insbesondere denen mit immersiven Benutzungsoberflächen, wenig unterstützt. Im RD-Konzept wird vorgesehen, Datenstrukturen des Arbeitsprozesses als RD-Datensatz zu implementieren (siehe Kapitel 4.5.2). Mit Hilfe dieser „Soll“-Klausel können die Überwachungs- und Berichtsfunktionalitäten leicht realisiert werden.

6. Der RD-Datensatz soll einen laufzeitunabhängigen Schlüssel besitzen, sodass er sowohl vom VR-System als auch von einem externen Prozess identifiziert werden kann.

Ein RD-Datensatz verknüpft die Repräsentation, die in der Regel ein Teil des baumartigen Szenengraphen ist, mit externen Datenstrukturen, die eine freie Topologie haben können. Um den Datenfluss oder die Interaktion einer VR-Anwendung festzulegen, spielt die persistente Identifizierung eines Objekts unter den beiden Topologien eine wichtige Rolle. Durch Einführung des laufzeitunabhängigen Schlüssels wird die Identität eines Datensatzes eindeutig bestimmt.

7. Der RD-Datensatz kann weitere RD-Datensätze beinhalten. Alle oben genannten Eigenschaften müssen bei der Verschachtelung gültig bleiben. Eine Verschachtelung darf zu keinem Zyklus führen.

Die Verschachtelung ist ein wichtiger Mechanismus im RD-Konzept, um auf Basis von existierenden RD-Datensätzen komplexere RD-Datensätze zu erstellen. Details über die Verschachtelung werden anhand eines Beispiels in Kapitel 4.4.3 vorgestellt.

Ein RD-Datensatz ist trivial, wenn dessen Inhalt und Repräsentationen nach der Erstellung nicht mehr geändert werden müssen. Ein Beispiel dafür ist eine 3D-Textbeschriftung. Für solche Datensätze gilt, dass sie lediglich die ersten vier Anforderungen, die mit „muss“ gekennzeichnet sind, erfüllen müssen. Informationen über Zustand, Ressourcen und Repräsentationen werden in einer Programmiereinheit gekapselt.

Sobald einem RD-Datensatz eine Semantik zugeordnet ist oder er relevant für den externen Prozess ist, müssen auch die fünfte und sechste Klauseln realisiert werden. Im Kontext der Prozessintegration können komplexe Datensätze aus dem externen Prozess in die VR-Systeme in Form von Richdata eingehen und durch immersive Interaktion modifiziert werden.

Angesichts der steigenden Komplexität wird für nicht-triviale Richdata eine weitere Struktur innerhalb des Datensatzes nötig. Erstens werden die RD-Repräsentationen getrennt vom persistenten RD-Inhalt behandelt. Dies ist insofern erforderlich, da der gleiche Inhalt in verschiedenen Formen präsentiert werden kann. Z. B. lässt sich die Beschreibung eines Hotels mit 3D-Text, Sprachaufzeichnung oder Bildern darstellen. Zudem kann die Repräsentation eines Dateninhalts eine höhere Komplexität aufweisen, da sie vom Layout- und Interaktionsdesign bzw. der Benutzbarkeit beeinflusst werden kann. Außerdem kann das Verhalten des Benutzers Einfluss auf den Zustand einer Repräsentation haben. Die Zustände von Repräsentationen haben meist nur Bedeutung während der Laufzeit. Zustände des Inhalts müssen dagegen persistent bleiben.

Zwischen Datensatzinhalt und dessen Repräsentationen gibt es zwar Abhängigkeiten, die allerdings nicht direkt in der Struktur des Dateninhalts zu erkennen sein müssen. Denn wie die Abbildung 4-4: MVC-Architekturmuster " zeigt, kennen die Repräsentationen den Dateninhalt durch direkte Assoziierung. Das bedeutet: Wird der Zustand des Dateninhalts geändert, muss die Repräsentation dies wissen und sich entsprechend ändern. Der Rückweg ist implizit: Die Änderung der Repräsentation wird vom Benutzer initiiert. Die erzeugten Ereignisse werden bewertet und eventuell gemäß der Programmlogik in eine Änderung des Dateninhalts übersetzt. Da die Steuerung die direkte Assoziation zum Dateninhalt hat, können die Methoden für Zustandsänderung explizit aufgerufen werden.

Die Logik, die die Zustandsänderung regelt, kann technisch gemeinsam mit dem Dateninhalt oder separat abgelegt werden. Welche Möglichkeit bevorzugt werden sollte, kommt darauf an, wie die Entscheidung für die funktionale Abstraktion getroffen wird. Wenn die Zustandsänderung nur lokale Wirkung hat, bietet sich an, die Logik mit dem Inhalt zusammenzufassen. Im Fall, dass die Zustandsänderung Auswirkungen auf andere Richdata hat oder verschiedene adaptierbare Logen infrage kommen, ist es sinnvoll, die Logik auszulagern. Für beide Varianten sind Verhaltensmuster wie State- oder Strategie-Pattern gut geeignet [Gamma 1995].

Folgendes Beispiel zeigt den nicht-trivialen RD-Datensatz „ExampleValue“ in Pseudocode. Zu den involvierten Datenressourcen gehören Geometrie, Material und Sound. Die Realisierung der oben genannten sieben Anforderungen wird im Kommentar des Pseudocodes gekennzeichnet. Die Zustände können in zwei Kategorien geteilt werden: dem laufzeitbezogenen Zustand *Sichtbarkeit* und dem laufzeitunabhängigen *Gewählt\_oder\_nicht*. Da die Programmlogik des Beispieldatensatzes einfach ist, wird sie nicht ausgelagert.

## Interface ExampleValue

```
// Instanziiieren & Serialisieren
ExampleValue (XML_INPUT) // Instanziierung aus text-basiertem Format (1, 4)
ToXML(XML_OUTPUT) // Save to XML (5)
// Resource
String URL_alle_Geometrie_Resource // (1, 2)
String URL_alle_Material_Resource // (1, 2)
String URL_Sound_or_other_Resource // (1, 2)
String Init_Information // (2)
// Member
StaticKey Laufzeit_unabhängige_Schlüssel // Es wird von extern gegeben (6)
RefOption Zugehörige_Option // Zur zugehörigen Option
Bool Gewählt_oder_nicht // Laufzeitunabhängiger Zustand
Bool Sichtbarkeit // Laufzeitabhängiger Zustand
RefSceneObjects All_Verknüpfungsknoten // Die Präsentation des Values
// kann auf mehrere Orte des
// Szenengraphs verteilt sein (2)

// Zugriff auf Datenzustand
Bool is_diese_value_Gewählt() // Zugriff des Zustands (3)
Bool setValueTrue(Parameters) // Setzen des Zustands (3)
Bool setVisible(Parameters) // Setzen der Sichtbarkeit (3)
// Präsentation erstellen
makePresentation (Parents) // Erstellung der Präsentation, mehr als
// ein Parent, da die Präsentation auf
// verschiedene Orte verteilt (2)
```

## End RDOptionValue

### 4.4.3 Verschachtelung

Richdata kann verschachtelt werden. Damit wird es möglich, mit wenigen Basiskomponenten auch komplexe Strukturen abzubilden. Die Struktur der Verschachtelung des RD-Inhalts und die der RD-Repräsentation wird mit unterschiedlichen Verfahren aufgebaut.

Die Abhängigkeiten der RD-Inhalte werden im Prinzip nach der Datenstruktur der Anwendungslogik gebildet. Dabei wird diese von den Anforderungen des Benutzers und dem zu integrierenden Prozess bestimmt. So besitzt z. B. die *RDOption* in Abbildung 4-6 einen Vektor von *RDOptionValue*. Die Benutzeraktion „Select“ wird aufgerufen, indem die Sichtbarkeit der Optionswerte bei der Repräsentation ein- und ausgeschaltet wird. Eine entsprechende Verschachtelung wird in Abbildung 4-6 mit der Methode *ChangeValueVisibility* in der Klasse *RDOptionRepresentation* angedeutet.



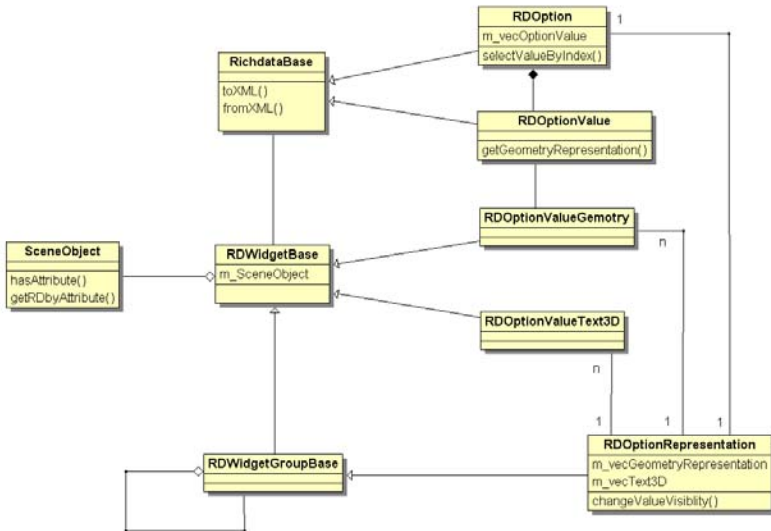


Abbildung 4-6: Klassendiagramm der Verschachtelung von Richdata-Datensätzen

Die Repräsentationen sind im Prinzip Teile des Szenengraphen und nicht-sichtbare Kommunikationsformen, z. B. Sound. Im Klassendiagramm (Abbildung 4-6) wird deutlich, dass die Basisklasse *RDWidgetBase*, aus der alle Repräsentationstypen abgeleitet werden, ein *SceneObject* als Membervariable besitzt. So hat jede RD-Repräsentation den Zugriff auf die Geometrie im Szenengraph. Im Fall der Verschachtelung wird der übergeordnete Repräsentationstyp aus der *RDWidgetGroupBase* abgeleitet, die mit der Basisklasse *RDWidgetBase* zusammenhängt. Die Repräsentation in einer Gruppe kann im Szenengraphen beliebig verteilt sein. Eventuell kann sich die Topologie der baumartigen Struktur zur Laufzeit auch ändern.

Während der Laufzeit werden die RD-Datensätze typischerweise folgendermaßen vom Benutzer erreicht: Der Benutzer selektiert ein Geometrie-Objekt mit dem Eingabe-Gerät.

Das *SceneObject* sucht nach eigenen Attributen. Falls die Attribute einen Schlüssel zum RD-Datensatz haben (siehe Anforderung 6 auf Seite 70), wird der RD-Datensatz erreicht und wird eine Aktion des RD-Datensatzes durchgeführt. Ein Beispiel zeigt *selectValueByIndex* in Abbildung 4-6. Nachdem die Zustandsänderung des Dateninhalts erfolgt ist, wird die entsprechende Änderung in der Repräsentation durchgeführt. So sieht der Benutzer auch das Feedback seiner Aktion.

## 4.5 Erzeugung der Richdata

Beim Erzeugungsprozess wird die Datenstruktur des zu integrierenden Prozesses in eine zum VR-System kompatible Form konvertiert und mit den Präsentationen im Szenen-graph verbunden. Wie diese Konvertierung im Detail abläuft, hängt von der jeweiligen Quelle der Richdata ab.

### 4.5.1 Automatisierte Erzeugung

Metadaten, die innerhalb eines Geometrie-Datensatzes eingebettet wurden, liegen meistens als Zahlen oder Text und in wenigen Fällen auch in Form von Bildern vor. Die Zugriffsschnittstellen zu den Daten werden durch die Spezifikation des Geometrie-Formats definiert. Der Dateninhalt ist überwiegend durch eine einfache Kombination mit 3D-Text darstellbar. Daher ist es vielfach möglich, die Metadaten des Geometrie-Datensatzes in einem automatisierten Prozess in RD-Datensätze umzuwandeln. Nach der Erstellung verändern sich die Metadaten während der Laufzeit nicht mehr. Außerdem haben sie meistens ein eindeutiges Bezugsobjekt in der Szene und damit ist auch die Repräsentation der Metadaten einfach umzusetzen.

Einige Austausch-Formate wie JT [Siemens 2008] und FBX [Besner 2005] verfügen über eine Container-Struktur, welche neben den Geometrie-Daten auch Informationen freier Formate speichern kann. Die Struktur innerhalb des Containers lässt sich nach eigenen Anforderungen zusammenstellen. Damit solche Informationen automatisiert in RD-Datensätzen konvertiert werden können, muss die VR-Anwendung die vorgegebene Struktur kennen.

Metadaten, die mit den Geometrie-Daten erstellt und danach nicht mehr geändert werden können, lassen sich auch mit dem Ladevorgang des CAD-Datensatzes laden. Weitere manuelle Eingriffe sind nicht notwendig. Die Automatisierung des Erzeugungs- und Ladevorgangs der Datensätze ist allerdings insofern eingeschränkt, dass dies nur in Verbindung mit einfacher Anwendungslogik bzw. einfachen Benutzerinteraktionen möglich ist.

Datenströme bzw. Datenformen wie Audio, Video oder interaktive Datenquellen sind dadurch gekennzeichnet, dass sie während der Laufzeit kontinuierlich Updates liefern. Eine Automatisierung ist auch hier sinnvoll, wenn viele solcher Datenquellen eingebunden werden sollen. Allerdings können die RD-Datensätze in diesem Fall nicht vollständig „erzeugt“ werden, da sich die Inhalte laufend ändern. Stattdessen werden RD-Datensätze erstellt, die alle Rahmenbedingungen, etwa die URL des Streamings, festlegen.

### 4.5.2 Erzeugung aus Anforderungen der Anwendung

RD-Datensätze können je nach Anforderung des VR-Integrationsprozesses konstruiert werden. Die Zustände des zu integrierenden Arbeitsprozesses werden dabei durch die

Zustände des RD-Inhalts dargestellt. Folgendes Diagramm (Abbildung 4-7) zeigt den Ablauf des Erzeugungsprozesses.

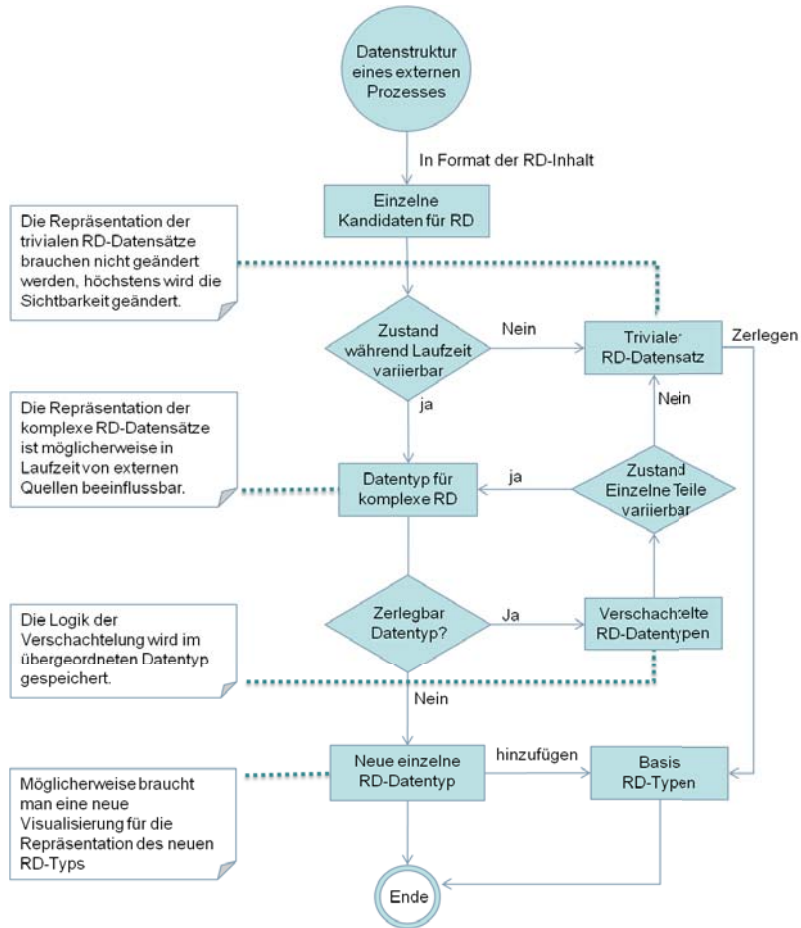


Abbildung 4-7: Ablauf des Erstellungsprozesses

Als Erstes muss ein RD-Datensatz die allgemeinen Voraussetzungen erfüllen, die in Absatz 4.4.2 in den Punkten eins bis vier beschrieben werden. Danach muss geprüft werden, ob der Datensatz während der Laufzeit variiert werden kann. Am Ende werden zwar alle Datensätze in Basis-Datentypen zerlegt. Die Repräsentationen können jedoch an unterschiedlichen Orten des Szenengraphen aufgehängt werden. In vielen Fällen werden die Daten bzw. auch die Geometrie nicht so positioniert, wie es dem Abbild der

realen Welt entspräche. Die Positionierung kann sich vielmehr auch während der Laufzeit ändern, um dem Interaktionsbedarf in der VR zu entsprechen.

Im nächsten Schritt wird geprüft, ob der Datentyp in Basis-Datentypen bzw. bekannte Datentypen, welche von VR-Plattformen angeboten werden, zerlegt werden kann. Diese Vorgehensweise verbessert die Wiederverwendbarkeit und Wartungsfreundlichkeit der Codes. Bei dieser Zerlegung kann es vorkommen, dass ein neuer Basis-Datentyp benötigt wird oder dass die Bestandteile eines Datentyps komplex zusammenhängen, z. B. mit einer Formel, deren Repräsentation nur als Ganzheit Sinn macht.

## 4.6 Richdata-Repräsentation

RD-Repräsentationen sind die Teile der Richdata, die in der Benutzungsoberfläche wahrnehmbar sind. Die Frage, wie eine Repräsentation aussehen bzw. implementiert werden soll, hängt nicht nur von technischen Spezifikationen ab, sondern ist vor allem eine Designfrage. Daher kann das RD-Konzept keine Standardlösung für das Design der RD-Repräsentation vorgeben.

Üblicherweise dauert die Erstellung einer immersiven Benutzungsoberfläche mehrere Monate. Im Unterschied dazu lässt sich das Grobdesign einer 2D-grafischen Benutzungsschnittstelle (GUI) innerhalb einiger Stunden erarbeiten. Eine der zentralen Ursachen dafür ist, dass es im Bereich 2D-GUI etablierte Designmuster gibt und zahlreiche Software-Tools für GUI-Design zur Verfügung stehen. Für immersive Benutzungsoberflächen dagegen gibt es bisher kein etabliertes Designmuster. In diesem Kapitel wird das etablierte WIMP-Designmuster für 2D-Benutzungsoberflächen analysiert und anhand der Ergebnisse ein für immersive Benutzungsoberflächen geeignetes Designmuster vorgeschlagen.

Dieser Vorschlag soll allerdings nicht als eine vorgegebene Richtlinie verstanden werden. Durch das RD-Konzept wird die Freiheit des Anwendungsentwicklers beim Design möglicher VR-Benutzungsoberflächen nicht eingeschränkt. Das vorgeschlagene Designmuster bietet aber eine Referenz, die ein Entwickler als Ausgangspunkt nutzen kann, um passend zu den jeweiligen Anwendungsspezifikationen eine geeignete immersive Benutzungsoberfläche zu kreieren.

### 4.6.1 Analyse der WIMP

Die Analyse in Abschnitt 4.3 zeigt, dass einzelne Repräsentationstypen wie Text, Audio oder Bilder schon heute in der 3D-Welt eingesetzt werden können. Dennoch verdeutlicht ein Vergleich zwischen ausgewählten immersiven Benutzungsoberflächen und der 2D-GUI von Desktop-Anwendungen, die bestehenden großen Unterschiede im Reifegrad. Benutzerstudien zeigen zudem, dass heutige 3D-Schnittstellen, obwohl eine zusätzliche Dimension genutzt werden kann, beim Thema Arbeitseffizienz der Leistung zweidimensionaler Benutzungsoberflächen deutlich hinterherhinken [Hauber 2006; Polys 2006a; Polys 2006b].

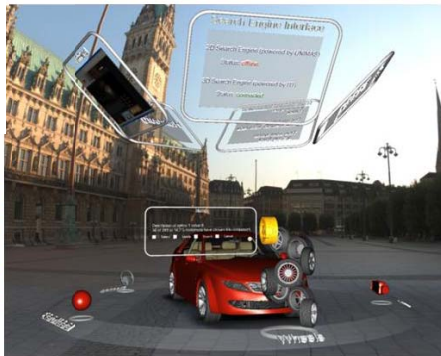


**Oben:** Ein realer Benutzereingang (Obi)

**Rechts:** Immersive UI mit Richdata-Konzept

**Unten:** Eine 3D-Benutzerschnittstelle (2nd Life)

**Rechts:**  
Die 2D-GUI von Windows 7



**Abbildung 4-8: Vergleich typischer Benutzeroberflächen**

Abbildung 4-8 stellt einige typische Benutzerschnittstellen zum Vergleich zusammen. Bei der ausgereiften 2D-GUI lässt sich erkennen, dass sie dem Benutzer nicht nur vielfältige Informationen bietet, sondern auch eine übersichtliche, ergonomische und ansprechende Arbeitsumgebung. Dies sind Voraussetzungen, um damit effizient arbeiten zu können.

Da es im immersiven Bereich noch zu wenig ausgereifte Anwendungen gibt, wird in der vorliegenden Arbeit versucht herauszufinden, welche Funktionalitäten bei einem ausgereiften 2D-GUI zur guten Benutzbarkeit beitragen. Aus diesen Funktionalitäten werden dann Prinzipien abstrahiert, die dazu dienen, eine gute Benutzbarkeit auch in einer 3D- bzw. einer immersiven Benutzeroberfläche zu realisieren. Am Ende der Arbeit (siehe Kapitel 1) wird in einer Benutzer-Studie erforscht, ob diese Methode bzw. das im nächsten Unterkapitel aufgestellte Konzept eine positive Wirkung auf die Benutzbarkeit der immersiven Benutzeroberfläche hat.

Das Entwurfsmuster mit *Windows*, *Icons*, *Menus* und *Pointer Devices* (WIMP) wurde im Jahr 1982 von Smith und Kollegen entwickelt [Smith 1989]. Es ist heute das gängige Designmuster für 2D-GUI [Dam 1997]. Die Benutzbarkeit und Benutzerfreundlichkeit wurde in den letzten 20 Jahren zwar ständig weiterentwickelt und verbessert. Die Grundstruktur des 2D-GUI hat sich aber nicht geändert. Nachfolgende Tabelle fasst die Analyse des WIMP-Konzepts zusammen und benennt einige Anregungen für das Design immersiver Benutzeroberflächen:

2D-GUI-Komponenten	Funktionalitäten und Prinzipien	Anregungen für immersive UI
Windows	<ul style="list-style-type: none"> <li>• Das Fenster (Window) ist ein virtueller Bereich mit einem markierten Rahmen. Alle Ressourcen zum Erledigen einer bestimmten Aufgabe sind innerhalb dieses Rahmens zu finden.</li> <li>• Ein Fenster kann von einer Position zu einer anderen bewegt werden. Alle Ressourcen kommen mit und bleiben in den gleichen Verhältnissen zu einander.</li> <li>• Es kann zu einem Symbol oder einer Leiste minimiert werden oder das ganze mögliche Sichtfeld (<i>field of view</i>) voll füllen.</li> <li>• Das Betriebssystem bietet die Möglichkeit, einen Überblick über alle offenen Fenster zu haben und unter ihnen zu wechseln.</li> </ul>	<ul style="list-style-type: none"> <li>• Logisch zusammenhängende Objekte sollen in einem dreidimensionalen Raum (Scope) mit deutlicher Grenze positioniert werden. Die Grenze kann explizit, z. B. als geteilter 3D-Raum, oder implizit, z. B. durch große Distanz oder unterschiedliche Beleuchtung, gesetzt werden.</li> <li>• Ein Scope soll alle benötigten Ressourcen für die vorgesehene Aufgabe zur Verfügung stellen.</li> <li>• Ein Scope soll minimiert oder unsichtbar gemacht werden können. Auch wenn sich der Benutzer selbst in diesen Scope befindet, soll er sich so jederzeit einen Überblick über die gesamte Anwendung verschaffen können.</li> <li>• In immersiver Umgebung sind gegenüber einem Desktop-Fenster zusätzliche Möglichkeiten denkbar, einen Scope zu aktivieren: z. B. durch Navigation oder Augenfokus.</li> <li>• Ein Scope kann als Ganzes bewegt werden. Allerdings muss beachtet werden, dass die Position und die Orientierung in der immersiven Welt zusätzlich eine semantische Bedeutung hat.</li> </ul>
Icon	<ul style="list-style-type: none"> <li>• Icons sind Bildzeichen in einer definierten Größe. Die Selektion eines Icons führt in der Regel zu ei-</li> </ul>	<ul style="list-style-type: none"> <li>• In immersiver Umgebung können Bildzeichen durch Geometrie ersetzt werden. Zur Unterscheidung gegenüber einem einfachen Icon</li> </ul>

2D-GUI-Komponenten	Funktionalitäten und Prinzipien	Anregungen für immersive UI
	<p>ner vordefinierten Aktion.</p> <ul style="list-style-type: none"> <li>• Icons haben meist eine spezielle Formgebung, damit der Benutzer sie deutlich von anderen Bildzeichen unterscheiden kann. Beschriftungen erleichtern zusätzlich die Erkennbarkeit.</li> <li>• Für Icons innerhalb eines Programms wird ein identisches Styling verwendet.</li> <li>• Wenn der Benutzerfokus auf ein Icon gerichtet ist, wird in der Regel eine ergänzende Text-Beschreibung als Floating-Menü angezeigt.</li> </ul>	<p>wird die Bezeichnung 3D-Icon gewählt.</p> <ul style="list-style-type: none"> <li>• Die 3D-Icons sollten eine uniforme Größe und optionale Beschriftungen haben. Sinnvoll ist zudem, weitere Kennzeichen zu kreieren, um eine klare Trennung von anderen Objekten in der Szene zu erreichen.</li> <li>• In immersiver Umgebung kann der Benutzer auf verschiedene Arten einen Fokus setzen. Jede Variante lässt sich mit unterschiedlichen Ereignissen verknüpfen. Folgende Beispiele sollen dies verdeutlichen:</li> <li>• Kopf-Orientierung oder Augenfokus führt dazu, dass eine Beschreibung angezeigt wird. Finger-Tracking dagegen führt zu Änderung von Position und Orientierung des 3D-Icons.</li> <li>• Mit einem 6DOF-Pointing-Gerät wird ein 3D-Icon selektiert und weitere Aktionen werden durchgeführt.</li> </ul>
Menü	<ul style="list-style-type: none"> <li>• Ein Menü besteht aus einer Liste von Befehlen. In der Regel werden diese in Form von Text präsentiert.</li> <li>• Ein Menü-System ist vielfach gemäß einer baumartigen Struktur aufgebaut.</li> <li>• Ein Menü kann sowohl</li> </ul>	<ul style="list-style-type: none"> <li>• In einer immersiven Umgebung gibt es keine Einschränkung des Sichtfeldes. Innerhalb eines Scopes kann einem Menü dennoch eine feste Position zugeordnet werden.</li> <li>• Ein Menü lässt sich ebenso mit dem Cursor des Eingabegerätes koppeln.</li> <li>• Floating- oder Kontext-Menüs können mit jedem Fokus (Augen-,</li> </ul>

2D-GUI-Komponenten	Funktionalitäten und Prinzipien	Anregungen für immersive UI
	mit einer festen Position im Fenster als auch mit dem Pointer-Fokus (Kontext-Menü, Floating-Menü usw.) verknüpft sein.	Finger- oder Input-Fokus) verknüpft werden.
Pointer-Device	<ul style="list-style-type: none"> <li>Die Maus ist das Standard-Pointing-Device in WIMP. Sie stellt den Fokus auf einer zweidimensionalen Oberfläche ein.</li> </ul>	<ul style="list-style-type: none"> <li>Pointer-Devices sind bei immersiven Systemen ebenfalls weit verbreitet. Sie werden meistens über Tracking-Systeme gesteuert und bieten daher sechs Freiheitsgrade.</li> </ul>

**Tabelle 4-3: Analyse des Designmusters WIMP und Übertragung auf die immersive Benutzungsoberfläche**

In der rechten Spalte von Tabelle 4-3 werden Prinzipien für immersive Benutzerschnittstellen aus dem WIMP-Konzept abstrahiert. Folgende Unterschiede zu 2D-GUI werden insbesondere festgestellt und bei der Umsetzung des immersiven UI berücksichtigt:

Der Benutzer hat in immersiven Systemen ein Sichtfeld von bis zu 360 Grad und die Möglichkeit zur freien Navigation in der gesamten Szene. Es ist daher für die meisten Anwendungen sinnvoll, die Hilfsmittel der Benutzerschnittstelle mit der Benutzerposition zu koppeln. Weil die Bewegung und Orientierung des Benutzers immer mitverfolgt wird, ist es damit auch möglich, dass die Hilfsmittel ebenfalls mit dem Benutzer bewegt werden. Diese Vorgehensweise ist vergleichbar mit einer Situation, in der der Benutzer vom Cockpit eines Flugzeugs aus die Außenwelt betrachtet. Auch bei einer Richtungsänderung bleiben die Instrumente immer um ihn herum an gleicher Position. Auch bei einer Bewegung des Benutzers in der VR bleiben die Hilfsmittel um ihn herum angeordnet. Diese Technik übernimmt in der immersiven Benutzungsoberfläche eine ähnliche Funktionalität wie die Kontext-Menüs in 2D-GUI.

Einige Eigenschaften von WIMP sind auch in derzeitigen immersiven Benutzerschnittstellen bereits vorhanden, z. B. das analog zur Maus eingesetzte *Pointer-Device*. VR-Systeme wie Lightning, VDP, NxGen oder VISTA bieten auch die Möglichkeit an, Menüs zu verwenden. Allerdings sind diese Menüs in der Regel ausschließlich mit dem Eingabegerät verbunden. Zudem können sie nur als einzelne baumartige Struktur definiert werden. Sie sind in der Gestaltung und in der Verwendung also nicht so flexibel und vielfältig wie die Menüs eines 2D-GUI (Fenstermenü, Kontext-Menü, Floating-Menü usw.). Im Ergeb-



nis bedeutet dies, dass für die Entwicklung eines immersiven UI noch viel Potenzial zur Verbesserung der Benutzbarkeit gegeben ist.

Dreidimensionale Symbole zur Aktivierung einer Aktion werden in einigen immersiven Anwendungen bereits eingesetzt. Ein Vergleich mit den zahlreichen Bearbeitungsprogrammen und Tools zur Wandelung eines normalen Bildes zum Icon, die für das Design von 2D-GUI verfügbar sind, zeigt das in Bezug auf die 3D-Umgebungen bestehende Defizit. Vor allem bietet die immersive Umgebung zusätzliche Möglichkeiten zur Gestaltung von Metaphern für Icons. Da in der VR neben dem Interaktionsgerät beispielsweise auch Kopfposition und –orientierung verfolgt werden, sind auch diese zur Aktivierung einsetzbar.

Die Benutzungsoberfläche von Windows ist als 2D-GUI so weit verbreitet, dass kaum mehr wahrgenommen wird, wie fundamental WIMP als Designmuster zur Benutzbarkeit und Arbeitseffizienz beiträgt. Der Umstand, dass es für immersive Benutzungsoberflächen noch kein vergleichbares Referenzmuster gibt, stellt nach Ansicht des Autors im Bereich der immersiven Benutzerschnittstelle das größte Defizit dar. Um dazu beitragen zu können, dieses Defizit abzubauen, werden im folgenden Abschnitt die Grundzüge eines Designmusters für ein immersives UI vorgestellt und erläutert.

#### 4.6.2 Immersives Repräsentationsmuster

In der Analyse des WIMP-Konzepts (Tabelle 4-3) wird als Analogie zu Window der *Scope* als abgegrenzter Raum in der virtuellen Welt eingeführt. Wie ein Desktop-Fenster lässt er sich in der VR-Umgebung bewegen, verkleinern, vergrößern oder modifizieren. Zusätzlich bietet ein solcher Raum die Möglichkeit, ihn nach dem Vorbild der realen Welt einzurichten, sodass der Benutzer in einer intuitiv erfassbaren Umgebung alle nötigen Informationen leicht erreichen und effizient arbeiten kann. Das vom Autor dafür vorgeschlagene Muster wird im Folgenden am Beispiel des Fahrzeugkonfigurators vorgestellt.

Im Fahrzeugkonfigurator werden zwei Gruppen von RD-Datensätzen nach dem Scope-Prinzip entworfen. Die Tabelle 4-4 zeigt jeweils das Designmuster des immersiven UI und in der rechten Spalte dessen Entsprechung in einer typischen 2D-Benutzungsoberfläche.

Elemente der Benutzungsoberfläche	Muster in immersiver Benutzungsoberfläche	Analoger Ansatz in WIMP
<p><b>Icon</b></p> <p>Jede Option kann zu einem 3D-Icon minimalisiert werden. Das Symbol hat einen festen Durchmesser von 0,5 Meter, einen Ring unter sich, eine namentliche Beschriftung und steht zu anderen Symbolen in einer festen gleichmäßigen Distanz.</p>		
<p><b>Scope</b></p> <p>Berührt das Pointing-Device ein 3D-Icon, wird dieses markiert. Erfolgt eine Selektion, wird der Scope für die Icon-Option geöffnet. Alle möglichen Werte der Option werden angezeigt. Zusätzlich wird nach dem Ablaufen der Übergangsanimation der Ring unter den Icon vergrößert, um zu verdeutlichen, welche Option gerade ausgewählt ist.</p>		
<p><b>Übergangsanimation</b></p> <p>Während der Scope geöffnet wird, wird eine Übergangsanimation angezeigt. Der unter dem Icon liegende Ring bewegt sich nach oben und das Icon verschwindet. Der Ring vergrößert sich und bewegt sich wieder nach unten. Währenddessen werden nacheinander alle Werte der Option angezeigt.</p>		


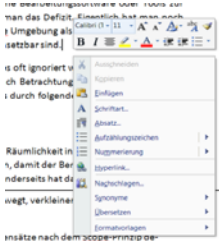
<p><b>Kontext-Menü</b></p> <p>Verharrt der Fokus auf einem Optionswert, werden die dazugehörigen Menüs mit statistischer Information als 3D-Text angezeigt. Gleichzeitig werden Dialogmöglichkeiten mit der Statistik eingeblendet. Die Statistikdaten werden auch bei einem Fernzugriff auf Datenbanken in Echtzeit aktualisiert.</p>		
--	---	---

Tabelle 4-4: Immersives Designmuster im Vergleich mit WIMP

## 4.7 Modellierungsprozess

Nach der Erzeugung eines RD-Inhalts (siehe Abschnitt 4.5) werden anschließend die entsprechenden RD-Repräsentationen konstruiert. Da die RD-Repräsentationen überwiegend in Form von Geometrie dargestellt werden, wird dieser Konstruktionsprozess im RD-Konzept als Modellierung bezeichnet. Um die Modellierung durchzuführen, stehen grundsätzlich zwei unterschiedliche Vorgehensweisen zur Wahl:

Einmal ist es möglich, die Repräsentation eines RD-Datensatzes unmittelbar zusammen mit dem RD-Inhalt zu erzeugen. Während der Datensatz angelegt wird, werden bereits alle Ressourcen, die zur Repräsentation notwendig sind, geprüft und geladen. Falls in einem Verarbeitungsschritt ein Fehler auftaucht oder benötigte Ressourcen nicht zur Verfügung stehen, bricht die gesamte Aktion ab. Ein Vorteil dieser Methode ist, dass damit zuverlässig vermieden wird, dass ein Datensatz zwar geladen, der Benutzer ihn aber nicht wahrnehmen oder darauf zugreifen kann. Das Prinzip dieser Modellierungsmethode zeigt das Sequenzdiagramm in Abbildung 4-9.

Ein Nachteil der gekoppelten Erzeugung ist deren Unflexibilität. In der Praxis ist es häufig nicht möglich, bei der Erstellung des Inhalts bereits alle Ressourcen für die Repräsentationen zur Hand zu haben. Insbesondere, wenn RD-Datensätze mehrere komplexe Verschachtelungen beinhalten, müssen zudem sämtliche Abhängigkeiten zwischen den Datensätzen exakt berücksichtigt werden. Aus diesen Gründen ist die gekoppelte Erzeugung von RD-Inhalten und Repräsentationen kaum wirtschaftlich zu realisieren.

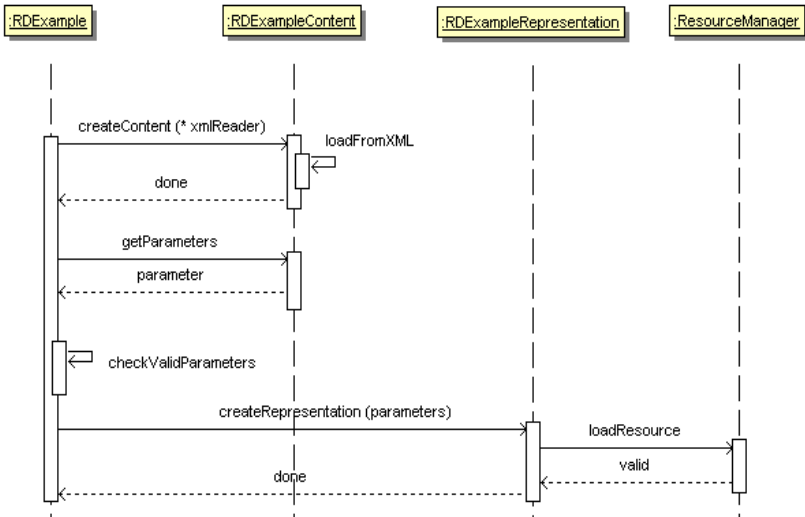
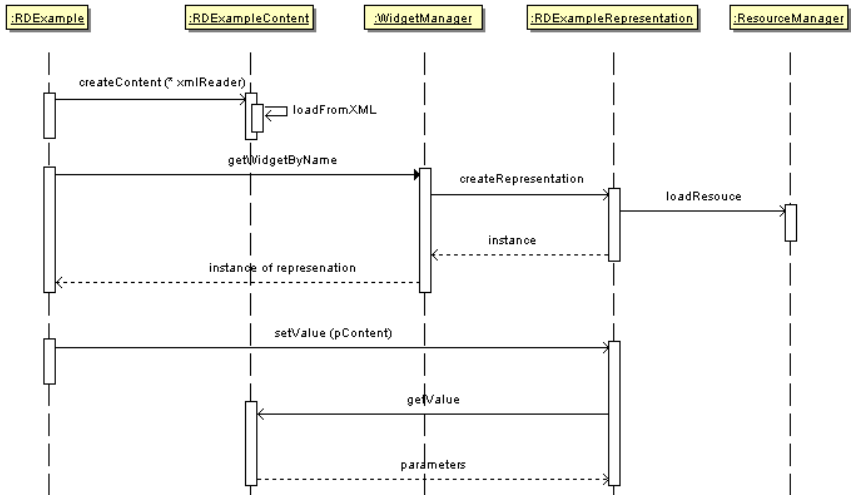


Abbildung 4-9: Sequenzdiagramm für die gekoppelte Erzeugung von Richdata-Inhalt und -Repräsentation

Als zweite Möglichkeit kann eine RD-Repräsentation auch unabhängig vom einzelnen RD-Inhalt erstellt werden. Diese RD-Repräsentation lässt sich dann während der Laufzeit je nach Bedarf mit einem RD-Inhalt verbinden. Ein großer Vorteil dieser Methode ist, dass es damit möglich wird, eine große Kollektion verschiedener Repräsentationen vorab und unabhängig von den RD-Inhalten zu erstellen. Der Anwendungsentwickler muss die einzelne Repräsentation dann nur noch für den jeweiligen Anwendungsfall instanzieren bzw. kombinieren. Das Prinzip dieser Modellierungsmethode ist im Sequenzdiagramm der Abbildung 4-10 ersichtlich. Als zusätzliches Element wird bei dieser Methode ein *WidgetManager* eingeführt, der die Verwaltung der Kollektion übernimmt. Die drei Schritte, die Erstellung der Repräsentation, die Erzeugung des RD-Inhalts und das Setzen des Werts, können vom Entwickler in beliebiger Reihenfolge flexibel durchgeführt werden. Dies wirkt sich allerdings unter Umständen nachteilig auf die Robustheit des Programms aus. Denn diese Methode schließt nicht zuverlässig aus, dass ein RD-Inhalt fehlerhaft mit einer Repräsentation verbunden wird oder einem RD-Inhalt keine Repräsentation zugeordnet wird. Außerdem fördert diese Methode ein Vorgehen, bei dem bei den verschiedenen Objekten mehr Funktionalitäten implementiert werden, als für einen konkreten Anwendungsfall benötigt werden. Dadurch werden zum Einen für die spätere Anwendung mehr Rechenressourcen notwendig. Zweitens ist die Gefahr von Programmierfehlern gegeben, da jede Repräsentation der Kollektion auch eine öffentliche Methode anbieten muss, um Anpassungen und Änderungen durchzuführen.



**Abbildung 4-10: Sequenzdiagramm für die separate Erzeugung von Richdata-Inhalt und -Repräsentation**

Im RD-Konzept sind beide Wege zur Modellierung von RD-Datensätzen vorgesehen. Damit bietet das Konzept Entwicklern der VR-Plattformen die Möglichkeit, eine Auswahl von Repräsentation bereits als Basisbausteine anzubieten. Es ist zu erwarten, dass die möglichen nachteiligen Auswirkungen auf Robustheit und Effizienz des Systems durch Erfahrung in der Entwicklung und Erfahrung aus Produkttests weiter minimiert werden können. Da das RD-Konzept beide Modellierungsmethoden ermöglicht, können die Entwickler einer Integrationsanwendung auf Basis von Repräsentationen der Kollektion eine komplexe Richdata nach der ersten Methode erstellen. Dies sichert zum einen, dass die Anwendung optimal auf den grundlegenden Prozess angepasst werden kann und zum anderen wird damit der spätere Wartungsaufwand verringert.

## 4.8 Interaktion

### 4.8.1 Interaktion mit Standardmethode

Interaktion ist eine der drei wichtigsten Merkmale der VR-Technologie [Burdea 2003]. Grundlegend dafür ist, dass die Augenposition und -orientierung des Benutzers in Echtzeit verfolgt werden. Die Ergebnisse dieser Verfolgung werden in hoher Frequenz zur Berechnung der simulierten Umgebung in das VR-System eingegeben. Damit wird erreicht, dass der Benutzer während der Navigation in der visualisierten Welt einen "realen" Eindruck erhält.

Diese Art von Interaktion gehört zur grundlegenden Basis der stereoskopischen Sicht. Sie ist unabhängig von der Struktur des Szenengraphen sowie den logischen Verbindungen

zwischen den Objekten der Anwendung und ihren dreidimensionalen Darstellungen. Wie im Kapitel 3.1.3 erläutert, dient dieser gesamte Bereich dem Zweck der Wahrnehmung der physikalischen Wirklichkeit und liegt nicht im Fokus der vorliegenden Ausarbeitung.

Die meisten VR-Plattformen bieten zur Interaktion darüber hinaus ein strukturiertes Menü-System. Prinzipiell können RD-Datensätze dem Benutzer auch auf diese Weise zur Verfügung gestellt werden. Zum Beispiel lässt sich die Sichtbarkeit einer Text-Beschreibung im Menü ein- bzw. ausschalten. Diese Methode hat Vorteile, aber bringt auch Einschränkungen mit sich.

Das Menü verfügt über ein standardisiertes Layout. Der Aufbau des Menüs folgt einer baumartigen Struktur und verbindet die Befehle immer mit einer eindeutigen Aktion. Die Erstellung eines Menü-Eintrags ist einfach und erfolgt gemäß eines vordefinierten Musters. Softwaretechnisch lässt sich dieser Prozess auch abstrahieren und automatisieren, sodass die benötigten Vorkenntnisse und der Aufwand der Erstellung minimiert werden. Die baumartige Struktur hilft dem Benutzer auch, den Überblick zu behalten und erleichtert ihm, eine bestimmte Aktion zu finden und durchzuführen.

Allerdings sollte das Menü-System eines immersiven UI nicht als einziger Weg zur Erreichung und Ausführung von Aktionen gewählt werden. Wenn alle möglichen Aktionen in die Menüs integriert werden, wird die Baumstruktur umfangreich. Die Menüs werden unübersichtlich und die Auswahl bestimmter Aktionen wird erschwert, sodass die Arbeitseffizienz signifikant sinkt. Insbesondere die RD-Datensätze sollen und müssen oft sensibel auf ihren Kontext reagieren. Sie können zum Beispiel auch Bedingungen enthalten, die sich aus dem Benutzerverhalten, aus dem Zustand der Anwendung oder aus der Eingabe externer Quellen ergeben. Letzten Endes liegt das Ziel des RD-Konzepts gerade darin, dem Benutzer eine vielfältige, intuitive und effiziente Umgebung anzubieten.

## 4.8.2 Anwendungsspezifische Interaktion

### 4.8.2.1 Bestandteile der Interaktion

Eine Programmeinheit für Interaktion nach dem RD-Konzept besitzt mindestens folgende Komponenten:

- Mindestens zwei Zustände: aktiv und inaktiv  
Wenn eine Interaktion aktiv ist, muss sie jedes an den Eingabegeräten auftretende Ereignis verfolgen bzw. filtern, ob das Ereignis für die Interaktion relevant ist. Der für diese Verfolgung notwendige Aufwand wird groß, wenn gleichzeitig viele Interaktionen alle Eingabeereignisse verfolgen (listen). Daher ist es notwendig, dass Interaktionen je nach Bedarf aktiviert und deaktiviert werden können.
- Signatur (Bedingungen) für das Aktivieren der Interaktion  
Anhand der Signatur wird geprüft, ob alle Ressourcen für eine Aktivierung der Interaktion vorhanden sind und ob sie fehlerfrei initialisiert.
- Identifizierung aller betroffenen RD-Repräsentationen

Die einzelnen an den Eingabegeräten auftretenden Ereignisse enthalten Informationen über die betroffenen Objekte im Szenengraphen. Die Programmeinheit für Interaktion muss identifizieren können, ob eine RD-Repräsentation von dem jeweiligen Ereignis betroffen ist.

- Zugriffsmöglichkeit auf alle betroffenen RD-Datensätze

Eine Programmeinheit für Interaktion kann den Zustand der betroffenen RD-Datensätze gemäß der Benutzeraktion ändern.

Der detaillierte Ablauf während der beiden letztgenannten Punkte wird im folgenden Kapitel 4.8.2.2 erklärt. Als Voraussetzung für das Starten einer Interaktionsmethode wird festgelegt, dass

- jeder einzelne RD-Dateninhalt bereits geladen bzw. generiert ist,
- die Repräsentationen nach dem Modellierungsprozess erzeugt oder aus einer Kollektion ausgewählt und instanziiert und
- die Abhängigkeiten und Änderungsregeln des Dateninhalts nach der Programmlogik des einzubettenden Prozesses nachgebildet wurden.

#### *4.8.2.2 Technischer Ablauf*

Im RD-Konzept sind die Geometriedaten, die der Benutzer direkt sehen und manipulieren kann, nicht mehr die zu behandelnden Gegenstände. Vielmehr sind sie lediglich ein Zugang zu den RD-Datensätzen. Daher ist der technische Ablauf der Interaktion komplexer als bei konventionellen VR-Anwendungen. Einen Überblick des Ablaufs gibt Abbildung 4-11. In der vorliegenden Arbeit wird davon ausgegangen, dass mindestens folgende Benutzeraktionen in der VR-Umgebung verfolgt werden:

1. Die Positionen und die Orientierungen der Augen des Benutzers
2. Die Position und die Orientierung der immersiven Eingabegeräte
3. Die Tasten-Ereignisse der Eingabegeräte

Die ersten beiden Aktionen werden kontinuierlich erfasst und umfassen sechs Freiheitsgrade (6-DoF). Die Informationen über die Benutzeraktionen können aus verschiedenen Quellen stammen. Üblich ist die Verwendung eines Tracking-Systems und von Desktop-Eingabegeräten, z. B. einer SpaceMouse. Alternativ können die sechs Freiheitsgrade auch durch Eingabegeräte wie einem Spiele-Controller oder einer Kombination aus Maus und Tastatur simuliert werden. In diesem Fall liegen zwar keine echten Positions- und Orientierungsinformationen vor, für Test- oder Demonstrationszwecke reicht die Simulation aber aus.

Die physikalischen Signale werden vom Eingabegerät zum VR-System gesendet. Dort werden sie ausgewertet und in interne Systemereignisse, die unabhängig von der physikalischen Eingabe sind, umgewandelt. Auch die simulierten sechsdimensionalen Signale, wie sie von dem im letzten Absatz erwähnten Spiele-Controller geliefert werden, werden in interne Systemereignisse umgewandelt.

Die Systemereignisse für kontinuierliche Interaktion, z.B. der Navigation, erhalten eine eigene Pipeline, da sie in hoher Frequenz (über 30 Hz) bearbeitet werden müssen [Bues 2009]. Benutzerbefehle, wie Ereignisse aufgrund der Betätigung einer Taste, erlauben dagegen eine relativ niedrige Bearbeitungsfrequenz. Die Untersuchungen von Piantanida und Cheshire kommen zu dem Ergebnis, dass eine minimale Frequenz von 12 Hz ausreicht [Piantanida 1993; Cheshire 1996], um das Gefühl einer fließenden Interaktion zu gewinnen.

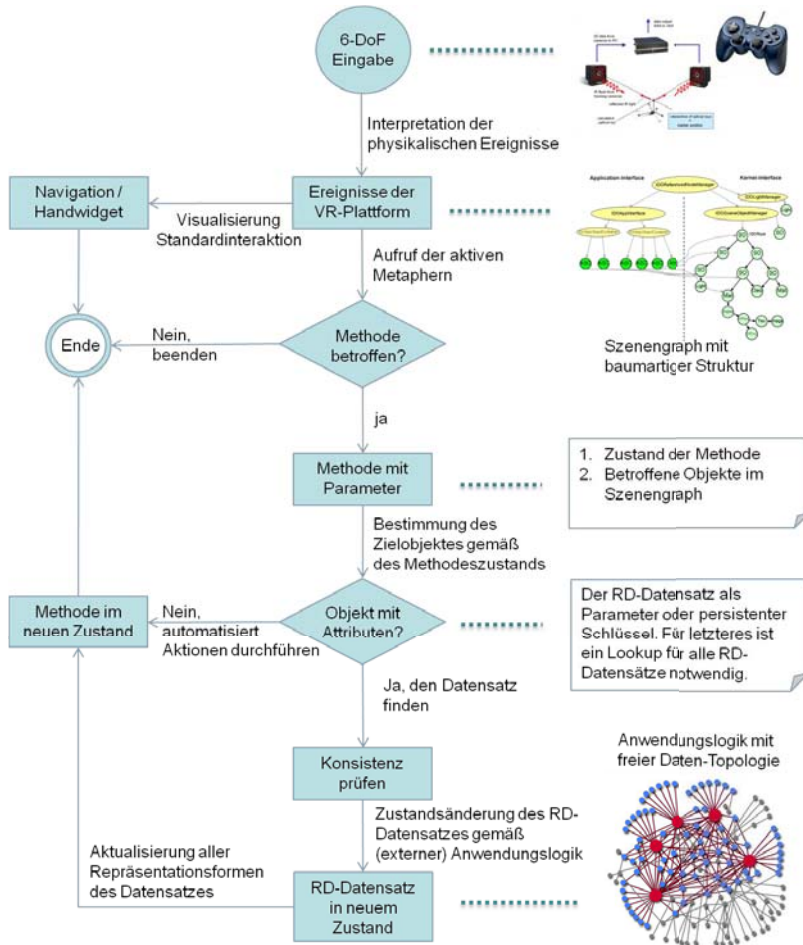


Abbildung 4-11: Technischer Ablauf bei der Interaktionsmethode



Im nächsten Schritt wird geprüft, ob der Typ des Systemereignisses mit den Eingangsbedingungen der aktiven Interaktionsmethode vereinbar ist. Dabei ist das *Picking* ein wichtiger Vorprozess. Dabei wird getestet, ob bzw. welche Objekte im Szenengraphen vom Benutzerfokus getroffen werden. Denn nicht die Bewegung des Benutzerfokusses, sondern das Treffen des Fokusses bzw. seiner Verlängerung auf ein bestimmtes Objekt aktiviert erst eine Zustandsänderung der Methode.

Besonders beachtet werden muss der Umstand, dass die Datenstrukturen der VR-Anwendung im Prinzip eine freie Topologie haben können, während der Szenengraph eine starre baumartige Struktur besitzt. Sollen nun die Anwendungsdaten durch Interaktion in der immersiven Benutzungsoberfläche manipuliert werden, müssen beide Topologien miteinander verbunden werden. Programmiertechnisch gibt es viele Möglichkeiten, diese Verbindung zu realisieren. Im RD-Konzept sind zwei Möglichkeiten vorgesehen, um übersichtliche und effiziente Verbindungen zu erstellen.

In der ersten Variante überprüft die Methode alle relevanten RD-Repräsentationen daraufhin, ob das gepickte Objekt in einer Repräsentationsliste auftritt. Wenn ja, kann die Methode auch den entsprechenden RD-Datensatz finden und nach der internen Logik die Aktionen durchführen. Diese Suche kann allerdings aufwendig werden, wenn eine Methode eine große Anzahl von Repräsentationen beeinflusst und das Prüfungsverfahren kontinuierlich in einer hohen Frequenz aufgerufen wird.

Die zweite Variante führt einen Schlüssel als Attribut ein. Der Schlüssel wird als Attribut den entsprechenden Szenegraph-Knoten zugeordnet und stellt damit einen Verweis auf die RD-Repräsentation und den Dateninhalt her. Wenn ein Objekt gepickt wird, genügt es zu prüfen, ob dem Objekt ein Attribut (der Schlüssel) zugeordnet ist. Wenn ja, lässt sich mit dem Schlüssel der RD-Datensatz finden. Dieser zweite Weg ist effizienter, setzt aber voraus, dass das zugrunde liegende VR-System den Attributierungsprozess unterstützt.

Nachdem die Aktion durchgeführt ist, aktualisiert die Methode den eigenen Zustand. Eventuell kann die Methode auch den Zustand anderer Methoden ändern, z. B. aktivieren oder ausschalten. Danach gibt sie dem System das Signal, dass diese Benutzeraktion bearbeitet wird.

#### **4.8.2.3 Implementierungsaspekte für Interaktion**

Eine Interaktion beinhaltet die Benutzeraktion, die Systemreaktionen sowie die Implementierung einer Zustandsänderung des zugrunde liegenden RD-Datensatzes. Eine Programmeinheit für Interaktion muss nicht nur dem in Abschnitt 4.8.2.2 beschriebenen technischen Ablauf folgen, sondern hat auch eine semantische Struktur, die je nach Anwendung festgelegt werden muss. Im Vergleich zur Manipulation reiner Geometrie-Objekte wirkt die Interaktion bei RD-Datensätzen selten isoliert. Zum Beispiel initiiert die Auswahl eines Optionswertes im Fahrzeugkonfigurator, dass

- die 3D-Ikone geändert,
- die Geometrie in der Konfiguration ausgetauscht,
- der 3D-Text für die Beschreibung der Option geändert,
- die Farbe der Beschreibung geändert und
- der Zustand des Datensatz-Inhalts geändert wird.

Alle diese Änderungen geschehen an verschiedenen Stellen des Szenengraphen und auf unterschiedlichen Ebenen. Bei der Konstruktion des Szenengraphen sollte daher auch beachtet werden, wie sich die Interaktion am effizientesten organisieren lässt.

Das RD-Konzept unterstützt das Design der Interaktion, wenn die Logik der Anwendung durch RD-Datensätze gekapselt wird. Das Layout des immersiven UI wird durch Kombinationen der RD-Repräsentationen erleichtert. Somit kann sich der Designer auf die Aktion und Reaktion des Benutzers sowie die Reihenfolge der Durchführung konzentrieren.

Zusätzlich können Aktionsmuster, z. B. Muster für Transformationen einer Gruppe von RD-Repräsentationen oder zeitliche Verläufe, im Zuge des Designs erstellt und angereichert werden. Nähere Erläuterungen enthält der folgende Abschnitt zu Möglichkeiten der Animation.

#### 4.8.3 Animation

Animation ist eine spezielle Interaktion im RD-Konzept, die nach dem Aufruf innerhalb einer zeitlichen Periode automatisch bis zum Ende durchgeführt wird und keine Zustandsänderung eines RD-Datensatzes verursachen kann.

Animation kann mit anderen Interaktionen kombiniert werden, um interaktive virtuelle Umgebungen mit lebhaften Effekten anzureichern. Jede Animation hat folgende drei Zustände: inaktiv, aktiv und pausiert. Damit ist es auch möglich, dass eine Interaktionsmethode verschiedene Animationen steuert. Transformation, Sichtbarkeit und Material der Geometrie-Daten können innerhalb der Animation in zeitlicher Reihenfolge abgewandelt werden. Zudem können dem Benutzer während oder an einem bestimmten Zeitpunkt der Animation Audio, Bilder oder bewegte Bilder eingespielt werden.

In der Praxis betrifft die Animation nur selten einen einzigen Geometrie-Datensatz. In den meisten Fällen betrifft sie eine ganze Gruppe von Datensätzen. Die Bewegung, die Orientierung und die Skalierung der Geometrie werden durch die Transformationsmatrix der zugeordneten Szenengraph-Knoten dargestellt. Dazu wird im RD-Konzept, wie in Abbildung 4-12 im Klassendiagramm illustriert, eine Abstraktionsschicht für die Transformationsstrategie verankert. Es können Kollektionen solcher Strategien erstellt werden, um die Erfahrungen vergangener Arbeiten zu sammeln und für zukünftige Arbeiten zur Verfügung zu stellen.

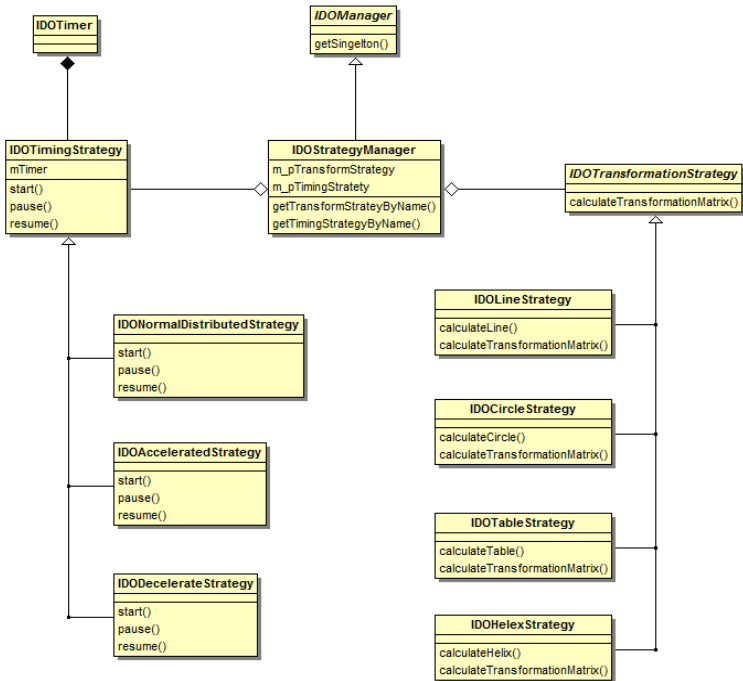


Abbildung 4-12: Klassendiagramm der Transformations- und zeitlichen Strategie im Richdata-Konzept

Ähnlich wird auch die zeitliche Abfolge durch eine nominierte Zeiteinheit attribuiert. Die Timing-Strategie läuft in einem eigenen Thread und besitzt einen *Timer*. Anders als die Transformationsstrategie ist die obere Klasse *IDOTimingStrategy* bereits instanzierbar. Ihr Zeitverlauf ist kontinuierlich und gleichmäßig.

## 4.9 Kommunikation mit externen Informationsquellen

Bidirektionale Kommunikation zwischen Anwendung und Backend ist eine Voraussetzung für eine integrierte Softwarelösung. Im Richdata-Konzept wird die Kommunikation in zwei Kategorien unterteilt:

- strukturierte Datenquellen,
- unstrukturierte Datenquellen.

### 4.9.1 Strukturierte Datenquellen

Wenn Informationsquellen ihre Daten strukturiert halten, ist es möglich, auf die Information gemäß der gegebenen Struktur bzw. Schnittstelle zuzugreifen oder diese zu än-

dern. Typische strukturierte Datenquellen sind Datenbanken. Sie sind in betrieblichen Informationssystemen hoch komplex. Ein Beispiel dafür ist das Produktdaten- bzw. Produktlebenszyklusmanagement in der Automobilindustrie [Stelzer 2009]. VR-Systeme können nur in den betrieblichen Arbeitsprozess integriert werden, wenn sie dazu in der Lage sind, sie während der Laufzeit mit solchen Datenbanken zu kommunizieren.

Eine Verbindung über die Datenbankschnittstelle ist effizient und exakt. Allerdings haben Datenbanken in der Regel proprietäre Schnittstellen, die nicht miteinander kompatibel sind. Zudem stützen sich moderne Datenbanken aus Organisations- und Sicherheitsgründen auf weltweit verteilte Strukturen. Die Folge davon ist, dass die Kommunikation zwischen den unterschiedlichen Datenquellen einschließlich der VR-Programme eine komplexe Aufgabenstellung ist. Die Service-Orientierte Architektur (SOA) ist ein Paradigma für die Strukturierung und Nutzung verteilter Funktionalitäten, die von unterschiedlichen Besitzern verantwortet werden. Webservices, die nach dem SOA-Prinzip aufgebaut sind, werden seit mehreren Jahren weit verbreitet eingesetzt. Sie vermindern vor allem die Problematik der Kompatibilität zwischen den unterschiedlichen Datenquellen.

Ziel des RD-Konzepts ist es, die Integration von VR zu erleichtern. Daher enthält das Konzept auch eine softwaretechnische Schnittstelle auf der Basis von SOA. Die Schnittstelle erlaubt den Zugriff auf Webservices und ermöglicht so, mit wenig Aufwand auf alle Datenbanken über das Internet zuzugreifen. Einzige Bedingung dabei ist, dass die jeweilige Datenquelle selbst einen Webservice unterstützt.

Für jeden Service wird ein Adapter erstellt, der eine Menge von Remote-Funktionsaufrufen beinhaltet. Innerhalb des Adapters wird der Funktionsaufruf zu einem lokalen Anknüpfungspunkt (Stub) weitergeleitet, der durch eine Webservice-Beschreibung (*Web Service Description Language, WSDL*) erzeugt wird und auf die Remote-Ressourcen zugreifen kann. In der Abbildung 4-13 wird ein entsprechendes Klassendiagramm am Beispiel der Fahrzeugkonfiguration vorgestellt.

Soweit Datenquellen mit integriert werden sollen, die keinen Zugriff über einen Webservice unterstützen, müssen die Entwickler nach wie vor für jeden konkreten Anwendungsfall eine Schnittstelle implementieren. Aber auch in diesem Fall bringt das RD-Konzept für die Datenkommunikation Vorteile. Die grundlegenden Datenstrukturen der externen Datenquellen werden als RD-Datensätze flexibel und robust in die VR-Anwendung implementiert.

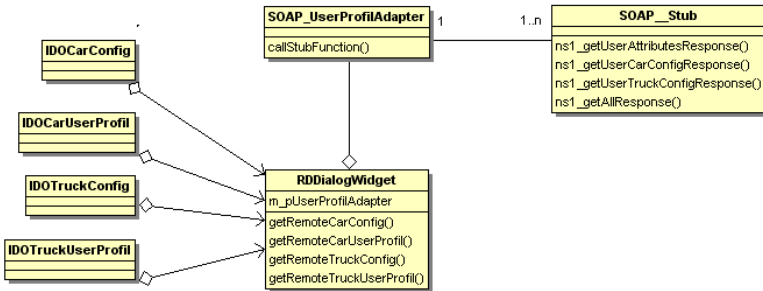


Abbildung 4-13: Klassendiagramm der Programmierschnittstelle für Webservices

## 4.9.2 Unstrukturierte Datenquellen

Unstrukturierte Datenquellen, z. B. Webinhalte, haben verschiedene Datenformate bzw. Schnittstellen. Sie werden auf unterschiedliche Weise auf konventionellen Computerbildschirmen angezeigt. Die Idee des RD-Konzepts ist, eine generelle Hilfsmethode zu entwickeln, die den Computerbildschirm in die immersive Umgebung bringt. Analog zur Anzeige am Desktopbildschirm wird damit der Zugriff auf unstrukturierte Datenquellen auch in der VR ermöglicht. Der Einsatz der Methode ist nicht davon abhängig, in welcher Form die Daten gespeichert oder angezeigt werden, und wo sich die Datenquelle befindet. Voraussetzung ist lediglich ein Computer, der das Zugriffsrecht auf die Datenquelle und auf eine 2D-Benutzungsoberfläche hat. Sofern der Computer vom VR-Programm direkt oder über ein Netzwerk erreichbar ist, lassen sich die Zieldaten in virtuellen Umgebungen anzeigen bzw. ändern.

Im RD-Konzept wird diese Methode unter der Bezeichnung *IVR-Panel* entwickelt. IVR steht dabei als Abkürzung für *Interactive Virtual Representation*. Technisch besteht die Methode aus drei Teilen:

1. Ein Modul, das in die zugrundeliegende VR-Plattform integriert wird. Dieses Modul verwaltet die Darstellung des ferngesteuerten Bildschirms und muss abhängig von der VR-Plattform und dem Betriebssystem realisiert werden.
2. Eine modifizierte Version des *RealVNCViewer*. Diese ist kompatibel zu dem *Remote Framebuffer Protocol*, das von Virtual Network Computing (VNC) definiert wird. Damit ist gewährleistet, dass das *IVRPanel* unabhängig vom Betriebssystem alle Remote-Displays anzeigen kann, sofern auf der Remote-Seite ein VNC-kompatibles Programm läuft.
3. Ein gemeinsamer Adressraum im Arbeitsspeicher, auf den sowohl das VR-Programm (Punkt 1) als auch der VNC-Viewer (Punkt 2) zugreifen kann. Dieser Teil der Methode ist unabhängig von VR-Plattform und Betriebssystem.

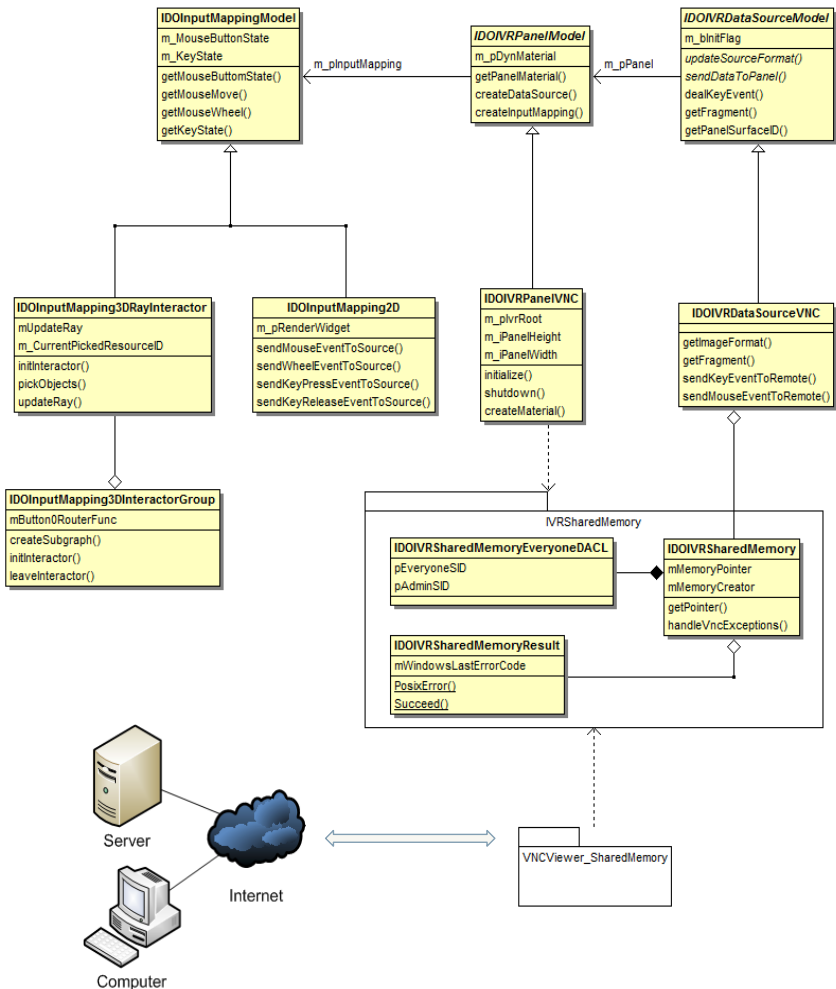


Abbildung 4-14: Struktur und Klassendiagramm für die Methode IVRPanel

Die zentrale Komponente des IVR-Panels ist eine dynamische Textur, die sowohl in hoher Frequenz als auch in kleinen Fragmenten aktualisiert werden kann. Diese Technik kann ebenfalls eingesetzt werden, um allgemein bewegte Bilder in der immersiven Umgebung wiederzugeben. Die Länge und die Breite der Textur sollten als Zweierpotenz angelegt sein. Der Renderer, der in industriellen Anwendungen meistens auf OpenGL basiert, muss die Textur dann nur einmal und nicht ein zweites Mal skalieren. Da das

originale Remote-Display selten die Größe einer Zweierpotenz hat, werden in IVRPanel die Textur-Koordinaten der Geometrie-Objekte an die Zielauflösung angepasst. Damit ist sichergestellt, dass die Textur in den richtigen Seitenverhältnissen vor dem Benutzer angezeigt wird.

Abbildung 4-14 zeigt die Klassenstruktur der Implementierung. Die Klasse *IDOIVRPanelVNC* hat auf einer Seite über den *IDOIVRDataSourceVNC* Zugriff auf den gemeinsamen Adressraum. Auf der anderen Seite besteht mit *InputMapping* eine direkte Zugriffsmöglichkeit auf Ereignisse aus dem VR-System. Solche Ereignisse können sowohl von immersiven Eingabegeräten, als auch vom 2D-GUI des Remote-Bildschirms stammen.



Abbildung 4-15: Metapher des IVR-Panels als  
eigenständige Anwendung (links) und integriert (rechts)

Die Übertragungen bzw. die Transformationen der Ereignisse laufen entlang zwei Linien:

- Zum einen kommen die Ereignisse aus dem Remote-Computer-Bildschirm über das VNC-Protokoll, *VNCViewer\_SharedMemory* und *IVRSharedMemory* zum dynamischen Material im *IDOIVRPanelVNC*. Dieses Material kann auf eine Oberfläche gelegt werden. Darauf wird dem Benutzer schließlich der Bildschirm angezeigt. Die Informationen sind Bilder bzw. Bildfragmente.
- Zum anderen gelangen die Benutzerereignisse vom *InputMapping* (entweder immersiv oder aus dem 2D-GUI) über *IDOIVRDataSourceVNC*, *IDOIVRSharedMemory* und den *VNC-Viewer* zum Remote-Computer. Die Informationen sind umgewandelte Button- und Maus-Ereignisse.

Das IVRPanel ist, wie in Abbildung 4-15 zu sehen, mit zwei Modi ausgestattet. Einmal dient die Metapher als eigenständige Anwendung. Die Oberfläche der Repräsentation wird gleichzeitig auch zum Bedienen des Remote-Computers eingesetzt. Der Benutzer kann alle Maus-Ereignisse mit einem 6-DoF-Eingabegerät simulieren. Im zweiten Modus

stellt das IVRPanel eine dynamische Textur für andere Anwendungen zur Verfügung. Dabei wird das Material mit dieser Textur erzeugt und in einer Materialbibliothek abgelegt.



## 5 Umsetzung des Richdata-Konzepts auf einer VR-Plattform

### 5.1 Vorgehensweise

Das in Kapitel 4 erläuterte RD-Konzept ist unabhängig von einer konkreten VR-Plattform. Bei der Realisierung des Konzepts müssen die technischen Rahmenbedingungen der zugrundeliegenden VR-Plattform jedoch mit einbezogen werden, denn sowohl die Repräsentationen der Datenformen als auch der Leitfaden des Entwicklungsprozesses der VR-Anwendung setzen bestimmte softwaretechnische Eigenschaften bzw. eine Infrastruktur voraus.

In diesem Kapitel erfolgt in vier Schritten die Umsetzung des Richdata-Konzepts auf einer VR-Plattform:

- Zuerst wird eine VR-Plattform ausgewählt und die Struktur der gewählten Ziel-Plattform analysiert.
- Anschließend werden die Bedingungen und Voraussetzungen des RD-Konzepts aufgezeigt und anhand der Ziel-Plattform festgestellt, ob bzw. welche Modifikationen, Ergänzungen bzw. Erweiterungen für die Realisierung des RD-Konzepts notwendig sind.
- Soweit Änderungen oder Erweiterungen notwendig sind, werden dafür technische Lösungen vorgeschlagen bzw. umgesetzt.
- Schließlich wird die technische Konzeption der Richdata-Anwendung auf der Ziel-Plattform realisiert.

In diesem Kapitel wird noch keine konkrete Anwendung auf Basis des RD-Konzepts entwickelt. Jedoch wird festgestellt, wie eine solche Anwendung entwickelt werden kann, damit diese mit der Ziel-Plattform vereinbar ist.

### 5.2 Auswahl der Zielform

Die *Virtual Decision Platform* (VDP) ist eine kommerzielle VR-Plattform, die von der Firma IC:IDO GmbH, dem Marktführer für VR-Lösungen in der deutschen Automobil- und Flugzeugbau-Industrie, angeboten wird. Auch in anderen europäischen Ländern sowie den USA und Asien wird VDP in zahlreichen Unternehmen der Maschinenbau-Branche im Entwicklungsprozess eingesetzt. Im Vergleich zu VR-Plattformen im Forschungsbereich hat die VDP als kommerzielle Lösung Stärken in Bezug auf die Robustheit und die Skalierbarkeit. Vor allem wird diese VR-Plattform in Wirtschaftsbereichen, für die ein großer Bedarf an VR-Integration angenommen werden kann, bereits von einem größeren Anwenderkreis benutzt. Deshalb wird die VDP in der vorliegenden Ausarbeitung als Referenz-Plattform für die Realisierung des RD-Konzeptes gewählt. Das RD-Konzept wird auf der Basis der Plattform technisch konzipiert.



- Die Plugins bzw. die Module, welche auf Basis einer der drei oben genannten Schnittstellen implementiert werden. VR-Anwendungen, die nach dem RD-Konzept entwickelt werden und die Arbeitslogik eines externen Prozesses in der virtuellen Umgebung nachbilden, sollen als Plugins auf der VDP implementiert werden.

### 5.3 Bedingungen bzw. Voraussetzungen des Richdata-Konzepts

In Kapitel 4.2.2 wurden die grundlegenden Funktionalitäten abgeleitet, die für die Anwendung des RD-Konzeptes erfüllt sein müssen. In Tabelle 5-1 wird untersucht, inwieweit die VDP, Version 2007, diese Voraussetzungen erfüllt.

Voraussetzungen	Stand in der VDP
Zugriff auf den Szenegraph – Erstellen, Löschen und Umstrukturieren von Knoten und Lichtquellen	<b>Erfüllt</b> Die VDP hat einen eigenständigen Szenegraphen und ein Dateiformat, das unabhängig vom Renderer ist. Daher ist ein Zugriff auf den Szenegraphen möglich.
Manipulieren des Szenegraphen – Transformation, Material und Shader	<b>Großteils erfüllt</b> Jeder Knoten im Szenegraphen besitzt zwei Transformationsmatrizen – für lokale und globale Koordinatensysteme. Für diese Materialien wird ein Manager implementiert, der Organisationsaufgaben übernimmt und mögliche Redundanzen vermeidet. Eine Manipulation innerhalb einer Textur (Image), wie sie etwa für bewegte Bilder notwendig ist, wird nicht unterstützt.
Identifizieren einzelner Knoten im Szenegraphen durch externen, eindeutigen, laufzeitunabhängigen Schlüssel	<b>Zum Teil erfüllt und nur mit erheblichem Ressourcenaufwand vollständig realisierbar</b> Für jeden Knoten des Szenegraphen ist es möglich, einen oder mehrere dynamische Parameter hinzuzufügen. Allerdings: Falls an einem Knoten ein Parameter hinzugefügt wird, müssen alle anderen Knoten ebenfalls diesen Parametertyp haben. Zudem sind die Parameter nur während der Laufzeit gültig. Es gibt keinen allgemeinen Mechanismus, um laufzeitunabhängige Schlüssel zu erzeugen.
Eventübertragung aus Benutzer-Aktion zur Anwendungslogik	<b>Erfüllt</b> Die VDP besitzt Mechanismen, um die Ereignisse von Eingabegeräten in eine Benutzeraktion umzuwandeln. Die zwei- und dreidimensionale Eingabe wird getrennt behandelt. Das dreidimensionale UI erhält Events von einem <i>Device-Server</i> , der eigenständig läuft.  Der Server gibt einen eigenen Adressraum für den Treiber der Eingabegeräte und das Hauptprogramm der VR-Plattform frei. Das Hauptprogramm greift in einer relativ niedrigen Frequenz auf den Adressraum des <i>Device-Servers</i> zu und holt sich die benötigten Events.

Voraussetzungen	Stand in der VDP
Verbinden logischer Strukturen mit Szenegraph-Knoten	<b>Erfüllt</b> Die Programmierschnittstelle für Anwendungen bietet Mechanismen an, um ein Alias von Szenegraph-Knoten zu erzeugen bzw. zu behalten. Durch das Alias kann auf den Knoten zugegriffen werden. Beim Herunterfahren des Systems werden alle Ressourcen für das Alias freigegeben.
Identifizieren logischer Strukturen durch Szenegraph-Knoten	<b>Kein allgemeiner Mechanismus angeboten</b> Durch eine Picker-Funktion kann ein Knoten des Szenegraphen identifiziert werden. Der Knoten hat zwar einen eindeutigen Schlüssel, der ist aber nur während der Laufzeit gültig.

**Tabelle 5-1: Checkliste zur Erfüllung der Voraussetzungen des RD-Konzepts auf VDP 2007**

In Kapitel 4.4 wurden Anforderungen an die Modularisierung gestellt. Die VDP erfüllt beide Anforderungen (siehe Tabelle 5-2).

Bedingungen	Stand in VDP
Das Konzept soll mit einer objektorientierten Programmiersprache implementiert werden.	<b>Erfüllt</b> Die VDP wird in C++ programmiert. Somit sind die Bedingungen der Objektorientierung erfüllt.
Das Konzept basiert auf einem modularisierten VR-System	<b>Erfüllt</b> Die VDP ist ein modularisiertes System.

**Tabelle 5-2: Checkliste zur Erfüllung der Anforderungen zur Modularität des Richdata-Konzepts auf VDP 2007**

In Kapitel 4.8.2.2 wurden die Anforderungen zum Interaktionsablauf behandelt. In der Tabelle 5-3 wird die VDP daraufhin überprüft.

Bedingungen	Stand in VDP
Bei der Interaktionsmethode wird davon ausgegangen, dass die Position und die Orientierung der Augen des Benutzers, der immersiven Eingabegeräte ebenso verfolgt werden wie die Tasten-Ereignisse der Eingabegeräte.	<b>Erfüllt</b> Eine Standard-VDP-Eingabe in immersiven Umgebungen erfüllt die Bedingungen.
Das zugrundeliegende VR-System soll den Attributierungsprozess des Szenegraphs <b>optional</b> unterstützen.	<b>Nicht erfüllt</b> VDP 2007 unterstützt die Attributierung nicht.

**Tabelle 5-3: Checkliste zur Erfüllung der Anforderungen zum Interaktionsablauf auf VDP 2007**

Die nicht vollständig erfüllten Bedingungen sind als Schwachstellen der VDP (Version 2007) als Zielplattform identifiziert. Um diese zu beseitigen, wird in Kapitel 5.4.1 eine Lösung ausgearbeitet.

## 5.4 Technische Umsetzung

In der technischen Umsetzung werden zuerst Lösungen für die entdeckten Schwachstellen der VDP entwickelt und vorgestellt (Kapitel 5.4.1). Anschließend wird die grundlegende Datenstruktur der RD-Datensätze auf der VDP realisiert (Kapitel 5.4.2). Anhand eines Beispiels wird die Datenstruktur der RD-Datensätze erläutert, die von konkreten Anwendungen plattformunabhängig angeboten werden sollen (Kapitel 5.4.3). Danach werden funktionale Hilfsmittel (Kapitel 5.4.4) und die programmiertechnischen Schnittstellen für die Anwendungen (Kapitel 5.4.5) vorgestellt.

### 5.4.1 Anpassung der Zielplattform

Im Kapitel 5.3 wurden einige Schwächen identifiziert, welche die Umsetzung des RD-Konzepts auf einer VDP 2007 erschweren. Sie werden während der Entwicklung der technischen Konzeption analysiert. Tabelle 5-4 zeigt, welche Lösungen jeweils erarbeitet werden.

Schwäche	Lösung
Manipulation innerhalb einer Textur (Image) ist nicht möglich.	<p>Das Problem einer Manipulation innerhalb einer Textur betrifft vor allem die bewegten Bilder. Konkret gibt es zwei Probleme.</p> <ul style="list-style-type: none"> <li>• Zum Ersten soll das Image der Textur aus Gründen der Effizienz inkrementell abgeändert werden können. Diese technische Verbesserung wird während der Entwicklung des RD-Konzepts vom Autor durchgeführt. In die Verbesserung involviert sind die Materialien, die Übertragung, die Synchronisation des Clusters und der Renderer.</li> <li>• Zum Zweiten wird eine Möglichkeit benötigt, um Textur-Koordinaten für einzelne Vertices zu bestimmen. Insbesondere ist dies wichtig, wenn Höhe und Breite von Bildern keinen Zweierpotenzen entsprechen (siehe Erläuterungen in Kapitel 4.9.2). Die VDP bietet hierfür keine allgemeine Lösung. Im Rahmen des RD-Konzepts wird eine spezielle Lösung implementiert, um eine Box mit individuell bestimmten Textur-Koordinaten erstellen zu können.</li> </ul>
Identifizieren einzelner Knoten im Szenegraphen durch externen, eindeutigen, laufzeitunabhängigen Schlüssel	Die links aufgeführten drei Schwachstellen beziehen sich alle auf den Ablauf der Eventübertragung bzw. der Eventbearbeitung von den Eingabegeräten über die Szenenobjekte bis zu den logischen RD-Datensätzen.

Schwäche	Lösung
Identifizieren logischer Strukturen durch Szenegraph-Knoten	Während der Entwicklung des RD-Konzepts wird die VDP 2008 vorgestellt und darin eine grundlegende Basis für die Attributierung geschaffen. Somit ist es auch möglich, einen laufzeitunabhängigen Schlüssel als Attribut für ein Szeneobjekt hinzuzufügen.  Im RD-Konzept werden diese Schlüssel gleich behandelt wie der Schlüssel des RD-Datensatzes, der außerhalb der Laufzeit im XML-Format gespeichert wird.
Optional soll das zugrundeliegende VR-System Attributierungsprozesse des Szenegraphen unterstützen.	

**Tabelle 5-4: Schwachstellen der VDP 2007 und deren Behandlung in der technischen Konzeption**

Mit den in der Tabelle beschriebenen Maßnahmen werden die technischen Hindernisse aus dem Weg geräumt. Die VR-Plattform VDP kann um das RD-Konzept erweitert werden.

#### 5.4.2 Basisstruktur der Richdata-Datensätze

Die zugrundeliegende Klasse der RD-Datensätze enthält zwei Eigenschaften, die in Kapitel 4.4.2 beschrieben sind. Zum einen muss jeder RD-Datensatz eine Möglichkeit anbieten, sich selbst in einem Text-Format zu speichern und sich aus einem textbasierten Format konstruieren zu lassen. Diese integrierte Funktionalität stellt sicher, dass die Richdata sich selbst oder den eigenen Zustand zum Zweck der Datenkommunikation eingeben oder ausgeben kann. In der Implementierung wird das Text-Format *Extensible Markup Language* (XML) festgelegt. XML ist ein etabliertes Standardformat zum Austausch von Daten zwischen Computersystemen [W3C-XML]. Ähnlich wie bei einem Adapter werden zwei Klassen nach dem gleichen Muster für die Ein- und Ausgabe von XML eingerichtet. Beide greifen auf die Methode eines XML-Parser zu (siehe Klassendiagramm in Abbildung 5-2). Dieser XML-Parser ist plattformunabhängig.

Zum anderen hat jeder RD-Datensatz einen laufzeitunabhängigen Schlüssel, der im Diagramm unten als *StaticKey* gekennzeichnet ist. Dieser Schlüssel kann auch einem Objekt im Szenegraphen als Attribut hinzugefügt werden. Der RD-Datensatz lässt sich bei der Erzeugung des *IDOCaterRichDataManagers* registrieren. Die Manager-Klasse wird nach dem Muster von *Singleton* erstellt, womit jedes Objekt auf den RD-Datensatz anhand seines *StaticKey* zugreifen kann.

Eine Richdata lässt sich auch direkt aus *IDORichDataBase* ableiten. In diesem Fall wird ein trivialer RD-Datensatz erzeugt. Wie in Kapitel 4.4.2 erläutert, kann sich der Zustand des trivialen Datensatzes nach der Erzeugung nicht mehr ändern. Er hat auch keinen *StaticKey* und muss daher bei seiner Erzeugung auch nicht beim *RichDataManager* registriert werden.

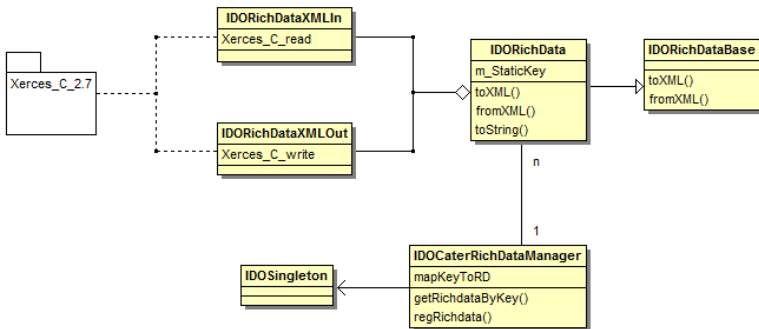


Abbildung 5-2: Klassendiagramm für die Basisklasse des Richdata-Datensatzes

### 5.4.3 Richdata-Datensätze auf VR-Plattformen

Nach dem RD-Konzept sollen VR-Plattformen nicht nur die Basistypen von Informationsformen wie 3D-Text oder Audio unterstützen, sondern wie in Kapitel 4.4 erläutert auch komplexe Templates in einer RD-Bibliothek anbieten, damit der Entwickler effizient neue Anwendungen erstellen kann. Die Bibliothek lässt sich für verschiedene VR-Projekte nutzen, kann aber auch angereichert werden, indem neue RD-Datensätze als Muster zur Bibliothek hinzugefügt werden.

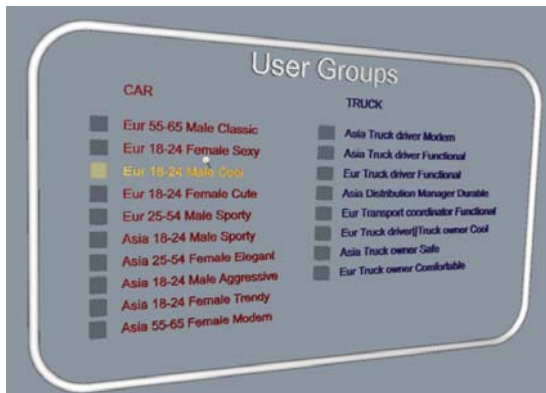


Abbildung 5-3: 3D-Text-Tabelle für die Fahrzeugkonfiguration

Bei der Fahrzeugkonfiguration wird z. B. eine Text-Tabelle benötigt. Diese wird aufgrund von Basistypen implementiert. Das in der Beispielanwendung verwendete Layout zeigt Abbildung 5-3. Wie in Abbildung 5-4 illustriert, bietet sie diese Umsetzung auch als Beispieldatensatz für weitere Spezifikationen an.

Die Klasse *IDORdTextTable* ist im Prinzip eine Matrix von *IDORdText3D*, einem der grundlegenden Basis-Typen. Mit der Ableitungskette von *IDORdTable* über *IDORdTextTable* bis hin zu *IDORdDialogWindow* wird eine Reihe von spezifischen RD-Datensätzen erzeugt. *IDORdTable* ist eine Tabelle mit beliebigen Inhalten. Bei *IDORdTextTable* wird der Inhalt jeder Zelle im 3D-Text-Format festgelegt. In *IDORdDialogWindow* wird einer oder mehreren Zellen eine besondere Form und Bedeutung zugeordnet. Werden bestimmte Zellen vom Benutzer selektiert, wird ein entsprechendes Ereignis ausgelöst.

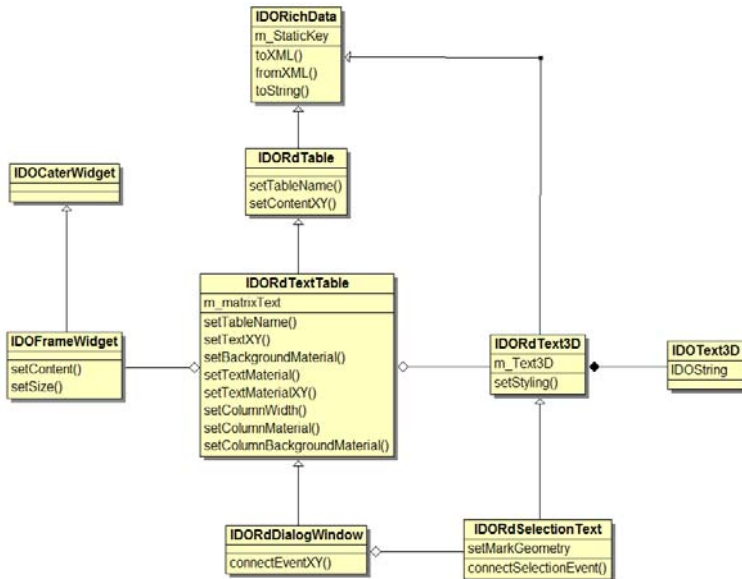


Abbildung 5-4: Klassendiagramm für Beispieldatensätze auf VR-Plattformen

*IDORdTextTable* bietet dem Benutzer viele Möglichkeiten, um individuelle Details bezüglich Textinhalt, Größe, Material, Hintergrund usw. für verschiedene Situationen zu bestimmen. Dabei spielt nicht nur der Inhalt eine Rolle, sondern auch das Aussehen, das normalerweise Teil der RD-Repräsentation ist.

Aus programmierertechnischer Sicht sind bei der Strukturierung folgende Punkte anzumerken:

- Im Muster-Datensatz kommen sowohl Kompositionen als auch Vererbungen vor. Eine lange Ableitungskette ist möglich, wenn viele spezifische funktionale Anforderungen implementiert und zur Bibliothek hinzugefügt werden.



- Die RD-Repräsentation wird in der Regel mit dem RD-Inhalt erzeugt. Damit wird erreicht, dass das Layout und der Inhalt so angezeigt werden, wie dies im Muster vorgesehen ist.
- Da es sich um ein Muster handelt, werden im Prinzip alle Einstellungsmöglichkeiten implementiert. Die Implementierung sowie die Verbindung zu den Repräsentationen müssen robust sein.

#### 5.4.4 Hilfsmethoden

Hilfsmethoden für das RD-Konzept, die allgemein einsetzbar sind, können auch während der Entwicklung von Anwendungen implementiert und angereichert werden. In der Tabelle 5-5 werden einige davon aufgelistet.

Die Hilfsmethoden werden auf Basis von VDP entwickelt und sind für eine Erweiterung bzw. Anreicherung offen.

Kategorie der Hilfsmethode	Beschreibung
Einordnung in ein immersives Layout	Mit den Methoden wird die Position des einzelnen Layout-Elements in vorgegebener Skalierung, Orientierung und dem entsprechenden Einordnungsverhältnis für lokale sowie globale Koordinationssysteme berechnet.
Transformationsstrategie (siehe auch Kapitel 4.8.3)	Mit den Methoden werden die Transformationsmatrizes einer Gruppe von Layout-Elementen in vorgegebenen Mustern berechnet. Diese Berechnung kann auch eingesetzt werden, um die Transformation eines einzelnen Layout-Elements in zeitlicher Reihenfolge zu bestimmen, z. B. für eine Animation.
Timing-Strategie (siehe auch Kapitel 4.8.3)	Mit den Methoden werden Ereignisse in vorgegebener zeitlicher Reihenfolge erzeugt.
Namenserstellung	Methode, um die Information aus dem externen Prozess (z. B. des Schlüssels) in der Namensgebung des Objekts im Szenegraphen automatisch sichtbar zu machen. Damit lässt sich z. B. kennzeichnen, bis zu welchem Knoten oder zu welcher Stufe der Szenegraph für einen Prozess relevant ist.
Automatische Optimierung der inneren Struktur der Geometrie	Methode, um die Struktur unterhalb der Stufe, die für den externen Prozess relevant ist, zu optimieren.

Tabelle 5-5: Kategorien von Hilfsmethoden

#### 5.4.5 Programmierschnittstelle für VR-Anwendungen

Die VDP besitzt eine programmiertechnische Schnittstelle für die Anwendungsentwicklung. Über diese Schnittstelle werden die einzelnen Anwendungen als Plugin zu dem System hinzugefügt. Da die für das RD-Konzept notwendigen Erweiterungen bereits in die VDP integriert sind, kann diese Schnittstelle problemlos zur VR-Integration nach dem RD-Konzept eingesetzt werden. Im nächsten Kapitel wird die Entwicklung von Anwendungen am Beispiel der Fahrzeugkonfiguration erläutert.

## 6 Anwendungsentwicklung am Beispiel der Fahrzeugkonfiguration

### 6.1 Rahmenbedingungen der Fahrzeugkonfiguration

#### 6.1.1 Überblick und Komponenten des Projektkontexts

Mit dem RD-Konzept soll ermöglicht werden, die VR-Anwendung effizient mit existierenden Prozessen zu integrieren und effektive immersive Benutzerschnittstellen erstellen zu können. Diese Ziele werden am Beispiel der Fahrzeugkonfiguration im Rahmen des Projekts *Computerised Automotive Technology Reconfiguration System for Mass Customization* (CATER) verifiziert.

Das CATER-Forschungsprojekt entwickelt prototypisch Werkzeuge für die kundenindividualisierte Massenfertigung. Wie die Abbildung 6-1 zeigt, bemüht sich das Projekt um Optimierungen unterschiedlicher Bereiche vom Kundenbeziehungsmanagement (Customer Engineering) über die Produktion und die Verfahrenstechnik bis hin zu Logistik und Vertrieb [Jiao 2007]. Eine wichtige Komponente des Projekts ist die VR-Benutzungsschnittstelle für das System-Engineering-Team. Die Mitarbeiter des Teams sollen in der VR auf unterschiedlichste Informationen von der Verbraucherstatistik bis zu den Designoptionen und den technischen Daten von Fahrzeugkomponenten zugreifen können, um Entscheidungen über Fahrzeugkonfigurationen im Prozess der individuellen Massenfertigung zu treffen. Da die Entwicklung, Produktion und Logistik heutzutage auf der ganzen Welt verteilt sind, bietet die VR-Technologie eine optimale Plattform, um Hindernisse in der Kommunikation wie kulturelle Unterschiede, verschiedener technischer Wissensstand sowie Sprachbarrieren zu überwinden.

Der Anwendungsfall des VR-Konfigurators umfasst drei Module, die im Rahmen des Projekts CATER entwickelt werden:

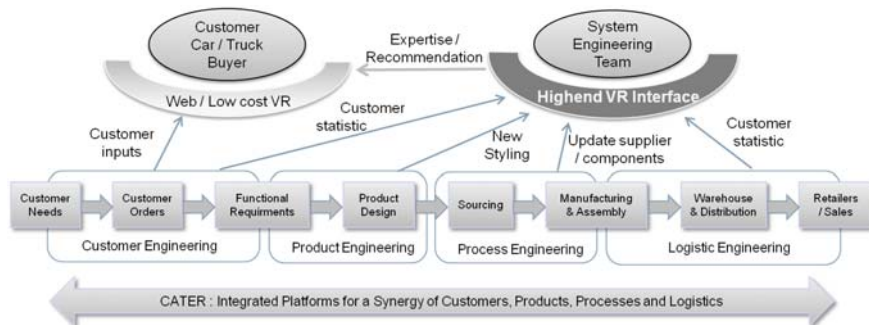


Abbildung 6-1: Anspruchsgruppen und Benutzungsoberfläche im Projekt CATER

Das erste Modul ist ein sogenanntes **Citarasa**-System [Lim 2007]. Im Malaiischen steht das Wort *Citarasa* für emotionale Bedürfnisse des Kunden. Die Emotionen können stark vom Benutzerprofil wie Kultur, Religion, Geschlecht oder Alter abhängig sein. Die Vorstellungen, wie ein *ästhetisches Auto* aussieht, können bei einem männlichen Europäer und einer Frau aus einem arabischen Land weit auseinander liegen. Das Citarasa-System dient dazu, die emotionalen Anforderungen der Kunden in eine funktionale Sprache zu übersetzen.

Das zweite Modul mit dem Name **DIYD** (do it yourself database) hilft dem Benutzer, eine eigene funktionale Auswahl durchzuführen. Die individuelle Massenfertigung im Fahrzeugbau bietet den Kunden die Möglichkeit, aus einer Vielzahl technischer Optionen auszuwählen. Er muss dabei aber komplexe Abhängigkeiten zwischen den Technikoptionen beachten. Ein Verbraucher ohne technische Vorkenntnisse kann die große Auswahl daher kaum nutzen. DIYD bietet dem Kunden konsolidierte Konfigurationen und Auswahlmöglichkeiten. Außerdem ist DIYD auch die zentrale Schnittstelle für die Web- bzw. VR-Benutzungsoberflächen, damit die Entscheidung des Benutzers von der funktionalen Auswahl in eine technisch korrekte Konfiguration umgewandelt und direkt an die Produktion weitergeleitet werden kann.

Das dritte Modul, **MaRK** (Middleware for Aggregated Knowledge), ist eine Schnittstelle zwischen den Produkt- und Produktionsdaten in der Automobilindustrie. MaRK ist in der Lage, mit den verschiedenen existierenden Informationssystemen von OEM bis hin zu den Zulieferern zu kommunizieren. Damit können individuelle technische Anforderungen in der Produktion besser und schneller berücksichtigt werden.

### 6.1.2 Anwendungsfall VR-Konfiguration

Im Rahmen des Aufbaus des CATER-Systems wird eine Benutzungsoberfläche für Fahrzeugkäufer entwickelt. Mit deren Hilfe kann der Käufer eigene Wünsche in das CATER-System eingeben und sich die entsprechenden Ergebnisse am virtuellen Fahrzeug ansehen. Diese Wünsche werden im Citarasa-System in Form von emotionalen Anforderungen erfasst und in funktionale Attribute übersetzt, die anschließend an das DIYD-Modul weitergegeben werden. DIYD gibt eine entsprechend der Wünsche erstellte Konfiguration zurück. Damit solch ein virtueller Vorschlag möglichst exakt die Erwartungen des einzelnen Fahrzeugkäufers erfüllen kann, müssen die OEMs mitentscheiden, wie diese „passenden Konfigurationen“ zusammengestellt werden. Die individuelle Massenfertigung bietet eine große Variantenvielfalt, ist aber gleichzeitig durch eine große Veränderungsdynamik gekennzeichnet. Die möglichen Fahrzeugkonfigurationen müssen daher periodisch immer wieder aktualisiert werden. Neue Technologien müssen dabei ebenso mitberücksichtigt werden wie sich ändernde Modetrends. Ein System-Engineering-Team sollte diese Aufgabe ausführen. Dies ist der Anwendungsfall für die VR-Konfiguration, welche auf Basis des RD-Konzepts entwickelt und implementiert wird.

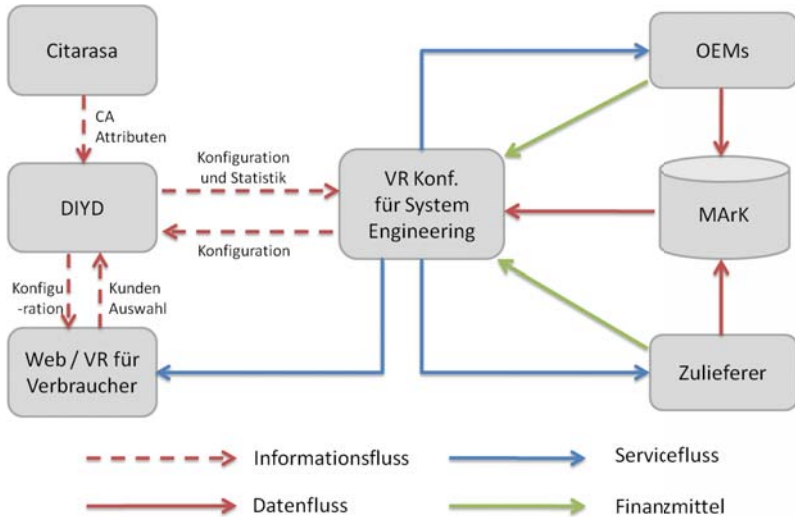


Abbildung 6-2: VR-Konfigurator und im Kontext des Projekts CATER

Abbildung 6-2 illustriert die Beziehung zwischen dem VR-Konfigurator, den involvierten Akteuren und den drei CATER-Modulen. Input für den Konfigurator liefert zum einen das Modul MArK in Form von Geometrie-Datensätzen und den zugehörigen technischen Beschreibungen. Zum anderen liefern die Module Citarasa und DIYD Informationen über die bisherigen Konfigurationen, über Benutzerprofile sowie aktuelle Statistiken. Wurde eine für eine bestimmte Verbraucherguppe typische Konfiguration vom System-Engineering-Team fertig aktualisiert, werden sie als neue Vorschlagsbasis an das DIYD-Modul weitergegeben und damit als aktualisierte Auswahlmöglichkeit für Fahrzeugkäufer freigegeben. Der im Rahmen der vorliegenden Arbeit entwickelte VR-Konfigurator fungiert als Schnittstelle zwischen Fahrzeugherstellern, Zulieferern und dem Fahrzeugvertrieb.

In der Realität ist das CATER-System ebenso wie der moderne Massenfertigungsprozess in der Automobilindustrie über die ganze Welt verteilt. Die Server der Module werden alle lokal bei dem Servicepartner aufgestellt. Sie bieten alle eigene Zugriffsschnittstellen in Form von Webservices. Das CATER-System ist damit weltweit über das Internet erreichbar. Mit anderen CATER-Modulen über ein Netzwerk zu kommunizieren ist auch für die VR-Schnittstelle eine grundlegende Anforderung.

Beim Design der VR-Schnittstelle wird davon ausgegangen, dass die avisierte Benutzergruppe, das System-Engineering-Team, aus professionellen Nutzern mit Erfahrung in VR besteht und diese langfristig mit der immersiven Umgebung arbeiten werden.

Aufgrund seiner Anforderungen ist die CATER-Fahrzeugkonfiguration gut als Beispielanwendung für das RD-Konzept geeignet. Im Rahmen des Anwendungsfalls sind insbesondere folgende Punkte, die mit den Zielen des RD-Konzepts übereinstimmen, von Bedeutung:

- Die System-Integration entlang der Wertschöpfungskette ist ein zentraler Bestandteil der Anwendung.
- Sowohl Geometrie-Daten als auch typische RD-Informationsformen, wie Statistiken, Modetrends und technische Beschreibungen, spielen eine wichtige Rolle und sollen in der immersiven Umgebung repräsentiert werden.
- Das System-Engineering-Team benötigt in der VR nicht nur perfekte Fahrzeugmodelle, sondern auch eine professionelle Benutzerschnittstelle, um in der immersiven Umgebung effizient arbeiten zu können.
- Die involvierten Module sind auf verschiedene Orte verteilt. Die Anforderungen an die Datenkommunikation in der Laufzeit sind demnach entsprechend hoch.

## 6.2 Konzeption einer Anwendung für die Fahrzeugkonfiguration

Der CATER-Fahrzeugkonfigurator wird als Modul in die VDP implementiert, um das RD-Konzept zu verifizieren. Wie Abbildung 6-3 illustriert, ist das Modul VR-Fahrzeugkonfigurator gegenüber der VDP ein Plug-In, basierend auf einer Programmierungsschnittstelle für die Anwendung, und andererseits die VR-Benutzungs Oberfläche für industrielle Benutzergruppen in CATER.

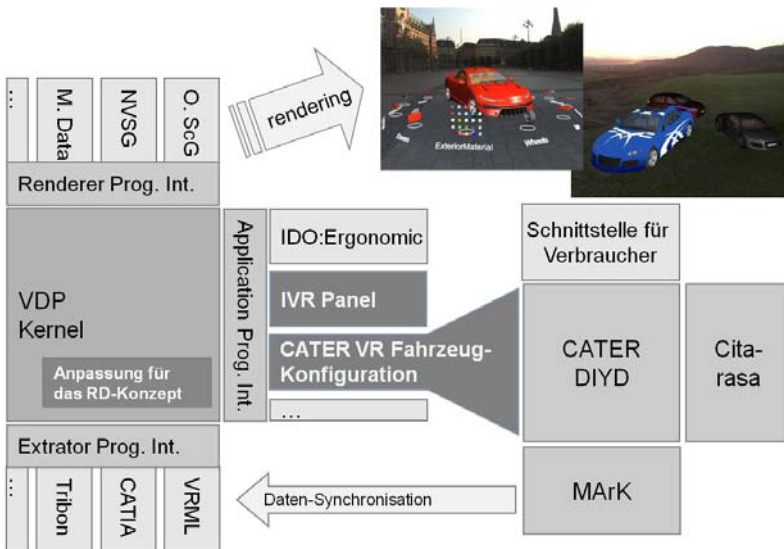


Abbildung 6-3: VR-Benutzungs Oberfläche für die Fahrzeugkonfiguration als VDP-Modul

Die Geometrie-Datensätze für die Fahrzeuge werden über MARK zur Verfügung gestellt. Die Informationen über die Konfiguration und weitere Metadaten liefert das DIYD-Modul. Die Darstellungen aus beiden Quellen werden im Szenegraphen zusammengeführt, der strukturiert im VDP-Kernel gehalten und durch einen Renderer in der immersiven Umgebung sichtbar gemacht wird.

Das IVR-Panel, das in Kapitel 4.9.2 vorgestellt wird, ist eine notwendige Ergänzung der VDP, um externe Ressourcen wie z. B. eine PowerPoint-Präsentation in die virtuelle Umgebung integrieren zu können. Das IVR-Panel wird als separates Modul implementiert, denn es bietet eine allgemeine Methode, die auch für weitere Anwendungen genutzt werden kann. Es ermöglicht allgemein die Anzeige eines Remote-Computers und bewegter Bilder.

Kernelement der Implementierung ist das Modul zur Fahrzeugkonfiguration. Es wird nach dem Prinzip des RD-Konzepts in mehrere Komponenten zerlegt. Eine Übersicht zeigt Abbildung 6-4.

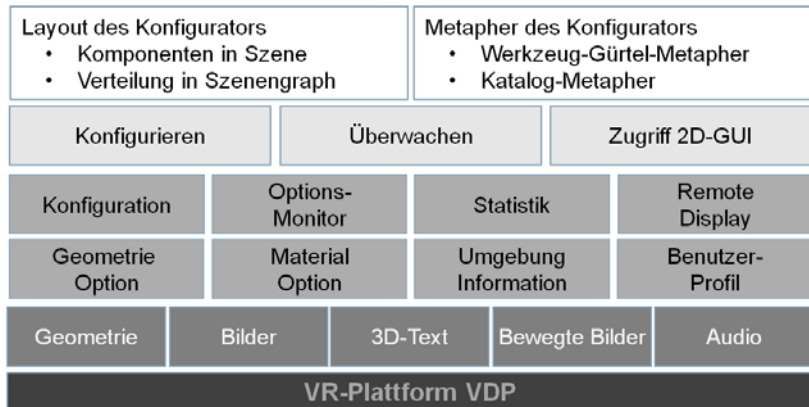


Abbildung 6-4: Komponenten des Moduls für die VR-Konfiguration

Die VDP bietet die notwendigen Basistypen von Repräsentationsformen als RD-Datensatz (siehe Kapitel 1). Dabei bestehen die RD-Datensätze entweder aus einer direkten Zusammenstellung der Basistypen (Bottom-up) oder aus der Abstraktion des Konfigurationsprozesses (Top-Down). Die funktionalen Teile bestehen aus Interaktionsmethoden und den dazugehörigen Komponenten. Dazu zählen beispielsweise das Konfigurieren, das Überwachen oder der Zugriff auf 2D-GUI durch das IVR-Panel. Das Layout des Konfigurators wird in einem iterativen Verfahren unter Einbeziehung des Feedbacks von Partnern und Kunden designed. Layout und Interaktionsmethode werden schließlich aufeinander abgestimmt und zusammen als Einheit in das VR-System integriert.

### 6.3 Richdata-Datensätze aus Abstraktion der Anwendungslogik

RD-Datensätze können von der VR-Plattform oder aus einer Anwendungslogik, z. B. der Fahrzeugkonfiguration, heraus erzeugt und instanziiert werden. Die Methodik wird in Kapitel 4.5 erläutert. In diesem Kapitel werden nicht alle Details der Erzeugung behandelt, sondern nur die Basisstruktur für die Implementierung von RD-Datensätzen aufgezeigt, und je ein typisches Beispiel für die beiden oben genannten Fälle der Erzeugung näher beleuchtet.

Wie im RD-Konzept vorgesehen (Kapitel 4.5.2), wird der Anwendungsfall dahingehend analysiert, wie sich die Datenstruktur des zu integrierenden Prozesses in RD-Datensätze umwandeln lässt. Zum Ablauf siehe Abbildung 4.7.

Die Konfiguration ist die zentrale Datenstruktur des Anwendungsfalls. Sie besteht aus Kombinationen von zwei verschiedenen Optionen:

- Die Geometrie-Option. Sie stellt die Austauschbarkeit der unterschiedlichen Fahrzeug-Komponenten dar.
- Die Material-Option. Sie stellt die Austauschbarkeit des Oberflächenmaterials bzw. von Materialkombinationen für Exterieur und Interieur dar.

Die Funktionen der beiden Optionstypen sind auf den ersten Blick ähnlich, unterscheiden sich aber in den Optionswerten und der Wirkung. Deshalb werden die beiden Optionswerte als eigenständige RD-Datensätze abstrahiert, obwohl sie beide das gleiche *OptionInterface* realisieren (siehe Abbildung 6-5).

Optionswerte für Fahrzeugkomponenten sind im Prinzip die Modelldaten der Komponenten und ihre Sichtbarkeit, die durch den Benutzer mittels Befehl ein- und ausgeschaltet werden. Optional kann ein Optionswert mehr als einen Modelldatensatz enthalten, um unterschiedliche Detaillevels darzustellen.

Optionen für Interieur- oder Exterieur-Material sind komplexer, denn sie speichern nicht nur lokale Materialien in den Optionswerten, sondern auch die möglichen Bezugsobjekte. Deshalb wird der RD-Datentyp *IDORdMatchingPath* eingeführt. Dieser Datentyp muss während der Laufzeit seinen Zustand nicht ändern. Er wird daher direkt aus der *IDORichDataBase* abgeleitet. Zusätzlich muss die Änderung der Umgebung auch innerhalb des reflektierenden Materials bekannt sein, damit das Exterieur-Material die Umgebung reflektieren kann.

Die Kette von der Konfiguration über die Optionen bis zu den Optionswerten zeigt eine typische Verschachtelung von RD-Datensätzen. Die Richdata stellen auch den Zustand des Konfigurators dar. Aus programmiertechnischer Sicht lassen sich bezüglich der Strukturierung folgende Punkte anmerken:

- Die Beziehungen zwischen den RD-Datensätzen sind überwiegend Kompositionen.



- Ist eine Ableitung notwendig, sollte sie nach Möglichkeit auf rein virtuellen Schnittstellen basieren.
- Die Repräsentation soll vom Inhalt getrennt erzeugt werden. Eine Ausnahme ist dann gegeben, wenn die Repräsentation aus Robustheitsüberlegungen ausschließlich an einen bestimmten Inhalt gebunden werden soll (siehe Kapitel 4.7).

Weitere RD-Datensätze aus dem CATER-Prozess sind z. B. Statistiken und Benutzerprofile. Ihre Erstellung erfolgt analog zur Konfiguration.

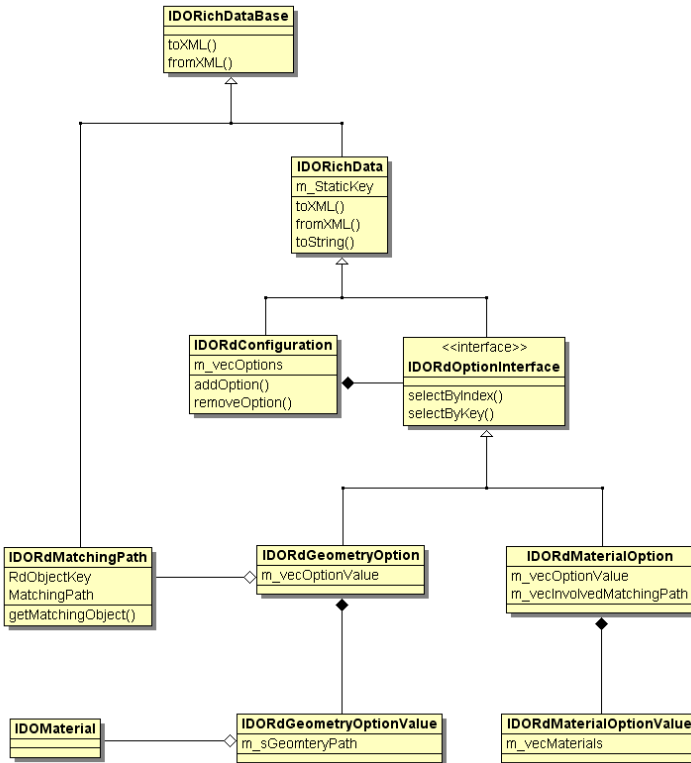


Abbildung 6-5: Klassendiagramm für RD-Datensätze zur Fahrzeugkonfiguration

## 6.4 Immersives Layout und Richdata-Repräsentation des Konfigurators

### 6.4.1 Layout des Fahrzeugkonfigurators

Das Layout des Fahrzeugkonfigurators besteht aus vier Komponenten: die Umgebung, das Fahrzeug, die Optionen und der Informationsbereich (siehe Abbildung 6-6).

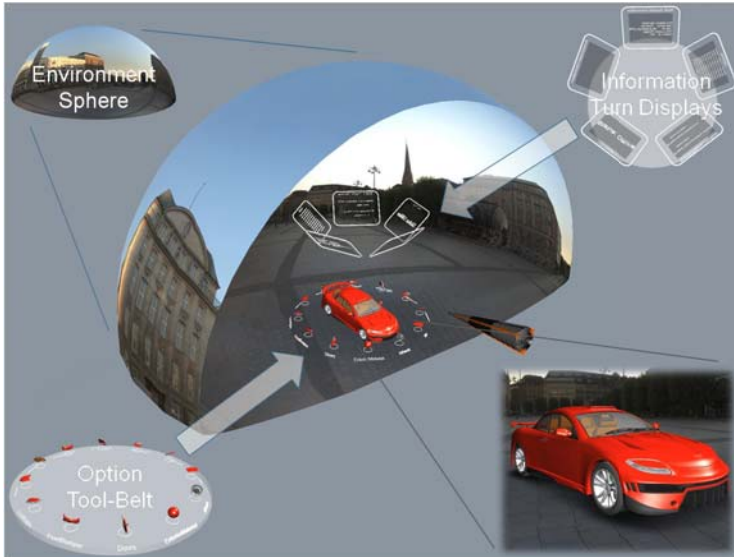


Abbildung 6-6: Gesamtlayout und Komponenten der Fahrzeugkonfigurator

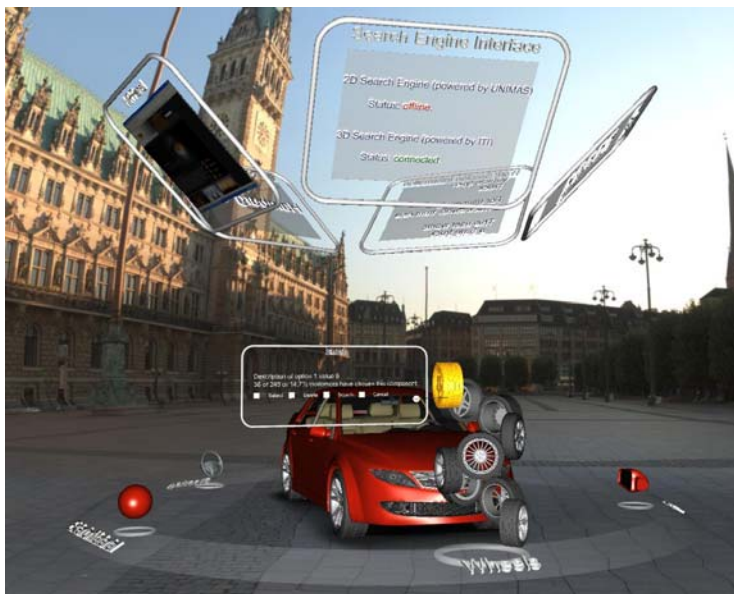



Abbildung 6-7: Layout des Fahrzeug-Konfigurator aus Benutzersicht





Das Layout stellt das Fahrzeug in die Mitte der Umgebung. 3D-Symbole für die Optionen sind rund um das Fahrzeug positioniert. Die Symbole haben Standardgröße, einen Ring unter sich und besitzen eine Textbeschilderung. Die Optionssymbole bilden wiederum einen Ring. Die Größe des Rings soll so bestimmt werden, dass der Benutzer an diesem Ring stehend das ganze Fahrzeug beobachten kann. Der detaillierte Szenenaufbau wird hier nicht nochmals erklärt, da dieser in Kapitel 4.6.2 „Immersive Repräsentationsmuster“ als Beispiel vorgestellt wurde.

Die Umgebung wird in der Implementierung durch Mapping eines Panoramabilds auf einer Halbkugel realisiert. Die Halbkugel hat in der virtuellen Welt einen Durchmesser von 20 Metern und eine Höhe von 11 Metern. Die Größenordnung der Darstellung muss mit der realen Welt, in der das Foto aufgenommen wird (in Abbildung 6-6 vor dem Hamburger Rathaus), übereinstimmen. Die Textur-Koordinaten werden auf der Halbkugel so eingestellt, dass das Bild eine realistische Szene zeigt, wenn der Benutzer in die Mitte der Kugel steht. Das Panoramabild wird zusätzlich im Shader für das Exterieur des Fahrzeugs eingesetzt, um die Umgebung „richtig“ zu reflektieren.

Der Informationsbereich besteht aus fünf virtuellen Fenstern in gleichem Stil. Für die Implementierung wurde ein Fensterrahmen mit runden Ecken gewählt. Während die Fenstergröße in vertikaler und horizontaler Richtung beliebig verändert werden kann, bleiben die Ecken in ihren Größen unverändert. Die Fenster können einen beliebigen Inhalt repräsentieren. Der Inhalt wird durch Aufruf der Methode *setContent* (siehe auch die Klasse *IDOFramWidget* in Abbildung 5-7) hinzugefügt. Ein einstellbarer Parameter bestimmt, ob dabei die Fenstergröße automatisch an den Inhalt angepasst oder umgekehrt der Inhalt der Fenstergröße entsprechend angezeigt wird.

Die fünf Fenster bilden im Konfigurator einen Kreis (siehe Abbildung 6-7). Sie stehen so geneigt über dem Fahrzeug, dass der Benutzer sie angenehm sehen kann, wenn er vor den Optionssymbolen steht und in Richtung des Fahrzeugs blickt. Vier der fünf Fenster repräsentieren Text-Informationen und im fünften werden über das IVR-Panel bewegte Bilder angezeigt (Erläuterungen dazu siehe Kapitel 4.9.2). Einen Überblick über die Inhalte der Fenster des Informationsbereichs gibt Tabelle 6-1.

Fenstername	Inhalt	Beispiel im Layout
Monitor für Optionen	Ein Monitor zeigt die Optionen an, die in der Szene geladen werden. Wenn ein Optionswert durch Benutzerinteraktion geändert wird, wird die aktuelle Änderung in diesem Monitor angezeigt.	

Fenstername	Inhalt	Beispiel im Layout
Umgebung	Ein Fenster beschreibt die Umgebung, in der sich der Benutzer in der virtuellen Welt aktuell befindet. Wird die Umgebung geändert, sollte sich die Fensteransicht entsprechend anpassen.	
Beschreibung der Arbeitsszene	Informationen über das Fahrzeug und die Profile und Anforderungen der Zielgruppe. Während der Laufzeit ändern sich diese Daten nicht.	
Monitor für Remote Server	Informationen über Datenbanken und deren aktuelle Zustände. Der letzte Zustand wird notiert. Zudem ist es möglich, dass der Zustand in regelmäßigen Abständen überprüft wird.	
IVR-Panel	Remote-Display für einen erreichbaren Computer im Netzwerk.	

**Tabelle 6-1: Inhalte in den Informationstafeln des Konfigurators**

## 6.4.2 Verteilung der Richdata-Repräsentationen im Szenegraphen

Bei der Entscheidung, wie die Elemente einer VR-Schnittstelle im baumartigen Szenegraphen einbetten werden sollen, müssen verschiedene Faktoren berücksichtigt werden. Besonders wichtig ist dabei:

- Die in der Interaktion zusammengehörenden Objekte sollen möglichst unter einem gemeinsamen Vater-Knoten hängen. Damit wird erreicht, dass eine Transformation immer nur an diesem Knoten durchgeführt werden muss.

- Die logisch zusammengehörenden Objekte sollen möglichst unter einem Knoten hängen, damit Änderungen lokal begrenzt bleiben.
- Der Szenegraph soll eine möglichst ausbalancierte Baumstruktur haben, damit die Render-Performance nicht unnötig reduziert wird.

Für den Fall, dass im konkreten Anwendungsfall diese Faktoren miteinander in Konflikt stehen, muss der Anwendungsentwickler Prioritäten setzen. Im Fahrzeugkonfigurator wird die Priorität entsprechend der Reihenfolge obiger Aufzählung festgesetzt. Daher werden die Umgebung, das Fahrzeugmodell, die Optionen und die Informationsbereiche nach dem ersten Prinzip je einem Knoten zugeordnet. Diese Zuordnung berücksichtigt hauptsächlich die Interaktion, z. B. die mögliche Rotation des Fahrzeugmodells, des Rings der Optionssymbole oder der Fenster auf horizontaler Ebene.

Logische Abhängigkeiten konnten teilweise nicht berücksichtigt werden. Dazu zählt zum Beispiel der Umstand, dass die Optionssymbole (Options-Marken), die Komponenten des Fahrzeugmodells und die Text-Beschreibung im Monitor zusammengehören und zusammen geändert werden müssen. In diesem Zusammenhang greifen jedoch die Vorteile des RD-Konzepts. Werden die Optionen im RD-Datensatz in einer logischen Beziehung zueinander dargestellt, lassen sich eigene Repräsentationen in den drei Unterbäumen (Fahrzeug, Option-Marken und Optionsmonitor) konsistent mit der Benutzer-Interaktion ändern.

Der dritte Faktor bezieht sich auf die Optimierung der Rendering-Performance. Wenn alle Knoten eines Baums nicht einzeln nachgefragt werden müssen, kann dieser Baum in einer flachen Struktur zusammengefasst werden. Der Fahrzeugkonfigurator wurde so angelegt, dass alle Fahrzeugkomponenten bzw. deren Options-Marken eine flache Baumstruktur besitzen. Die Blätter des flachen Baums werden nach ihrem Material sortiert.

## 6.5 Interaktion

### 6.5.1 Werkzeuggürtel-Metapher

Die Werkzeuggürtel-Metapher stellt die Beziehung zwischen dem Fahrzeugmodell und den zugehörigen Optionen her. Wie in der Abbildung 6-6 ersichtlich, werden die Optionen eines Fahrzeugs räumlich in einem Kreis um das Fahrzeugmodell herum positioniert. Steht der Benutzer an einem der Optionssymbole, kann er das Fahrzeug in angemessener Distanz beobachten und gleichzeitig Aktionen der jeweiligen Option durchführen. Das Ergebnis seiner Aktion, etwa den Austausch von Komponenten oder Materialien, sieht der Benutzer unmittelbar am virtuellen Fahrzeugmodell. Der Umgang mit den Optionen erinnert an einen Werkzeuggürtel. Je nach Arbeitsaufgabe wird ein entsprechendes Werkzeug aus dem Gürtel herausgeholt, um Fahrzeugkomponenten zu montieren. Vor dem Hintergrund dieser Analogie wird der Begriff der Werkzeuggürtel-Metapher eingeführt. Im Unterschied zu einem klassischen Werkzeuggürtel ist der Gürtel mit den

Optionen allerdings nicht um den Benutzer herum, sondern um das Objekt herum angeordnet.

Das Klassendiagramm der Abbildung 6-8 verdeutlicht die Bestandteile und die Beziehungen zwischen der Metapher, den betroffenen RD-Datensätzen und der CATER-Programmlogik. Die RD-Repräsentationen werden aus der Metapher erzeugt, mit den RD-Inhalten verbunden und anschließend entsprechend des Metapher-Layouts aufbereitet. Somit stellt die Metapher auch die Verbindung zwischen dem Szenegraphen und der Anwendungslogik her.

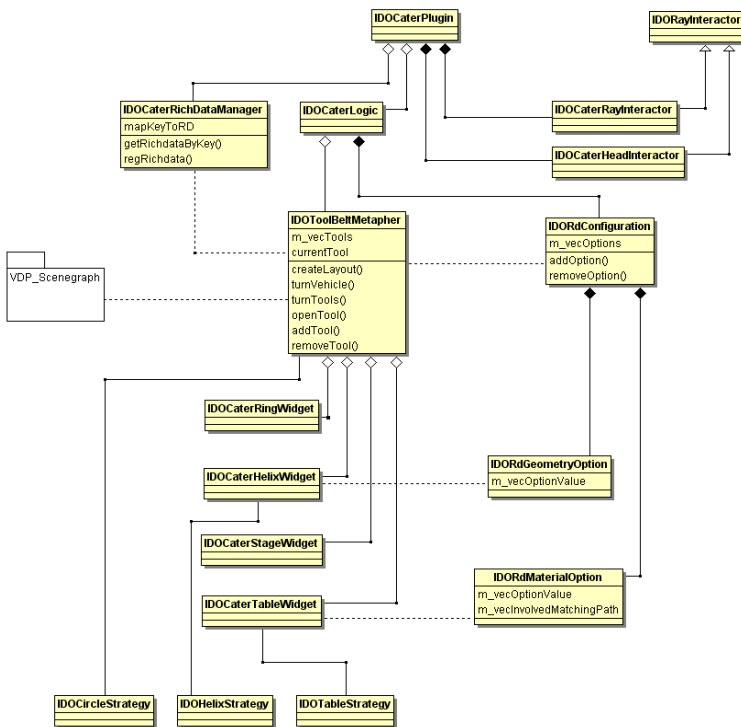


Abbildung 6-8: Klassendiagramm der Werkzeuggürtel-Metapher

Der technische Ablauf der Interaktionsmethode wird in der Zustandsmaschine in Abbildung 6-9 illustriert. Da die Fahrzeugkonfiguration das zentrale Objekt der Anwendung ist, stellt der Werkzeuggürtel die Haupt-Metapher dar. Sie wird nach Eingabe des Benutzerprofils automatisch von der Anwendung gestartet.

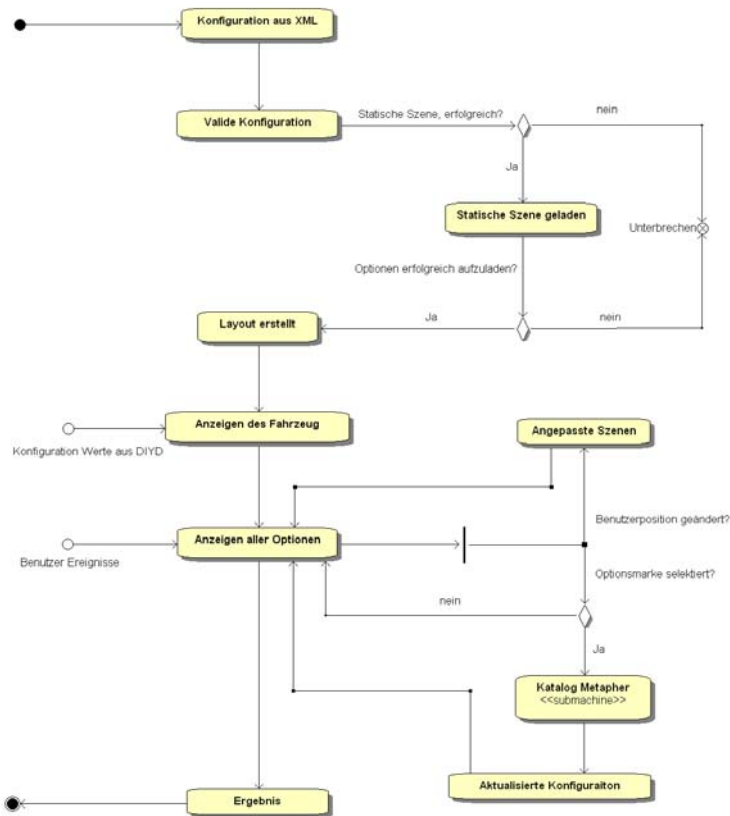


Abbildung 6-9: Zustandsmaschine der Werkzeuggürtel-Metapher

Die Konfiguration wird aus der zugehörigen XML-Datei geladen. Zuerst wird dabei geprüft, ob die Konfiguration erfolgreich initialisiert werden kann. Dies bedeutet, dass analysiert wird, ob die XML-Datei valide ist und ob alle Geometrie-Dateien für das Fahrzeug und alle möglichen Optionen zur Verfügung stehen. Im nächsten Schritt werden die Daten in den Arbeitsspeicher geladen, aber noch nicht angezeigt. Falls beim Ladevorgang ein Fehler auftritt, bricht die Erstellung der Metapher ab. Ist der Ladevorgang erfolgreich, werden die Daten nach Vorgabe des Layouts der Metapher aufbereitet. Die Metapher ist nun vollständig initialisiert.

Nach der Initialisierung fordert die Anwendung bei der DIYD-Datenbank Informationen dazu an, welche Optionswerte als Default-Werte in der Konfiguration ausgewählt wer-

den sollen. Danach werden das Fahrzeug und die Optionen in der VR-Umgebung angezeigt. Die Anwendung wartet auf die Benutzeraktionen.

Der Benutzerfokus und die Benutzerselektion werden im *CaterRayInteractor*, welcher den Standard *RayInteractor* der VDP erweitert, realisiert. Er leitet die Systemereignisse an die CATER-Anwendungslogik weiter. Dort wird das Ereignis interpretiert und die Zustandsänderung der Interaktionsmetapher bestimmt.

Durch die Eingabe werden zwei Schleifen in der Metapher gebildet. Eine sorgt für die ständige Anpassung an die Benutzernavigation oder den Benutzerfokus. Z. B. werden bestimmte 3D-Texte immer senkrecht zur Benutzerorientierung ausgerichtet. Die zweite Schleife überwacht, ob der Benutzer ein Optionssymbol selektiert. Wird eine Option ausgewählt, aktiviert dieses Ereignis die Katalog-Metapher. Diese ermöglicht es schließlich, eine Option zu modifizieren. Der Benutzer kann die Ergebnisse jederzeit speichern und die Arbeitsperiode beenden.

### 6.5.2 Katalog-Metapher

Sobald der Benutzer eines der Optionssymbole selektiert, wird ein 3D-Katalog mit den möglichen Optionswerten geöffnet. Ein Katalog kann in verschiedenen Layouts angezeigt werden, da kein Layout für jeden Anwendungsfall eine optimale Präsentation bieten kann. In der Implementierung wird standardmäßig eine Helix-Form für Geometrie-Optionen (Abbildung 6-10 links) und eine Tabellenform für Material-Optionen (Abbildung 6-10 rechts) eingesetzt.



Abbildung 6-10: Layout der Katalog-Metapher

Bei der Auswahl bzw. dem Design des Layouts werden folgenden Faktoren berücksichtigt:

- Anzahl der darzustellenden Optionswerte



Je mehr Optionswerte die einzelne Option hat, desto schwieriger ist es, dem Benutzer einen Überblick über die Auswahlmöglichkeiten zu geben. Die für die Fahrzeugkonfiguration zur Verfügung gestellten Testdatensätze haben maximal 9 Optionswerte bei den Geometrie-Optionen und 24 Optionswerte bei der Material-Option.

- **Detaillierungsgrad der 3D-Anzeige der Optionswerte**  
Die 3D-Anzeige der Geometrie-Optionen stellt die austauschbaren Fahrzeugkomponenten dar. Daher muss die Anzeige so detailliert sein, dass der Benutzer die Unterschiede zwischen den Komponenten deutlich erkennen kann. Für die Anzeige der Material-Optionen ist dagegen eine Darstellung mit einfacher Geometrie ausreichend. Denn das Zielobjekt der Auswahl ist in diesem Fall nicht die Geometrie, sondern das verwendete Material.
- **Räumliche Einschränkung beim immersiven Layout**  
Werden alle Optionswerte angezeigt, sollen sie dennoch so angeordnet sein, dass sie den Blick des Benutzers auf die einzelnen Optionen oder auf das Fahrzeugmodell stören oder sogar verhindern.

Im Rahmen der Implementierung wurde für die Darstellung und Auswahl der Geometrie-Optionswerte ein dynamisches Menü in Helix-Form entwickelt. Die Parameter der Helix-Form werden in der Tabelle 6-2 erläutert und in der Abbildung 6-11 illustriert.

Parameter	Beschreibung	Wert in der Anwendung
x	Anzahl der Optionswerte	Variable
d	Diameter des Optionswerts	$d = 0,5 \text{ Meter}$
$\alpha$	Polar-Offset zwischen nacheinander kommenden Optionswerten auf horizontaler Ebene	$\alpha = 2 * \pi / 5 \text{ Grad}$
t T	Vertikaler Offset zwischen nacheinander kommenden Optionswerten im Verhältnis zu d	$t = 0,5$ sodass absoluter Offset: $T = d * t = 0,25 \text{ Meter}$
D	Diameter des geöffneten Katalogs	$D = 2 * d = 1 \text{ Meter}$
H	Höhe des geöffneten Katalogs	$H = (x + 1) * T$

**Tabelle 6-2: Parameter der Helix-Form des Katalogs**

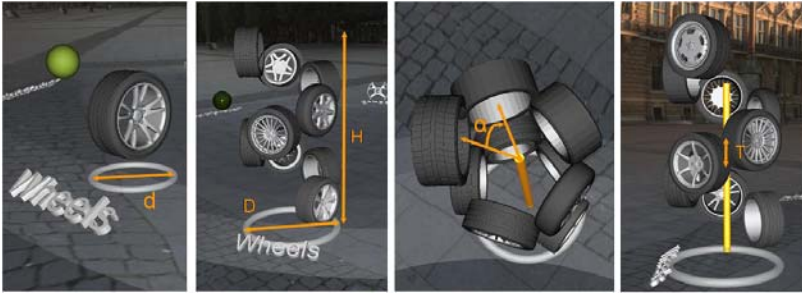


Abbildung 6-11: Helix-Layout für Katalog-Metapher und deren Parameter

Die Positionierung wird durch eine entsprechende Transformationsstrategie berechnet. Die Helix-Form bringt mehrere Vorteile mit sich:

- Sie nutzt den vertikalen Raum, um die Fahrzeugkomponenten anzuzeigen. Dies erzeugt eine kompakte Wirkung und benötigt nur wenig Platz auf der horizontalen Ebene.
- Die Wendelgeometrie ermöglicht durch die Verteilung sowohl auf der horizontalen Ebene ( $\alpha$ ) als auch längs der vertikalen Achse (T) eine kompakte Darstellung der Optionswerte.
- Die rotierende Animation versetzt den Benutzer in die Lage, alle Objekte zu beobachten, ohne sich zu bewegen.

Eine Einschränkung der Helix-Form besteht darin, dass die gesamte Höhe (H) nicht beliebig hoch sein darf. In der Regel sollte sie nicht viel höher als die Augenhöhe des Benutzers sein. Andernfalls muss der Benutzer nach oben schauen, um alle Elemente auf der Helix erfassen zu können, und verliert somit den Überblick über die Optionswerte. Im Fall des Fahrzeugkonfigurators bedeutet dies: Wird der Durchmesser des Optionswerts (d) auf 0,5 Meter und der vertikale Abstand (Offset) auf 0,25 Meter gesetzt, kann die Helix bis zu 8 Fahrzeugkomponenten optimal darstellen.

Der Ablauf der Metapher wird in der Zustandsmaschine in Abbildung 6-12 beschrieben. Mit einer Animation wird der 3D-Katalog geöffnet oder geschlossen. Bei der Helix-Form wird zusätzlich eine Animation für die Rotation des gesamten Katalogs auf horizontaler Ebene implementiert, damit der Benutzer alle Komponenten betrachten kann, ohne sich selbst zu bewegen.

Liegt der Benutzerfokus auf einer Komponente (Optionswert), wird automatisch über das Netzwerk die zugehörige Statistik beim CATER-Backend nachgefragt und im Informationsbereich angezeigt (siehe auch Abbildung 6-7). Falls gleichzeitig eine Animation abläuft, wird diese unterbrochen. Die Text-Informationen kann somit in Ruhe gelesen werden. Die Options-Beschreibungen enthalten auch Aktionen bezüglich des gewählten Optionswerts, die der Benutzer auswählen kann. Erfolgt eine Auswahl, wird die entspre-

chende Aktion durchgeführt. Bestätigt der Benutzer die Auswahl eines Optionswerts, wird der Katalog geschlossen und die gewünschte Option am Fahrzeugmodell angezeigt.

Das Prinzip der Animation wird im Kapitel 4.8.3 erläutert. Im Layout der Fahrzeugkonfiguration werden, wie in der Katalog-Metapher (Kapitel 6.5.2) erläutert, zwei Animationen implementiert.

Zusätzlich werden weitere Animationen nach dem RD-Konzept erstellt, die aber nicht direkt in der Fahrzeugkonfiguration eingesetzt werden. Abbildung 6-13 zeigt die Zustandsmaschine am Beispiel der Animation bei der Öffnung eines Katalogs. Der Bereich zwischen „Start“ und „Katalog geöffnet“ zeigt die speziellen Zustände für die Katalog-Öffnung. Die sonstigen Zustände gelten allgemein für alle Animationen.

Bei der Implementierung von Animationen ist es häufig notwendig, Punkte im Ablauf zu definieren, an denen technisch unabhängige Verläufe synchronisiert werden. Bei der Animation Katalog-Öffnung findet die Synchronisation nicht nur zu den Initial- und Endzuständen statt, sondern auch in der Mitte der Animation an dem Punkt, an dem der Ring seinen höchsten Punkt erreicht, bevor er wieder nach unten sinkt.

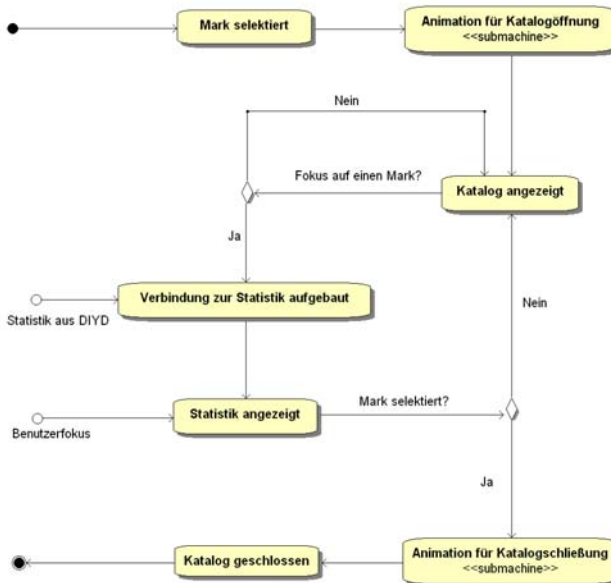


Abbildung 6-12: Zustandsdiagramm der Katalog-Metapher

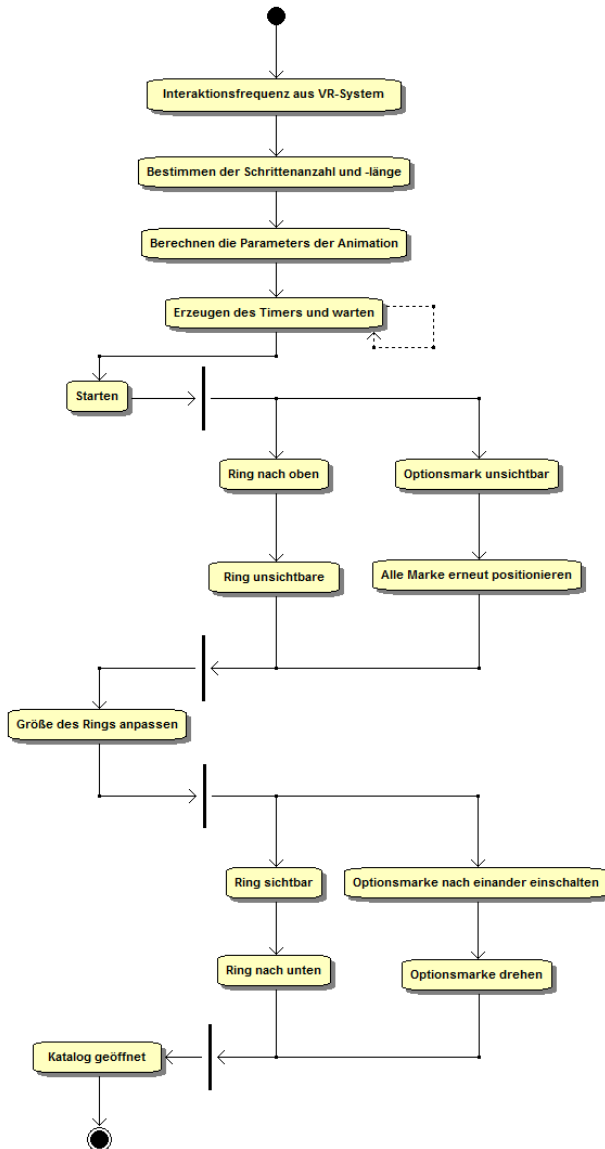


Abbildung 6-13: Zustandsmaschine der Animation zum Öffnen der Kataloge

## 6.6 Verbindung mit Datenquellen

Der Fahrzeugkonfigurator verbindet die Datenbanken im CATER-System. Die Datenbanken werden von dem für das DIYD-Modul zuständigen Projektpartner bereitgestellt. Mit einem Webservice kann über das Internet von jedem Ort der Welt aus auf die Datenbank zugegriffen werden. Im Konfigurator werden zwei Funktionalitäten über die Datenbankverbindung realisiert: Zum Ersten die Anfrage der typischen Konfiguration für ausgewählte Benutzergruppen und zum Zweiten die Anzeige von aktuellen Statistiken für einzelne Fahrzeugkomponenten.

Die technische Spezifikation des Webservices ist in Tabelle 6-3 zusammengefasst:

Spezifikation	Beschreibung
Zweck	Ein Dienst für den Zugriff auf die Fahrzeugkonfiguration und Statistiken für bestimmte Benutzergruppen
Adresse des Test-Netzwerkdienstes	<a href="http://160.40.10.35:8080/CATERproject/services/GetPredictionClass?wsdl">http://160.40.10.35:8080/CATERproject/services/GetPredictionClass?wsdl</a>
Eingabe für Fahrzeugkonfiguration	Eine Kundengruppe aus folgendem Umfang <ul style="list-style-type: none"> <li>• 10 Kundengruppen für Personenkraftwagen</li> <li>• 8 Kundengruppen für Lastkraftwagen</li> </ul>
Ausgabe für Fahrzeugkonfiguration	Je ein Beispielmodell für Personal- und Lastkraftwagen
Eingabe für Statistik der Fahrzeugkomponenten	Auswahl der Optionswerte
Ausgabe für Statistik der Fahrzeugkomponenten	<ul style="list-style-type: none"> <li>• Statistik für 62 Optionswerte in Personenkraftwagen</li> <li>• Statistik für 35 Optionswerte in Lastkraftwagen</li> </ul>

**Tabelle 6-3: Spezifikation des Webservices für Konfiguration und Statistik**

Technisch werden folgende vier Anfragen von dem Webservice angeboten:

Remote Methode	Eingabe	Ausgabe
getCarConfig	<ul style="list-style-type: none"> <li>• Region der Kundengruppe</li> <li>• Altersintervall der Kundengruppe</li> <li>• Geschlecht der Kundengruppe</li> <li>• Spezielle Anforderung der Kundengruppe</li> </ul>	Ein Vektor des Schlüssels der gewählten sechs Optionswerte für Personenkraftwagen
getTruckConfig	<ul style="list-style-type: none"> <li>• Region der Kundengruppe</li> <li>• Beschreibung der Kundengruppe</li> </ul>	Ein Vektor des Schlüssels der gewählten fünf Opti-

Remote Methode	Eingabe	Ausgabe
	<ul style="list-style-type: none"><li>pe</li><li>• Spezielle Anforderung der Kundengruppe</li></ul>	onswerte für Lastkraftwagen
getCarStat	<ul style="list-style-type: none"><li>• Region der Kundengruppe</li><li>• Altersintervall der Kundengruppe</li><li>• Geschlecht der Kundengruppe</li><li>• Schlüssel der Fahrzeugkomponente</li></ul>	Anzahl der Kunden in der Kundengruppe, die sich für diesen Optionswert entschieden haben
getTruckStat	<ul style="list-style-type: none"><li>• Region der Kundengruppe</li><li>• Beschreibung der Kundengruppe</li><li>• Schlüssel der Fahrzeugkomponente</li></ul>	Anzahl der Kunden in der Kundengruppe, die sich für diesen Optionswert entschieden haben

**Tabelle 6-4: Methode des Webservices für Konfiguration und Statistik**

## 7 Evaluierung

### 7.1 Vorgehensweise

Am Beispiel der Fahrzeugkonfiguration wurde eine VR-Anwendung nach dem Leitfaden des RD-Konzepts entwickelt. Es ist beabsichtigt zu verifizieren, ob das RD-Konzept die in Kapitel 3 festgestellten Defizite schließen kann und ob das RD-Konzept dazu beiträgt, die Schwierigkeiten bei der Entwicklung von integrierten VR-Anwendungen zu vermindern. Konkret ist zu evaluieren, ob

1. das RD-Konzept zur Entwicklung von integrierten VR-Anwendungen effektiv einsetzbar ist;
2. eine auf der Basis des RD-Konzepts entwickelte Anwendung einen industriellen Teilprozess abbilden, und dieser in einen Gesamtprozess integriert werden kann;
3. vielfältige Informationsformen mit Unterstützung des RD-Konzepts produktiv in der immersiven UI verwendet werden können;
4. mit einer immersiven Anwendungsbenutzungsoberfläche, die nach dem Leitfaden des RD-Konzepts entwickelt wurde, eine im Vergleich zu herkömmlichen VR-Anwendungen signifikante Verbesserung der Benutzbarkeit, insbesondere der Arbeitseffizienz, erzielt werden kann.

Die erste Fragestellung ist eine grundlegende Anforderung, die durch die erfolgreiche Umsetzung des RD-Konzepts auf der VR-Plattform VDP und die Entwicklung der Anwendung für die Fahrzeugkonfiguration beantwortet wird. Dabei ist allerdings anzumerken, dass die Effektivität eines Konzepts als softwaretechnische Infrastruktur nur durch langfristigen Einsatz in verschiedenen Anwendungen nachzuweisen ist. Die Erkenntnis, die mit dem Anwendungsfall der Fahrzeugkonfiguration erworben wurde, ist dabei zwar der wichtigste, jedoch lediglich der erste Schritt.

Die zweite und dritte Fragestellung betrifft die Überprüfung des Funktionsumfangs, der für integrierte VR-Anwendungen notwendig sind. In der zweiten Fragestellung geht es speziell um den Funktionsumfang auf der Prozessebene, in der dritten um den Funktionsumfang auf Datenebene.

Die letzte Fragestellung beinhaltet die Evaluierung der Benutzbarkeit. Vorhergehende Benutzbarkeitsstudien zeigen überwiegend negative Ergebnisse für 3D-GUI bzw. immersive Benutzungsoberflächen im Vergleich zu konventionellen 2D-GUI. Insbesondere beim Thema Arbeitseffizienz wird die 2D-Variante mit Abstand als besser eingestuft [Polys 2005; Hauber 2006]. In der Evaluierung ist daher festzustellen, ob eine auf dem RD-Konzept basierende Anwendung eine bessere Benutzbarkeit bieten kann als bisherige VR-Anwendungen. Diese wichtige Aufgabe der Evaluierung wird in einer Benutzbarkeitsstudie (siehe Kapitel 0) geklärt. Auch während der Anwendungsentwicklung ist die inter-

aktive Evaluierung der Benutzbarkeit ein wichtiges Thema. In den Kapiteln 7.2 und 7.3 wird die Vorgehensweise und die Details vorgestellt.

Die obengenannten Fragen können nur anhand einer konkreten Anwendung, bei deren Entwicklung das RD-Konzept umgesetzt wird, beantwortet werden. So erfolgt die Evaluierung des RD-Konzepts in der vorliegenden Arbeit am Beispiel des Fahrzeugkonfigurators.

Der Fahrzeugkonfigurator dient dazu, die kundenindividuelle Massenproduktion (MC) in der Automobilindustrie zu unterstützen. Ziel ist es, den Fahrzeugkonfigurator als integrierten Teil des Prozesses zu installieren. Allerdings ist die kundenindividuelle Massenproduktion in der Fahrzeugherstellung derzeit noch nicht weit verbreitet. Eine detaillierte Spezifikation, welche die Anforderungen des angestrebten Zielprogramms beschreibt, konnte daher am Anfang der Arbeit nicht vorgegeben werden. Es ist also nicht möglich, die Anwendung für die Fahrzeugkonfiguration nach sequenziellen Vorgehensmodellen wie dem Wasserfallmodell [Royce 1987] zu entwickeln.

Stattdessen wurde für die Anwendungsentwicklung eine iterative Vorgehensweise mit einem operationalen Prototyp gewählt [Davis 1992]. Jeder Optimierungszyklus des Prozesses enthält folgende Aktivitäten, die in Abbildung 7-1 zusätzlich illustriert werden [Boehm 1986]:

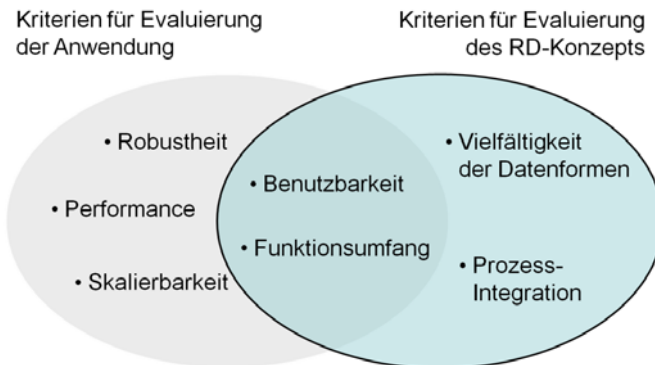


Abbildung 7-1: Spiral-Vorgehensmodell nach Boehm [Boehm 1986]



- Festlegung von Zielen, Identifikation von Alternativen und Beschreibung der Rahmenbedingungen
- Evaluierung der Alternativen und das Erkennen, Abschätzen und Reduzieren von Risiken, z. B. durch Analysen, Simulationen oder Prototyping
- Realisierung und Überprüfung des Prototyps
- Planung des nächsten Zyklus

Die Aktivitäten zur Evaluierung des RD-Konzepts begleiten den Entwicklungsprozess und finden parallel zur Evaluierung der Anwendung in jeder Phase der Spirale statt. Jedoch sind die Objekte der beiden Evaluierungsaktivitäten nicht vollständig identisch. Während die Robustheit, die Performance und die Skalierbarkeit bei der Evaluierung der Anwendung eine wichtige Rolle spielen, liegt hinsichtlich der Evaluierung des RD-Konzepts der Fokus auf den Datenformen und der Prozessintegration. Die Benutzbarkeit und der Funktionsumfang sind in beiden Evaluierungsprozessen enthalten (siehe Abbildung 7-2).



**Abbildung 7-2: Kriterien in der Evaluierung der Anwendung und des RD-Konzepts**

In diesem Kapitel werden hauptsächlich die Aktivitäten der Evaluierung des RD-Konzepts erklärt. Sie finden in zeitlicher Reihenfolge gemäß dem Spiral-Vorgehensmodell in den folgenden drei Phasen statt:

- Anforderungsphase (Kapitel 7.2)
- Grobentwurfsphase (Kapitel 7.3)
- Feinentwurf und betriebsfähiger Prototyp (Kapitel 0)

In jeder Phase werden die folgenden Punkte geklärt und erläutert:

- Kriterien und Hypothesen
- Setup
- Durchführung
- Ergebnisse

## 7.2 Evaluierungsaktivitäten in der Anforderungsphase

### 7.2.1 Kriterien und Hypothesen

In der ersten Anforderungsphase der Anwendungsentwicklung wird der Funktionsumfang für den Fahrzeugkonfigurator festgelegt. In Bezug auf das RD-Konzept ist dabei zusätzlich zu bewerten, ob bzw. inwieweit das RD-Konzept anhand dieser zu entwickelnden VR-Anwendung evaluiert werden kann. Insbesondere sind dabei folgende drei Kriterien entscheidend:

- Funktionsumfang: Er wird durch die Anforderungsphase bestimmt. Es wird angenommen, dass der Funktionsumfang die Fragestellungen des RD-Konzepts (Punkt 1 bis 3 in Kapitel 7.1) decken kann.
- Vielfältigkeit der Datenformen: Es wird angenommen, dass vielfältige Datenformen von den Benutzern der Anwendung gefordert werden.
- Prozessintegration: Es wird angenommen, dass
  - die Anwendung ein integrierter Teil eines übergeordneten Prozesses sein soll und
  - die zugrundeliegende VR-Plattform die Prozessintegration unterstützt.

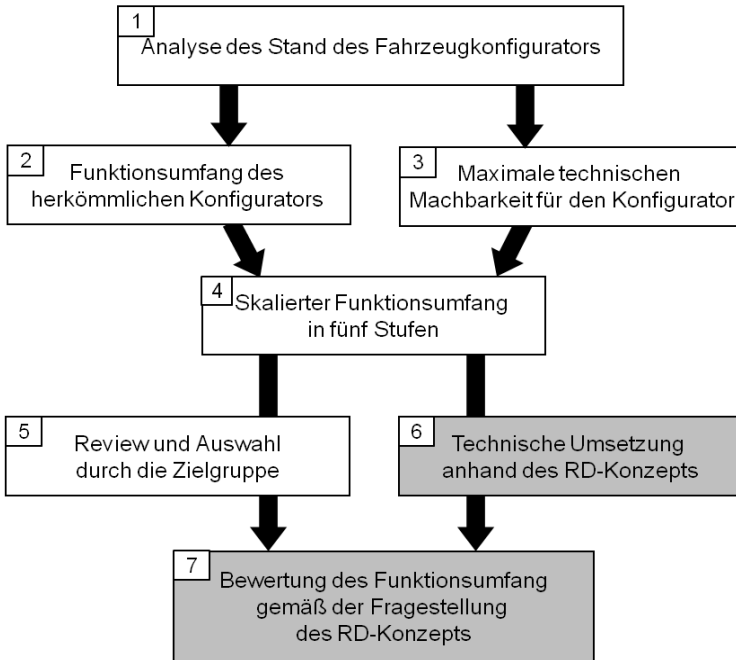
### 7.2.2 Setup

Die Arbeiten zur Bestimmung und Bewertung des Funktionsumfangs werden auf einem Desktop-Computer durchgeführt. Zudem wird ein Web-Arbeitsplatz für die Arbeitsgruppe aufgebaut, um den Meinungsaustausch sowie die Verwaltung des Know-how über die Grenze des einzelnen Teilnehmers der Arbeitsgruppe hinaus zu organisieren.

### 7.2.3 Durchführung

Zwei Gruppen von Personen, das Entwicklerteam und Mitarbeiter von Automotive OEM, nahmen an den Aktivitäten teil. Das Entwicklerteam bestand aus vier Fachleuten mit langjähriger Erfahrung im Forschungsbereich VR. Aus dem Bereich Automotive OEM beteiligten sich vier Personen, die zukünftig für die VR-Konfiguration zuständig sein sollen. Die Aktivitäten und der Ablauf zur Bestimmung und Bewertung des Funktionsumfangs werden in Abbildung 7-3 illustriert.

Das Entwicklerteam recherchierte den Stand der Technik für die Fahrzeugkonfiguration (Schritt 1 in Abbildung 7-3). Aus den erworbenen Kenntnissen wurden der Funktionsumfang eines herkömmlichen Konfigurators (Schritt 2) und der mit heutiger Technik maximal realisierbare Funktionsumfang (Schritt 3) bestimmt. Weiter wurden fünf Abstufungen zwischen den beiden Extremkategorien eingefügt (Schritt 4). Die jeweils höhere Stufe beinhaltet den Funktionsumfang der nächsttieferen Stufe und zusätzlich eigene Funktionalitäten.



**Abbildung 7-3: Ablauf zum Bestimmen und zur Bewertung des Funktionsumfangs**

Der fünfstufige Funktionsumfang wurde den Mitarbeitern der Automotive OEM zur Auswahl vorgelegt (Schritt 5). Soweit sie es für notwendig erachteten, konnten sie auch neue funktionale Anforderungen hinzufügen. Außerdem wurde die technische Umsetzung für jede Stufe gemäß dem RD-Konzept geklärt (Schritt 6). Zum Abschluss wurde bewertet, ob die ausgewählte Stufe bzw. die entsprechende Umsetzung die Fragestellung des RD-Konzepts validieren kann (Schritt 7).

Im Rahmen der Schritte 1 bis 5 wird der Funktionsumfang des Fahrzeugkonfigurator bestimmt. Die Schritte 6 und 7 evaluieren, ob der Funktionsumfang die Fragestellung des RD-Konzepts (Punkt 1 bis 3 in Kapitel 7.1) decken kann. In der Abbildung 7-3 sind die beiden Evaluierungsschritte mit dunklem Hintergrund gekennzeichnet. Die Tabelle 7-1 erläutert den fünfstufigen Funktionsumfang sowie die mögliche Umsetzung mit dem RD-Konzept.

Stufe	Funktionale Anforderung aus Anwendung	Technische Umsetzung mit RD-Konzept
Stufe 1 Minimaler VR-Konfigurator	<ul style="list-style-type: none"> <li>Immersive Darstellung des Fahrzeugs mit auswählbaren Komponenten.</li> <li>Ein digitaler Katalog mit Fahrzeugkomponenten steht als Datenstruktur zur Verfügung.</li> <li>Es dürfen nur Anfragen an den Katalog gestellt werden. Änderungen des Katalogs dürfen nicht möglich sein.</li> <li>Benutzer können entsprechend einer vordefinierten Reihenfolge die Konfiguration ändern.</li> </ul>	<ul style="list-style-type: none"> <li>Datenaustausch auf Datenebene</li> <li>Der Katalog kann starr ins Programm geschrieben werden.</li> <li>Robust, nahezu wartungsfrei einsetzbar.</li> <li>Ähnlicher Funktionsumfang wie bei schon existierenden Web-Konfiguratoren für Fahrzeuge.</li> </ul>
Stufe 2	<ul style="list-style-type: none"> <li>Der Inhalt des Katalogs kann über eine externe Datenbank hinzugefügt, gelöscht und modifiziert werden.</li> <li>Eine Interaktionsmetapher wird erstellt, um die Selektion mit 6DOF-Eingabegeräten durchzuführen.</li> </ul>	<ul style="list-style-type: none"> <li>Die Konfigurationen sollten modifiziert werden können, ohne das Programm selbst ändern zu müssen.</li> </ul>
Stufe 3	<ul style="list-style-type: none"> <li>Benutzer können in freier Reihenfolge die Konfiguration ändern.</li> <li>Es sollten reichlich Metadaten zur Verfügung gestellt werden, um die Auswirkung von Veränderungen der Konfiguration zu erklären.</li> </ul>	<ul style="list-style-type: none"> <li>Die Abhängigkeiten zwischen den Optionen sollten als Datenstruktur erfasst werden.</li> <li>Das Layout der Metapher kann eine große Rolle spielen.</li> </ul>
Stufe 4	<ul style="list-style-type: none"> <li>Der Inhalt des digitalen Katalogs kann während der Laufzeit von externen Datenbanken beeinflusst oder bestimmt werden.</li> </ul>	<ul style="list-style-type: none"> <li>Flexible Datenstruktur und Datenbankverbindung soll realisiert werden können.</li> </ul>

Stufe	Funktionale Anforderung aus Anwendung	Technische Umsetzung mit RD-Konzept
	<ul style="list-style-type: none"> <li>Benutzeraktionen bzw. Arbeitsergebnisse können während der Laufzeit an die Datenbank zurückgegeben werden.</li> </ul>	<ul style="list-style-type: none"> <li>Eine Remote-Datenbank ist für die Aktualisierung einer Konfiguration sinnvoll.</li> </ul>
Stufe 5 Maximale Funktionalitäten	<ul style="list-style-type: none"> <li>Darstellung der Topologie der Datenstruktur.</li> <li>Eventuelle Topologieänderung der Datenstruktur während der Laufzeit.</li> <li>Kollaboration verschiedener Standorte.</li> </ul>	<ul style="list-style-type: none"> <li>Betrifft die Visualisierung der Datenstruktur.</li> <li>Die Topologie darf nicht beliebig sein.</li> <li>Das derzeitige TCP-IP Protokoll bietet keine Garantie für eine bestimmte Bandbreite.</li> </ul>

**Tabelle 7-1: Analyse des Funktionsumfangs zur ersten Erprobung**

In der Anforderungsphase wurde eine Zielplattform ausgewählt und bewertet, ob bzw. inwieweit diese Plattform die technischen Rahmenbedingungen des RD-Konzepts erfüllen kann. Dabei wurde

- zuerst eine VR-Plattform zur technischen Konzeption gewählt und die Struktur der gewählten Ziel-Plattform analysiert.
- Anschließend wurden die Bedingungen und Voraussetzungen des RD-Konzepts aufgezeigt und anhand der Ziel-Plattform festgestellt, ob bzw. welche Modifikationen, Ergänzungen bzw. Erweiterungen für die Realisierung des RD-Konzepts notwendig sind.

Die Durchführung der Evaluierungsaktivitäten bezüglich Auswahl und Analyse der Zielplattform werden in den Kapiteln 5.2 *Auswahl der Zielplattform* und 5.3 *Bedingungen bzw. Voraussetzungen des Richdata-Konzepts* ausführlich vorgestellt und daher hier nicht wiederholt.

#### 7.2.4 Ergebnisse

Die Ergebnisse der Bewertung des Funktionsumfangs umfassen folgende Punkte:

- Der Funktionsumfang des Konfigurators entspricht größtenteils den Anforderungen von Stufe 4.
- Die Metadaten in Stufe 3 sollten wie die Konfiguration während der Laufzeit aktualisiert werden können.

- Die Erstellung neuer Optionen oder Optionswerte wird vom CATER-Backend nicht unterstützt, da dies aus Datenschutzgründen in der Industrie nicht möglich ist.
- Es sollte möglich sein, verschiedene Sprachen, auch aus dem asiatischen Raum, darzustellen.

Die Anforderungen des ausgewählten Funktionsumfangs stimmen mit den Zielen des RD-Konzepts (Punkt 1 bis 3 in Kapitel 7.1) weitgehend überein. Insbesondere folgende Punkte sind in dem Anwendungsfall von Bedeutung:

- Die System-Integration entlang der Wertschöpfungskette ist ein zentraler Bestandteil der Anwendung.
- Sowohl Geometrie-Daten als auch typische RD-Informationsformen, wie Statistiken, Modetrends und technische Beschreibungen, spielen eine wichtige Rolle in der immersiven Umgebung.
- Das System-Engineering-Team benötigt in der VR nicht nur perfekte Fahrzeugmodelle, sondern auch eine professionelle Benutzerschnittstelle, um in der immersiven Umgebung effizient arbeiten zu können
- Die involvierten Module sind auf verschiedene Orte verteilt. Die Anforderungen an die Datenkommunikation in der Laufzeit sind demnach entsprechend hoch.

Diese Ergebnisse sind auch in Kapitel 5.3 *Bedingungen bzw. Voraussetzungen des Rich-data-Konzepts* zu finden. Durch die Evaluierungsaktivitäten wird bestätigt, dass

- die Fragestellungen des RD-Konzepts (Punkt 1 bis 3 in Kapitel 7.1) durch die Anwendung für die Fahrzeugkonfiguration beantwortet werden,
- vielfältige Informationsformen von den Benutzern der VR-Anwendung explizit gefordert werden,
- die Anwendung in einen übergeordneten Prozess integriert werden soll und
- die zugrundeliegende VR-Plattform mit kleinen Erweiterungen die Prozessintegration unterstützt.

## 7.3 Evaluierungsaktivitäten in der Grobentwurfsphase

### 7.3.1 Kriterien und Hypothesen

Nach der Erarbeitung der Anforderungen wurde ein lauffähiger Prototyp entworfen und implementiert. Kriterien der Evaluierungsaktivitäten in der zweiten Phase sind:

- Die Benutzbarkeit: Es wird angenommen, dass ein immersives Layout der VR-Benutzungsoberfläche realisiert wird, das die Ergebnisse der letzten Phase bzw. die Anforderung des RD-Konzepts zum Teil oder vollständig erfüllt.
- Die Funktionalitäten: Es wird angenommen, dass die in der letzten Phase festgelegten Funktionalitäten zum Teil oder vollständig realisiert werden.

Bezüglich der Vorgehensweise nach dem Spiral-Modell wird allgemein angenommen, dass der Prototyp dabei kontinuierlich verbessert wird. Werden ein oder beide oben genannten Punkte nicht vollständig erfüllt, sollen Verbesserungsvorschläge bzw. neue Anforderungen für die nächste Entwicklungsphase aufgestellt werden.

### 7.3.2 Setup

Zur Evaluierung werden drei Test-Systeme eingesetzt. Die Ausstattungen der in dieser Phase verwendeten Test-Systeme 1 und 2 werden in der Tabelle 7-2 beschrieben. System 3 wird im Kapitel 7.4. vorgestellt.

Anzeige-System	System 1: Desktop	System 2: Mobiles VR-System
Anzeigegerät	Monitor oder Mono-Projektor	Stereoprojektor
Auflösung	1680 x 1050	1280 x 1024
Eingabe		
Tracking	Simuliert mit Logitech Dual Action	ART optisches Tracking-System mit 2 Infrarot-Kameras
Eingabe	Simuliert mit Logitech Dual Action	Optische IC:Control
Computer-System		
CPU	Xeon 3.0 GHz, 1 core 2 thread	Xeon 3.06 GHz, 2 core
Arbeitsspeicher	2 GB	4 GB
Graphikkarte	NVIDIA FX 4500	NVIDIA FX 3700
Grafikspeicher	512 MB 256 Bit GDDR3	512 MB 256 Bit GDDR3
Software-System		
Betriebssystem	Windows XP Professional 64 bit	Windows XP Professional 64 bit
VR Plattform	VDP 2007 / VDP 2008	VDP 2008

**Tabelle 7-2: Ausstattung der Test-Systeme**

Das erste Test-System ist das Entwicklungssystem. Es wird für Erprobungen während der Entwicklung und zur Validierung des immersiven Layouts eingesetzt. Das zweite System ist ein mobiles System, das vor allem für den Test des ersten lauffähigen Prototyps verwendet wird.

Für die Evaluierung stehen drei Test-Datensätze zur Verfügung (siehe Tabelle 7-3). Der Datensatz Car-Design-Kit umfasst eine Material- und 11 Geometrie-Optionen. Aufgrund

seiner Größe ist dieser Datensatz besonders geeignet für Performance-Tests. Von den übrigen beiden Datensätzen beinhaltet der eine ein Modell eines Personenkraftwagens und der andere die Daten eines Lastkraftwagens. Für diese beiden Datensätze werden von allen Partnern, die in die Prozessintegration involviert sind, Metadaten und Statistiken zur Verfügung gestellt. Die zwei letztgenannten Datensätze werden daher speziell für den Test der Systemintegration und im Rahmen der Benutzbarkeitsstudie eingesetzt.

	Car-Design-Kit	Personenkraftwagen	Lastkraftwagen
Geometrie-Optionen	11	4	4
Material-Optionen	1	2	1
Geometrie-Optionswerte	44	30	17
Material-Optionswerte	24	32	18

**Tabelle 7-3: Test-Datensätze**

### 7.3.3 Durchführung

Die Evaluierungsaktivitäten von der Grobentwurfsphase bis zur Feinentwurfsphase folgen dem iterativen Verfahren des Spiral-Vorgehensmodell (siehe Kapitel 7.1). Die konkreten Schritte der Umsetzung werden in Abbildung 7-4 vorgestellt.

An den Evaluierungsaktivitäten werden drei Personengruppen beteiligt:

- Der Entwickler
- Vier Probanden aus dem Personenkreis der Mitarbeiter bei Automotive OEM
- Fünf Probanden aus dem Personenkreis der Fachleute mit VR-Entwicklungserfahrung

Zuerst demonstrierte der Entwickler beiden Probandengruppen die Arbeit mit dem Prototypen (Schritt 1). Dabei wurden das immersive Layout und die implementierten Funktionalitäten grundlegend erklärt (Schritt 2 und 3). Danach bewerteten die Probanden diese prototypisch implementierte Version des Fahrzeugkonfigurators. Änderungsvorschläge bzw. neue funktionale Anforderungen wurden aufgelistet bzw. diskutiert (Schritt 6). Parallel dazu testeten die Probanden selbst den Umgang mit der Benutzungsoberfläche (Schritt 5). Vorschläge in Bezug auf die Benutzbarkeit wurden notiert und anschließend die Benutzungsoberfläche verbessert (Schritt 7). Zum Ende wurden die Änderungsvorschläge gemäß dem RD-Konzept in den Prototyp implementiert (Schritt 8).

Die durch diese Änderungen und Ergänzungen entstandene neue Version des Prototyps wird im nächsten Zyklus des Spiral-Vorgehensmodells weiter evaluiert. Die Schritte 6, 7 und 8 betreffen die Evaluierung des RD-Konzepts. In der Abbildung 7-4 sind sie mit dunklem Hintergrund gekennzeichnet. In diesem Kapitel werden die immersiven Layouts für



jeden Evaluierungszyklus erklärt. Sie bilden auch die Grundlage für die Beurteilung der Benutzbarkeit des UI.

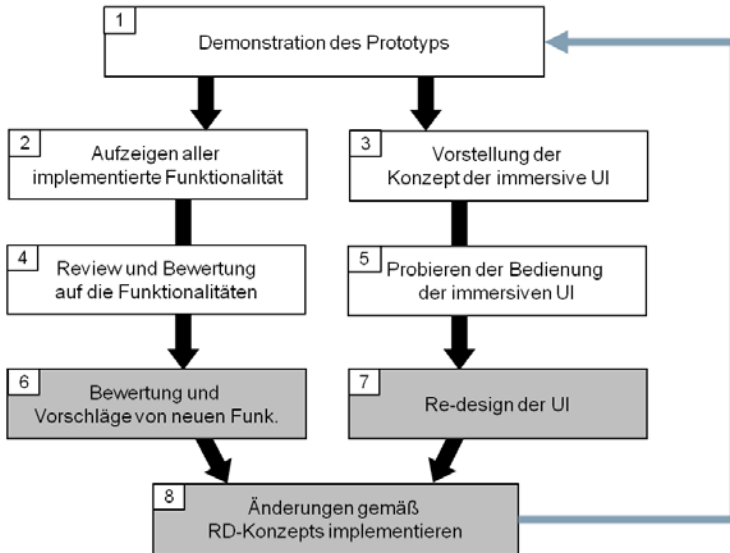


Abbildung 7-4: Ablauf der Evaluierungsaktivitäten in Grob- bis Feinentwurfsphase

Die erste Version des immersiven UI besteht aus vier wesentlichen Bestandteilen (siehe auch Abbildung 7-5):

- Das Fahrzeug  
Das zu konfigurierende Fahrzeug steht in der Mitte der VR-Benutzeroberfläche auf einer runden Bühne. Alle möglichen Optionssymbole sind ringförmig um das Fahrzeug herum angeordnet. Bei Bedarf kann die dargestellte Szene um ein weiteres Fahrzeug bzw. eine zusätzliche Bühne erweitert werden. Damit lassen sich z. B. zwei Modellvarianten direkt miteinander vergleichen.
- Bühne und Ring  
Die Größe der Bühne bzw. des Rings sollte sich automatisch an die reale Größe des Fahrzeugs anpassen. Die Bühne sollte von dem Benutzer gedreht werden können, um das Fahrzeug von verschiedenen Standpunkten aus optimal betrachten zu können.
- Metadaten der Fahrzeuge  
Die Metadaten zum Fahrzeug werden hinter dem Fahrzeug angezeigt. Sie können sich automatisch an die Benutzerposition anpassen, damit die zweidimensionalen Informationen in der immersiven Umgebung besser erkennbar sind.



Abbildung 7-5: Layout der ersten Version der immersiven Benutzungsschnittstelle

- Fernstehender Rechner  
Mittig hinter dem Fahrzeug wird ein Anzeigefenster positioniert, das das Bild eines fernstehenden Rechners wiedergibt. Die technischen Details dazu werden in Kapitel 4.9.2 erklärt.

Die Zwischenergebnisse der Bewertung durch die Probanden sind in folgenden Punkten zusammengefasst:

- Das Feedback der Probanden fällt überwiegend positiv aus. Alle benötigten Elemente für die Benutzungsoberfläche sind im Aufriss zu finden und auch für den Benutzer leicht zugänglich.
- Das Farb-Konzept muss verbessert werden. Für die Einfärbung von Hintergrund und Bühne sollten keine dominierenden Farben verwendet werden. Die farbliche Gestaltung der Umgebung kann verfälschende Auswirkungen darauf haben, wie der Benutzer seine Auswahl trifft.
- Die Werte der Varianten (in Form von Geometrie-Datensätzen) dürfen nicht zu nahe beieinander stehen.
- Informationen über die verschiedenen Varianten sollten sich nahe bei den Objekten befinden.
- Die Anzeige des fernstehenden Rechnerbildschirms sollte beweglich sein. Idealerweise sollte sie der Benutzer steuern können.

Obige Verbesserungsvorschläge werden in der nächsten Phase des iterativen Entwicklungsprozesses berücksichtigt. Auf dieser Basis wird der erste lauffähige Prototyp implementiert. Für die Anwahl der einzelnen Optionen werden Interaktionsmetaphern integriert. Die Metadaten sind im Prototyp allerdings noch nicht mit der Programmlogik verbunden (siehe Abbildung 7-6).



Abbildung 7-6: Layout des ersten lauffähigen Prototyps

Das Grundkonzept des ersten Layouts wird beibehalten. Die Verbesserungsvorschläge der vorigen Evaluierungsphase werden wie in Tabelle 7-4 erläutert umgesetzt.

Verbesserungsvorschläge	Umsetzung
Verbesserung des Farbkonzepts	Hintergrund und Hauptbühne erhalten neutrale Farben (Grau in verschiedener Intension).
Varianten nicht zu nahe beieinander positionieren	Die Distanz zwischen den Varianten wird vergrößert.
Informationen über die verschiedenen Varianten nah an den Objekten anzeigen	Wenn der Fokus der Benutzereingabe auf einer Variante liegt, wird der Name als 3D-Text angezeigt. Weitere Informationen hinzuzufügen ist ebenfalls möglich.
Bildschirm des fernstehenden Rechners beweglich und steuerbar anzeigen	Der Rechner-Bildschirm wird in einem beweglichen Rahmen angezeigt. Maus-Events können durch Eingabegeräte simuliert werden.

Abbildung 7-7: Verbesserung in der Entwurfsphase

Alle in der Evaluierung des Funktionsumfangs festgestellten Funktionalitäten mit Ausnahme der Verbindungen zu VR-externen Prozessen werden in einen lauffähigen Prototyp implementiert. Die Ergebnisse der Erprobung sind in Tabelle 7-4 zusammengestellt.

Funktionalität	Methode	Ergebnisse
Immersive Darstellung des Fahrzeugs mit auswählbaren Komponenten	Erprobung des Prototyps	Voll funktionsfähig. Geprüft anhand eines Datensatzes mit 11 Optionen und 68 Alternativen bzgl. Komponenten oder Material.

Funktionalität	Methode	Ergebnisse
Digitaler Katalog mit auswählbaren Alternativen	Implizit	Voll funktionsfähig
Der Inhalt des Katalogs kann durch eine externe Datenbank hinzugefügt, gelöscht und modifiziert werden.	Implizit	Funktionalität für Prozessintegration noch nicht vorhanden
Selektion der Alternativen durch 6DoF-Eingabegeräte	Erprobung des Prototyps	Voll funktionsfähig
Benutzer kann die Konfiguration in freier Reihenfolge ändern.	Erprobung des Prototyps	Voll funktionsfähig
Es sollten reichlich Metadaten zur Verfügung gestellt werden, um die Auswirkung von Veränderungen der Konfiguration zu erklären.	Erprobung des Prototyps	3D-Text in fester Position oder an den Objekten verankert. Anzeige von Bildern bzw. bewegten Bilder über Bildschirmansicht eines ferngesteuerten Rechners
Der Inhalt des digitalen Katalogs kann während der Laufzeit von externen Datenbanken beeinflusst oder bestimmt werden.	Implizit	Funktionalität für Prozessintegration noch nicht vorhanden
Benutzeraktionen bzw. Arbeitsergebnisse können während der Laufzeit an die Datenbank zurückgegeben werden.	Implizit	Funktionalität für Prozessintegration noch nicht vorhanden

**Tabelle 7-4: Checkliste der Anforderungen an Funktionalitäten**

### 7.3.4 Ergebnisse

In der Erprobung werden folgende Defizite der Benutzbarkeit festgestellt:

- Im Layout des Prototyps sind die meisten RD-Repräsentationen am Rand der gesamten Szene positioniert. Dieser Szenenaufbau ist nur geeignet für eine immersive Anzeige mit weitem Blickwinkel, z. B. einer PowerWall mit zwei oder drei Kanälen oder einer CAVE™. Bei einem System mit einem einzigen Kanal, wie der für die Erprobung eingesetzten mobilen VR-Anlage, muss sich der Benutzer ständig neu orientieren, um alle RD-Repräsentation sehen zu können.
- Navigiert sich der Benutzer zwischen den Modellen und den RD-Repräsentationen hindurch, muss er in zwei verschiedene Richtungen blicken, um alle Elemente der Szene sehen zu können. Z. B. für den Vergleich der beiden Fahrzeugmodelle ist dies ungünstig.

- Der ferngesteuerte Bildschirm steht weit vom Benutzer entfernt und hat eine Länge von über vier Metern. Eine Steuerung mit simulierten Maus-Ereignissen ist technisch möglich. Jedoch lässt sich ein mehr als zwei Meter langer virtueller Strahl vom Benutzer nur mühsam kontrollieren.
- Durch Navigation, insbesondere im Fly-Mode, verliert der Benutzer leicht die Orientierung. Benutzern mit wenig Erfahrung in der VR ist es nur schwer möglich, wieder in die orthogonale Orientierung zurückzugelangen.
- Die VR-Umgebung vermittelt dem Benutzer keinen Eindruck davon, wie das Fahrzeug in einer realen Umgebung aussieht.

Folgende Tabelle listet die Funktionalitäten auf, die verbessert bzw. ergänzt werden sollten:

Funktionalität	Beschreibung
Der Inhalt des Katalogs kann durch eine externe Datenbank hinzugefügt, gelöscht und modifiziert werden.	Funktionalität für Prozessintegration
Der Inhalt des digitalen Katalogs kann während der Laufzeit von externen Datenbanken beeinflusst oder bestimmt werden.	Funktionalität für Prozessintegration
Benutzeraktionen bzw. Arbeitsergebnisse können während der Laufzeit an die Datenbank zurückgegeben werden.	Funktionalität für Prozessintegration
Realitätsnahe Umgebungen für die Konfigurationsaktivitäten, um dem Benutzer einen realistischen Eindruck seiner Arbeitsergebnisse zu vermitteln.	Anforderungen aus Erprobung der Benutzbarkeit
Animation zum Übergang der Zustandsänderung für RD-Datensätze	Anforderungen des RD-Konzepts
Deutliche Kennzeichnung des Rands einer Funktionsgruppe und Positionierung von RD-Elementen innerhalb dieser Rahmenelemente (Scope)	Anforderungen des Scope-Konzepts

**Tabelle 7-5: Verbesserungsvorschläge für Funktionalitäten am Ende der Entwurfsphase**

## 7.4 Evaluierungsaktivitäten mit betriebsfähigem Prototyp

### 7.4.1 Kriterien und Hypothesen

Für die Aktivitäten in der finalen Phase sind folgende Kriterien zu beachten:

- Funktionsumfang: Es wird angenommen, dass der Funktionsumfang vollständig implementiert und vollständig funktionsfähig ist.

- Vielfältigkeit der Datenformen: Es wird angenommen, dass vielfältige Datenformen in der Benutzungsoberfläche der Anwendung gefordert werden.
- Prozessintegration: Es wird angenommen, dass die Anwendung in dem Gesamtprozess integriert ist.
- Benutzbarkeit des Configurators. Es wird angenommen, dass mit einer immersiven Anwendungsbenutzungsoberfläche, die nach dem Leitfaden des RD-Konzepts entwickelt wurde, im Vergleich zu herkömmlichen VR-Anwendungen eine signifikante Verbesserung der Benutzbarkeit, insbesondere der Arbeitseffizienz, erzielt werden kann.

Da die Funktionalitäten, Vielfältigkeit der Datenformen sowie Prozessintegration im Kapitel „Anwendungsentwicklung am Beispiel der Fahrzeugkonfiguration“ ausführlich erläutert werden, werden sie in diesem Kapitel nicht als Schwerpunkt behandelt. Die Evaluierungsaktivitäten fokussieren sich auf die Benutzbarkeitsstudie.

#### 7.4.2 Setup

Als drittes Test-System wird die PowerWall im ICIDO Demo-Center eingesetzt (siehe Abbildung 7-8).

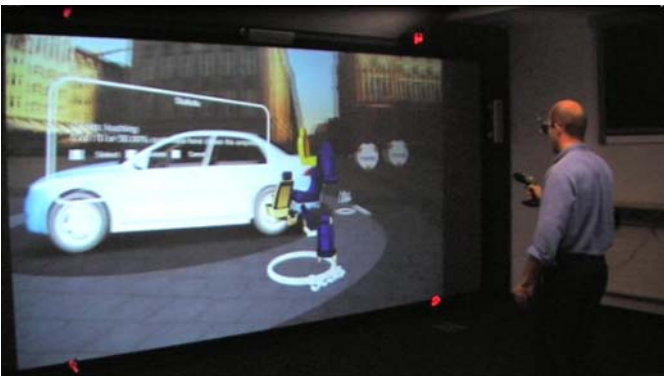


Abbildung 7-8: Testsystem für die finale Evaluierung

Die technische Spezifikation ist in Tabelle 7-6 aufgeführt. An dem PowerWall-System werden die Tests zu Performance und Benutzbarkeit der Schlussphase durchgeführt.

Anzeige-System	System 3: Stationäre VR-System
Anzeige Anlage	Stereoprojektor Voll HD
Auflösung	1920 x 1080

<b>Eingabe</b>	
Tracking	ART optisches Tracking-System mit 4 Infrarot-Kameras
Eingabe	ART Control
<b>Computer-System</b>	
CPU	Xeon 3.06 GHz, 2 core
Arbeitsspeicher	4 GB
Graphikkarte	NVIDIA FX 4800
Grafikspeicher	768 MB 384 Bit GDDR3
<b>Software-System</b>	
Betriebssystem	Windows XP Professional 64 bit mit SP2
VR Plattform	VDP 2008

**Tabelle 7-6: Konfiguration des Test-Systems mit stationärer VR-Anlage**

### 7.4.3 Durchführung

Der Prototyp wurde überarbeitet und angepasst, sodass zusätzlich die Funktionalitäten realisiert wurden, die in den vorangegangenen Entwicklungs- und Evaluierungsphasen beschrieben und als erforderlich festgestellt wurden. Die detaillierte Vorstellung dieses finalen Prototyps erfolgt in Kapitel 6. In der Schlussphase wird eine weitere Evaluierung durchgeführt. Schwerpunkt dabei ist die Benutzbarkeit des integrierten Fahrzeugkonfigurators unter Verwendung der Richdata. Neun Mitarbeiter aus drei Automobilwerken haben an der Benutzbarkeitsstudie teilgenommen. Statistische Daten zu den Probanden werden in Tabelle 7-7 aufgelistet.

Laut Nielsen lassen sich durch eine Benutzbarkeitsstudie mit sieben Probanden bereits viele wertvolle Erkenntnisse gewinnen [Nielsen 1994]. An der Studie für den VR-Konfigurator nahmen neun Probanden aus drei Automobilwerken teil. Damit ist ein ausreichender Umfang gegeben, um signifikante Ergebnisse erzielen zu können.

Test Personen	Statistik
Aus Forschungseinrichtung des OEM	<ul style="list-style-type: none"> <li>• Centro Ricerche Fiat (2),</li> <li>• Mercedes Benz Car Develop (5)</li> <li>• Volvo Technology (2)</li> </ul>
Beruflicher Hintergrund	<ul style="list-style-type: none"> <li>• Product planer (1),</li> <li>• Marketing (1),</li> <li>• Designer bzw. Ergonom (4),</li> <li>• VR-Experten (3)</li> </ul>
Geschlecht	<ul style="list-style-type: none"> <li>• Männlich (8)</li> <li>• Weiblich (1)</li> </ul>

**Tabelle 7-7: Probanden für die Benutzbarkeitsstudie**

Die Erprobung des Prototyps umfasste bei jedem Probanden folgende Punkte:

- Der Entwickler stellt das CATER-Storyboard und den Anwendungsfall für die Fahrzeugkonfiguration vor.
- Der Entwickler stellt den Funktionsumfang der Fahrzeugkonfiguration vor.
- Der Entwickler demonstriert und erklärt die immersive Benutzungsoberfläche.
- Der Proband erhält 20 Minuten Zeit, um sich in der virtuellen Umgebung zu rechtzufinden, falls er/sie wenig Erfahrung mit VR bzw. mit VDP hat.
- Der Proband wählt eine Kundengruppe aus und versucht für diese Kundengruppe eine Fahrzeugkonfiguration zu erstellen.
- Der Proband konfiguriert ein Fahrzeug nach eigenen Wünschen.
- Der Proband füllt den Fragebogen aus. Der Entwickler steht für eventuelle Rückfragen zur Verfügung.
- Der Proband und der Entwickler führen eine formlose Diskussion. Anregungen und Vorschläge aus der Diskussion werden im Fragebogen dokumentiert.

Die Ergebnisse der Benutzbarkeitsstudie basieren auf der Auswertung der Fragebögen. Ein leerer Musterfragebogen ist in Anhang A beigefügt. Der Fragebogen umfasst 78 Fragen, wobei der Proband einzelne Fragen auch unbeantwortet lassen darf, wenn er nicht antworten will oder kann. Einige Fragestellungen und die Ergebnisse bezüglich der Benutzbarkeit werden in den folgenden Tabellen aufgelistet. Die von den Probanden abgegebenen Einstufungen sind jeweils in absoluten Häufigkeiten angegeben. Zudem wird als Lagemaß der Median angegeben. Für die ordinal skalierten Werte ist der Median zwar weniger differenzierend als das arithmetische Mittel. Er bildet aber die Lage der Werte präziser ab.



Ergänzend wurde der einseitige nicht-parametrische Wilcoxon-Vorzeichentest [Siegel 2001] durchgeführt, um zu überprüfen, ob sich der Median vom Median von drei signifikant unterscheidet. Drei ist der mittlere Wert der verwendeten Skala und steht damit bei vergleichenden Bewertungen für Neutralität.

Wurde Signifikanz auf dem 5%-Niveau ( $p < 0,05$ ) erreicht, ist der Median mit einem Stern markiert. Bei Signifikanz auf dem 1%-Niveau ( $p < 0,01$ ) erfolgt eine Kennzeichnung mit zwei Sternen. Diese Vorgehensweise wird gewählt, um auszuschließen, dass die Ergebnisse durch zufällige Effekte zustande gekommen sind. Die Software *Statistical Analysis and Graphics* [NCSS 2010] wird verwendet, um die Parameter zu berechnen. Detaillierte Ergebnisse der Berechnungen mit NCSS enthält Anhang B.

#### 7.4.4 Ergebnisse

Die für diese Arbeit relevanten Ergebnisse der Benutzbarkeitsstudie sind im Folgenden zusammengestellt. Die übrigen im Fragebogen behandelten Fragestellungen sind im Kontext des Projekts CATER erforderlich.

- *Wie bewerten Sie den VR-Konfigurator im Vergleich zu ähnlichen Anwendungen, z. B. dem Web-Konfigurator mit konventionellem 2D-UI, bezüglich folgender Indikatoren?*

Indikator		1	2	3	4	5		Median
Nützlichkeit	useless	1		1	6	1	useful	4*
Anmutung	annoying		1	3	3	2	nice	4
Effektivität	superfluous			6	2	1	effective	3
Effizienz	inefficient		1	1	6	1	efficient	4*

**Tabelle 7-8: Absolute Häufigkeiten von Einstufungen der VR-Benutzungsoberfläche im Vergleich mit einer 2D-GUI**

Vom CATER-Projekt wird ein Web-Konfigurator zur Verfügung gestellt, der mit derselben Datenbasis ausgestattet ist, wie der zu testende Prototyp. Die Probanden von *Centro Ricerche Fiat* und *Volvo Technology* haben diesen Web-Konfigurator [CATER 2008] im Rahmen des Projekts bewertet. Die Probanden von *Mercedes-Benz Car Develop* haben den Web-Konfigurator von CATER nicht getestet und bewertet. Sie konnten jedoch auf einen eigenen Web-Konfigurator von Mercedes-Benz als Referenz zurückgreifen [Mercedes-Benz 2010]. Allerdings muss angemerkt werden, dass in diesem Fall die beiden Konfigurationen nicht mit derselben Datenbasis bestückt sind.

Die in der Tabelle gelisteten Ergebnisse zeigen, dass die Probanden der immersiven Benutzungsoberfläche in Bezug auf die Nützlichkeit und die Effizienz deutlich den Vorzug

geben. Von den neun Probanden stufen sieben die VR-Benutzungsoberfläche als besser ein, ein Proband stellt keinen Unterschied fest und ein Proband beurteilt die VR-Anwendung schlechter als den Web-Konfigurator. Fünf Probanden empfinden das VR-Interface als schöner. Sechs der neun Probanden vertreten die Auffassung, dass das VR-Interface ebenso effektiv ist wie die 2D-Variante.

- Wie bewerten Sie den VR-Konfigurator im Vergleich zur selben Anwendung auf dem Desktop-Computer bezüglich folgender Indikatoren?

Indikator		1	2	3	4	5		Median
Nützlichkeit	<i>useless</i>		2	1	5	1	<i>useful</i>	4
Anmutung	<i>annoying</i>			3	6		<i>nice</i>	4*
Effektivität	<i>superfluous</i>			4	5		<i>effective</i>	4*
Effizienz	<i>inefficient</i>			7	1	1	<i>efficient</i>	3

**Tabelle 7-9: Absolute Häufigkeiten von Einstufungen der VR-Benutzungsoberfläche im Vergleich mit dem Desktop-Computer**

Dank der Skalierbarkeit der VDP kann der VR-Konfigurator auch ohne Änderung auf einem Desktop-Computer mit normalem Monitor angezeigt und bedient werden. Das Bewertungsverfahren ist identisch mit dem der vorherigen Fragestellung. Die Auswertung der Antworten zeigt Tabelle 7-9.

Die in der Tabelle gelisteten Häufigkeiten belegen, dass die Probanden in Bezug auf die Nützlichkeit und die Anmutung die immersive Umgebung deutlich bevorzugen. Von den neun Probanden stufen sechs das VR-Interface als besser ein, drei sehen keinen Unterschied bzw. bewerten es schlechter als die Arbeit mit der Desktop-Version. Fünf Probanden vertreten die Meinung, dass die Arbeit in der virtuellen Umgebung effektiver ist als am Desktop, die übrigen vier stellen keinen Unterschied fest. Allerdings bevorzugt keiner von ihnen die Nutzung des 3D-UI am Desktop. Sieben Probanden geben an, dass die Arbeitseffizienz beider Systeme gleich ist. Zwei beurteilten das VR-Interface in diesem Punkt als besser.

- Welche Funktionalitäten des VR-Konfigurators sind für Sie in welchem Maß interessant?

Indikator		1	2	3	4	5		Median
1:1-Verhältnis	<i>Abso- lutely not inter- esting</i>		1		3	4	<i>Abso- lutely inter- esting</i>	4,5*
3D-Animation				3	4	1		4*
Immersives Feeling			1		3	4		4,5*
Verbindung mit der Datenbank			1	2	2	3		4
Zugriff auf Remote-Computer		1		3	3	1		3,5
Statistische Daten			2	3		3		3

**Tabelle 7-10: Bewertungen der im VR-Konfigurator implementierten Funktionalitäten**

Als Beantwortung dieser Frage hat ein Proband alle Funktionalitäten mit „absolutely not interesting“ bewertet. Er notierte auf dem Fragebogen, dass die Selektionsklicks zu lange dauern würden. Seine Bewertung wird bei der detaillierten Analyse der Beantwortung dieser Frage nicht berücksichtigt. Die Ergebnisse der Bewertungen der übrigen acht Probanden enthält Tabelle 7-10. Auf das größte Interesse stößt bei den Probanden die 3D-Animation. Acht der neun Probanden bewerten sie auf der fünfstufigen Skala mit den Werten 3, 4 oder 5. Auch die Faktoren reales Größenverhältnis und immersives Empfinden werden äußerst positiv eingestuft: Sieben von neun Probanden bewerten beide Aspekte als interessant oder interessant. Auch die Verbindung mit der Datenbank und der Zugriff auf den Remote-Computer stoßen mehrheitlich auf Interesse. Den statistischen Daten wird dagegen eine verhältnismäßig geringe Bedeutung zugemessen. Allerdings muss in diesem Zusammenhang angemerkt werden, dass die Statistik ein spezifisches Feature ist, das nur für bestimmte Benutzergruppen interessant ist.

- *Welcher Zugang zu einer Funktionalität (in der immersiven Umgebung) ist Ihnen sympathisch?*

Indikator		1	2	3	4	5		Median
Das Kugelmanü der VDP	<i>Abso- lutely not</i>	1	2	3	2	1	<i>Abso- lutely yes</i>	3
Das 2D-GUI auf dem Desktop		2		4	3			3
Das Symbol (3D-Icon) in der Szene				3	4	3		4**

**Tabelle 7-11: Bewertungen der unterschiedlichen Wege zu einer Funktionalität**

Bei dieser Frage sind die Bewertungen für das immersive und das konventionelle Menü jeweils gleichmäßig verteilt, sodass keine eindeutige Priorisierung festgestellt werden

kann. Im Vergleich dazu wird das Symbol in der Szene als Zugangsweg zu einer Funktion deutlich bevorzugt. Eine Optionsauswahl über Symbole in der Szene sind als 3D-Icons im RD-Konzept enthalten (siehe Kapitel 4.6.2).

- *Wenn Sie Metadaten oder Fakten ansehen möchten, um besser Entscheidungen treffen zu können, wo sollen diese (in der immersiven Umgebung) positioniert werden?*

Indikator		1	2	3	4	5		Median
An einer fest stehenden Position	<i>Absolutely not</i>	1	2	2	3	1	<i>Absolutely yes</i>	3
Nahe am beschriebenen Objekt			1	3	5			4
An der Stelle, an der der Benutzer steht			1	3	3	2		4

**Tabelle 7-12: Bewertungen der unterschiedlichen Möglichkeiten zur Positionierung von Metadaten**

Der Konfigurator nutzt alle drei verschiedenen Möglichkeiten zur Anzeige von Metadaten oder Fakten. Die Bewertungen der Probanden hinsichtlich der Varianten sind relativ gleichmäßig verteilt.

- *Wie wichtig ist es für Sie, dass die folgenden Faktoren noch weiter verbessert werden?*

Indikator		1	2	3	4	5		Median
Real-like material	<i>Less important</i>		1	1	3	4	<i>More important</i>	4*
Details on geometries			1	2	5	1		4
Ease to use			1	3	1	4		4
More meta-information			1	4	3	1		3
Better real environment		3	1	2	3			3

**Tabelle 7-13: Bewertungen der Funktionalitäten im VR-Konfigurator**

Bei der Bewertung der Faktoren, die verbessert werden sollten, werden den realitätsnahen Materialien, den Details der Geometrie sowie der Einfachheit der Bedienung große Bedeutungen zugemessen. Die Qualität der derzeitigen virtuellen Umgebung (die Halbsphäre) und der Meta-Informationen ist nach Auffassung der Probanden für den konkreten Anwendungsfall bereits ausreichend.

#### 7.4.5 Reliabilität und Analyse der Ergebnisse

Folgende Punkte fassen die laut Wilcoxon-Vorzeichentest als zuverlässig eingestuften Ergebnisse der Benutzbarkeitsstudie zusammen:

- Der VR-Konfigurator mit immersivem UI ist nützlicher und effizienter als vergleichbare Softwarelösungen mit 2D-UI.
- Der VR-Konfigurator mit immersivem UI ist effektiver und bietet bessere Anmutung als die gleiche Softwareanwendung auf einem Desktop-Anzeigesystem.
- Insbesondere die Funktionalitäten 1:1-Verhältnis, 3D-Animation und immersives Feeling erachten die Benutzer als wichtig.
- Realitätsnahes Material ist nach Meinung der Probanden ein wichtiger Faktor, der noch verbessert werden sollte.
- Zum Aufruf einer Funktion in der immersiven Umgebung werden Symbole in der Szene (3D-Icon) gerne genutzt.

Bezüglich der Beurteilung der **Arbeitseffizienz** fallen die Resultate bisheriger Studien (siehe Kapitel 1) überwiegend negativ für VR- bzw. 3D-Benutzungsoberflächen aus. Dabei wurden sie im Rahmen der Erledigung gleicher Aufgaben mit konventionellen 2D-UI verglichen. In der vorliegenden Studie stufen die Probanden die VR-Benutzungsoberfläche der Fahrzeugkonfiguration im Vergleich zur Nutzung eines 2D-UI für die gleichen Aufgaben als effizienter ein. Laut Wilcoxon-Test hat das Ergebnis einen P-Wert von 0,0351. Dieser Wert spricht für ein durchaus hohes Maß an Zuverlässigkeit. Dieses, von den bisherigen Studien abweichende Ergebnis zeigt, dass auch in immersiver Umgebung eine effiziente Arbeit möglich ist. Die Ergebnisse zeigen auch, dass die Benutzungsoberfläche exakt nach den besonderen Anforderungen der immersiven Anwendung designed und implementiert werden muss.

Als grobes Maß für die Arbeitseffizienz kann die Anzahl der notwendigen Operationen für einen Arbeitsschritt herangezogen werden. Um einen Optionswert auszuwählen, benötigt der Benutzer am CATER-Web-Konfigurator vier Maus-Klicks, beim VR-Konfigurator reichen drei Klicks (mit dem immersiven Eingabegerät). Zusätzlich sind beim Web-Konfigurator weitere Maus-Klicks notwendig, um zwischen den vorgegebenen Kamera-Positionen zu wechseln. Eine freie Änderung der Ansichtsposition wird in der VR durch natürliches Navigieren (Gehen und Körperbewegung) erreicht. In der Benutzerstudie wird aus zwei Gründen auf eine genaue Zeiterfassung verzichtet: Erstens benötigen die Probanden in vielen Fällen lange Zeit zum Lesen der Statistik, bevor sie sich für eine Option entscheiden. Der erste Testdurchgang dauert daher wesentlich länger, als der nachfolgende Vergleichstest. Zweitens ist das Erfahrungsniveau der Probanden in Bezug auf die Arbeit in immersiver Umgebung unterschiedlich. Ein Vergleich zwischen den Probanden liefert daher keine verwertbaren Ergebnisse.

Welchen Eindruck eine Benutzungsoberfläche auf den Benutzer macht, ist eine subjektive Fragestellung. Da das VR-System in der Regel mit High-End-Computerhardware aus-

gestattet ist und dem Benutzer so ein hochwertiges immersives Erlebnis bieten kann, punktet die VR-Technologie bei der **Anmutung** gegenüber 2D- und 3D-UI auf Desktop-Computern. Die Ergebnisse der vorliegenden Studie können dies bestätigen.

Für die Fahrzeugkonfiguration ist eine realitätsnahe optische Sicht eine wichtige Voraussetzung, um effektiv arbeiten zu können. Dabei können die Stereoskopie und 1:1-Skalierung für den Benutzer wichtig sein. Allerdings haben vorherige VR-Anwendungen oft Mängel beim Funktionsumfang, sodass der Vorsprung bei der **Effektivität** zum Großteil neutralisiert wird. Zu den Mängeln gehört z. B. manuelles Laden von verschiedenen Modellvarianten. Auch muss für Aktionen wie das Wechseln von Materialien die VR-Umgebung verlassen und die Änderungen an einem 2D-UI eingegeben werden. Mit der Entwicklung und Umsetzung des RD-Konzepts integriert der Fahrzeugkonfigurator der vorliegenden Arbeit alle notwendigen Funktionalitäten und Datenkommunikationen in das Anwendungsprogramm. Entsprechend positiv wird die Effektivität der VR-Technologie bewertet.

Bei den Funktionalitäten werden VR-spezifische Faktoren wie immersives Feeling und die 1:1-Skalierung als besonders wichtige Punkte angesehen. Ähnlich hoch wird die Datenbankverbindung eingeschätzt. Sie ist schließlich eine Voraussetzung für eine integrierte VR-Anwendung. Konkrete Datenformen wie die Statistik und die Anzeige eines Remote-Computers werden zwar ebenfalls von vielen Probanden hoch bewertet, können aber bei der Zuverlässigkeitsprüfung nicht punkten. Außerdem gibt es Probanden, die einzelne Datenformen nicht mögen oder nicht brauchen. Bezüglich der Statistik hat z. B. der Proband aus dem Marketing beim Gespräch nach der Erprobung geäußert, dass dieses Feature eine zentrale Bedeutung für die Entscheidungsfindung hat. Der Proband aus der Designabteilung beurteilt die Statistik dagegen als störend. Für seine Arbeit am virtuellen Fahrzeugmodell reicht das eigene Fachwissen aus.

Die Kanäle, über die verschiedene Beschreibungen erreichbar sind, werden von den Probanden als problemlos nutzbar angesehen und schon während des Testlaufs als selbstverständlich angenommen. Sie werden daher auch nicht signifikant als verbesserungswürdig bewertet. Dieses Ergebnis muss keineswegs negativ gewertet werden. Denn das Ziel des RD-Konzepts ist nicht, die Aufmerksamkeit des Benutzers auf die Informationskanäle zu ziehen. Vielmehr dienen die Informationskanäle lediglich als Hilfsmittel, um eine effektive, effiziente und einfach zu bedienende Benutzungsoberfläche zu erstellen.

Die **3D-Animation** stieß bei den Probanden auf das größte Interesse, obwohl sie im Fahrzeugkonfigurator nur einen Übergang zwischen zwei System-Zuständen darstellt (siehe Kapitel 6.5.2). Dieses überraschende Ergebnis wurde noch dadurch verstärkt, dass in den Gesprächen nach der Erprobung alle Teilnehmer die Animation als besonders beeindruckendes Feature angesprochen haben.

Die Vorteile der **3D-Icons**, die im RD-Konzept in Kapitel 1 eingeführt werden, erzielen in der Studie eine gute Zuverlässigkeit. Sie erreichen im Wilcoxon-Test den einzigen P-Wert unter der 0.01-Grenze (P-Wert in 0,007813) im Rahmen der Studie. Dieses Ergebnis ist allerdings auch nicht dahingehend zu werten, dass alle Funktionen in der 3D-Welt nur durch Icons aufgerufen werden sollten. Denn auch die anderen abgefragten Möglichkeiten zur Auswahl von Funktionen erhalten keine negativen Bewertungen. Zu einem ähnlichen Urteil kommt die Studie in Bezug auf die Varianten zur Anzeige von Informationen. Auch hier zeigen die Ergebnisse weder ausgeprägte Ablehnung noch Priorisierung der im RD-Konzept umgesetzten Möglichkeiten.

## 8 Zusammenfassung und Ausblick

Die Integration ist einer der wichtigsten Erfolgsfaktoren für den VR-Einsatz im industriellen Entwicklungs- und Produktionsprozess. Ausgereifte integrierte VR-Lösungen benötigen vielfältige Informationsformen, um neben der visuellen Szene auch die Beschreibung, Logik, Überwachung und Kontrollierbarkeit des zu integrierenden Prozesses in der VR-Anwendung darzustellen. In VR wurden sie allerdings bisher nicht in der gleichen Intensität wie die Erzeugung, Verarbeitung und Anzeige von Geometriedaten unterstützt bzw. erforscht, da die VR-Technologie von Anfang an gemeinsam mit der Darstellung von 3D-Modellen entwickelt wurde.

Zu Beginn der Arbeit wurde der Schwerpunkt darauf gelegt, die technische Realisierbarkeit vielfältiger Repräsentationsformen in VR-Systemen aufzuzeigen und deren Benutzbarkeit zu verbessern. Nach Recherchen zum neuesten Stand der Technik und anschließender Analyse der technischen Möglichkeiten wurde festgestellt, dass die meisten Repräsentationsformen schon in früheren Projekten einzeln untersucht bzw. realisiert wurden. Dass sie trotzdem nicht verbreitet eingesetzt werden, hat mehrere Ursachen: Zum einen konnte oder kann die Computer-Hardware die Randbedingungen nicht alle erfüllen. Z. B. war die Auflösung der Anzeigesysteme in der Vergangenheit noch so gering, dass ausführlichere Texte beim Layout der immersiven Anwendung bewusst vermieden wurden. Um solche Hindernisse zu überwinden, ist technischer Fortschritt notwendig. Zum anderen sind die Entwickler, die eine immersive Benutzungsoberfläche für spezielle Anwendungen auf Basis einer VR-Plattform konzipieren sollen, noch zu sehr den Grundformen verhaftet. Bislang lässt sich die Erstellung eines immersiven UI mit der Aufgabenstellung vergleichen, ein professionelles 2D-GUI unter den Vorbedingungen zu erstellen, dass nur eine Programmierschnittstelle mit Alphabet, Linien und Rastergraphik zur Verfügung steht. Auch die in der 2D-GUI selbstverständlichen Elemente wie Window-Rahmen, Menüs, Bilder usw. müsste der Entwickler bei Null beginnend selbst programmieren bzw. die Grundstrukturen erst noch „erfinden“. Daher dauert das Design eines immersiven UI in der Regel Monate, während das Layout eines 2D-GUI mit Unterstützung einer Klassenbibliothek innerhalb weniger Stunden erstellt wird.

Ausgehend von dieser Bestandsaufnahme wurde der Schwerpunkt der vorliegenden Arbeit bzw. der Entwicklung des **Richdata**(RD)-Konzepts auf drei zentrale Aspekte gelegt:

- Datenkonzept mit den zentralen Bestandteilen RD-Inhalt und RD-Repräsentation  
Mit Anwendung des RD-Konzepts werden die Datenstrukturen der Anwendung über vielfältige Repräsentationsformen (auch Geometriedaten) mit der Datenstruktur der virtuellen Szene verknüpft. Die RD-Datensätze können durch Komposition miteinander eine baumartige Hierarchie bilden, wobei die Koppelung zwischen der Anwendung und den Repräsentationen gültig bleiben muss. Auf diese Weise ist garantiert, dass die Zustände des zu integrierenden Prozesses



durch den Benutzer in der VR-Anwendung immer sichtbar bzw. kontrollierbar sind. Zusätzlich wird eine XML-Beschreibung in jeden RD-Datensatz integriert. So ist sichergestellt, dass die Inhalte der RD-Datensätze über eine Datenkommunikation mit externen Datenquellen bearbeitet und ausgetauscht werden können.

- Leitfaden zur Vorgehensweise

Parallel zur Ausarbeitung des RD-Konzepts wird ein Leitfaden erstellt, der erläutert, wie eine integrierte VR-Anwendung mit Hilfe des RD-Konzepts konstruiert werden kann. Auf Datenebene werden einerseits grundlegende Informationstypen (wie 3D-Text oder Audio) und Bausteine (wie 3D-Menü oder Tabellen) von der VR-Plattform angeboten. Andererseits wird die Datenstruktur des Anwendungszustands von oben nach unten zerlegt. Für das immersive Layout und die Interaktion gibt es bisher kein etabliertes Standard-Paradigma. Im RD-Konzept werden Vorschläge erörtert, die den Entwicklern bei der Orientierung helfen. Der Leitfaden umfasst auch die Analyse und Bestimmung der Voraussetzungen, welche eine VR-Plattform erfüllen muss, damit sie um das RD-Konzept erweitert werden kann.

- Verbindung mit VR-externen Quellen.

Als universell einsetzbare Kommunikationsmethode wird eine Schnittstelle für den Zugriff auf einen Webdienst implementiert. Datenbanken, die eine Service-Orientierte Architektur (SOA) besitzen bzw. eine Webdienst-Schnittstelle anbieten, können so von mit dem RD-Konzept erstellten VR-Anwendungen erreicht werden. Für unstrukturierte Informationsquellen wird eine allgemeine Methode implementiert, um den Remote-Bildschirm durch ein Netzwerk innerhalb der virtuellen Umgebung anzuzeigen und bedienen zu können.

Das RD-Konzept legt besonderen Wert auf nachhaltige Weiterentwicklung und Wiederverwendung. Auf Datenebene ist das Konzept offen für die Ergänzung um weitere Informationsformen. Eine Analyse unterschiedlicher Informationsformen wird in Bezug auf ihren technischen Reifegrad und ihre Verwendbarkeit im Rahmen der VR-Integration durchgeführt. Auf Prozessebene werden die beiden dominierenden Entwicklungsstrategien für VR-Lösungen, die Entwicklung von Standard-Software einerseits und die auf Projektbasis andererseits, miteinander verglichen. Die Ergebnisse bezüglich den Vor- und Nachteilen der beiden Strategien sind bei der Konzeption berücksichtigt. Das RD-Konzept bemüht sich, die Vorteile beider Entwicklungslinien zu kombinieren. Das Basissystem lässt sich mit Software-Bausteinen aus Projektanwendungen anreichern, wenn diese nach dem RD-Leitfaden entwickelt wurden.

Die Umsetzung des RD-Konzepts erfolgt sowohl auf Plattform- als auch auf Anwendungsebene. Im Rahmen der Ausarbeitung wird das Konzept auf der *Virtual Decision Plattform* (VDP) umgesetzt und anschließend als Anwendung ein Fahrzeugkonfigurator für das Projekt CATER entwickelt. Die Plattform VDP wird nach den im Leitfaden be-

schriebenen Kriterien untersucht. Dabei wurden einige bestehende Schwächen in der Verteilung und Ereignisübertragung entdeckt. Zur Behebung dieser Mängel wurden Lösungen erarbeitet und umgesetzt.

Der Fahrzeugkonfigurator wurde auf Basis der VDP implementiert. Die zentrale Datenstruktur der Anwendung ist die Fahrzeugkonfiguration, die in Optionen, Optionswerten, Benutzergruppen usw. bis hin zu den grundlegenden RD-Typen zerlegt werden kann. Der Konfigurator ist während der Laufzeit über die Schnittstelle für den Webdienst mit der CATER-Datenbank verbunden. Für den Zugriff auf konventionelle Benutzerschnittstellen wie Webseiten oder Präsentationsprogramme wird das Modul IVR-Panel implementiert. Dieses ist ein in die 3D-Szene integriertes interaktives 2D-Display.

Mit der erfolgreichen Umsetzung des RD-Konzepts auf der Ziel-Plattform und der Entwicklung der Anwendung wurde verifiziert, dass das RD-Konzept effektiv für die Erstellung integrierter VR-Anwendungen einsetzbar ist. Die Evaluierung des RD-Konzepts fand während und nach der Entwicklung der Anwendung für die Fahrzeugkonfiguration statt. Diese Anwendung wurde durch einen iterativen Prozess entworfen und implementiert. Bereits in der Anfangsphase wurde geprüft, ob sich die Anforderungen des konkreten Anwendungsfalls mit dem Ziel der Verifizierung des RD-Konzepts vereinbaren lassen. Nach jedem Entwicklungszyklus wurde die Vollständigkeit der Integration und der Nutzwert des Einsatzes von Richdata geprüft.

In der Schlussphase wurde eine Benutzerstudie unter Beteiligung von drei Unternehmen der Automobilindustrie durchgeführt. Insgesamt haben neun Probanden von Daimler, Fiat und Volvo (Truck) an der Studie teilgenommen.

Der VR-Konfigurator wurde durch Probandenbefragung hinsichtlich Effektivität und Effizienz mit konventionellen 2D-GUI verglichen. Der VR-Konfigurator konnte sich dabei bewähren. Insbesondere ist das positive Ergebnis hinsichtlich der Arbeitseffizienz von Bedeutung, da bisherige Benutzerstudien in ihren Probandentests eine deutliche Präferenz für 2D-GUI beobachtet hatten. Die Hypothese, dass auch in der VR eine effiziente Arbeit möglich ist, wenn die Benutzungsoberfläche exakt nach den besonderen Anforderungen der immersiven Anwendung konzipiert und implementiert wurde, kann zumindest für die implementierte Anwendung des Fahrzeugkonfigurators bestätigt werden.

RD-Funktionalitäten, wie z. B. die Datenbankverbindung und die 3D-Animationen, wurden von den Probanden gut akzeptiert. Allerdings haben einzelne RD-Repräsentationsformen, z. B. Statistikanzeigen, die aus 3D-Text und -Zahlen bestehen, und der Remote-Bildschirm weder signifikant positive noch signifikant negative Bewertungen erhalten. Dies muss allerdings nicht unbedingt als negativ gewertet werden, denn das Ziel des RD-Konzepts ist nicht, die Aufmerksamkeit des Benutzers auf die Informationskanäle zu ziehen, sondern mit deren Hilfe eine effektive, effiziente und einfach zu bedienende Benutzungsoberfläche zu erzielen.

Die Anwendung des RD-Konzepts liefert eine softwaretechnische Infrastruktur, um die Entwicklung von integrierten VR-Anwendungen zu fördern. Beim Vergleich mit den ausgereiften Softwarelösungen für ein 2D-GUI wird erkennbar, dass die hier vorgestellte Arbeit nur ein erster Schritt sein kann, der zu einer etablierten Methode führen könnte. Die Konzeption konzentriert sich auf die RD-Datenstruktur und den Leitfaden und weniger auf die Breite möglicher Anwendungsbausteine bzw. die Vielfalt der möglichen Repräsentationsformen. Eine ausgereifte Infrastruktur muss eine breite Basis anbieten können, sodass in diesem Bereich noch viel Arbeit zu leisten ist. Dies kann durch die Erprobung des RD-Konzepts in weiteren Anwendungen erreicht werden. Außerdem liegt der Leitfaden in der vorliegenden Arbeit noch in einer verhältnismäßig groben Form vor. Im Rahmen der Ausarbeitung der grundlegenden Konzeption musste er so allgemein gehalten werden, um die Allgemeingültigkeit nicht zu beeinträchtigen. Für die praktische Anwendung sollte er für die jeweiligen Anwendungsfelder konkretisiert und weiter detailliert werden. Dies würde auch dazu beitragen, die Arbeitseffizienz der Entwickler weiter zu steigern.

Der Fahrzeugkonfigurator wurde gemäß den Anforderungen von Ingenieuren der Automobilindustrie implementiert. Mit kleinen Änderungen bzw. Erweiterungen kann er problemlos auch im Bereich des Vertriebs oder als Anwendung für Fahrzeugkunden eingesetzt werden. Ebenso ist eine Anwendung in anderen Produktsegmenten vorstellbar. Ein immersives Erleben und die Einbindung vielfältiger Informationsformen stoßen auch bei Endverbrauchern auf großes Interesse. Ein Blick auf die Neuentwicklungen im Bereich der Spielkonsolen und die Verbreitung von 3D-Bildschirmen bzw. 3D-Fernsehern deutet bereits an, dass auch auf Systemen des Konsumentenmarktes ein VR-Konfigurator zur Anwendung kommen könnte.

## Abstract

*This dissertation develops a software infrastructure called the **Rich Data Concept**, in order to enrich the virtual environments with diverse representation forms and to foster the development of integrated Virtual Reality (VR) applications. Guidelines are elaborated and describe how to realize this concept through conventional VR platforms, employing the example of a commercial platform. This work then verifies the concept by using it to develop an application for vehicle configuration. Finally, the author conducted a user study involving potential users from the automotive industry. In this study the VR configurator's 3D interface is compared with conventional 2D graphical interfaces. The VR configurator based on the Rich Data Concept showed an improvement in usability and efficiency. The results could partly be verified to be statistically significant. The hypothesis that it will be possible to work efficiently in the VR if the user interface is designed and implemented for the particular requirements could be confirmed - at least for this application of the vehicle configurator.*

Integration is one of the key factors for successful Virtual Reality (VR) applications deployed in industrial development and production processes. Sophisticated, integrated VR solutions are supposed to support various forms of information, so that not only visualized scenes but also descriptions, business logic, and process monitoring can be integrated in a VR environment. In VR research, however, they have not been investigated with the same intensity as the rendering of geometric data, because the VR technology has been researched and developed primarily for stereoscopic representation of 3D models.

This research initially focused on technical feasibilities which enrich VR systems with a variety of representation forms and improve their usability. After researching the state of the art and performing subsequent analysis of the technical possibilities, it was recognized that most forms of representation had already been investigated or realized in previous projects. But even so, they are still not widely used in virtual environment for the following reasons: first, there were/are some technical challenges for computer hardware. For example, user interfaces of VR applications avoided text for a long time because the resolution of display systems was not big enough. Technological progress is the only way to overcome such obstacles. Second, developers designing an immersive user interface have to begin at a very basic level. It is comparable to someone trying to develop a professional 2D-GUI with only an alphabet and graphic lines. Layout patterns or elements like window frames, menus or icons are obvious in 2D-GUI, but they have to be "invented" by the developer himself for each VR application. Therefore, it typically

takes the designer of an immersive UI months, while one can create the layout of a 2D GUI with the support of a programming library within a few hours.

Based on this assessment, the investigation was diverted to provide a software infrastructure which enriches the virtual environment with diverse representation forms and fosters the development of integrated VR applications. This infrastructure, the **Rich Data Concept**, includes the following three key aspects:

- Data concept

Data structures of application logic, which could be represented in various forms, have to be able to associate with the data structure of the virtual scene. Rich data records can be composited into a tree-like hierarchy, where the coupling between the content and representations remains valid. This guarantees that the process state, in which the VR application should be integrated, is always visible and controllable by the user standing in a virtual environment. In addition, XML description is integrated in every rich data record. This ensures that the content of rich data can be processed and exchanged with external data sources.

- Guidelines

Parallel to the development of the rich data concept itself, guidelines are created to describe how an integrated VR application can be designed using this concept. On the one hand, fundamental information types (such as 3D text or audio) and patterns (such as 3D menu or tables) should be provided by VR platforms. On the other hand, data structures from application logic can be split top-down. Thus, no established standard paradigm for layout and immersive interactions is available. The rich data concept offers a proposal to help the developers find a first orientation. The guidelines also include a description which helps to determine whether a VR platform could be extended with rich data concept or if not, what is missing.

- Connection with external sources

This research implements an interface for accessing a web service as a universal method for communication with external data sources. If a database has a service-oriented architecture (SOA) and provides a web service interface, a rich data concept-based virtual reality application can call it up easily. For unstructured information, one implements a general method. Using remote display techniques, it is possible to transport monitor information through the network into the virtual environment and control applications running on the remote computer.

The rich data concept emphasizes extendibility and reusability. The concept remains open to additional types of representation forms. The research analyses different forms in terms of their technical maturity and their applicability in the context of VR integration. At the process level, it compares the two dominant development strategies for VR-

solutions: the development of both standard and project-based software. The work considers the advantages and disadvantages of both strategies in terms of the design of the concept, which seeks to combine the advantages of both lines. The basic system could be accumulated with software modules and features from project applications if they were developed following the guidelines of the rich data concept.

The rich data concept has to be deployed at both platform- and application-level before the first use. The investigation deploys the concept on the Virtual Decision Platform (VDP), a commercial virtual reality platform. That followed an integrated VR application – a vehicle configurator - has been developed using and thereby verifying the rich data concept. The VDP platform was analysed according to the criteria described in the guidelines. Weaknesses in terms of by events distribution and transmission were detected and improved.

The vehicle configurator was then implemented on the basis of the VDP. The central data structure of the vehicle configurator is the configuration which can be broken into options, option values, user groups, etc. down to the basic data types. The configurator is connected at run time with the CATER database via the web services interface. Furthermore, an integrated 2D display in the virtual environment is implemented to access conventional user interfaces, such as web pages, on desktop computers.

The successful development of the rich data concept on the target platform and the implementation of the application have verified that the rich data concept can be used effectively for the development of integrated VR applications. Evaluation of the rich data concept took place during and after the development of the application for vehicle configuration. This application was designed and implemented by an iterative process, checking in the initial phase whether or not it would be possible to verify the rich data concept in this specific use case. After each iterative cycle, research examined whether rich data were completely integrated and how users could benefit from the integrated rich data.

In the final phase, a user study was conducted by three companies in the automotive industry. Nine test persons from Daimler, Fiat and Volvo (trucks) participated in the study.

In a survey the VR-Configurator was compared with conventional 2D GUI for the same or similar use cases for effectiveness and efficiency. The VR-configurator proved itself. In particular, the positive result in terms of efficiency is important, because previous studies had observed that testers expressed a clear preference for 2D GUI. The hypothesis that it would be possible to work efficiently in the VR if the user interface were designed and implemented for the particular requirements could be confirmed - at least for this application of the vehicle configurator.

The tester assessed typical rich data functionalities such as database connectivity and 3D animations, as well. Some representation forms, such as remote screen or statistical

data consisting of 3D text and numbers, got either significantly positive or negative reviews. This should not be understood as negative, however, because the aim of the rich data concept is not to draw the attention of the user to the information channels, but to build effective, efficient and easy to use user interfaces easily.

The rich data concept provides a software infrastructure to foster the development of integrated virtual reality applications. When it is compared with the sophisticated software solutions for 2D GUI, it is clear that the work presented here is only a first step that could lead to an established method. The concept focuses on the rich data structure and guidance, and puts less emphasis on the spectrum of possible representation forms and patterns. A mature infrastructure needs a broad basis. This can be achieved by using the rich data concept in further projects or applications. In addition, the guidelines in the present work are relatively coarse, because the basic concept must be so general as not to affect general applicability. For a particular application field, it should be more specific and have a finer structure. This would also help to increase the efficiency of the developer.

The car configurator has been developed to meet the requirements of engineers from the automotive industry. With minor modifications or enhancements, one can easily apply it in the area of vehicle sales. It is also possible to use this configurator in other product segments. An immersive experience and the integration of diverse forms of information in VR could generate much interest among consumers, too. Looking at the new developments in the field of game consoles and the proliferation of 3D monitors and 3D TVs, one can already suggest that a VR configurator might also be applied to consumer markets.

## Literaturverzeichnis

- [A.R.T. 2007] A.R.T.: *Produktionsbeschreibung*, 2007. Online im Internet unter: <http://www.ar-tracking.de/A-R-T-cameras.142.0.html>. Letzter Besuch am: 06.01.2010.
- [Adobe 2006] Adobe: *Get started with intelligent 3D Adobe PDF documents*. Online im Internet unter: [http://www.adobe.com/designcenter/acrobat3d/articles/ac3dit\\_workflow08/ac3dit\\_workflow08.pdf](http://www.adobe.com/designcenter/acrobat3d/articles/ac3dit_workflow08/ac3dit_workflow08.pdf). Letzter Besuch am 30.09.2010.
- [AiLive 2008] AiLive: *LiveMove 2: Motion recognition and tracking*. Online im Internet unter: <http://www.aillive.net/lm2wii.html>. Letzter Besuch am: 06.01.2010.
- [AliceStreet 2009] AliceStreet: *Alice Street Conference Center - A real conference room on your desktop*. Online im Internet unter: [http://www.alicestreet.com/Library\\_2008/AliceStreet\\_Brochure.pdf](http://www.alicestreet.com/Library_2008/AliceStreet_Brochure.pdf). Letzter Besuch am 01.04.2010.
- [Andujar 2006] Andujar, C. und F. Argelaguet: *Friction surfaces: scaled ray-casting manipulation for interacting with 2D GUIs*. In: 12th Eurographics Symposium on Virtual Environments, Seite 101-108, Lisboa, 2006.
- [Andujar 2007] Andujar, C. und F. Argelaguet: *Virtual Pads: Decoupling Motor Space and Visual Space for Flexible Manipulation of 2D Windows within VEs*. In: Second IEEE Symposium on 3D User Interfaces, Charlotte, NC, 2007.
- [Ascension 2009] Ascension: *Flock of Birds*. Online im Internet unter: <http://www.ascension-tech.com/realtime/RTflockofBIRDS.php>. Letzter Besuch am: 05.01.2010.
- [ASME-Y14.26M] American Society Of Mechanical Engineers: *ASME-Y14.26M, Initial Graphics Exchange Specification (IGES)*. Letzter Besuch am: 1993.
- [Assenmacher 2006] Assenmacher, I., B. Hentschel, C. Ni, C. Bischof und T. Kuhlen: *Interactive Data Annotation in Virtual Environments*. In: 12th Eurographics Symposium on Virtual Environments, Lisbon, Portugal, 2006.
- [Barsoum 2008] Barsoum, E. und F. Kuester: *Towards adaptive Web scriptable user interfaces for virtual environments*, in: *Virtual Reality*, Band 12(1), Seite 55-64, 2008.



- [Bell 2001] Bell, B., S. Feiner und T. Höllerer: *View Management for Virtual and Augmented Reality*. In: Proceedings of the ACM symposium on User interface software and technology, Seite 101 - 110, 2001.
- [Bell 2002] Bell, B., T. Höllerer und S. Feiner: *An Annotated Situation-Awareness Aid for Augmented Reality*. In: Proceedings of the 15th annual ACM symposium on user interface software and technology, Seite 27-30. ACM, New York, 2002.
- [Bender 2009] Bender, M., P. Rally, C. Rimpau und M. Wiedemann: *Einleitung* im Buch: G. Reinhart und D. Spath (Hg.), *Auftragsprozesse in der kundenintegrierten Montage*, Seite 1-3. Fraunhofer IRB Verlag, ISBN: 978-3-8167-7882-0, Stuttgart, 2009.
- [Besner 2005] Besner, M.: *Alias FBX Initiative White Paper*. Online im Internet unter: [www.microcad.info/brochure\\_productos/FBX\\_BRO.pdf](http://www.microcad.info/brochure_productos/FBX_BRO.pdf). Letzter Besuch am 20.10.2010.
- [Blach 1998] Blach, R., J. Landauer, A. Rösch und A. Simon: *A highly flexible virtual reality system*, in: *Future Gener. Comput. Syst.*, Band 14(3-4), Seite 167-178, 1998.
- [Blau 2007] Blau, H., S. Eicker, A. Hofmann und T. Spies: *Reifegradüberwachung von Software* im *ICB Research Report*, Band 20, Seite 1-40. Institut für Informatik und Wirtschaftsinformatik der Universität Duisburg Essen, 2007.
- [Boehm 1986] Boehm, B.: *A spiral model of software development and enhancement*, in: *SIGSOFT Softw. Eng. Notes*, Band 11(4), Seite 14-24, 1986.
- [Börner 2000] Börner, R.: *Autostereoskopische Einprosonenbildschirme mit Trackingsystem*. Heinrich-Hertz-Institut für Nachrichtentechnik, Berlin, 2000.
- [Bowman 2004] Bowman, D., E. Kruijff, J. J. LaViola und I. Poupyrev: *3D User Interfaces - Theory and Practice*, Addison-Wesley, ISBN: 978-0201758672, 2004.
- [Bowman 1999] Bowman, D. A., L. F. Hodges, D. Allison und J. Wineman: *The Educational Value of an Information-Rich Virtual Environment*, in: *Teleoperators and Virtual Environments*, Band 8(3), Seite 317-331, 1999.
- [Bowman 2003] Bowman, D. A., C. North, J. Chen, N. F. Polys, P. S. Pyla und U. Yilmaz: *Information-rich virtual environments: theory, tools, and research agenda*. In: Proceedings of the ACM symposium on virtual reality software and technology, Seite 81 - 90. ACM, 2003.
- [Brüderlin 2007] Brüderlin, B., M. Heyer und S. Pfützner: *Interviews3D: A Platform for Interactive Handling of Massive Data Sets*, in: *IEEE Comput. Graph. Appl.*, Band 27(6), Seite 48-59, 2007.

- [Bues 2003] Bues, M., R. Blach und F. Haselberger: *Sensing surfaces: bringing the desktop into virtual environments*. In: Proceedings of the workshop on Virtual environments 2003. ACM, Zurich, Switzerland, 2003.
- [Bues 2009] Bues, M.: *Ein System zum visuell hochwertigen Echtzeitrendering für Virtuelle Umgebungen* Institut für Arbeitswissenschaft und Technologiemanagement Basel, Universität Stuttgart, Dissertation, ISBN: 978-3-939890-49-2, 2009.
- [Bullinger 1999] Bullinger, H. J., R. Blach und R. Breining: *Projection Technology Applications in Industry - Theses for the design and use of the current Tools*. In: 3. International Immersive Projection Technology Workshop, Seite 23-32. Springer, Berlin, 1999.
- [Burdea 2003] Burdea, G. C. und P. Coiffet: *The Three I's of Virtual Reality* im Buch: *Virtual Reality Technology* Band 1, Seite 3-4. John Wiley & Sons Inc., ISBN: 978-0471086321, New Jersey, 2003.
- [Card 1999] Card, S. K., J. Mackinlay und B. Shneiderman: *Readings in Information - Visualization: Using Vision to Think (Interactive Technologies)*, Academic Press, ISBN: 978-1558605336, San Diego, 1999.
- [CATER 2008] CATER: *CATER will help you find the car just right for you*, 2008. Online im Internet unter: <http://uranus.iti.gr/cater/car/index.html>. Letzter Besuch am: 12.12.2008.
- [Cheshire 1996] Cheshire, S.: *Lantency and the Quest for Interactivity*. In: The Synchronous Person-to-Person Interactive Computing Environments Meeting. Volpe Welty Asset Management, San Francisco, 1996.
- [Christensen 2001] Christensen, E., F. Curbera, G. Meredith und S. Weerawarana: *Web Services Description Language (WSDL) 1.1*. Online im Internet unter: <http://www.w3.org/TR/wsdl>. Letzter Besuch am: 05.01.2010.
- [Cruz-Neira 1993] Cruz-Neira, C., D. J. Sandin und T. A. DeFanti: *Surround-screen projection-based virtual reality: the design and implementation of the CAVE*. In: Proceedings of the 20th annual conference on Computer graphics and interactive techniques. ACM, Anaheim, CA, 1993.
- [CyberGlove 2009] CyberGlove: *Homepage*. Online im Internet unter: <http://www.cyberglovesystems.com>. Letzter Besuch am: 05.01.2010.
- [Dahm 2003] Dahm, P. und P. Widmann: *GraLab: Das Graphenlabor* Institut für Softwaretechnik der Universität Koblenz, 2003. Online im Internet unter: <http://www.uni-koblenz->

landau.de/koblenz/fb4/institute/IST/AGEbert/MainResearch/Graphentechnologie/GraLab. Letzter Besuch am: 15.02.2009.

- [Dam 1997] Dam, A. v.: *Post-WIMP user interfaces*, in: *Communications of the ACM*, Band 40(2), Seite 63-67, 1997.
- [Dangelmaier 2001] Dangelmaier, M.: *Ein Verfahren zur ergonomischen Bewertung des Fahrerplatzes von Personenkraftwagen* Institut Arbeitswissenschaft und Technologiemanagement, Stuttgart, Universität Stuttgart, Dissertation, ISBN 3-931388-69-7, 2001.
- [Dangelmaier 2008] Dangelmaier, M. und F. Haselberger: *Immersive virtuelle Begehung und Inspektion*. In: *Augmented & Virtual Reality in der Produktentstehung: Grundlagen, Methoden und Werkzeuge*, Paderborn, 2008.
- [Davis 1992] Davis, A. M.: *Operational Prototyping: A New Development Approach*, in: *IEEE Softw.*, Band 9(5), Seite 70-78, 1992.
- [Davis 1997] Davis, S.: *Future Perfect*, Addison-Wesley, ISBN: 978-0201327953, Reading, Massachusetts, 1997.
- [Eberle 2007] Eberle, J.: *Entwicklung eines Datenmanagementprozesses zur automatisierten Implementierung von CAD-Daten in Virtual Reality bei der DaimlerChrysler AG* Studiengang Informatik, Stuttgart, Hochschule Mannheim, Bachelorarbeit, 2007.
- [Elmqvist 2003] Elmqvist, N.: *3Dwm: A Platform for Research and Development of Three-Dimensional User Interfaces*. Department of Computing Science, Chalmers University of Technology and Göteborg University, Göteborg, 2003.
- [Ferwerda 2001] Ferwerda, J. A.: *Elements of Early Vision for Computer Graphics*, in: *IEEE Comput. Graph. Appl.*, Band 21(5), Seite 22-33, 2001.
- [Fishkind 2009] Fishkind, A.: *Made in IBM Labs: IBM Creates Software for Holding Face-to-Face Meetings in Virtual Worlds*, in: *IBM Media Relations*, 2009.
- [Fröhlich 2000] Fröhlich, B. und J. Plate: *The Cubic Mouse - A new Device for Three-Dimensional Input*. In: *ACM CHI 2000*, Seite 526-531, 2000.
- [Gamma 1995] Gamma, E., R. Helm, R. Johnson und J. Vlissides: *Design Patterns - Elements of Reusable Object-Oriented Software*, Addison Wesley, 1995.
- [Grimm 2009] Grimm, S.: *Fotorealismus - interaktiv und in Echtzeit*, in: *IDO:Communicate*, Band 2, Seite 16-17, 2009.
- [Gu 2002] Gu, X., S. J. Gortler und H. Hoppe: *Geometry images*, in: *ACM Trans. Graph.*, Band 21(3), Seite 355-361, 2002.
- [Häfner 2002] Häfner, U.: *Entwicklung eines kabellosen Eingabesystems für immersive Umgebungen* Institut für Arbeitswissenschaft und

Technologiemanagement IAT, Stuttgart, Universität Stuttgart, Dissertation, 2002.

- [Harmon 1996] Harmon, R., W. Patterson, W. Ribarsky und J. Bolter: *The Virtual Annotation System*. In: Proceedings of the IEEE Virtual Reality Annual International Symposium, Seite 239-245, 1996.
- [Hauber 2006] Hauber, J., H. Regenbrecht, M. Billinghurst und A. Cockburn: *Spatiality in videoconferencing: trade-offs between efficiency and social presence*. In: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work. ACM, 2006.
- [He 2003] He, H.: *What is Service-Oriented Architecture*. Online im Internet unter: <http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html>. Letzter Besuch am: 06.01.2010.
- [Herbig 2005] Herbig, G.: *Infos rund um die Stereofotografie*. Online im Internet unter: [http://www.herbig-3d.de/german/stereofotografie\\_frame.htm](http://www.herbig-3d.de/german/stereofotografie_frame.htm). Letzter Besuch am: 27.06.2005.
- [Herrmann 2009] Herrmann, M., F. Hamisch, B. Eschrich, H. Leitner und P. Gaschler: *3D-Tracking using Wii Remotes*, 2009. Online im Internet unter: <http://www.youtube.com>
- [HHI 2010] HHI: *3D Displays für stereoskopische Inhalte*. Interactive media human factors. Heinrich-Hertz-Institut, 2010. Online im Internet unter: <http://www.hhi.fraunhofer.de/de/abteilungen-am-hhi/interaktive-medien-human-factors/uebersicht/>. Letzter Besuch am: 08.07.2010.
- [Hills 2005] Hills, A., J. Hauber und H. Regenbrecht: *Videos in space: a study on presence in video mediating communication systems*. In: Proceedings of the 2005 international conference on Augmented tele-existence, 2005.
- [ICIDO 2009] ICIDO: *Handbuch des Visual Decision Platform 2008 SP1*. ICIDO GmbH, Stuttgart, 2009.
- [InvenSense 2009] InvenSense: *Homepage*. Online im Internet unter: <http://www.invensense.com/>. Letzter Besuch am: 06.01.2010.
- [ISO-9241-110] International Organization for Standardization: *ISO-9241-110, Ergonomie der Mensch-System-Interaktion*. Letzter Besuch am: 2006.
- [ISO-10303] International Organization for Standardization: *ISO-10303, Standard for the exchange of product model data*. Letzter Besuch am: 2004.

- [ISO-14772] International Organization for Standardization: ISO-14772, *The Virtual Reality Modeling Language - VRML 97*. Letzter Besuch am: 2003.
- [ISO-19775-1.2] International Organization for Standardization: ISO-19775-1.2, *X3D Architecture and base components Edition 2*. Letzter Besuch am: 2008.
- [ITU 1992] ITU: *Terminal Equipment and Protocols for Telematic Services*. Information Technology - Digital Comopression and Coding of Continuous-tone Still Images - Requirements and Guidelines, Joint Photographic Experts Group (JPEG).1992.
- [Jiao 2007] Jiao, R. J., Q. Xu, J. Du, Y. Zhang, M. Helander, H. M. Khalid, P. Helo und C. Ni: *Analytical affective design with ambient intelligence for mass customization and personalization*, in: *International Journal of Flexible Manufacturing Systems*, Band 19, Seite 26, 2007.
- [JT-Open] JT-Open, *JT Open - An open platform for visualization, collaboration and data sharing across the product lifecycle*. Online im Internet unter: [http://www.plm.automation.siemens.com/en\\_us/Images/2949\\_tcm1023-7642.pdf](http://www.plm.automation.siemens.com/en_us/Images/2949_tcm1023-7642.pdf). Letzter Besuch am: 2008.
- [Kreimer 2004] Kreimer, R.: *Altenpflege: menschlich, modern und kreativ - Grundlagen und Modelle einer zeitgemäßen Prävention, Pflege und Rehabilitation*, Schlütersche Verlagsgesellschaft mbH & Co. KG, ISBN: 3-89993-105-x, Neuenkirchen, 2004.
- [Larson 2000] Larson, K., M. v. Dantzich, M. Czerwinski und G. Robertson: *Text in 3D: some legibility results*. In: CHI '00 extended abstracts on Human factors in computing systems. ACM, The Hague, The Netherlands, 2000.
- [Lee 2007] Lee, J. C.: *Head Tracking for Desktop Virtual Reality Displays using the Wii Remote*, 2007. Online im Internet unter: <http://www.youtube.com/>. Letzter Besuch am: 06.01.2010.
- [Lentz 2007] Lentz, T., D. Schröder, M. Vorländer und I. Assenmacher: *Virtual reality system with integrated sound field simulation and reproduction*, in: *EURASIP Journal on Advances in Signal Processing*, Band 2007, Seite 187, 2007.
- [Lim 2007] Lim, T. Y., H. M. Khalid und C. Ni: *Citarasa based vehicle planning system*, in: *Industrial Engineering and Engineering Management*, Seite 1297-1301, 2007.
- [LindenLab 2009] LindenLab: *2nd Life*. Online im Internet unter: <http://secondlife.com/>. Letzter Besuch am: 15.10.2009.
- [Losasso 2003] Losasso, F., H. Hoppe, S. Schaefer und J. Warren: *Smooth geometry images*. In: *Proceedings of the 2003 Eurographics/ACM*

- SIGGRAPH symposium on Geometry processing. Eurographics Association, Aachen, Germany, 2003.
- [Marks 1991] Marks, J. und S. Shieber: *The Computational Complexity of Cartographic Label Placement*, 1991.
- [Mechdyne 2008] Mechdyne: *Produktinformation*. Online im Internet unter: <http://www.mechdyne.com/integratedSolutions/displaySystems/peripherals.htm>. Letzter Besuch am: 05.01.2010.
- [Mercedes-Benz 2010] Mercedes-Benz: *Konfigurieren Sie Ihr Modell*, 2010. Online im Internet unter: [http://www.mercedes-benz.de/content/germany/mpc/mpc\\_germany\\_website/de/home\\_mpc/passengercars/home/new\\_cars/configurator\\_showroom.flash.html](http://www.mercedes-benz.de/content/germany/mpc/mpc_germany_website/de/home_mpc/passengercars/home/new_cars/configurator_showroom.flash.html). Letzter Besuch am: 09.02.2010.
- [Nagl 1998] Nagl, M.: *Softwaretechnik - Methodisches Programmieren im Großen*, Springer-Verlag GmbH, 1998.
- [NCSS 2010] NCSS: *Statistical Analysis and Graphics*, 2010. Online im Internet unter: <http://www.ncss.com/ncss.html>. Letzter Besuch am: 14.02.2010.
- [Ni 2006] Ni, C.: *Entwicklung eines Dokumentationssystems fuer die interaktive Annotation von Stroemungsmerkmalen in virtuellen Umgebungen* Virtual Reality Centre Aachen, Aachen, RWTH University Diplomarbeit, 2006.
- [Nielsen 1994] Nielsen, J.: *Usability Engineering*, Morgan Kaufmann, ISBN: 978-0125184069, 1994.
- [Nielsen 1998] Nielsen, J.: *2D is Better Than 3D*. Online im Internet unter: <http://www.useit.com/alertbox/981115.html> Letzter Besuch am: 01.04.2010.
- [Nintendo 2006] Nintendo: *Produktinformation für Wii-Konsole*. Online im Internet unter: [http://www.nintendo.de/NOE/de\\_DE/wii\\_54.html](http://www.nintendo.de/NOE/de_DE/wii_54.html). Letzter Besuch am: 06.01.2010.
- [Piantanida 1993] Piantanida, T., D. K. Boman und J. Gille: *Human perceptual issues and virtual reality*, in: *Virtual Reality Systems 1*, Band 1, Seite 10, 1993.
- [Piller 2002] Piller, F. T. und C. M. Stotko: *Mass Customization - four approaches to deliver customized products and services with mass production efficiency*. In: *Proceeding 2002 IEEE International Engineering Management Conf. Managing Technology for the New Economy*, Seite 773-778, 2002.
- [Piller 2006] Piller, F. T.: *Mass Customization - Ein wettbewerbsstrategisches Konzept im Informationszeitalter*, Arnold Picot, Ralf Reichwald und Egon Franck (Hg.), Deutscher Universitäts-Verlag, ISBN: 3-8350-0355-0, 2006.

- [Polys 2004a] Polys, N., M. Moldenhauer, A. Ray, C. Dandekar, C. North und D. Bowman: *Snap2Diverse: Coordinating Information Visualizations and Virtual Environments*. In: Proceedings of SPIE Visualization and Data Analysis, 2004a.
- [Polys 2004b] Polys, N. F., D. A. Bowman und C. North: *Information-Rich Virtual Environments: Challenges and Outlook*. In: NASA Workshop on the Knowledge Integrating Virtual Iron Bird, 2004b.
- [Polys 2005] Polys, N. F., S. Kim und D. A. Bowman: *Effects of information layout, screen size, and field of view on user performance in information-rich virtual environments*. In: Proceedings of the ACM symposium on Virtual reality software and technology. ACM, Monterey, CA, USA, 2005.
- [Polys 2006a] Polys, N. F., S. Kim und D. A. Bowman: *Effects of information layout, screen size, and field of view on user performance in information-rich virtual environments*. Computer Animation and Virtual Worlds %L 077:Polys:2006. Band 18: Seite. 19-38.2006a.
- [Polys 2006b] Polys, N. F., L. Shupp, J. Volpe, V. Glina und C. North: *The Effects of Task, Task Mapping, and Layout Space on User Performance in Information-Rich Virtual Environments*. Computer Science of VirginiaTech, Blacksburg, 2006b.
- [Polys 2007] Polys, N. F., M. Shapiro und K. Duca: *IRVE-Serve: A Visualization Framework for Spatially-Registered Time Series Data*. In: The Web3D 2007 Symposium, Seite 15-18. ACM SIGGRAPH, 2007.
- [Reenskaug 1979] Reenskaug, T.: *Models-Views-Controllers*, in, 1979.
- [Regenbrecht 2003] Regenbrecht, H., C. Ott, M. Wagner, T. Lum, P. Kohler, W. Wilke und E. Müller: *An Augmented Virtuality Approach to 3D Videoconferencing*. In: Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality, 2003.
- [Regenbrecht 2004] Regenbrecht, H., T. Lum, P. Kohler, C. Ott, M. Wagner, W. Wilke und E. Mueller: *Using augmented virtuality for remote collaboration*, in: *Presence - Teleoperators and virtual environments*, Band 13(3), Seite 338-354, 2004.
- [Reimersdahl 2000] Reimersdahl, T., T. Kuhlen, A. Gerndt, J. Henrichs und C. Bischof: *ViSTA: a multimodal, platform-independent VR-Toolkit based on WTK, VTK and MPI*. In: Proceedings of Fourth International Immersive Projection Technology Workshop, 2000. Online im Internet unter: [http://www.rz.rwth-aachen.de/aw/cms/rz/Themen/Virtuelle\\_Realitaet/research/~piz/vr\\_software\\_vista/?lang=de](http://www.rz.rwth-aachen.de/aw/cms/rz/Themen/Virtuelle_Realitaet/research/~piz/vr_software_vista/?lang=de).
- [Reinhard 2006] Reinhard, E., G. Ward, S. Pattanaik und P. Debevec: *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*, Elsevier, ISBN: 978-0-12-585263-0, San Francisco, 2006.

- [Reitmayr 2001] Reitmayr, G. und D. Schmalstieg: *A Wearable 3D Augmented Reality Workspace*. In: Proceedings of IEEE Wearable Computers (ISCW 2001), Seite 165. IEEE Computer Society Washington, 2001.
- [Reitmayr 2003] Reitmayr, G. und D. Schmalstieg: *Data Management Strategies for Mobile Augmented Reality*. Online im Internet unter: <http://www.ims.tuwien.ac.at/media/documents/publications/reitmayrstars2003.pdf>. Letzter Besuch am 05.09.2010.
- [Royce 1987] Royce, W. W.: *Managing the development of large software systems: concepts and techniques*. In: Proceedings of the 9th international conference on Software Engineering. IEEE Computer Society Press, Monterey, California, United States, 1987.
- [Schilling 2006] Schilling, A., S. Kim, D. Weissmann, Z. Tang und S. Choi: *CAD-VR geometry and meta data synchronization for design review applications*, in: *Journal of Zhejiang University - Science A*, Band 7, Seite 1482-1491, 2006.
- [Schirra 2008] Schirra, R.: *Entwicklung eines Systems zur immersiven Arbeitssystemgestaltung mit CAD-Schnittstelle*. Betreuer: D. Spath und E. Westkämper Institut Arbeitswissenschaft und Technologiemanagement, Stuttgart, Universität Stuttgart, Dissertation, 2008.
- [Schupp 1994] Schupp, G., K. Naßhan, U. Stephenson und W. Fasold: *Hineinhören in geplante Räume*, in: *IBP-Mitteilung*, Band 21(250), 1994.
- [Selvage 2005] Selvage, M., D. Wolfson und J. Handy-Bosma: *Information management in Service-Oriented Architecture, Part 2: Explore the different approaches to information management in SOA*. Online im Internet unter: <http://www.ibm.com/developerworks/webservices/library/ws-soa-ims2/>. Letzter Besuch am: 14.03.2007.
- [Siegel 2001] Siegel, S.: *Nichtparametrische statistische Methoden*, Verlag Dietmar Klotz, ISBN: 3-88074-102-6., 2001.
- [Siemens 2008] Siemens: *JT Open - An open platform for visualization, collaboration and data sharing across the product lifecycle*. 2008.
- [smartMeeting 2006] smartMeeting: *Web conferencing and live presentation tools*. Online im Internet unter: <http://www.smartmeeting.com/>. Letzter Besuch am: 21.12.2006.
- [Smith 1989] Smith, D. C., C. Irby, R. Kimball, B. Verplank und E. Harslem: *Designing the Star user interface (1982)* im Buch: *Perspectives on the computer revolution*, Seite 261-283. Ablex Publishing Corp., ISBN: 0-89391-369-3, 1989.



- [Sodnik 2006] Sodnik, J., S. Tomazic, R. Grasset und A. Duenser: *Spatial Sound Localization in an Augmented Reality Environment*. In: Proceedings of the 18th Australia conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments, Seite 111 - 118, 2006.
- [Spath 2008] Spath, D., J. Lentjes und F. Haselberger: *Virtual Engineering - mit virtuellen Techniken die Produktentwicklung beschleunigen*. In: F&K 2008 - Fertigungstechnisches Kolloquium, Seite 361-376, Stuttgart, 2008.
- [Spath 2009] Spath, D. und J. Lentjes: *Virtuelle Realität als Werkzeug in der Digitalen Produktion*, in: ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb, Band 01-02, Seite 84-88, 2009.
- [Stefani 2003] Stefani, O. und J. Rauschenbach: *3D input devices and interaction concepts for optical tracking in immersive environments*. In: Proceedings of the ACM Virtual environments 2003, Seite 317 - 318, 2003.
- [Stein 2008] Stein, T. und X. Decoret: *Dynamic label placement for improved interactive exploration*. In: Proceedings of the 6th international symposium on Non-photorealistic animation and rendering. ACM, Annecy, France, 2008.
- [Stelzer 2009] Stelzer, R., D. Petermann, B. Saske und W. Steger: *Kollaborationsumgebung für CAD-VR*, in: ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb, Band 01-02, 2009.
- [Streubel 2008] Streubel, F.: *Analyse und Umsetzung bekannter und neuer Interaktionstechniken unter Verwendung des Fingertracking Device der Firma A.R.T.* Professur für Graphische Datenverarbeitung und Visualisierung, Stuttgart, Technische Universität Chemnitz, Diplomarbeit, 2008.
- [Sturman 1989] Sturman, D. J., D. Zeltzer und S. Pieper: *Hands-on interaction with virtual environments*. In: Proceedings of the 2nd annual ACM SIGGRAPH symposium on User interface software and technology. ACM, Williamsburg, Virginia, United States, 1989.
- [TixeoSoft 2008] TixeoSoft: *Product Sheet for Meeting3D*, 2008. Online im Internet unter: [www.tixeo.com](http://www.tixeo.com). Letzter Besuch am: 01.04.2010.
- [Tsang 2003] Tsang, M., G. W. Fitzmzurice, G. Kurtenbach, A. Khan und B. Buxton: *Boom Chameleon: Simultaneous capture of 3D viewpoint, voice and gesture annotations on a spatially-aware display*. In: Proceedings of the ACM SIGGRAPH '03 (Special issue), Seite 698 - 698, 2003.
- [Tsang 2002] Tsang, M. C.: *Boom Chameleon: Simultaneous capture of 3D viewpoint, voice and gesture annotations on a spatially-aware display*. 2002.

- [UltraVNC 2009] UltraVNC: *UltraVNC - Remote Control Software for all*, 2009. Online im Internet unter: [www.uvnc.com](http://www.uvnc.com). Letzter Besuch am: 01.04.2010.
- [Vasudevan 2001] Vasudevan, V.: *A Web Services Primer*. Online im Internet unter: <http://www.xml.com/lpt/a/760>. Letzter Besuch am: 13.03.2010.
- [Verlinden 1993] Verlinden, J., J. Bolter und C. v. d. Mast: *Voice Annotation: Adding Verbal Information to Virtual Environments*. In: *Proceeding of the European Simulation Symposium*, Seite 60-69, 1993.
- [ViaVoice 2005] ViaVoice: *IBM ViaVoice Standard Edition, Version 10*. . Online im Internet unter: <http://www.nuance.com/viavoice/standard/>. Letzter Besuch am: 12.02.2010.
- [W3C-XML] W3C XML Core Working Group: *W3C-XML, Extensible Markup Language (XML) 1.0 (Fourth Edition)*. Online im Internet unter: <http://www.w3.org/TR/2006/REC-xml-20060816/>. Letzter Besuch am: 2006.
- [Waack 1985] Waack, F. G.: *Stereo Photography*. An Introduction to Stereo Photo Technology and practical Suggestions for Stereo Photography. Berlin, Selbstverlag.1985.
- [Wanda 2009] Wanda: *Produktinformation*. Online im Internet unter: <http://www.ascension-tech.com/realtime/wanda.php>. Letzter Besuch am: 05.01.2010.
- [Wandell 1995] Wandell, B. A.: *Foundations of vision*, Sinauer Associates, ISBN: 0-87893-853-2, Sunderland, 1995.
- [Wernecke 1994] Wernecke, J. und O. I. A. Group: *The Inventor Toolmaker - Extending Open Inventor, Release 2*, ISBN: 978-0201624939, 1994.
- [Westkämper 2007] Westkämper, E., R. Becker, A. Grzesiak, A. Henning und M. Kempf: *Qualitätsmanagement im Rapid Prototyping* im Buch: B. Bertsche und H.-J. Bullinger (Hg.), *Entwicklung und Erprobung innovativer Produkte - Rapid Prototyping*, Seite 159-184. Springer, ISBN: 978-3-540-69879-1, 2007.
- [Wikipedia 2010] Wikipedia: *Wikipeda - Die freie Enzyklopädia*. Online im Internet unter: [www.wikipedia.de](http://www.wikipedia.de). Letzter Besuch am: 15.10.2010.
- [Wimmer 2004] Wimmer, P.: *Aufnahme und Wiedergabe stereoskopischer Videos im Anwendungsbereich der Telekooperation* Institut für Telekooperation Linz, Johannes Kepler Universität, Diplomarbeit, 2004.
- [Wright 2008] Wright, K. und D. Gracanin: *An approach to study of hypermedia information navigation in virtual environments (Poster)*. In: *Proceedings of the 2008 IEEE Symposium on 3D User Interfaces*, 2008.

- [Zimmerman 1987] Zimmerman, T. G., J. Lanier, C. Blanchard, S. Bryson und Y. Harvill:  
*A hand gesture interface device*, in: *SIGCHI Bull.*, Band 18(4), Seite  
189-192, 1987.

## Anhang

### Anhang A: Fragebogen der Benutzbarkeitsstudie

# Information Society Technologies (IST) Programme



Computerised Automotive Technology Reconfiguration System for Mass Customization

CATER

IST-5-035030-STREP

USER ACCEPTANCE & USABILITY ASSESSMENT – OEM REPRESENTATIVE

User ID (Name/Surname if not ID): \_\_\_\_\_

Date: \_\_\_\_\_

Site: \_\_\_\_\_

Evaluator/supervisor name: \_\_\_\_\_

Time and day of test: \_\_\_\_\_

ID's of Real-Life scenarios tried: \_\_\_\_\_

## Part 1: User Acceptance and Functionalities utility assessment

This part of the questionnaire aims to detect the usefulness of the functionalities of the CATER system you tried as well as your willingness to have and/or pay in order to own it. This questionnaire is to be completed for each single module you will assess and after you run all foreseen real-life evaluation scenarios for this specific module.

What is your overall judgment of the system you tried, on the following eight items?

### 1. My judgements of the system are ... (please tick a box on every line)

- |   |             |         |             |
|---|-------------|---------|-------------|
| 1 | useful      | _ _ _ _ | useless     |
| 2 | pleasant    | _ _ _ _ | unpleasant  |
| 3 | bad         | _ _ _ _ | good        |
| 4 | nice        | _ _ _ _ | annoying    |
| 5 | effective   | _ _ _ _ | superfluous |
| 6 | irritating  | _ _ _ _ | likeable    |
| 7 | assisting   | _ _ _ _ | worthless   |
| 8 | undesirable | _ _ _ _ | desirable   |

### 2. Would you consider using the system in your work?

Absolutely not ☐☐☐☐☐ Absolutely, yes

Comment: .....  
.....

### 3. Up to how much would you be willing to pay as maximum to have the system you have just tried?

For installation. Please specify in Euros: .....€

For maintenance. Please specify in Euros: .....€

**4. In what situation(s) would you consider to use this CATER module?**

- ☐ Getting inspiration
- ☐ Follow market trends
- ☐ Looking for information on specific customer preferences
- ☐ Making a design choice
- ☐ Other, please specify: .....
- ☐ Other, please specify: .....

**5. Would the system help improving efficiency/accuracy/quality in your work?**

Absolutely not ☐☐☐☐☐ Absolutely, yes

Comment: .....

.....

**6. Would you rely on the system to give you relevant information?**

Not at all ☐☐☐☐☐ Yes, very much

Comment: .....

.....

**7. Does the system handle any information that you think is sensitive, that you would like not to be associated with your name/person/company?**

☐ Yes      ☐ No

If yes, please explain what .....

.....

**8. Do you think that this module contradicts with the organisational, e-business or other regulations of your company and, thus, is not usable or beneficial for you?**

☐ Yes    ☐ No

If yes, in which way? Please specify:

.....

.....

.....

.....

**9. Do you think that this module concerns users from other departments of your company/organisation?**

☐ Yes    ☐ No

If yes, which ones? Please specify:

.....

.....

.....

.....

**10. Which are, in your opinion, the main advantages of the CATER module?**

.....

.....

.....

.....

.....

.....

**11. Which are, in your opinion, the main disadvantages of the CATER module?**

.....

.....

.....

.....

.....

.....



**12. Do you have any ideas to improve the CATER module? If yes, please specify.**

.....

.....

.....

.....

.....

.....

**13. Are you aware of any competitive systems?**

☐ No                      ☐ Yes, please specify: .....

If yes, which are the main *strengths* and *weaknesses* of the CATER module in comparison with these competitive systems?

.....

.....

.....

.....

## Part 2: System Usability assessment

### 14. System Usability Scale rating

Your first impulse is usually correct; please answer the questions without thinking about them for a long time. All items must be checked. If you feel you cannot respond to a particular item, please mark the centre point of the scale.

SUS questions	-2 Strongly Disagree	-1	0	1	2 Strongly agree
1. I think I would like to use this system frequently.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2. I found the system unnecessarily complex.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. I thought the system was easy to use.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. I think that I would need the support of a technical person to be able to use this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5. I found the various functions in this system were well integrated.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6. I thought there was too much inconsistency in this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. I would imagine that most people would learn to use this system very quickly.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8. I found the system very cumbersome to use.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

SUS questions	-2 Strongly Disagree	-1	0	1	2 Strongly agree
9. I felt very confident using the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10. I need to learn a lot of things before I could get going with this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11. The system would fit well into my daily routines	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12. Using the system, I would lose a lot of time.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13. I am in control of the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14. I can learn to operate the system by trial-and-error.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15. It's easy to remember how to operate the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16. Text and icons are difficult to perceive.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
17. The screen layout is confusing.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18. The system is logically consistent.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19. There are a lot of misunderstandings between me and the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20. I would recommend the system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### **Part 3: General comments**

**15. Please provide any further comments you may have.**

.....

.....

.....

.....

Thank you for your participation!

# Information Society Technologies (IST) Programme



Computerised Automotive Technology Reconfiguration System for Mass Customization

CATER

IST-5-035030-STREP

## **USER ACCEPTANCE & USABILITY ASSESSMENT** **– VR Interface for Reconfiguration**

User ID (Name/Surname if not ID): \_\_\_\_\_

Date: \_\_\_\_\_

Site: \_\_\_\_\_

Evaluator/supervisor name: \_\_\_\_\_

Time and day of test: \_\_\_\_\_

ID's of Real-Life scenarios tried: \_\_\_\_\_

## Part 1: Compare configure vehicles in 2D, 3D and immersive user interfaces

*Please compare vehicle configurators with traditional 2d GUI (for example web configurator), 3d GUI using desktop computer and immersive VR interfaces*

### 1. What is your overall judgment of the immersive VR system you tried, on the following eight items in compare with 2d GUI, for example web interface?

My judgements of the VR system are ... (please tick a box on every line)

- |   |             |         |             |
|---|-------------|---------|-------------|
| 1 | useful      | _ _ _ _ | useless     |
| 2 | pleasant    | _ _ _ _ | unpleasant  |
| 3 | bad         | _ _ _ _ | good        |
| 4 | nice        | _ _ _ _ | annoying    |
| 5 | effective   | _ _ _ _ | superfluous |
| 6 | irritating  | _ _ _ _ | likeable    |
| 7 | assisting   | _ _ _ _ | worthless   |
| 8 | undesirable | _ _ _ _ | desirable   |
| 9 | inefficient | _ _ _ _ | efficient   |

### 2. What is your overall judgment of the immersive VR system you tried, on the following eight items in compare with 3d GUI using desktop computer?

My judgements of the VR system are ... (please tick a box on every line)

- |   |             |         |             |
|---|-------------|---------|-------------|
| 1 | useful      | _ _ _ _ | useless     |
| 2 | pleasant    | _ _ _ _ | unpleasant  |
| 3 | bad         | _ _ _ _ | good        |
| 4 | nice        | _ _ _ _ | annoying    |
| 5 | effective   | _ _ _ _ | superfluous |
| 6 | irritating  | _ _ _ _ | likeable    |
| 7 | assisting   | _ _ _ _ | worthless   |
| 8 | undesirable | _ _ _ _ | desirable   |
| 9 | inefficient | _ _ _ _ | efficient   |

**3. In the VR interface you tried, which factor of following is interesting for you?**

- |  |   |
|--|---|
| <b>1:1 scale:</b>                      | Absolutely not <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Absolutely, yes |
| <b>3D Animation:</b>                   | Absolutely not <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Absolutely, yes |
| <b>Immersived feeling:</b>             | Absolutely not <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Absolutely, yes |
| <b>Statistic data:</b>                 | Absolutely not <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Absolutely, yes |
| <b>Connection with other database:</b> | Absolutely not <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Absolutely, yes |
| <b>Access of remote computer:</b>      | Absolutely not <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Absolutely, yes |
| <b>Other _____:</b>                    | Absolutely not <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Absolutely, yes |

**Part 2: Design of immersive VR user interface**

*Please answer following questions with experience of the immersive applications in VDP product line and the CATER Module.*

**4. When you want to access a functionality, it would be likeable if you can find it:**

- |                                   |   |
|-----------------------------------|---|
| <b>In the sphere menu system:</b> | Absolutely not <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Absolutely, yes |
| <b>On the 2D menu of desktop:</b> | Absolutely not <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Absolutely, yes |
| <b>A symbol in the scene:</b>     | Absolutely not <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Absolutely, yes |

Comment: .....

**5. When you want to see some metadata or facts, so that you could make your decision better, such information should be found:**

- |  |   |
|--|---|
| <b>In a fix location on the scene:</b> | Absolutely not <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Absolutely, yes |
| <b>In the near of objectives:</b>      | Absolutely not <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Absolutely, yes |
| <b>where I'm standing:</b>             | Absolutely not <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Absolutely, yes |

Comment: .....

6. Which following factor should be improved:

<b>Real-like material:</b>	less important <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> more important
<b>Details on geometries:</b>	less important <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> more important
<b>Ease to use:</b>	less important <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> more important
<b>More meta-information:</b>	less important <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> more important
<b>Better real environment:</b>	less important <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> more important

Comment: .....

.....

### Part 3: Advice

.....

.....

.....

.....

.....

.....

.....

Thank you for your participation!



## Anhang B: Wilcoxon-Test über Ergebnisse der Benutzbarkeitsstudie

Die Ergebnisse wurden durch die statistische Software NCSS berechnet.

### Wilcoxon Test für Diss Cheng Nützlichkeit\_Immersive\_vs\_2D

Page/Date/Time 1 09.02.2010 17:27:25  
Database c:\Dokumente und Einstellung ... rsiv gegen 2D und Desktop.S0  
Variable Nützlichkeit\_Immersiv\_gegen\_2D

#### Descriptive Statistics Section

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Nützlichkeit	9	3,666667	1,118034	0,372678	2,80727	4,526063

#### Nonparametric Tests Section

##### Quantile (Sign) Test

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	1	7	0,070313	0,996094	0,035156

### Wilcoxon Test für Diss Cheng Anmutung\_Immersive\_vs\_2D

Page/Date/Time 3 09.02.2010 17:27:25  
Database c:\Dokumente und Einstellung ... rsiv gegen 2D und Desktop.S0  
Variable Anmutung\_Immersiv\_gegen\_2D

#### Descriptive Statistics Section

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Anmutung	9	3,666667	1	0,3333333	2,897999	4,435335

#### Nonparametric Tests Section

##### Quantile (Sign) Test

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	1	5	0,218750	0,984375	0,109375

### Wilcoxon Test für Diss Cheng Effektivität\_Immersive\_vs\_2D

Page/Date/Time 5 09.02.2010 17:27:25  
Database c:\Dokumente und Einstellung ... rsiv gegen 2D und Desktop.S0  
Variable Effektiv\_Immersiv\_gegen\_2D

#### Descriptive Statistics Section

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Effektiv	9	3,444444	0,7264832	0,2421611	2,88602	4,002869

#### Nonparametric Tests Section

##### Quantile (Sign) Test

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	0	3	0,250000	1,000000	0,125000

### Wilcoxon Test für Diss Cheng Effizienz\_Immersive\_vs\_2D

Page/Date/Time 7 09.02.2010 17:27:26  
Database c:\Dokumente und Einstellung ... rsiv gegen 2D und Desktop.S0  
Variable Effizienz\_Immersiv\_gegen\_2D

#### Descriptive Statistics Section

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Effizienz	9	3,777778	0,8333333	0,2777778	3,137221	4,418334

#### Nonparametric Tests Section

##### Quantile (Sign) Test

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	1	7	0,070313	0,996094	0,035156

**Wilcoxon Test für Diss Cheng**  
**Nützlichkeit\_Immersive\_vs\_Desktop**

Page/Date/Time 9 09.02.2010 17:27:26  
 Database c:\Dokumente und Einstellung ... rsiv gegen 2D und Desktop.S0  
 Variable Nützlichkeit\_Immersiv\_gegen\_Desktop

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Nützlichkeit	9	3,555556	1,013794	0,3379312	2,776285	4,334826

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	2	6	0,289063	0,964844	0,144531

**Wilcoxon Test für Diss Cheng**  
**Anmutung\_Immersive\_vs\_Desktop**

Page/Date/Time 11 09.02.2010 17:27:26  
 Database c:\Dokumente und Einstellung ... rsiv gegen 2D und Desktop.S0  
 Variable Schönheit\_Immersiv\_gegen\_Desktop

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Anmutung	9	3,666667	0,5	0,1666667	3,282333	4,051001

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	0	6	0,031250	1,000000	0,015625

**Wilcoxon Test für Diss Cheng  
Effektivität\_Immersive\_vs\_Desktop**

Page/Date/Time 13 09.02.2010 17:27:26  
Database c:\Dokumente und Einstellung ... rsiv gegen 2D und Desktop.S0  
Variable Effektivität\_Immersiv\_gegen\_Desktop

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Effektivität	9	4,111111	1,054093	0,3513642	3,300864	4,921359

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	0	5	0,062500	1,000000	0,031250

**Wilcoxon Test für Diss Cheng  
Effizienz\_Immersive\_vs\_Desktop**

Page/Date/Time 15 09.02.2010 17:27:26  
Database c:\Dokumente und Einstellung ... rsiv gegen 2D und Desktop.S0  
Variable Effizienz\_Immersiv\_gegen\_Desktop

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Effizienz	9	3,333333	0,7071068	0,2357023	2,789803	3,876864

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	0	2	0,500000	1,000000	0,250000

### Wilcoxon Test für Diss Cheng Feature\_Importance\_1\_to\_1\_Scale

Page/Date/Time 1 09.02.2010 17:58:50  
Database c:\Dokumente und Einstellung ... lcoxon feature importance.S0  
Variable \_1\_1\_Scale

#### Descriptive Statistics Section

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
_1_1_Scale	8	4,25	1,035098	0,3659625	3,384636	5,115364

#### Nonparametric Tests Section

##### Quantile (Sign) Test

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	1	7	0,070313	0,996094	0,035156

### Wilcoxon Test für Diss Cheng Feature\_Importance\_3D\_Animation

Page/Date/Time 3 09.02.2010 17:58:51  
Database c:\Dokumente und Einstellung ... lcoxon feature importance.S0  
Variable \_3D\_Animation

#### Descriptive Statistics Section

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
_3D_Animation	8	3,75	0,7071068	0,25	3,158844	4,341156

#### Nonparametric Tests Section

##### Quantile (Sign) Test

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	0	5	0,062500	1,000000	0,031250

**Wilcoxon Test für Diss Cheng**  
**Feature\_Importance\_Immersive\_feeling**

Page/Date/Time 5 09.02.2010 17:58:51  
Database c:\Dokumente und Einstellung ... lcoxon feature importance.S0  
Variable Immersive\_feeling

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Immersive_feeling	8	4,25	1,035098	0,3659625	3,384636	5,115364

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	1	7	0,070313	0,996094	0,035156

**Wilcoxon Test für Diss Cheng**  
**Feature\_Importance\_Statistic\_data**

Page/Date/Time 7 09.02.2010 17:58:51  
Database c:\Dokumente und Einstellung ... lcoxon feature importance.S0  
Variable Statistic\_data

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Statistic_data	8	3,5	1,309307	0,4629101	2,405392	4,594608

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	2	3	1,000000	0,812500	0,500000

**Wilcoxon Test für Diss Cheng**  
**Feature\_Importance\_Database\_Connection**

Page/Date/Time 9 09.02.2010 17:58:51  
 Database c:\Dokumente und Einstellung ... lcoxon feature importance.S0  
 Variable Connection\_with\_database

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
DB_connection	8	3,875	1,125992	0,3980982	2,933647	4,816352

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	1	5	0,218750	0,984375	0,109375

**Wilcoxon Test für Diss Cheng**  
**Feature\_Importance\_Access\_of\_Remote\_Computer**

Page/Date/Time 11 09.02.2010 17:58:51  
 Database c:\Dokumente und Einstellung ... lcoxon feature importance.S0  
 Variable Access\_of\_remote\_computer

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Remote_computer8		3,375	1,187735	0,4199277	2,382029	4,367971

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	1	4	0,375000	0,968750	0,187500

**Wilcoxon Test für Diss Cheng**  
**Feature\_to\_Improve\_Real\_like\_material**

Page/Date/Time 1 11.02.2010 14:13:14  
Database c:\Dokumente und Einstellung ... ant, 5 for more important.S0  
Variable Real\_like\_material

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Real_like_material	9	4,111111	1,054093	0,3513642	3,300864	4,921359

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	1	7	0,070313	0,996094	0,035156

**Wilcoxon Test für Diss Cheng**  
**Feature\_to\_Improve\_Detail\_on\_Geometry**

Page/Date/Time 3 11.02.2010 14:13:14  
Database c:\Dokumente und Einstellung ... ant, 5 for more important.S0  
Variable Details\_on\_geometries

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Detail_geometries	9	3,666667	0,8660254	0,2886751	3,000981	4,332353

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	1	6	0,125000	0,992188	0,062500



**Wilcoxon Test für Diss Cheng  
Feature\_to\_Improve\_Ease\_to\_Use**

Page/Date/Time 5 11.02.2010 14:13:14  
Database c:\Dokumente und Einstellung ... ant, 5 for more important.S0  
Variable Ease\_to\_use

**Descriptive Statistics Section**

		Standard		Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Ease_to_use	9	3,888889	1,166667	0,3888889	2,99211	4,785668

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	1	5	0,218750	0,984375	0,109375

**Wilcoxon Test für Diss Cheng  
Feature\_to\_Improve\_Meta-Information**

Page/Date/Time 7 11.02.2010 14:13:14  
Database c:\Dokumente und Einstellung ... ant, 5 for more important.S0  
Variable More\_meta\_information

**Descriptive Statistics Section**

		Standard		Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Meta_information	9	3,444444	0,8819171	0,2939724	2,766543	4,122346

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	1	4	0,375000	0,968750	0,187500

**Wilcoxon Test für Diss Cheng**  
**Feature\_to\_Improve\_Real\_Environment**

Page/Date/Time 9 11.02.2010 14:13:14  
Database c:\Dokumente und Einstellung ... ant, 5 for more important.S0  
Variable Better\_real\_environment

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Real_environment	9	2,555556	1,333333	0,4444444	1,530665	3,580446

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	4	3	1,000000	0,500000	0,773438

**Wilcoxon Test für Diss Cheng**  
**Access\_of\_Functionality\_Sphere\_Menu**

Page/Date/Time 1 11.02.2010 12:46:20  
Database c:\Dokumente und Einstellung ... not, 5 for absolutely yes.S0  
Variable In\_the\_sphere\_menu

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Sphere_menu	9	3	1,224745	0,4082483	2,058578	3,941422

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	3	3	1,000000	0,656250	0,656250

**Wilcoxon Test für Diss Cheng**  
**Access\_of\_Functionality\_2D\_GUI\_Menu**

Page/Date/Time 3 11.02.2010 12:46:20  
 Database c:\Dokumente und Einstellung ... not, 5 for absolutely yes.S0  
 Variable On\_the\_2D\_Menu\_of\_Desktop

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
2D_GUI_Menu	9	2,888889	1,166667	0,3888889	1,99211	3,785668

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	2	3	1,000000	0,812500	0,500000

**Wilcoxon Test für Diss Cheng**  
**Access\_of\_Functionality\_3D\_Icon**

Page/Date/Time 5 11.02.2010 12:46:20  
 Database c:\Dokumente und Einstellung ... not, 5 for absolutely yes.S0  
 Variable A\_symbol\_in\_the\_scene

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
3D-Icon	9	4,111111	0,781736	0,2605787	3,510216	4,712007

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	0	7	0,015625	1,000000	0,007813

**Wilcoxon Test für Diss Cheng**  
**Access\_of\_Meta-Info\_Fix\_Location**

Page/Date/Time 1 11.02.2010 14:05:31  
Database c:\Dokumente und Einstellung ... formation should be found.S0  
Variable In\_a\_fixed\_location\_in\_the\_scene

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Fixed_location	9	3,111111	1,269296	0,4230985	2,135444	4,086778

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	3	4	1,000000	0,773438	0,500000

**Wilcoxon Test für Diss Cheng**  
**Access\_of\_Meta-Info\_Near\_to\_Object**

Page/Date/Time 3 11.02.2010 14:05:31  
Database c:\Dokumente und Einstellung ... formation should be found.S0  
Variable In\_the\_near\_of\_objects

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
Near_to_objects	9	3,444444	0,7264832	0,2421611	2,88602	4,002869

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	1	5	0,218750	0,984375	0,109375

**Wilcoxon Test für Diss Cheng**  
**Access\_of\_Meta-Info\_User's\_Position**

Page/Date/Time 5 11.02.2010 14:05:31  
 Database c:\Dokumente und Einstellung ... formation should be found.S0  
 Variable Where\_I\_m\_standing

**Descriptive Statistics Section**

			Standard	Standard	95,0% LCL	95,0%
UCL						
Variable	Count	Mean	Deviation	Error	of Mean	of Mean
User's Position	9	3,666667	1	0,3333333	2,897999	4,435335

**Nonparametric Tests Section**

**Quantile (Sign) Test**

Null	Quantile	Number	Number	H1:Q<>Q0	H1:Q<Q0	H1:Q>Q0
Quantile (Q0)	Proportion	Lower	Higher	Prob Level	Prob Level	Prob Level
3	0,5	1	5	0,218750	0,984375	0,109375

