

Virtual Reality - Technologie für eine zentrale Schnittstelle im Produktentwicklungsprozess

Von der Fakultät für Maschinenwesen

der

Technischen Universität Dresden

zur

Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommene und genehmigte Dissertation

Dipl.-Ing. Petermann, Dirk

geb. am 19.01.1980 in Frankfurt (Oder)

Tag der Einreichung: 19.09.2017

Tag der Verteidigung: 20.11.2018

Gutachter: Prof. Dr.-Ing. habil. Ralph Stelzer

Prof. Dr. Dr.-Ing. Dr. h. c. Jivka Ovtcharova

Vorsitzender der Promotionskommission: Prof. Dr.-Ing. Martin Schmauder

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl Konstruktionstechnik/CAD des Instituts für Maschinenelemente und Maschinenkonstruktion der Technischen Universität Dresden sowie während meiner darauffolgenden Tätigkeit bei der XPLM Solution GmbH.

Mein besonderer Dank gilt meinem Doktorvater Herrn Prof. Dr.-Ing. habil. Ralph Stelzer. Ich danke ihm sehr für die stete Förderung, die fachlichen Diskussionen sowie die wertvollen Hinweise und Anregungen. Ich danke ihm außerdem für die geschaffenen Freiräume und insbesondere dafür, dass er bereits während meines Studiums mein Interesse für das Product-Lifecycle-Management und die VR-Technologie geweckt und gefördert hat. Prof. Dr. Dr.-Ing. Dr. h. c. Jivka Ovtcharova danke ich für die Übernahme des Koreferats.

Meinen Kollegen am Lehrstuhl und bei XPLM danke ich für das außerordentliche gute Arbeitsklima und deren Hilfsbereitschaft. Insbesondere Dr.-Ing. W. Steger und Dr.-Ing. B. Saske haben durch vielfältige Anregungen und tatkräftige Unterstützung wesentlich zum Gelingen dieser Arbeit beigetragen. Stellvertretend für alle anderen Kollegen möchte ich mich vor allem bei Annett Kupka und Andreas Heller für die vielen ermunternden Gespräche bedanken.

Zu guter Letzt danke ich allen Freunden und meiner Familie für das entgegengebrachte Verständnis und den Beistand während des Entstehens dieser Arbeit. Besonders herzlich und nachdrücklich danke ich meinen Eltern und meinem Bruder, die mich stets gefördert und unermüdlich unterstützt haben.

Dresden, Januar 2019

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen und Zielstellung	4
2.1	Design Review nach DIN EN 61160.....	4
2.2	VR-Technologie.....	6
2.3	VR-Schnittstellen im Design Review	9
2.3.1	Modellaufbereitung	12
2.3.2	Organisation und Handhabung	24
2.3.3	Rückführung der Ergebnisse	26
2.4	Wertung der bestehenden Situation und Zielstellung	29
3	VR-PLM Integrationskonzept	32
4	Daten- und Prozessmodellierung.....	38
4.1	Konventionen zur Beschreibung der Datenmodelle und des Systemverhaltens.....	38
4.1.1	Klassendiagramm	38
4.1.2	Aktivitätsdiagramm	41
4.2	PES-Produktmodell	45
4.2.1	Abbildung der Produktstruktur	46
4.2.2	Identifizierende Metadaten	49
4.2.3	Transformation.....	50
4.2.4	Erscheinungsbild.....	53
4.2.5	Simulation und Animation.....	64
4.2.6	Präsentationsumgebung	67
4.3	SES-Modell	68
4.4	PLM-Integration des PES-Produktmodells und SES-Modells.....	72
4.4.1	Generierung.....	75
4.4.2	Rekonstruktion	78
4.5	Klassifizierende Metadaten	80
4.6	Bidirektionale Kopplung	84
4.7	Wechsel des Erscheinungsbildes	87

4.8	Lokalisierung von Komponenten.....	88
4.9	Import strukturfremder virtueller Prototypen.....	90
4.10	Handhabung und Organisation	91
4.10.1	Sitzungskonzept.....	91
4.10.2	Autorisierung.....	93
5	Realisierung der Middleware	95
5.1	VR-PLM System.....	95
5.2	Client-Server Modell.....	96
5.3	Session	98
5.4	Adapter	100
5.5	VRPLM-Befehle	103
5.6	Benutzungsschnittstellen	105
6	Anwendungsszenario	110
6.1	Modellaufbereitung.....	110
6.2	Durchführung.....	114
6.3	Rückführung der Ergebnisse	116
6.4	Umsetzung von Maßnahmen	118
7	Zusammenfassung	120
8	Literaturverzeichnis	123

1 Einleitung

Das systematische Vorgehen zur Erzeugung technischer Produkte ist allgemein als Produktentstehungsprozess bekannt. Die Summe der hierin enthaltenen Arbeitsschritte zwischen der am Anfang stehenden Produktidee bis hin zur Erzeugung der Fertigungsunterlagen wird als Entwicklungs- und Konstruktionsprozess (EKP) bezeichnet [PBF13]. Die VDI-Richtlinie 2221 versteht den EKP als Bestandteil der Produktentwicklung [VDI2221].

Zentrale Aufgabe der Produktentwicklung ist die Umsetzung von technischen Produktanforderungen in ein reales technisches System. Dabei werden bis zur abschließenden Lösung Modelle iterativ generiert, bewertet und überarbeitet. Dieser Prozess ist in der DIN EN 61160 beschrieben und wird als Entwicklungsbewertung bzw. Design Review bezeichnet. Ferner folgt die moderne Produktentwicklung spezifischen Konzepten, wie dem Systems Engineering, der Integrierten Produktentwicklung sowie der Kollaboration (Bild 1-1). Ziel der Konzepte ist es, Wechselwirkungen zwischen Entwicklungsdomänen, Phasen im Produktlebenszyklus sowie Entwicklungspartnern zu beschreiben und zu steuern.

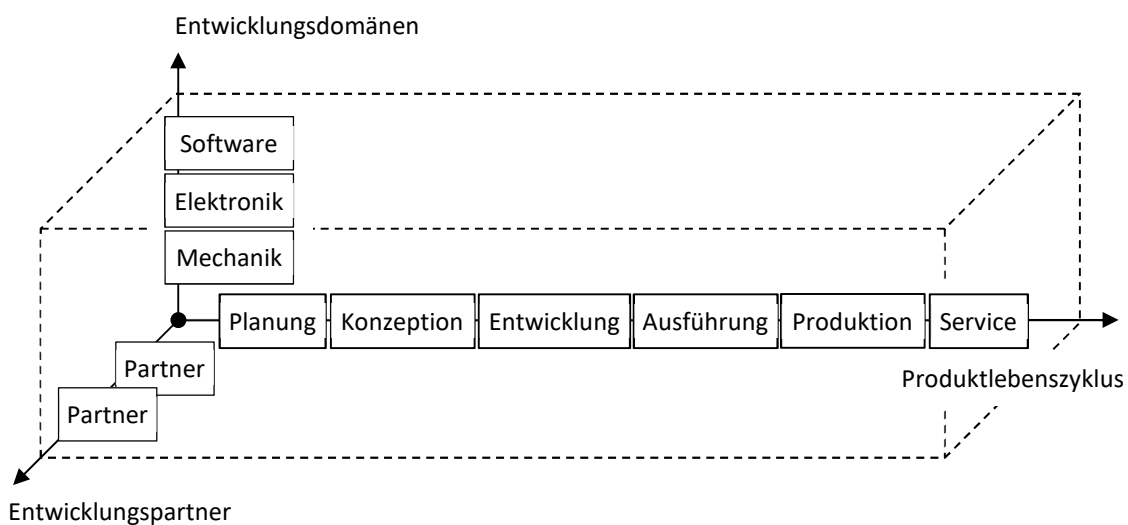


Bild 1-1: Mehrdimensionale Wechselwirkungen im Produktentwicklungsprozess nach [EiS09]

Mit der Anwendung der genannten Konzepte entsteht ein erhöhter Koordinierungsaufwand, um Informationen zur richtigen Zeit, in der richtigen Reihenfolge und dem richtigen Ort zur Verfügung zu stellen. Hinsichtlich dieser Herausforderung haben sich Lösungen des Product-Lifecycle-Managements (PLM) etabliert. PLM ist ein Konzept, das sämtliche produktrelevanten Informationen, Prozesse und Systeme sowohl innerhalb über verschiedene

Standorte als auch außerhalb (zu Zulieferern, Partnern und Kunden) eines Unternehmens integriert [EiS09].

War in der nahen Vergangenheit die Erprobung von Produktentwürfen maßgeblich durch physische Prototypen geprägt, werden heutzutage Produktmerkmale wie zum Beispiel Gestalt, Baustruktur und Verhalten informationstechnisch abgebildet. Man spricht vom virtuellen Prototyp. Virtuelle Prototypen entstehen durch die Anwendung rechnergestützter Methoden und Werkzeuge und werden analog den physischen Prototypen für die Absicherung von Produkteigenschaften im Design Review verwendet. Bezüglich der Übertragung von Bewertungsprozessen auf den virtuellen Prototyp besteht der Anspruch, Produktmerkmale möglichst real wahrzunehmen.

Für die realistische Präsentation von virtuellen Prototypen bietet sich die Virtual Reality (VR) Technologie an. Heutzutage sind stereoskopische Präsentationen, die entweder einzeln (Powerwall) bzw. in Kombination zu einem geschlossenen Raum (Cave) verwendet werden, fester Bestandteil moderner Entwicklungsabteilungen von Maschinenbauunternehmen. Aktueller Trend sind stereoskopische Brillen, die sich durch einen hohen Mobilitätsfaktor sowie geringe Anschaffungskosten auszeichnen.

VR unterstützt die Entwicklung von Produkten im Maschinenbau in hohem Maße. Maßstabsgerechte, fotorealistische und funktionale virtuelle Prototypen sowie Interaktionsmöglichkeiten in Echtzeit ermöglichen es, Produkte fernab von physischen Prototypen rein virtuell zu erleben bzw. zu erproben. Dies macht VR zu einem idealen Werkzeug für Bewertungsaufgaben. Vor allem innerhalb der Domäne Mechanik wird es derzeit intensiv als Visualisierungsplattform für Ergonomie-, Einbau-, Funktions- und Designuntersuchungen verwendet.

Mit der Nutzung von VR für Bewertungsaufgaben am virtuellen Prototyp ergibt sich nach [Kun12] eine erhöhte Transparenz komplexer Produkte. In der Folge werden etwaige Mängel früher erkannt, Iterationsschleifen vermindert und die Anzahl erforderlicher physischer Prototypen reduziert. Ergebnis sind eine gesteigerte Produktqualität, eine verringerte Zeit, die das Produkt bis zur Platzierung am Markt benötigt sowie gesenkte Entwicklungskosten. Dies verschafft Unternehmen einen Wettbewerbsvorteil.

Im Sinne des Design Reviews nach DIN EN 61160 ist die Nutzung von VR in einem durch PLM geprägten Produktentwicklungsprozess nur mangelhaft unterstützt. Die Abbildung im PLM verwalteter Information sowie die Rückführung von im Design Review mit VR gewonnenen Ergebnissen ins PLM sind derzeit durch aufwendige manuelle Arbeiten und Medienbrüche geprägt.

In der DIN EN 61160 existieren keine Empfehlungen zur Nutzung moderner Visualisierungstechnologien wie VR, oder zu deren Integration in ein PLM-Konzept. Es sind ausschließlich der administrative Prozess des Design Reviews und Empfehlungen bezüglich der durchzuführenden Untersuchungen beschrieben. DIN EN 61160 weist nur darauf hin, dass notwendige Ressourcen und Informationen bereitzustellen, Tätigkeiten zu überwachen sowie Ergebnisse zu messen und zu analysieren sind.

Im Sinne der modernen Produktentwicklung ist das Design Review ein andauernder Prozess, der sich über den gesamten Produktlebenszyklus und alle Entwicklungsdomänen erstreckt. Das bedeutet, Bewertungsgegenstand sind alle im PLM verwalteten Informationen.

Ziel dieser Arbeit ist die Entwicklung eines Konzeptes, das auf Basis von VR-Technologie den Zugang zu Daten und Prozessen des PLM ohne manuelle Vorarbeiten und Medienbrüche im Design Review beschreibt (Bild 1-2). Die mit VR dargestellten Informationen sind im Kontext des PLM zu belassen, damit die im Design Review gewonnenen Ergebnisse konsistent in den Produktentwicklungsprozess zurückgeführt werden können. Aufgrund der derzeitigen Nutzung von VR speziell für Ergonomie-, Einbau-, Funktions- und Designuntersuchungen steht die Domäne Mechanik hierbei im Vordergrund. Gleichzeitig wird bei der Konzeptentwicklung eine mögliche Ausweitung auf weiterführende Domänen berücksichtigt.

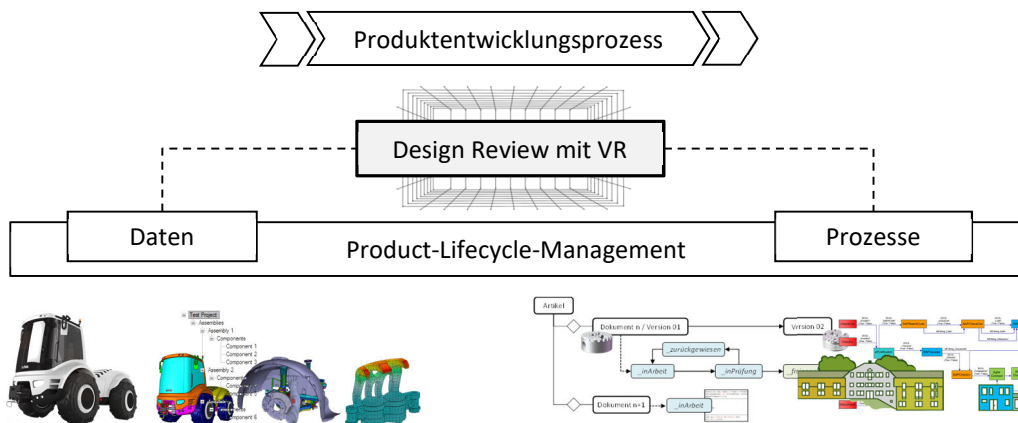


Bild 1-2: VR-basiertes Design Review im Produktentwicklungsprozess

2 Grundlagen und Zielstellung

2.1 Design Review nach DIN EN 61160

In der von [Ehr09] beschriebenen „Definitionsphase“ bestehend aus Planung und Entwicklung, wird infolge der Festlegung von Wirkprinzipien die Gestalt und das Verhalten des Produktes bestimmt. Nach [ScP15] sind ein Großteil aller Fehler am Produkt auf Unzulänglichkeiten in dieser Phase zurückzuführen.

Die Mängelerkennung geht mit der Bewertung des aktuellen Entwicklungsstandes einher. Anerkannte Methodik zur Bewertung von Produktmerkmalen bzw. -anforderungen ist die in den Produktentwicklungsprozess eingebundene Entwicklungsbewertung (Design Review). DIN EN 61160 beschreibt diese als geplante, dokumentierte und unabhängige Bewertung einer existierenden Konstruktion oder eines vorgeschlagenen Konstruktionsentwurfs. Neben dem geplanten Design Review finden in den Entwicklungsabteilungen auch spontane Beurteilungen, so genannte Walkthroughs statt. Sie sind weniger formal und dienen vorrangig der Diskussion während der Entwicklungsphase [CuC08].

Nach DIN EN 61160 leiten sich die zu besprechenden Themen aus der Art der Entwicklungsbewertung, deren Zielen und Zwecken sowie aus der Art des Produktes ab. Eine allgemeingültige Tagesordnung ist nicht gegeben. Übliche Themen sind beispielsweise die Verifizierung der:

- Produktmerkmale,
- Funktionsfähigkeit,
- Instandhaltung,
- menschlichen Einflüsse,
- Sicherheit sowie
- Kosten.

In Abhängigkeit des Themas werden Prozesse bzw. Merkmale des aktuellen Entwicklungsstandes betrachtet und mit Sollzuständen der Anforderungsliste verglichen. Prozess- und merkmalsbezogene Sollzustände sind nach DIN EN ISO 9000 entweder quantitativ in Form alphanumerischer Werte bzw. Verhältnisse, oder qualitativ durch eine Tendenz gegeben. Während quantitative Produkthanforderungen mittels mathematischer Methoden bewertet werden, basiert die Validierung qualitativer Eigenschaften, wie zum Beispiel die Erreichbarkeit eines Bauraums, häufig auf der Erprobung durch den Menschen. Dieser Prozess ist von subjektiver Wahrnehmung geprägt.

Im Ergebnis der Entwicklungsbewertung werden Empfehlungen ausgesprochen und Maßnahmen festgelegt. Anschließend werden Änderungen an der Entwurfskonzeption und den Spezifikationen vorgenommen. Ein Design Review dauert an, bis die angeordneten Maßnahmen wie vorgesehen behandelt und abgeschlossen sind. Die Ergebnisse des Design Reviews sind revisionssicher zu dokumentieren, um das erworbene Wissen zu sichern und den Entwicklungsablauf transparent zu gestalten.

Während des Design Reviews werden Bewertungsprozesse zunehmend vom physischen auf den virtuellen Prototyp übertragen. Die Nutzung virtueller Prototypen zeichnet sich durch einen schnellen Zugriff, eine vereinfachte Verteilung, Bearbeitung und Reproduktion aus. Ansätze eines kontextabhängigen „intelligenten“ Verhaltens ermöglichen temporäre auf das Design Review beschränkte Änderungen am Produkt, die die Diskussion von Empfehlungen und Maßnahmen unterstützen. Die frühe und regelmäßige Bewertung des aktuellen Entwicklungsstandes am virtuellen Prototyp führt nach [WaB13] zu einer Verkürzung des Produktentstehungsprozesses (Bild 2-1) sowie zu einer Steigerung der Produktqualität.

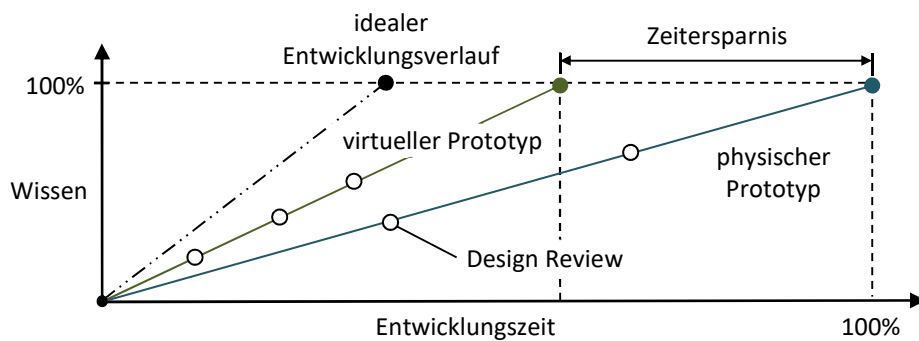


Bild 2-1: Reduzierung der Entwicklungszeit durch den Einsatz virtueller Prototypen im Design Review nach [Ovt05]

2.2 VR-Technologie

Die VDI-Richtlinie 4499 beschreibt die VR-Technologie als Anwendung einer hochmodernen Mensch-Maschine-Schnittstelle, die den Benutzer durch den Einsatz innovativer Endgeräte in eine dreidimensionale rechnerinterne Welt einbezieht [VDI4499]. Dies umfasst Technologien zur Definition eines echtzeitfähigen, rechnergestützten, dreidimensionalen Modells, mit dem der Benutzer durch Einbeziehung seiner Person und infolge multimodaler Rückkopplungen interagieren kann. Ähnliche Ansätze sind in [Rhe91] zu finden. Hier wird VR als das Erleben einer dreidimensionalen computergenerierten Umgebung, in der Navigation und Interaktion möglich sind, beschrieben.

Nach [Bur06] ist das Empfinden einer virtuellen Realität aufgrund der subjektiven Wahrnehmung nutzerspezifisch. Die Wahrnehmung von virtueller Realität ist maßgeblich von den sich gegenseitig beeinflussenden Faktoren Immersion, Interaktion und Imagination abhängig.

Immersion beschreibt inwieweit der dargestellte virtuelle Inhalt als real wahrgenommen wird. Eine in [Lin11] durchgeführte Studie zeigt, dass sich der Mensch seiner visuellen Wahrnehmung am stärksten bewusst ist. Auditive und haptische Reize folgen mit einigem Abstand (Bild 2-2). Unter Berücksichtigung von [Lin11] ist davon auszugehen, dass die visuelle Stimulierung für das Empfinden von Immersion maßgebend ist.

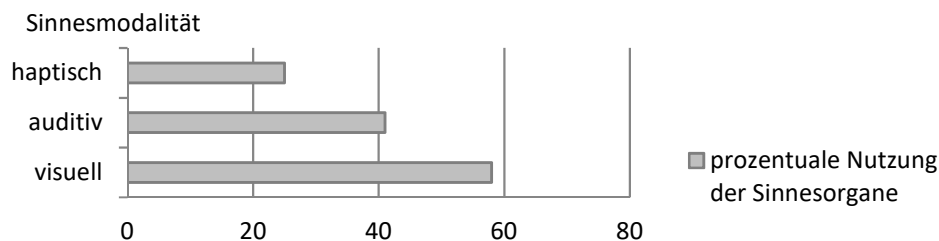


Bild 2-2: Nutzung visueller, auditiver und haptischer Sinnesorgane

Unter Interaktion wird verstanden, dass der Nutzer den virtuellen Prototyp verändern bzw. beeinflussen kann. Grundlegende Modellinteraktionen in virtuellen Umgebungen werden nach [LKM17] in:

- Manipulation (Kontrolle von Objekteigenschaften wie Position, Orientierung, Farbe, Transparenz und Textur; Selektion),
- Navigation (Ansichtseinstellung) und
- Systemkontrolle (Steuerung der Umgebungsparameter wie Kollisionserkennung, Beleuchtung und Hintergrundeigenschaften).

untergliedert. Bei der Modellinteraktion wird zwischen direkter und indirekter Interaktion unterschieden [SSP11]. Indirekte Interaktion bedeutet, dass der Nutzer mittels technischer Eingabegeräte wie beispielsweise Tastatur und Maus das Modell steuert. Bearbeiter und Modell befinden sich hierbei in einer beliebigen relativen Position zueinander. Von direkter Interaktion ist die Rede, wenn das Nachführen des Modellobjekts anhand einer Änderung von Position und/oder Orientierung des Nutzers erfolgt [MZF10].

Als Kennziffern der Imagination leitet [She00] unter anderem die Quantität der wiedergegebenen Informationen hinsichtlich Pixeldichte, Farbraum und Bilder pro Sekunde ab, sowie die Möglichkeit der relativen Positionierung zwischen Objekt und Beobachter.

In der modernen Produktentwicklung wird VR-Technologie heutzutage meist für Beurteilungsaufgaben innerhalb der Domäne Mechanik eingesetzt. Nach [EiS09] werden hierbei Erprobungen durchgeführt, die einen statischen bzw. dynamischen Charakter besitzen.

Zu den statischen Erprobungen mit VR gehören beispielsweise Beurteilungen von Designkonzepten, die sich aus dem Wechselspiel von Umgebung, Oberflächen und Materialien ergeben. Dies ist möglich, da die Renderer moderner VR-Systeme das Verhalten von Licht [ReP06] bei der Interaktion mit der virtuellen Umgebung annähernd physikalisch korrekt approximieren. Je nach zugewiesenem Material und Oberflächeneigenschaften werden so bei Lichteinfall physikalische Effekte wie Absorption, Reflexion und Streuung am virtuellen Prototyp wiedergegeben. Es entsteht eine fotorealistische Darstellung (Bild 2-3), die entsprechend einer Studie aus [RDK13] nahezu identisch zum Abbild eines physischen Prototyps verwendet werden kann.

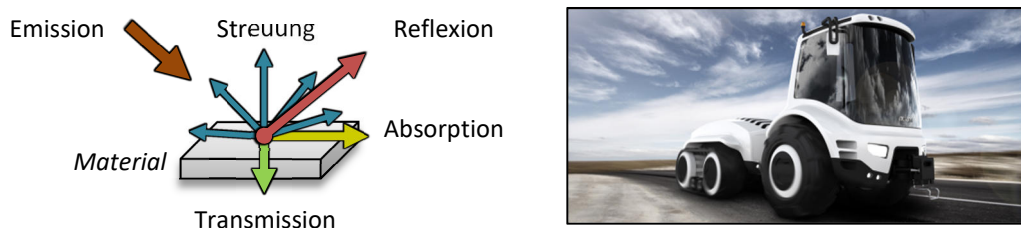


Bild 2-3: Simulation des physikalischen Verhaltens von Licht an Oberflächen

Neben der Beurteilung von Designkonzepten, zählen beispielsweise auch Untersuchungen hinsichtlich der Ergonomie zu den statischen Erprobungen. So werden in der Montageplanung Montagestraßen mit VR maßstabsgerecht dargestellt und ergonomische Aussagen bezüglich der Bewegungs- und Sichträume von Mitarbeitern generiert. Bestehende Produktionslinien werden im Rahmen kontinuierlicher Verbesserungsprozesse mit VR weiter optimiert, da

vorgeschlagene Änderungen ausgiebig virtuell getestet werden können, ohne dass die laufende Produktion beeinträchtigt wird [VDI5600], [AHO06].

Bei dynamischen Untersuchungen werden Komponenten unter Beachtung ihres Freiheitsgrades entlang eines zu prüfenden Pfades betrachtet. Hieraus ergeben sich die zweckgerecht bedingten Bewegungen von Komponenten und etwaige Kollisionen mit der Umgebung. Man spricht von so genannten Ein- und Ausbauuntersuchungen.

Dynamische Untersuchungen werden für die frühzeitige Einbindung des Service herangezogen, indem manuelle Tätigkeiten der Wartung bereits in der Gestaltfindungsphase simuliert werden. Finden hierzu beispielsweise physische Prototypen als tangible Interface in virtuellen Umgebungen Anwendung, können aus ergonomischer Sicht verhältnismäßig realistische Ergebnisse erzielt werden [SSP11]. Dieser Ansatz schlägt die Brücke zwischen der vollständig virtuellen Erprobung und der traditionellen Erprobung mit physischen Prototypen (Bild 2-4).

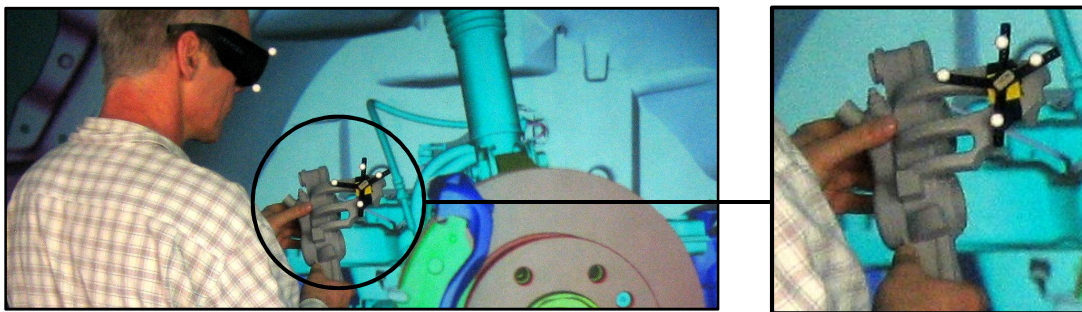


Bild 2-4: Bewertung von Servicearbeiten mittels physischer Prototypen in virtuellen Umgebungen

Nach [JeS10] eignet sich VR auch für die Erprobung von Instandhaltungsarbeiten, die fundierte Kenntnisse hinsichtlich der Funktionen und Wirkungsweisen der Betriebsmittel verlangen. So werden zum Beispiel Prüfungen an strom- und spannungsführenden Bauelementen der Energietechnik risikofrei durchgeführt, indem relevante technische Abläufe und physikalische Größen mittels VR visualisiert werden.

2.3 VR-Schnittstellen im Design Review

Bild 2-5 zeigt die neun Teilprozesse einer Entwicklungsbewertung nach DIN EN 61160. Hinsichtlich der Nutzung von VR als Präsentationsschicht für Beurteilungsaufgaben des technischen Systems und der damit verbundenen Generierung von Empfehlungen ist der Teilprozess 6.6 maßgebend.

- **Definition Präsentationsschicht:** Verschiedene Ausprägungen an Hard- und Software, die im Design Review zum Erleben des Produktmodells dienen.

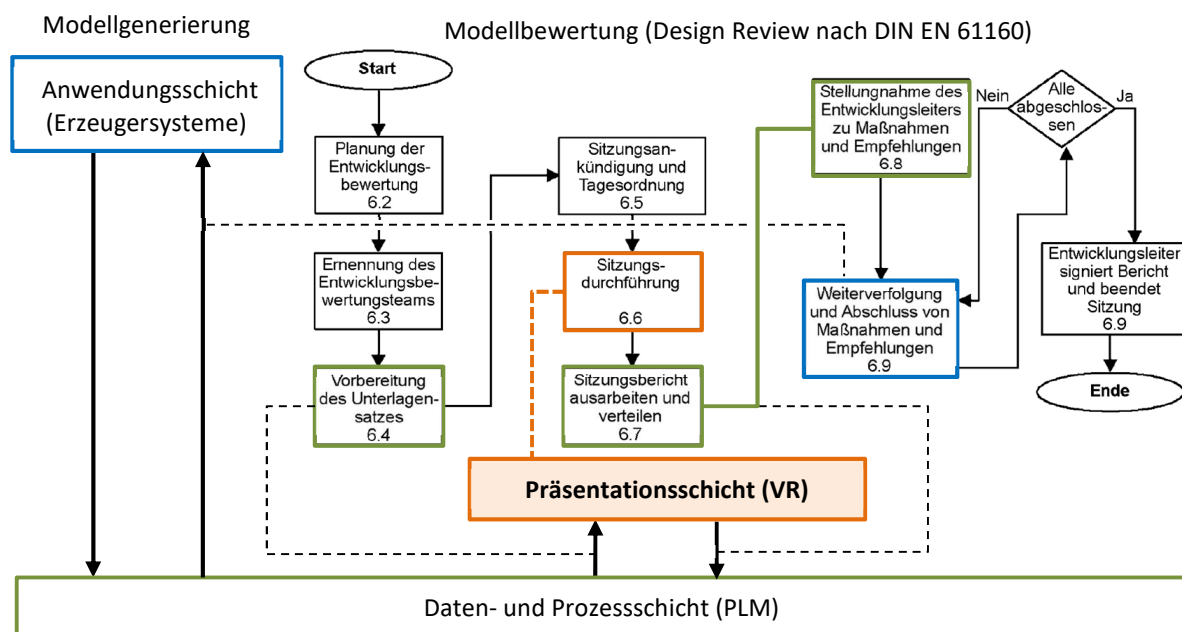


Bild 2-5: Integration der VR-Technologie im Design Review nach DIN EN 61160 - Bild 2

Die Beurteilung des Produktmodells während des Teilprozesses 6.6 erfolgt durch einen erweiterten Personenkreis. Hieraus resultiert, dass die Nutzung von VR als Präsentationsschicht einer Mehrbenutzerstrategie folgen muss. Schwerpunkte sind die Unterstützung eines örtlich verteilt agierenden Personenkreises im Sinne der Kollaboration, die Unterstützung eines Rollenmodells entsprechend den Anforderungen nach DIN EN 61160 sowie Methoden, die eine eindeutige Kommunikation im Kontext des Produktmodells gewährleisten. Dem angepasst müssen administrative Kontrollfunktionen zur Bestimmung des Dargestellten und zur Rückführung von Informationen in den Produktentwicklungsprozess existieren.

Die während der Beurteilungstätigkeit verwendeten Informationen werden durch den vorgelagerten Teilprozess 6.4 bereitgestellt. Zur Anwendung kommen informationstechnische

Modelle aus den verschiedenen Entwicklungsdomänen, die unter anderem Wechselwirkungen bezüglich Produktion, Einsatz und Recycling des späteren Produktes beschreiben. Rechnergestützte Anwendungen, die Modelle dieser Art generieren, werden als Erzeugersysteme bezeichnet und im Kontext dieser Arbeit unter dem Begriff Anwendungsschicht zusammengefasst.

- **Definition Anwendungsschicht:** Verschiedene Ausprägungen an Hard- und Software, die im Produktentwicklungsprozess der Generierung und Bearbeitung des Produktmodells dienen.

Im Sinne des PLM-Konzeptes sind die Daten der Erzeugersysteme konsistent in einer Datenschicht, die den Unternehmensprozessen entspricht, abgebildet. Zusammenfassend wird im weiteren Verlauf von einer Daten- und Prozessschicht gesprochen.

- **Definition Daten- und Prozessschicht:** Verschiedene Ausprägungen an Hard- und Software, die zur Verwaltung des Produktmodells dienen.

Die aus der Daten- und Prozessschicht hervorgehenden prozessualen und strukturellen Beschreibungen des Produktes bilden die Grundlage einer konsistenten Beurteilungstätigkeit. Zusammenfassend muss Teilprozess 6.4 gewährleisten, dass alle erforderlichen Informationen für die Präsentation mit VR geeignet sind und hinsichtlich der Rückführung von Informationen in den Produktentwicklungsprozess im Kontext der Daten- und Prozessschicht stehen. Darüber hinaus sind der personenspezifische Zugriff auf die Informationen der Daten- und Prozessschicht sowie die Organisation der Teilnehmer im Design Review zu steuern.

Im Ergebnis der Beurteilungstätigkeit erfolgt die Generierung von Empfehlungen hinsichtlich der Optimierung einzelner Konstruktionsentwürfe (Teilprozess 6.7). Die Empfehlungen sind konsistent zur Daten- und Prozessschicht und in Form eines Sitzungsberichts zu dokumentieren und innerhalb der Daten- und Prozessschicht zu verwalten. Entsprechend der Empfehlungen wird im nachfolgenden Teilprozess 6.8 der Sitzungsbericht um verbindliche Maßnahmen zur Optimierung erweitert. Die Teilprozesse 6.7 und 6.8 werden aus Sicht der Optimierung des technischen Systems im Rahmen dieser Arbeit als Einheit betrachtet.

Bezüglich der einleitend erwähnten Walkthroughs besteht der Anspruch, Optimierungen im Vorfeld des Design Reviews unverbindlich zu erproben. Dieser Prozess zeichnet sich für den Nutzer durch das direkte Wechselspiel zwischen der Anwendungs- und der Präsentationsschicht aus. Man spricht von einer bidirektionalen Kopplung der Schichten. Modelle werden hierbei mit VR beurteilt, anschließend in das Erzeugersystem zurückgeführt und überarbeitet sowie abschließend mit VR aktualisiert zur Anzeige gebracht. In diesem Fall

entfallen die Teilprozesse 6.7 und 6.8. Der Teilprozess 6.9 folgt direkt nach der Beurteilungstätigkeit.

Im Rahmen dieser Arbeit wird die Qualität der Integration von VR im Design Review an der Unterstützung der Teilprozesse 6.4, 6.6 und 6.7/6.8 gemessen und nachfolgend näher untersucht. Sie werden synonym als:

- Modellaufbereitung (6.4),
- Organisation und Handhabung (6.6) sowie
- Rückführung der Ergebnisse (6.7 und 6.8)

bezeichnet. Die verbleibenden vor- und nachgelagerten Teilprozesse des Design Reviews finden ihre Anwendung abseits der Bewertung des technischen Systems.

Zusammenfassend wird festgestellt, dass der Prozess des Design Reviews verschiedene Ausprägungen an Hard- und Software bezüglich Präsentations-, Anwendungs- sowie Daten- und Prozessschicht integriert. Die Integration von VR in den Design Review Prozess muss gewährleisten, dass sie unabhängig von einer spezifischen Ausprägung ist.

2.3.1 Modellaufbereitung

Der virtuelle Prototyp

Während der Produktentwicklung wird das Produkt durch eine Vielzahl partieller Modelle abgebildet. Informationen, die ein Modell spezifizieren, werden allgemein als Produktdaten bezeichnet. Die Zuordnung der Produktdaten im Kontext des Produktes erfolgt mittels der Produktstruktur. Diese entsteht mit dem Erkennen der prinzipiellen Lösung [VDI2221] und ist von der Sicht auf das Produkt abhängig. So liegt der Fokus in der Produktentwicklung auf der physischen Baustruktur in Form von Einzelteilen und Baugruppen sowie funktionalen Zusammenhängen. Bei der Montage beispielsweise ist neben der physischen Baustruktur die Reihenfolge des Zusammenbaus ein Gliederungskriterium. Weitere Kriterien ergeben sich aus der Forderung einer einfachen Fertigung und einem hohen Grad der Wiederverwendung von Produkten. In der Praxis werden mehrere Produktstrukturen sowohl disziplinär als auch interdisziplinär gedacht und gelebt. Die Summe aller Produktstrukturen und -daten resultiert nach [SpK97] in einem ganzheitlichen übergeordneten Produktmodell (Bild 2-6).

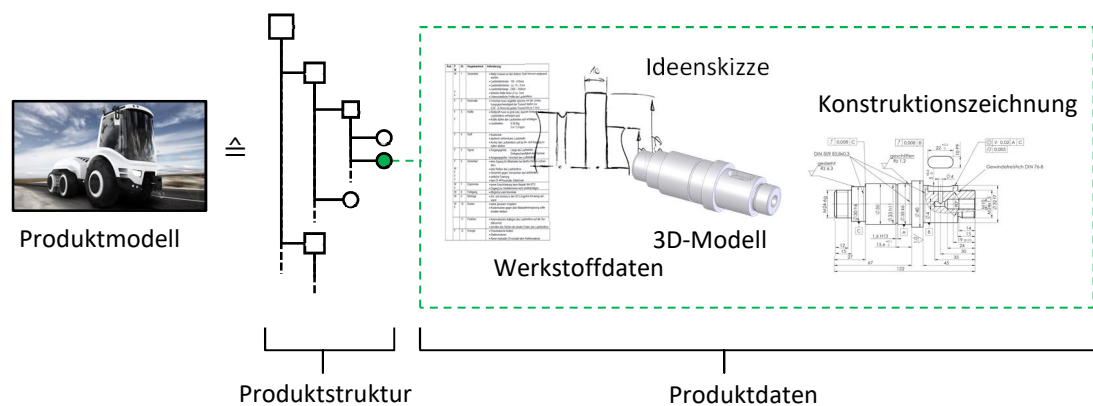


Bild 2-6: Sicht auf das Produktmodell in der Entwicklungsphase

Heutzutage sind Produktstrukturen und Produktdaten informationstechnisch abgebildet. Man spricht vom virtuellen Prototyp (VP). Dieser ergibt sich aus der Nutzung rechnergestützter Methoden mittels Anwendungssoftware. Virtuelle Prototypen stehen entsprechend des Frontloading-Konzeptes frühzeitig für Beurteilungsaufgaben zur Verfügung. Hierdurch ist es den Unternehmen möglich, ökonomisch auf konzeptionelle Mängel zu reagieren.

Nach [Mey07] ist die Produktentwicklung maßgeblich durch die rechnergestützten Methoden der Gestaltfindung (CAD), Berechnung (CAE) sowie Fertigung (CAM) beeinflusst (Bild 2-7). Zusammenfassend werden die rechnerunterstützten Methoden im Produktentwicklungsprozess allgemein als CAX-Methoden bezeichnet. Virtuelle Prototypen, die speziell für Funktions- und Einbauuntersuchungen genutzt werden, werden auch als Digital Mock-Up (DMU) bezeichnet [DIN199].

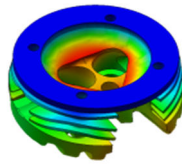
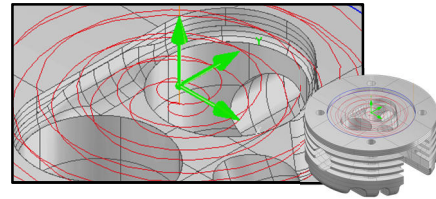
virtueller Prototyp CAD
(Gestaltbeschreibung)virtueller Prototyp CAE
(Verformungsuntersuchung)virtueller Prototyp CAM
(CNC-Programm Ebenenschruppen)

Bild 2-7: interdisziplinäre virtuelle Prototypen am Beispiel eines Zylinderkopfs

Innerhalb jeder Methode existiert eine Vielzahl kommerzieller Erzeugersysteme, die anwendungsspezifische Spezialisierungen aufweisen. Aus dieser differierenden Systemlandschaft resultieren virtuelle Prototypen, die unterschiedliche Datenformate aufweisen. Diese so genannten nativen Datenformate sind derart gespeichert, dass sie nicht über standardisierte, herstellerunabhängige Prozeduren ausgewertet werden können.

Die aus der Anwendung der verschiedenen CAx-Methoden abgeleiteten virtuellen Prototypen stehen in Wechselwirkung untereinander. Eine Änderung im Gestalt-, Verhaltens- oder Fertigungsmodell hat stets Rückwirkungen auf die jeweils anderen Modelle. Keine der Methoden darf daher im Design Review für sich allein gesehen werden. Diesem Umstand steht entgegen, dass native virtuelle Prototypen in den Erzeugersystemen nur nebeneinander existieren. Wird das Produkt durch unterschiedliche native virtuelle Prototypen beschrieben, ist im Rahmen dieser Arbeit von einem heterogenen virtuellen Prototyp die Rede.

- **Definition heterogener virtueller Prototyp:** Summe verschiedener nativer virtueller Prototypen, die im Kontext des Produktes wechselwirken.

Hinsichtlich des Erkennens interdisziplinärer Wechselwirkungen im Design Review erschweren heterogene virtuelle Prototypen die Anwendung der rechnergestützten Methoden. In [KFG07] wird festgestellt, dass die mangelhafte Daten- und Systemintegration im Produktentwicklungsprozess als größtes Hindernis bei der durchgängigen Anwendung der CAx-Methoden zu sehen ist (Bild 2-8).

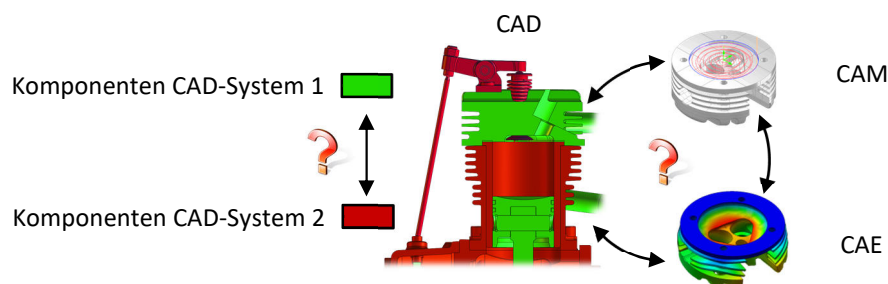


Bild 2-8: Wechselwirkung der CAx-Methoden im Design Review

Probates Mittel der zusammenhängenden Visualisierung heterogener virtueller Prototypen im Design Review ist die Konvertierung nativer Daten in ein neutrales Datenformat. Anerkannte neutrale Datenformate, die von den meisten VR-Systemen unterstützt werden, sind beispielsweise STEP, JT und VRML. Im Vergleich zu nativen Daten ist der Informationsgehalt bzw. Abbildungsraum neutraler Datenformate als gering anzusehen [VSG11]. Parametrische anwendungsspezifische Beschreibungen, wie zum Beispiel Features und geometrische Zwangsbedingungen, die ein „intelligentes“ Verhalten realisieren, sind nicht vorhanden.

Während der Erprobung bzw. Bewertung mit VR ist die mit der Modellgenerierung einhergehende Parametrik nativer virtueller Prototypen unbedeutend. Datenkonsumierende Prozesse wie das Design Review beschränken sich nach [FLL11] auf die Verwendung geometrischer bzw. geometriebezogener Modell- sowie Metadaten. Dem entspricht die Nutzung neutraler Datenformate hervorragend.

Neutrale Datenformate differieren in ihren Abbildungsräumen [BGW11], [BFS11]. Formate, die beispielsweise die Geometrie analytisch exakt beschreiben sind hervorragend für unternehmensinterne Untersuchungen geeignet. Ist hinsichtlich verteilter Arbeit der Schutz des geistigen Eigentums von Interesse, sind diskrete, abstrahierte neutrale Datenformate einzusetzen. Steht die Bewertung ästhetischer Aspekte im Vordergrund des Design Reviews, müssen neutrale Datenformate, die Farben und Texturen abbilden, zum Einsatz kommen. Neutrale Datenformate, die Metadaten wie Produktfertigungsinformationen (PMI) unterstützen, können darüber hinaus auch im Bereich der Fertigung genutzt werden. Tabelle 1 zeigt ausgewählte Eigenschaften der weit verbreiteten neutralen Datenformate STEP, JT und VRML.

Tabelle 1: spezifische Eigenschaften neutraler Datenformate nach [FLL11]

	STEP 214	VRML 97	JT 9.5c
Geometrie			
diskrete Repräsentation	✗	✓	✓
analytische Repräsentation	✓	✗	✓
Farbinformationen	✓	✓	✓
Texturen	✗	✓	✓
Transparenz	✗	✓	✓
Metadaten			
Volumen, Dichte, Masse	✓	✗	✓
Produktfertigungsinformationen	✓	✗	✓
Baugruppenstruktur	✓	✓	✓

Die Praxis zeigt, dass in Abhängigkeit der verwendeten Konvertierungssoftware und deren Konfiguration die Abbildungsräume der neutralen Datenformate unterschiedlich genutzt werden. Dies führt dazu, dass Eigenschaften wie beispielsweise Produktstruktur, Farbe, Materialtextur oder Produktfertigungsinformationen unberücksichtigt bleiben oder differenziert beschrieben sind. So können zum Beispiel Angaben zur Position und Orientierung in VRML-Daten zum einen in der Produktstruktur, zum anderen in der diskreten Geometrierepräsentation berücksichtigt werden. Auch die Art der mathematischen Beschreibung differenziert. Während JT beispielsweise für Position und Orientierung eine Transformationsmatrix implementiert, nutzt VRML 4-dimensionale Vektoren (Bild 2-9). Gehen Informationen während der Konvertierung verloren bzw. können aufgrund unterschiedlicher Beschreibungsarten nicht in einem gemeinsamen Kontext verwendet werden, sind manuelle Anpassungen durchzuführen.

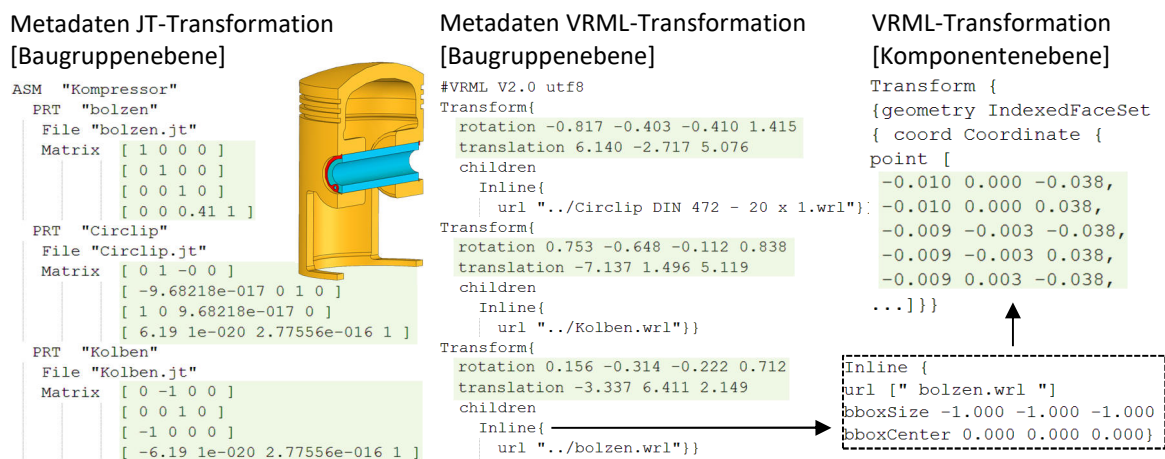


Bild 2-9: Transformationsangaben in JT und VRML

In Anbetracht der Tatsache, dass neutrale Datenformate abgeleitete Momentaufnahmen des Entwicklungsprozesses darstellen, ist nach Änderungen am nativen Original stets eine erneute Konvertierung und Aufbereitung erforderlich. Die Modellvorbereitung stellt somit einen zeitlichen, sich wiederholenden Aufwand dar, der die Nutzung virtueller Prototypen in Echtzeit im Design Review einschränkt.

Weitere Einschränkungen ergeben sich, wenn Produkte im Design Review aus verschiedenen neutralen Datenformaten bestehen. Die in den Spezifikationen der neutralen Datenformate beschriebene Produktstruktur unterstützt ausschließlich Referenzen auf neutrale Datenformate desselben Typs. Heterogene Produktstrukturen sind nicht unterstützt, weshalb neutrale Datenformate in einer heterogenen System- bzw. Formatlandschaft nur eingeschränkt nutzbar sind.

Zusammenfassend wird festgestellt, dass die Verwendung von VR im Design Review einer einheitlichen, systemunabhängigen Beschreibung von Gestalt, Erscheinungsbild und Funktion

heterogener virtueller Prototypen bedarf. Der hiermit einhergehende allgemeine Konvertierungsaufwand ist derart zu minimieren, dass die Nutzung nativer virtueller Prototypen im Design Review mit VR möglichst ad hoc erfolgt.

PDM-Metadaten

Die Produktentwicklung ist ein Prozess, der infolge inhaltlich abgeschlossener, zeitlicher und sachlogischer Funktionen Prozessobjekte generiert oder verändert. So werden beispielsweise virtuelle Prototypen initial erstellt, infolge des Design Reviews (bis zur optimierten Lösung) iterativ bearbeitet und abschließend an die Produktion weitergegeben. Die konsistente Verwaltung der virtuellen Prototypen erfolgt innerhalb der PLM-Lösung durch das Produktdatenmanagement (PDM).

Bild 2-10 zeigt wie das Produktmodell im PDM mittels so genannter Artikel und Dokumente beschrieben ist. Artikel sind jene Komponenten des Produktes, die im späteren Produktionsprozess erstellt, montiert oder zugepackt werden. Hierzu zählen Einzelteile, Baugruppen beliebiger Komplexität oder das gesamte Produkt. Beziehungen zwischen den Artikeln, wie zum Beispiel die klassische Stückliste, sind mittels der Artikelstruktur abgebildet. Während der Entwicklung generierte dateibasierte Informationen (Produktdaten) entsprechen Dokumenten. Beziehungen zwischen Dokumenten resultieren aus der Dokumentstruktur. Die den Dokumenten zugeordneten Dateien sind aus Gründen der Performance, Wartung und Verfügbarkeit getrennt von den Artikel- und Dokumentstammdaten in so genannten electronic vaults gespeichert. Beziehungen bzw. Relationen zwischen den Entitätstypen Artikel und Dokument sowie Datei und Vault sind in der Regel vom Typ n:m. Das bedeutet, jede Entität kann mit beliebig vielen Instanzen einer anderen Entität in Verbindung stehen. Die Entität Dokument referenziert beliebig viele Entitäten vom Typ Datei (1:n).

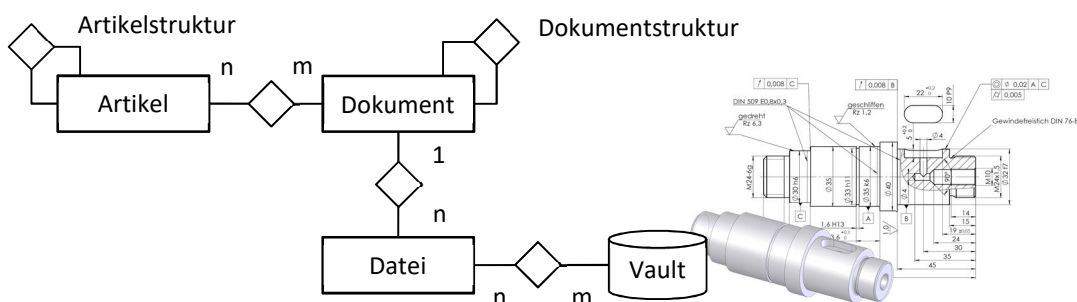


Bild 2-10: Verwaltung des Produktmodells mit PLM als Entity-Relationship Modell

Entitäten besitzen identifizierende und klassifizierende Merkmale, bezüglich der Verwaltung, Prozesssteuerung und Organisation im Produktentwicklungsprozess. Diese Merkmale werden als Metadaten bezeichnet. Sie bestehen aus alphanumerischen Zeichen mit einem Speicherbedarf von wenigen Kilobytes und werden innerhalb des PDM erstellt, verwaltet und verändert. Grundlagen für die Ausgestaltung der Metadaten sind unter anderem Anforderungen des Qualitätsmanagements nach DIN EN ISO 9001, Sachmerkmale

entsprechend DIN 4000, Spezifikationen des Dokumentenmanagements nach DIN EN ISO 11442 sowie Aspekte der Nummerierung nach DIN 6763.

Im Design Review sind Metadaten zur Identifizierung von Bedeutung, wenn dateibezogene Empfehlungen und Maßnahmen konsistent in den Produktentwicklungsprozess zurückzuführen sind. Aus Sicht des PDM sind Dokumentnummer und Änderungsindex die entscheidenden Identifizierungsmerkmale. Die Dokumentnummer ist hierbei das primäre Identifizierungsmerkmal. Sie ist eindeutig und durch ein Nummernsystem gegliedert (z. B. 001 ... 999). Der Änderungsindex bzw. die Versionsnummer ermöglicht den Zugriff auf Revisionsstände eines Dokumentes, die sich im Laufe der Lebenszeit ergeben. Dieser Index ist meist numerisch und wird automatisch inkrementell erhöht, wenn kompatible Änderungen an Dokumenten im PDM eingepflegt werden. Erfordern Prozessobjekte unterschiedliche Metadaten bzw. Ausprägungen von Metadaten, so wird dem durch voneinander getrennten Ordnungsbereichen entsprochen. Man spricht dann von Dokumententypen (Bild 2-11).

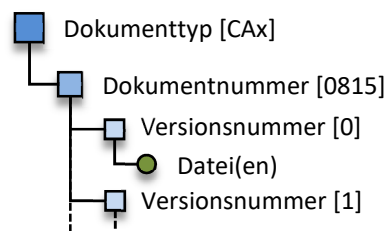


Bild 2-11: Metadaten zur Identifizierung von Dokumenten bzw. Dateien

Klassifizierende Metadaten dienen dem Zusammenfassen bzw. Abgrenzen von Entitäten die einander ähnlich sind. Klassifizierungen beziehen sich zum einen auf Eigenschaften, die die Entität selbst beschreiben, wie beispielsweise Farbe, Masse, Werkstoff und Teilefamilie. Zum anderen auf Beziehungen, die Entitäten zu ihrem Umfeld haben, wie zum Beispiel Hersteller, Produktreifegrad und Bestellnummer.

Klassifizierende Metadaten sind bei der Generierung von Maßnahmen im Design Review von Bedeutung. Sie dienen zur Identifizierung bereits existierender Komponenten, die für die Umsetzung von Maßnahmen in Frage kommen. Hierzu werden zum Beispiel die Komponenten einer Teilfamilie ermittelt und auf Kompatibilität getestet. Das Ziel der Unternehmen ist, Anpassungs- bzw. Variantenkonstruktionen den kosten- und zeitintensiven Neukonstruktionen vorzuziehen.

Metadaten, die das Produkt zu seinem Umfeld beschreiben, gewährleisten im Design Review die Absicherung der Unternehmensprozesse. Wichtige prozessuale Entscheidungsgrundlage für die Generierung von Empfehlungen und Maßnahmen ist beispielsweise das Freigabe- und Änderungswesen. Metadaten zur Freigabe zeigen auf wie Maßnahmen im Ergebnis des Design

Reviews umgesetzt werden können. So dürfen Dokumente, die beispielsweise den Status *freigegeben* besitzen, nicht bearbeitet werden, da sie in den nachgelagerten Phasen der Produktentwicklung bereits Anwendung finden. Änderungen sind in diesem Fall durch eine neue Version des Produktes zu realisieren. Hierbei wird unter Berücksichtigung einer Änderungskennung ein weiteres Dokument im PDM erzeugt. Das Freigabe- und Änderungswesen dient im Design Review auch dazu, den Produktentwicklungsprozess transparent nachzuvollziehen. Jede Freigabe und Änderung ist im PDM durch Informationen zu Datum, Zuständigkeit und Grund dokumentiert. Bild 2-12 zeigt das Freigabe- und Änderungswesen am Beispiel eines Zylinderkopfes. Der vereinfachte Freigabeablauf eines Dokumentes besteht aus den Status *inArbeit*, *inPrüfung* und *freigegeben*.

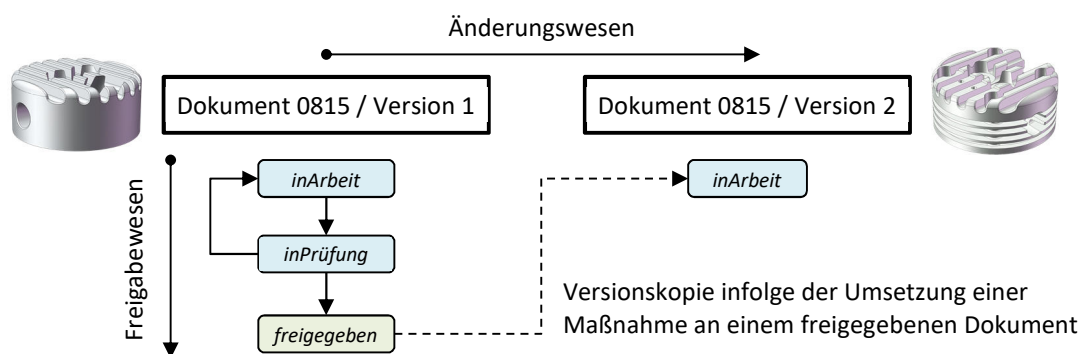


Bild 2-12: Beispiel Freigabe- und Änderungswesen

Mit steigendem Produktumfang (z. B. Airbus A340-600 mit ca. 4.000.000 Einzelteilen [Zer00]) entsteht eine große Anzahl von Entitäten, Prozessen und zugehörigen Metadaten im PDM. Die Folge sind komplexe Modelle, deren Gesamtverhalten nur schwer zu erfassen ist, auch wenn vollständige Informationen zu Prozessen und Prozessobjekten vorliegen. Dieser Zustand gründet maßgebend darauf, dass die Benutzerschnittstellen kommerzieller PDM-Lösungen durch listenartige Ansichten geprägt sind. Gleichzeitig ist es unmöglich im Rahmen der Modellaufbereitung komplexer Produkte alle eventuell erforderlichen Metadaten im Vorfeld zu bestimmen.

Identifizierende und klassifizierende Metadaten sind im Design Review von essentieller Bedeutung. Sie ermöglichen die konsistente Rückführung von Ergebnissen, geben Auskunft über die Realisierung von Empfehlungen und Maßnahmen im Rahmen der Unternehmensprozesse und beschreiben die Entwicklungshistorie. In Anbetracht der Bedeutung und Komplexität der existierenden PDM-Metadaten ist im Design Review mit VR ein intuitiver, schneller und vollständiger Zugang zu gewährleisten.

Bekannte Konzepte zur Modellaufbereitung

Die bekannten Lösungsansätze zur Modellaufbereitung unterscheiden sich grundlegend in der Art wie die stereoskopische VR-Szene erzeugt wird. Es werden entweder die in der Anwendungsschicht generierten nativen virtuellen Prototypen direkt verwendet oder zusätzliche Datenbestände erzeugt, die speziell für die Verwendung mit VR geeignet sind.

Die Nutzung nativer virtueller Prototypen im VR erfolgt mittels so genannter OpenGL Intercept Techniques (OGLIT) [LLR13]. Hierzu wird eine Zwischenanwendung (Middleware) verwendet, die in der Schnittstelle zwischen dem Erzeugersystem und der verwendeten Grafikkarte zur Bilderzeugung wirkt. Die Middleware fängt hierbei den initialen Aufruf des Erzeugersystems zur Bilderzeugung ab und generiert einen parallelen Aufruf, der die Grafikkarte ein stereoskopisches Bild berechnen lässt. Die Visualisierung erfolgt über einen Viewer, der mit der Middleware geliefert wird. OpenGL steht im Rahmen dieser Arbeit stellvertretend für die diversen Programmierschnittstellen der Grafikkarte (Bild 2-13).

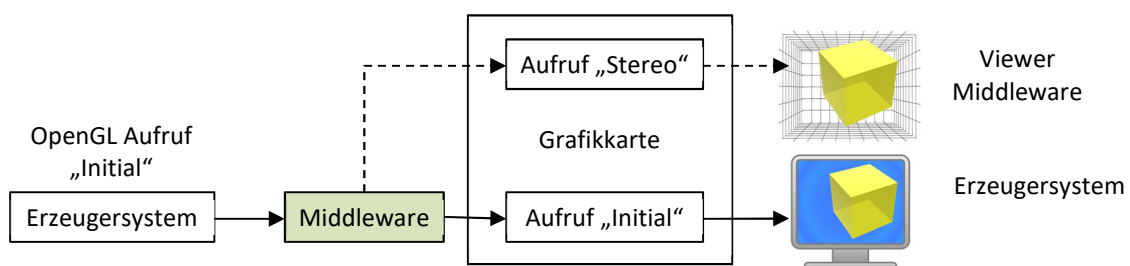


Bild 2-13: Schematische Darstellung OpenGL Intercept Techniques

Mittels OGLIT verfügen Erzeugersysteme auf sehr einfache Weise über die Möglichkeit einer VR-Anbindung. Die Viewer kommerzieller Middleware bieten Tools für beliebige Displaykonfigurationen. Diese positionieren und skalieren abgefangene OpenGL-Aufrufe relativ zueinander und bieten Navigations- und Interaktionsfunktionen. Bezüglich etwaiger Kollaborationsprozesse im Design Review bieten OGLIT den Vorteil, dass die nativen Daten beim jeweiligen Erzeuger verbleiben und nur Pixelinformationen verteilt bzw. an verteilten Standorten genutzt werden. Hierdurch wird der Schutz des geistigen Eigentums unterstützt, denn das native Modell kann nur unter Aufwand rekonstruiert werden [StW13].

Nachteilig am OGLIT Ansatz ist das Fehlen von Metadaten hinsichtlich der Daten- und Prozessschicht des PLM. Metadaten sind kein Bestandteil von OpenGL-Aufrufen, weshalb eine zum PLM konsistente Abbildung des Produktes nicht möglich ist. Darüber hinaus muss das Produkt notwendigerweise parallel zur Präsentationsschicht im CAx-System geladen sein. Bei umfangreichen Modellen resultieren hieraus lange Ladezeiten und träge Reaktionen bei Nutzereingaben [YGK08] im Design Review mit VR. Ursache ist die Visualisierung des nativen virtuellen Prototyps auf pixelbasierten Ausgabegeräten. Grafikschnittstellen wie OpenGL und

Direct3D verarbeiten ausschließlich Geometriebeschreibungen, bei denen die Oberflächen mittels einer endlichen Anzahl von Polygonen beschrieben sind. Man spricht von diskreten Oberflächenbeschreibungen. Native virtuelle Prototypen hingegen sind analytisch beschrieben, d.h. mittels parametrischer Gleichungen. Vor der Visualisierung eines analytisch beschriebenen Prototyps muss zur Visualisierung die Konvertierung in eine diskrete Beschreibung erfolgen (Bild 2-14). Dieser Prozess wird auch als Diskretisierung bezeichnet. Diskrete Oberflächenbeschreibungen, die in ihrer Genauigkeit den analytischen ähnlich sind, zeichnen sich durch eine hohe Anzahl von Polygonen aus.

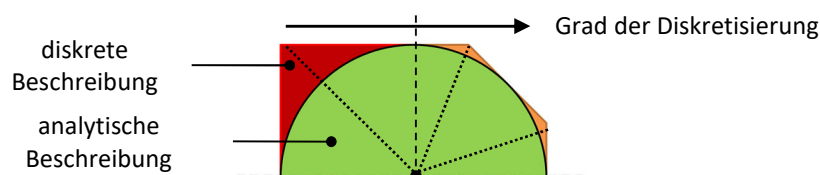


Bild 2-14: Diskretisierung einer analytisch beschriebenen Fläche mittels Dreiecken

Der Berechnungsaufwand der Visualisierung steigt proportional zur Anzahl der Einzelteile und zum Grad der Diskretisierung. Je höher der Berechnungsaufwand, desto weniger Bilder können pro Sekunde ausgegeben werden. Liegt die Anzahl der Bilder pro Sekunde unter ca. 15, wird eine träge Interaktion wahrgenommen.

Eine gegenüber dem OLGIT Ansatz verbesserte Performanz bieten alleinstehende VR-Anwendungen mit integrierten Importfunktionen für native CAx-Daten. Produktstrukturen und Geometrie werden im CAx-Modell analysiert und automatisch in der VR-Szene wiedergegeben. Die aus CAx-Daten abgeleiteten VR-Szenen repräsentieren stets einen festen Entwicklungsstand. Veränderungen im Produktmodell haben zur Folge, dass eine neue VR-Szene generiert werden muss. Dieser Prozess gestaltet sich bei umfangreichen Produktmodellen sehr zeitintensiv. Hinsichtlich spontaner Walkthroughs, bei denen im Vorfeld des Design Reviews ad hoc Empfehlungen am aktuellen Produktmodell ausprobiert werden, ist diese Vorgehensweise nicht praktikabel. Des Weiteren ist beim Import nativer CAx-Daten nicht gewährleistet, dass Informationen zu identifizierenden und klassifizierenden PLM-Metadaten im Design Review zur Verfügung stehen.

Die zweite bekannte Gruppe an Lösungsansätzen, basierend auf Datenbeständen, die speziell für die Verwendung mit VR geeignet sind, unterscheidet sich maßgeblich in der Verwaltung der erzeugten Daten.

In [KaH08], [HSW08], [GrS05] und [NWG10] sind Forschungs- und Entwicklungsprojekte publiziert, die sich der direkten Verbindung von CAx-Systemen und VR-Werkzeugen mittels bilateraler Schnittstellen, spezieller Konvertierungs- und Aufbereitungsprozessen und damit

proprietärer Integrationslösungen widmen. Weiterführend wird in [WLW10] ein Modellansatz benutzt, um VR-Szenen mit CAD-Komponenten so zu koppeln, dass sofortige punktuelle Aktualisierungen in VR nach CAD-Änderungen möglich sind. In den zitierten Arbeiten werden die Strukturbeschreibung sowie die VR-konformen Daten zunächst unabhängig von einer konsistenten unternehmensweiten Prozess- und Datenschicht dargestellt. Lösungsansätze dieser Art zeichnen sich durch redundante Daten aus, die es erschweren, einen aktuellen Datenbestand zu pflegen.

In [GBS02] taucht erstmals die grundlegende Idee der Verknüpfung von Produktstrukturen der unternehmensweiten Prozess- und Datenschicht mit VR auf. Dieser Idee folgend, werden in [SKW06] Produktstrukturen und VR-konforme Daten aus im PDM gepflegten CAD-Daten abgeleitet und in einem lokalen VR-Datenmodell referenziert. Das in [SKW06] eingeführte Memphis System gewährleistet, dass die VR-Szene stets dem aktuellen Entwicklungsstand entspricht. Im Zusammenhang mit der Lösungsbewertung im Design Review werden ähnlich zu [WLW10] Modellmodifikationen direkt im VR-System vorgenommen und die Änderungen ins lokale VR-Datenmodell übertragen. Das lokale VR-Datenmodell wird hierbei jedoch unabhängig von der PDM-Lösung und der darin integrierten Erzeugersysteme verstanden (Bild 2-15). VR dient somit nur als Präsentationsschicht, aus der keine Ergebnisse in die unternehmensweite Prozess- und Datenschicht zurückgeführt werden können.

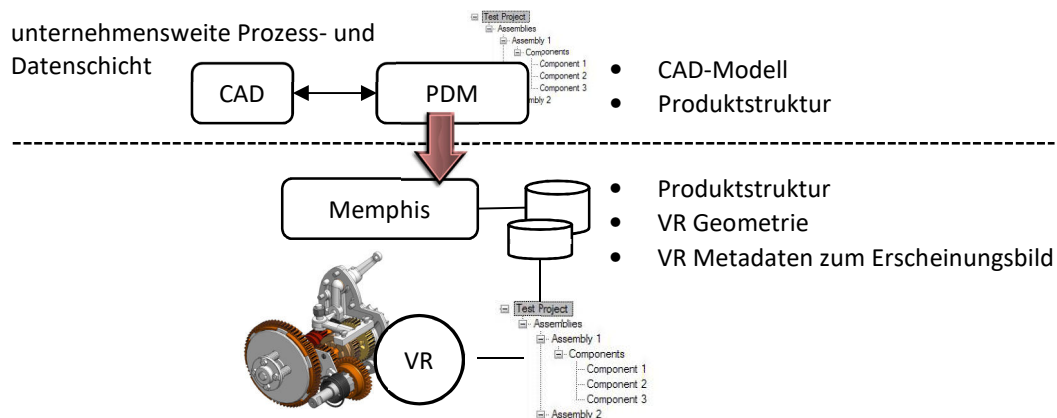


Bild 2-15: Memphis System nach [SKW06]

Die kommerzielle Lösung vPLM [ReW12] beschreibt einen Ansatz zur Integration der VR-Technologie in eine PLM-Lösung, der die in [SPS09A], [SPS09B] und [PSS11] beschriebenen Konzepte zur Entwicklung eines Integrationskonzepts anwendet. Soweit in der kommerziellen Lösung beschrieben, gehören hierzu Objekte der VR-Szene vom Typ Shader, Lichter, Kameras und Umgebungen. Diese sind im PDM über eine Klassifizierung strukturiert verwaltet. Die zur Visualisierung erforderlichen JT-Geometriedaten werden während des Hochladens der nativen CAD-Modelle im PDM erzeugt und um Materialinformationen erweitert. Die

Verwaltung von Produktstruktur, Position und Orientierung von Baugruppen und Einzelteilen erfolgt systemspezifisch in der PDM-Lösung. [ReW12] integriert die VR-Technologie vollständig im PLM. Bezüglich einer heterogenen Systemlandschaft im Produktentwicklungsprozess ist jedoch davon auszugehen, dass neutrale Datenformate wie VRML, STEP und IGES ebenfalls ihre Anwendung finden. Deren Metadatenmodell differenziert system- und unternehmensspezifisch von der geforderten JT Beschreibung. Hinsichtlich verschiedener im Einsatz befindlicher kommerzieller PLM-Lösungen und VR-Systeme muss ebenfalls von heterogenen Systemlandschaften ausgegangen werden. Die Nutzung eines spezifisch erweiterten neutralen Datenformats, einer spezifischen PLM-Lösung und eines spezifischen VR-Systems beschränkt die Anwendung der angebotenen Lösung in einer heterogenen Systemlandschaft maßgeblich.

Zusammenfassend wird festgestellt, dass keiner der bekannten Lösungsansätze einem kollaborativen Produktentwicklungsprozess, der heterogene Daten aus verschiedenen Anwendungsbereichen nutzt, austauscht, transformiert und verwaltet, entspricht.

2.3.2 Organisation und Handhabung

Design Reviews zeichnen sich durch die Zusammenarbeit mehrerer Teilnehmer aus. Der erforderliche Personenkreis und deren mitwirkende Tätigkeit ist in der DIN EN 61160 durch ein Rollenmodell beschrieben. Folgende Rollen sind vorgesehen:

- Sitzungsleiter (Leitung des Design Reviews),
- Schriftführer (Dokumentation von Fragen, Antworten, Empfehlungen und Maßnahmen),
- Entwickler (Präsentation der Produktdaten und Beantwortung von Fragen),
- Fachkraft (Bewertung) und
- Kunde/Anwender (Bewertung).

[SBD10] konzipiert ein für Mehrbenutzer-Design-Reviews abweichendes Rollenmodell, das sich aus den Anforderungen der Automobilindustrie ergibt. Hinsichtlich der Nutzung virtueller Prototypen wird die Rolle eines technischen Masters beschrieben, der für die Steuerung der Hard- und Software verantwortlich ist.

Verteilte Entwicklungsarbeiten bedingen im Sinne der Kollaboration einen erhöhten Aufwand, den erforderlichen Personenkreis im Design Review zur richtigen Zeit an den richtigen Ort zu berufen. Lange Intervalle zwischen Bedarf und Durchführung können die Folge sein. Dieser Umstand wird vermieden, wenn jedem Teilnehmer des Design Reviews ein verteilter Zugriff auf die Präsentationsschicht gewährt wird. Diese Art der Nutzung von VR entspricht dem Ansatz der virtuellen verteilten Umgebung (Distributed Virtual Environments - DVEs) [Mil11]. Hinsichtlich eines großen Personenkreises im Design Review entsprechen aus technologischer Sicht derzeit nur DVEs einer hochauflösenden, echtzeitfähigen und getrackten Darstellung des Produktes mit VR [VDC13]. Deren Leistungsfähigkeit wird derzeit maßgeblich durch die Spieleindustrie demonstriert.

Heutzutage nutzen verteilte Entwicklungsteams meist informationstechnische Infrastrukturen und Technologien, wie Instant Messaging, Internet- und Videotelefonie, virtuelle Skizziertafeln und geteilte Bildschirminhalte, um Informationen im Design Review in Echtzeit zu teilen. Stand der Technik sind so genannte virtuelle Projekträume bzw. Portale, in denen Informationen entsprechend den Bedürfnissen der Nutzer bereitgestellt und verwaltet sind [WaS07]. Hinsichtlich der Anforderungen im Design Review existieren personenspezifische Rollen zur Beschreibung der mitwirkenden Tätigkeit einzelner Teilnehmer sowie sichere Authentifizierungsverfahren und Speicherbereiche. Mittels elementarer, vordefinierter Arbeitsabläufe werden Informationen und Ressourcen wiederholt und prozesssicher aktiviert [SPS10]. Virtuelle Projekträume folgen somit der Strategie des PLM.

Die Abbildung verteilter Entwicklungsarbeiten in einem virtuellen Projektraum bzw. virtuellen verteilten Umgebungen setzt voraus, dass die Modelle und Daten sowie internen Vorgehensmodelle der zusammenarbeitenden Partner einander entsprechen [PSS11]. Dieser Anspruch ist derzeit nur mangelhaft unterstützt, sodass virtuelle Projekträume nur eingeschränkt nutzbar sind. Für Projektpartner ist der Aufwand zu hoch, die unternehmensspezifischen Projektstrukturen temporär in vorgegebene zu integrieren. Die Abstimmung unternehmensübergreifender Prozesse erfolgt daher manuell und bedingt damit immer wieder Unterbrechungen und Missverständnisse während der Zusammenarbeit. Von [HeM06] geführte Interviews mit Industrieunternehmen zeigen, dass:

- kurzfristige Änderungen in den Entwicklungszielen,
- externe und interne Interaktionen sowie
- fehlende oder unvollständige Informationen

immer wieder Anlass für kritische Situationen sind. Diese sind laut Studie als „Norm“ anzusehen. Standardisierte Integrationsschnittstellen bzw. Kommunikationsprotokolle sind soweit nicht bekannt. Mit dem Ansatz der Open Services for Lifecycle Collaboration sind eine Reihe von Spezifikationen beschrieben, um die Integration von Daten verschiedener Software-Tools zu unterstützen [FuP16]. Bezüglich der in der Produktentwicklung verwendeten Modelle und Daten steht eine Spezifikation aus.

Zusammenfassend wird festgestellt, dass die Durchführung eines Design Reviews mit VR zum einen Anforderungen an eine unternehmensübergreifende Produktmodellbeschreibung stellt. Zum anderen werden personenspezifische Rollen zur Beschreibung der mitwirkenden Tätigkeit einzelner Teilnehmer sowie sichere Authentifizierungsverfahren benötigt. Im Sinne der Kollaboration muss das Design Review mit VR dem Konzept der virtuellen verteilten Umgebung folgen.

2.3.3 Rückführung der Ergebnisse

DIN EN 61160 sieht vor, die Ergebnisse aus Teilprozess 6.6 mittels eines Sitzungsberichtes zu dokumentieren. Initial enthält er Informationen zu Teilnehmern, Untersuchungsgegenstand, Referenzdokumenten, Fragen und Mängel. Während des Design Reviews werden vom Schriftführer sukzessiv weitere Informationen hinsichtlich der Empfehlungen und Maßnahmen hinzugefügt. Nach DIN EN 61160 sollte der Sitzungsbericht u. a. folgende Informationen enthalten:

- zu Teilnehmern,
- zur Kennung des bewerteten Produktes,
- eine Zusammenfassung der Erkenntnisse,
- eine Liste der Empfehlungen,
- eine Kurzfassung der geplanten Maßnahmen,
- zur Sache gehörende Fragen und Stellungnahmen,
- eine Liste der Maßnahmen zu unbeantworteten und ungelösten Fragen;
- zur Person, die für eine Maßnahme verantwortlich gemacht wurde;
- zum Datum für die Erledigung von Maßnahmen,
- Status bzw. Erledigung von Maßnahmen und Empfehlungen aus früheren Sitzungen sowie
- eine Liste von Bezugsdokumenten.

Gewonnenes Erprobungswissen, das die Durchführung von Teilprozess 6.6 methodisch beschreibt, ist nach DIN EN 61160 nicht Bestandteil des Sitzungsberichts.

Inhalt und Ausgestaltung des Sitzungsberichts sind unternehmensspezifisch und orientieren sich an den Anforderungen an Qualitätsmanagementsysteme nach EN ISO 9001. Als Informationsträger sind Textdokumente und Tabellen in digitaler Form weit verbreitet. Darüber hinaus sind dem Sitzungsbericht häufig digitale Bildschirmfotos des virtuellen Prototyps zugeordnet, in denen die Ergebnisse notiert sind. Kommerzielle Anwendungen wie Präsentationssoftware und Textverarbeitung bieten hierfür umfangreiche Gestaltungsmöglichkeiten. Weit verbreitet ist die Einbettung von Text, Grafik und Symbolen (Bild 2-16). Editierte Bildschirmfotos transportieren bei Bedarf einen sehr detaillierten Informationsgehalt.

Einbauuntersuchung					Schneidwerk SE43 10000056432_CAD_000_02				
Holzernter H931 10000055713_CAD_000_01					Thematik – Einbauuntersuchung				
Datum 01.01.2014 232B					Winkelstellung des Resthebers reduziert. Ab 20° Neigung kollidiert Grundplatte des Resthebers mit Klemmbügel des Schutzbügels.				
SL	Petermann	BK	Saske						
SF	Walther		Steger						
EW	Böhmer		Heller						
LFDN	Komponente	ID	Status	Maßnahme	Per				
1	Fahrwerk	10000055789	Grün	-					
2	Rahmen	10000055793	Gelb	Versteifung entfernen	Sem				
3	Hydraulik	10000055842	Grün	-					
4	Bündel	10000055863	Grün	-					
5	Schneidwerk	10000056432	Rot	Überarbeitung Grundplatte	Holtzhausen	DRS	25.04.2014		

Bild 2-16: Tabellarischer Sitzungsbericht mit referenziertem und kommentiertem Bildschirmfoto

Neben der Dokumentation von Ergebnissen anhand von Bildschirmfotos, können Empfehlungen und Maßnahmen mittels Annotationen am virtuellen Prototyp dokumentiert werden. Beispiel hierfür sind Produktfertigungsinformationen (Bild 2-17). Die Darstellungsart und der Inhalt der PMI sind in der DIN ISO 16792 standardisiert, um die Konsistenz der Informationen zwischen virtuellem Prototypen und Zeichnung in einem Datensatz sicherzustellen. Bezüglich der Anforderungen an die Dokumentation des Design Reviews enthält DIN ISO 16792 keine expliziten Standardisierungen. Die Gestaltungsmöglichkeiten der Dokumentation von Empfehlungen und Maßnahmen im Design Review sind daher auf die genannten Produktfertigungsinformationen begrenzt. Hinsichtlich der Rückführung von Ergebnissen zeichnen sich Annotationen gegenüber separaten Dokumenten, durch einen geringeren Verwaltungsaufwand aus.

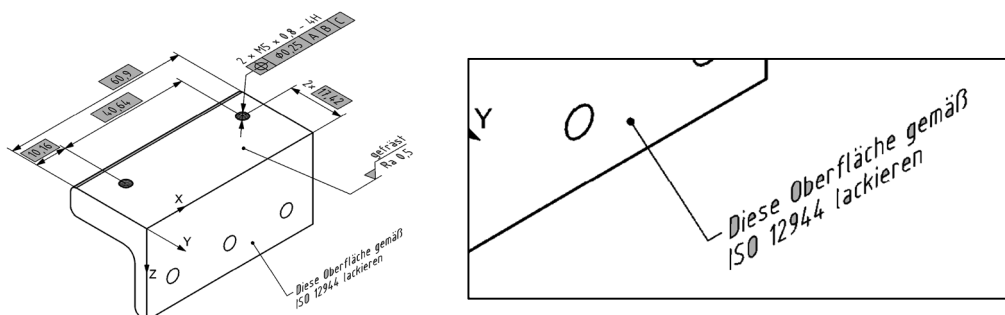


Bild 2-17: Produktfertigungsinformationen am virtuellen Prototyp nach [DIN16792]

Die Praxis zeigt, dass der während des Design Reviews erstellte Sitzungsbericht unabhängig von der gewählten Dokumentationsart einer redaktionellen Nachbearbeitung unterliegt. Hierzu zählen beispielsweise die Verknüpfung von Bildschirmfotos zu Empfehlungen, die

Ermittlung von zuständigen Sachbearbeitern sowie die Integration der erzeugten Dokumente in den Kontext des unternehmensweiten Informations- und Prozessmodells.

Die redaktionelle Nachbearbeitung erfordert das Detailwissen der Mitglieder des Design Reviews. Ist dieses nicht greifbar, sind Verzögerungen und Unsicherheiten im Prozess des Design Reviews die Folge. [SKK12] beschreibt beispielsweise eine Untersuchung der Dokumentation von Fehlerereignissen bei der ca. ein Fünftel der detektierten Fehler nicht eindeutig dokumentiert wurden. Nach der Durchführung von Maßnahmen sind in der Praxis meist weitere redaktionelle Nacharbeiten am Sitzungsbericht erforderlich, mittels derer der Abschluss dokumentiert wird.

Zusammenfassend wird festgestellt, dass die bekannten Ansätze für die Ausgestaltung des Sitzungsberichts den Anforderungen an Qualitätsmanagementsysteme entsprechen. Nachteilig ist, dass der Sitzungsbericht aufgrund fehlender Information während der Bewertung im Design Review einer redaktionellen Nachbearbeitung unterliegt. Die manuelle Auswertung der im Design Review generierten Informationen führt in Anbetracht umfangreicher Unterlagensätze zu Verzögerungen und Unsicherheiten im Prozess.

2.4 Wertung der bestehenden Situation und Zielstellung

Gegenüber den klassischen Erzeugersystemen, bietet VR als Präsentationsschicht für eine Vielzahl der im Design Review zu bewertenden Merkmale Vorteile (siehe Kapitel 2.2). Hinsichtlich der ganzheitlichen Darstellung des virtuellen Prototyps wird VR jedoch nur nach Entwicklungsetappen eingesetzt.

Die diskontinuierliche Verwendung von VR im Design Review ist dem Umstand geschuldet, dass die VR-Technologie nicht als Teil des PLM-Konzepts verstanden wird und damit keine effizienten Abläufe im Design Review unterstützt. Die VR-Szene selbst sowie die im Design Review gewonnenen Ergebnisse bedürfen einer zeitintensiven manuellen Aufbereitung.

Wie in Bild 2-18 dargestellt, werden native Produktdaten im Vorfeld des Design Reviews während der Überführung ins PDM zunächst in ein neutrales Datenformat konvertiert, sofern diese noch nicht in einem VR-konformen Format vorrätig sind. Anschließend werden die konvertierten Daten für eine spezifische Anwendung der Präsentationsschicht im Kontext der Strukturen des PDM manuell aufbereitet. Hierbei entsteht ein spezifisches Produktmodell für die Verwendung mit VR-Systemen. Dieses wird im Rahmen dieser Arbeit als VR-Produktmodell bezeichnet. Entspricht die im Design Review mit VR gewünschte Darstellung nicht dem Informationsgehalt des VR-Produktmodells, werden abschließend fehlende Informationen manuell im VR-System ergänzt. Der benötigte Zeitaufwand für die Generierung des VR-Produktmodells steigt proportional mit dem Umfang des PDM-Produktmodells und des geforderten Detaillierungsgrades. Im Ergebnis existiert ein statisches, lokal gespeichertes VR-Produktmodell, das im Design Review mit VR abgerufen werden kann. Erfolgen zwischen Generierung und Abruf des VR-Produktmodells Änderungen am ursprünglichen PDM-Produktmodell, sind diese im Design Review nicht berücksichtigt. Gleiches gilt für Informationen die während der Generierung des VR-Produktmodells unberücksichtigt blieben. Die Infolge der Bewertung entstehenden Ergebnisse werden abschließend manuell nachbereitet. Hierbei werden zunächst die Inhalte des Sitzungsberichtes inklusiver etwaiger Bezugsdokumente sowie Maßnahmen in den Kontext des PDM gebracht. Anschließend wird der Sitzungsbericht selbst im PDM gespeichert und mit dem bewertenden Produkt verknüpft.

Aus Sicht der Kollaboration ergeben sich weitere Herausforderungen, die die Verwendung von VR im Design Review derzeit einschränken. Hierzu gehören die Bewertung heterogener virtueller Prototypen, differierende Daten- und Prozessschichten der Kollaborationspartner, die Initiierung anwendungsübergreifender Arbeitsabläufe sowie der Einsatz von VR im Sinne einer virtuellen verteilten Umgebung.

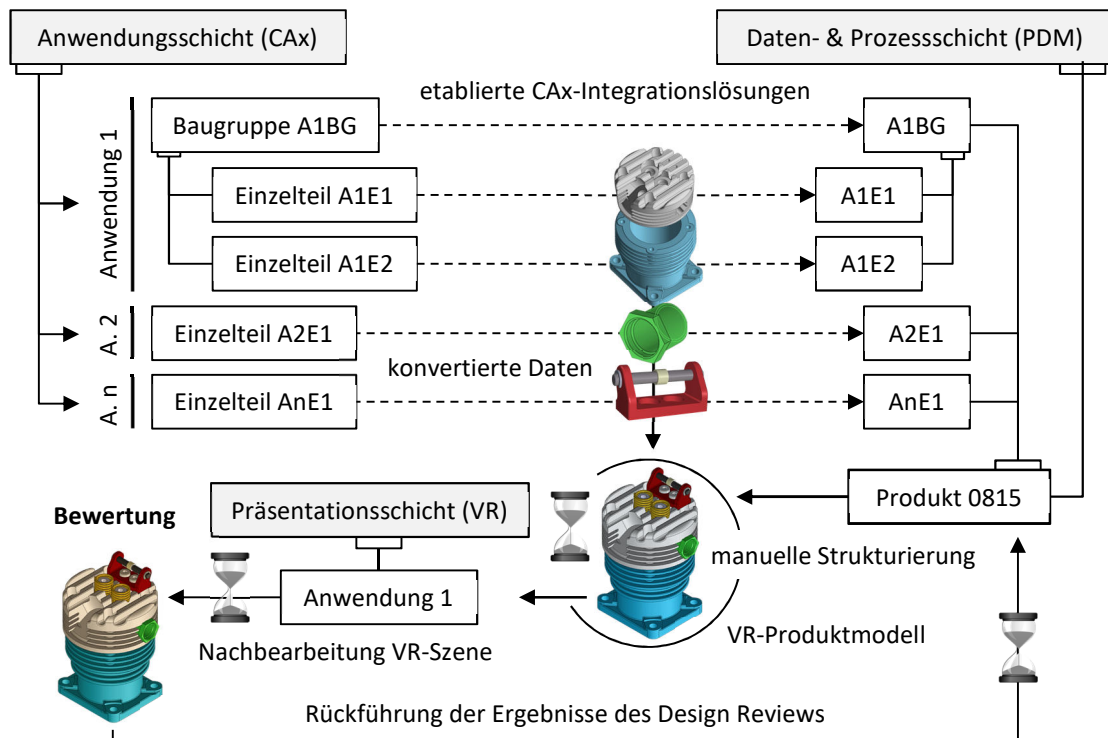


Bild 2-18: Visualisierungsprozess heterogener, virtueller Prototypen mit VR

In DIN EN 61160 sind hinsichtlich der Nutzung informationstechnisch verwalteter, verteilt entwickelter, nativer virtueller Prototypen im Design Review keine Lösungsvorschläge gegeben. Es ist ausschließlich darauf hingewiesen, dass die für die Durchführung der Entwicklungsbewertung notwendigen Ressourcen und Informationen bereitzustellen, Entwicklungsbewertungstätigkeiten zu überwachen und Ergebnisse zu messen und zu analysieren sind.

Ziel dieser Arbeit ist die Ausarbeitung eines Konzeptes, das das Design Review mit VR entlang der Wertschöpfungskette der modernen Produktentwicklung entsprechend der herausgearbeiteten Teilprozesse Modellaufbereitung, Organisation und Handhabung sowie Rückführung der Ergebnisse im PLM integriert. Im Mittelpunkt der Konzeptentwicklung stehen die Anforderungen:

- beliebige in der Daten- und Prozessschicht des PLM verwaltete virtuelle Prototypen für die gemeinsame Verwendung innerhalb der Präsentationsschicht abzubilden,
- gleiche virtuelle Prototypen der Anwendungs- und Präsentationsschicht für anwendungsübergreifende Arbeitsabläufe bidirektional zu koppeln,
- PLM-Metadaten innerhalb der Präsentationsschicht zu integrieren,
- die konsistente Rückführung von Ergebnissen aus dem Design Review in den Produktentwicklungsprozess zu gewährleisten sowie
- die praxistaugliche Anwendung des Design Reviews mit VR im Sinne einer verteilten Kollaboration.

Die Summe der aufgeführten Anforderungen wird im Rahmen dieser Arbeit als VR-PLM Integrationskonzept bezeichnet. Das VR-PLM Integrationskonzept gründet ausschließlich auf Integrationen der Daten- und Objektebene (Bild 2-19). Das bedeutet, die Beschreibung des unternehmensspezifischen Prozesses zur Lösungsfindung, bestehend aus Bewertung und Maßnahmengenerierung entsprechend DIN EN 61160:6.6, ist aufgrund der unterschiedlich gelebten, teils einzigartigen Natur vom Konzept ausgenommen.

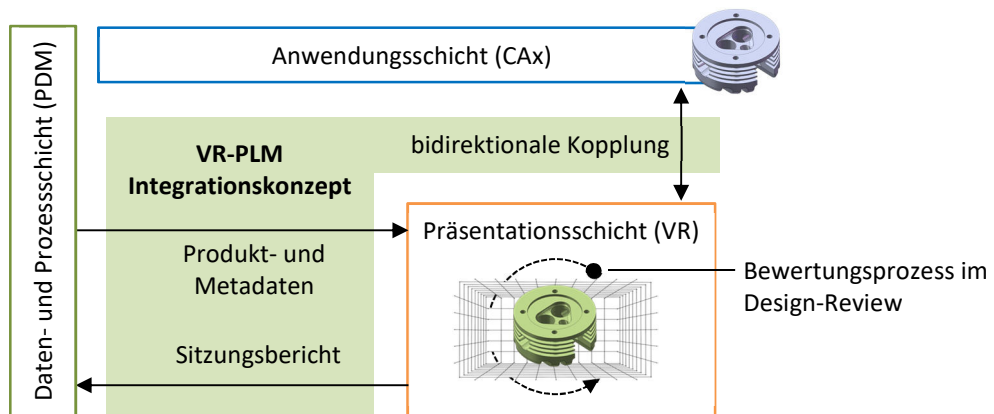


Bild 2-19: Abgrenzung des VR-PLM Integrationskonzepts vom Bewertungsprozess

Datenintegrationen, welche virtuelle Prototypen im Sinne des Produktentwicklungsprozesses verwalten, jedoch nicht interpretieren (Bild 3-2).

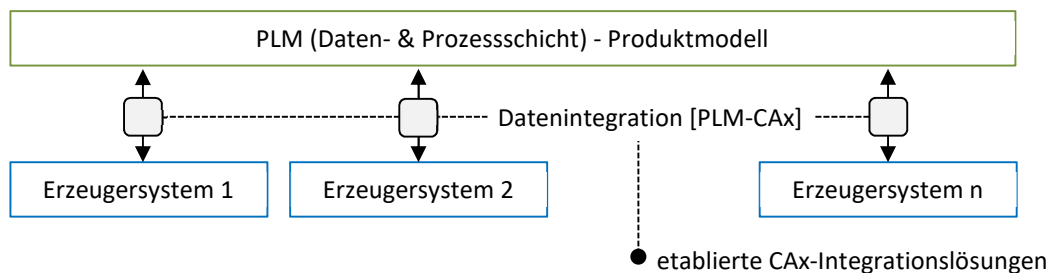


Bild 3-2: klassische CAx-Integrationslösungen

Die im Design Review mit VR darzustellenden Informationen leiten sich aus den nativen CAx-Modellen ab, die innerhalb des PDM als Dokumente verwaltet sind. CAx-Modelle bzw. virtuelle Prototypen sind mittels systemspezifischer Datenelemente und -strukturen beschrieben, die im Objektmodell des CAx-Systems definiert sind. VR-Systeme definieren ebenfalls systemspezifische Objektmodelle. Obwohl die Objektmodelle hinsichtlich der Gestalt und des Erscheinungsbildes des virtuellen Prototyps gleiche Informationen abbilden, sind sie aufgrund ihrer systemspezifischen Art inkompatibel.

Die Interoperabilität der systemspezifischen Objektmodelle von CAx- und VR-Systemen realisiert das VR-PLM Integrationskonzept durch ein Datenmapping. Das bedeutet, Informationen werden im CAx-System ausgelesen, für die Verwendung im VR-System aufbereitet und abschließend im VR-System geschrieben. Da entsprechend DIN EN 61160 die Teilprozesse Modelaufbereitung und Durchführung des Design Reviews zeitlich getrennt durchgeführt werden können, sind die während des Datenmappings aufbereiteten Informationen in Form eines Datenmodells persistent und konsistent zu speichern (Bild 3-3).

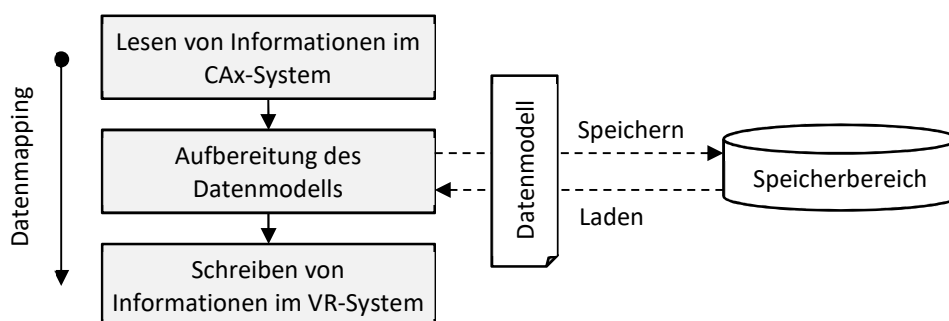


Bild 3-3: Datenmapping mit PES-Produktmodell

Darüber hinaus muss das Datenmodell derart gestaltet sein, dass die enthaltenen Informationen unabhängig von einer spezifischen VR-Lösung beschrieben sind. Der Umfang

des Datenmodells resultiert aus den im Design Review mit VR erforderlichen visuellen und strukturellen Informationen des nativen virtuellen Prototyps. Aufgrund der Art der verwalteten Informationen wird das Datenmodell im weiteren Verlauf dieser Arbeit als PES-Produktmodell bezeichnet. Die Abkürzung PES leitet sich aus dem Englischen ab und steht für die wesentlichen Produktmerkmale („**P**roduct-**E**ssentials“) im Design Review. Die Entwicklung des PES-Produktmodells steht im Fokus dieser Arbeit.

Damit sich die im Design Review gewonnenen Ergebnisse konsistent in den Produktentwicklungsprozess zurückführen lassen, ist es erforderlich die im VR-System visualisierten Objekte im Kontext des PDM zu belassen. Das bedeutet, neben den strukturellen und visuellen Informationen muss das PES-Produktmodell identifizierende PDM-Metadaten enthalten. Hierdurch ist es möglich den mit VR visualisierten virtuellen Prototyp innerhalb der Daten- und Prozessschicht des Unternehmens zu identifizieren. Darüber hinaus können auf Basis der identifizierenden PDM-Metadaten anwendungsübergreifende Arbeitsabläufe zwischen den integrierten Programmen der Präsentations-, Anwendungs- sowie Daten und Prozessschicht realisiert werden.

- **Definition PES-Produktmodell:** Datenmodell, das einen virtuellen Prototyp innerhalb der Daten- und Prozessschicht des PLM identifiziert und aus visueller sowie struktureller Sicht für die Verwendung mit VR abbildet.

Klassifizierende PDM-Metadaten, die projekt- und unternehmensspezifische prozessuale Abhängigkeiten im Produktentwicklungsprozess abbilden, werden nicht als Bestandteil des PES-Produktmodells aufgefasst. Klassifizierende PDM-Metadaten werden zur Laufzeit des Design Reviews auf Grundlage der identifizierenden PDM-Metadaten ermittelt und mittels geeigneter Methoden zur Ansicht gebracht.

Das VR-PLM Integrationskonzept setzt voraus, dass für jedes im VR-System zu visualisierende Objekt eine aktuelle diskrete Geometriebeschreibung im PDM verwaltet ist. Das Laden der diskreten Geometriebeschreibung erfolgt während des Datenmappings, auf Basis der im PES-Produktmodell verwalteten identifizierenden PDM-Metadaten. Diskrete Geometriebeschreibungen werden im VR-PLM Integrationskonzept nur auf Objekte des VR-Systems angewendet, die keine Produktstruktur darstellen. Hierdurch lassen sich virtuelle Prototypen darstellen, die verschiedene Arten von diskreten Geometriebeschreibungen nutzen.

Das PES-Produktmodell entsteht sukzessiv mit dem Verlauf der Produktentwicklung, wenn virtuelle Prototypen neu erzeugt bzw. bearbeitet werden und im PDM gespeichert werden. Parallel zum Speichern wird der virtuelle Prototyp analysiert und ein PES-Produktmodell erzeugt, das die visuellen und strukturellen Informationen inklusive der erforderlichen identifizierenden PDM-Metadaten abbildet. Weist der virtuelle Prototyp eine Produktstruktur

auf, wird für jedes rekursiv ermittelte Strukturelement ein zusätzliches partielles PES-Produktmodell erzeugt. In diesem Fall bildet die Gesamtheit der partiellen Modelle das PES-Produktmodell (Bild 3-4).

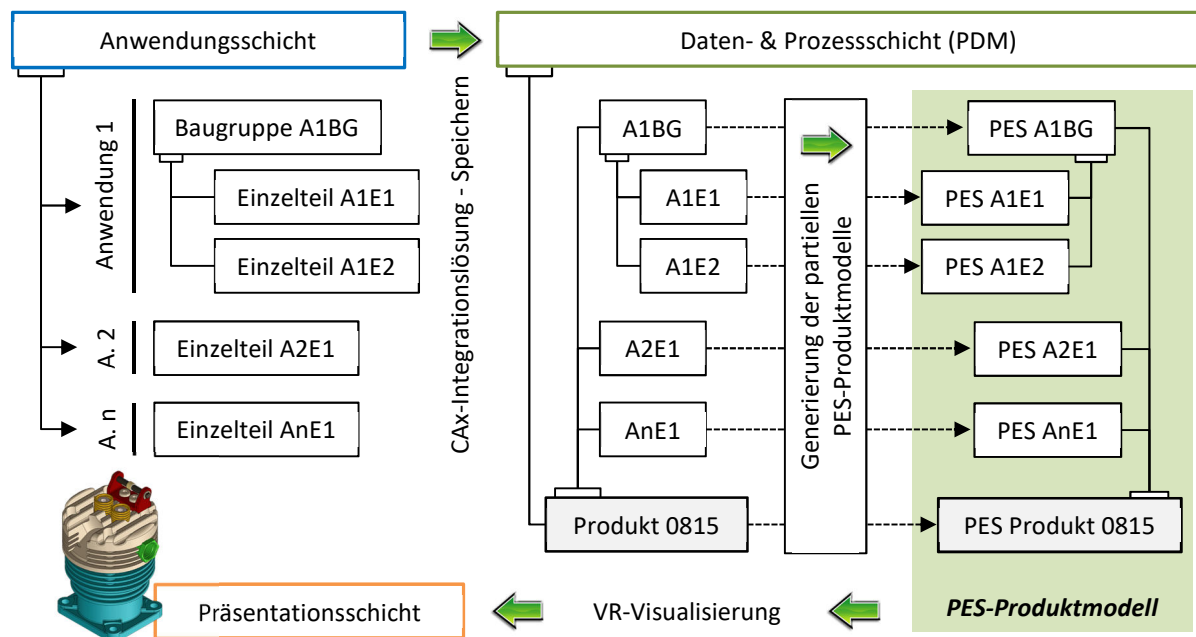


Bild 3-4: Entstehung und Verwaltung des PES-Produktmodells

Die konsistente Verwaltung der partiellen PES-Produktmodelle erfolgt im PDM. Mit dieser Vorgehensweise entfällt die zeitaufwendige Konvertierung nativer virtueller Prototypen im Vorfeld eines Design Reviews mit VR, wie sie noch in vielen Unternehmen üblich ist. So ist praktisch jederzeit und zeitnah möglich, die VR-Visualisierung des aktuell verfügbaren Datenbestands zu einem Produkt mittels der partiellen PES-Produktmodelle durchzuführen, ohne dass besondere Aktivitäten zur Vorbereitung erforderlich sind (Bild 3-4). Eine Ausnahme bildet das erstmalige Einchecken von Daten eines Zulieferers oder eines externen Kooperationspartners. Hier kann der einmalig hohe Aufwand der PES-Produktmodellgenerierung durch konfigurierte Batch-Prozesse außerhalb der Spitzenzeiten erbracht werden [PSS13].

Die im Design Review mit VR ermittelten Empfehlungen und Maßnahmen werden im VR-PLM Integrationskonzept mittels eines informationstechnischen Sitzungsberichtes abgebildet. Hierzu wird im Rahmen dieser Arbeit ein weiteres Datenmodell entwickelt, das den Anforderungen an die Dokumentation nach DIN EN 61160 entspricht. Aufgrund der Art der verwalteten Informationen wird das Datenmodell im weiteren Verlauf dieser Arbeit als SES-Modell bezeichnet. Die Abkürzung SES leitet sich aus dem Englischen ab und steht für die wesentlichen Ergebnisse („**S**ession-**E**ssentials“) im Design Review.

Das SES-Modell entsteht sukzessiv während der Bewertungsphase im Design Review, indem Empfehlungen und Maßnahmen mittels geeigneter Benutzungsschnittstellen aufgezeichnet

werden. Für die konsistente Verwaltung der Empfehlungen und Maßnahmen bildet das SES-Modell alle Ergebnisse des Design Reviews im Kontext des PDM ab. Das bedeutet, jede Empfehlung oder Maßnahme ist um PDM-Metadaten ergänzt, die das im VR-System bewertete Objekt innerhalb der unternehmensweiten Daten- und Prozessschicht identifizieren. Darüber hinaus muss das SES-Modell analog des PES-Produktmodells durch das VR-PLM Integrationskonzept im PLM verwaltet sein.

- **Definition SES-Modell:** Datenmodell, das die Ergebnisse des Design Reviews entsprechend DIN EN 61160 informationstechnisch abbildet und mit den virtuellen Prototypen der Daten- und Prozessschicht des PLM verknüpft.

Ziel des SES-Modells ist zum einen, die redaktionelle Nachbearbeitung der während des Design Reviews mit VR generierten Ergebnisse zu minimieren bzw. aufzuheben. Zum anderen können nach Abschluss der Bewertungstätigkeit, Maßnahmen automatisiert initiiert und innerhalb der Präsentationsschicht überwacht werden.

Im Hinblick auf ein verteilt stattfindendes Design Review, das mehrere Systeme der Anwendungs- und Präsentationsschicht eingliedert, ist das VR-PLM Integrationskonzept um eine Middleware erweitert (Bild 3-5). Sie ist erforderlich, um über die Komplexität und Heterogenität der zugrundeliegenden IT-Infrastruktur mit zum Beispiel einer Vielzahl von Rechnerarchitekturen, Betriebssystemen, Netzwerkprotokollen und Programmiersprachen zu abstrahieren. Sie bildet die informationstechnische Grundlage für die Vermittlung von Funktionsaufrufen zwischen entkoppelten Softwarekomponenten im Design Review mit VR. Voraussetzung hierfür ist, dass die zu integrierenden Anwendungen über Programmierschnittstellen verfügen (Application-Programming-Interfaces API).

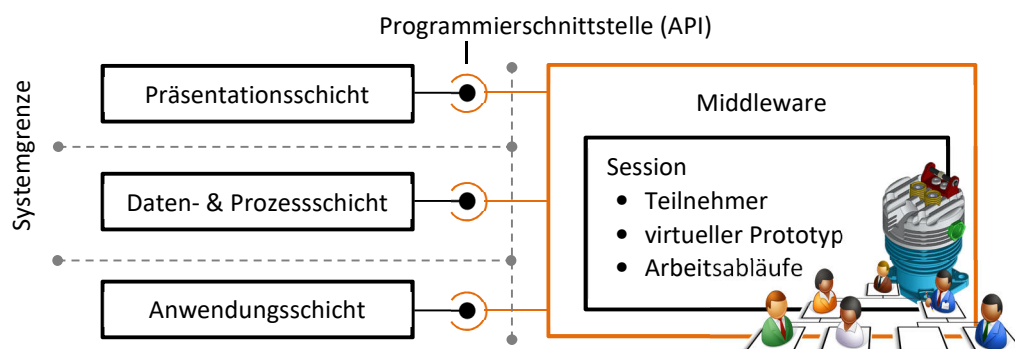


Bild 3-5: Informationstechnisches Umfeld für das Design Review mit VR

Bild 3-5 zeigt ebenfalls, dass neben der Middleware das VR-PLM Integrationskonzept um ein zweites informationstechnisches Umfeld ergänzt ist. Die Session bildet zum einen das Design Review hinsichtlich der integrierten Anwendungen, verwendeten virtuellen Prototypen und Teilnehmer ab. Zum anderen überwacht und kontrolliert sie die Interaktionen der Teilnehmer

im Design Review. Die Session bildet somit Aspekte der Organisation und Handhabung im VR-PLM Integrationskonzept ab.

Während der Entwicklung des VR-PLM Integrationskonzepts besteht die allgemeine Randbedingung, dass alle am Design Review beteiligten Anwendungen wie beispielsweise VR- und CAx-Systeme anforderungskonform einzusetzen sind. Das bedeutet, alle am Design Review beteiligten Systeme werden in ihren Domänen wirksam belassen und nicht durch Funktionen erweitert, über die eines der jeweils anderen Systeme ohnehin verfügt. So steht zum Beispiel CAD für die Erstellung und Bearbeitung exakter parametrischer 3D-Geometriemodelle einschließlich funktionaler und fertigungstechnischer Produktmerkmale. VR nutzt diese Informationen, um anhand von Darstellungen in hoher Wahrnehmungsqualität die Beurteilung digitaler Produktmodelle zu verbessern [PSS11].

Im Rahmen des Abbildungsraumes des PES-Produktmodells können beliebig viele VR-Anwendungen als Präsentationsschicht mit dem gleichen Produktmodell genutzt werden. Das VR-PLM Integrationskonzept unterstützt aus Daten- und Prozesssicht eine durch Systems Engineering und integrierte Produktentwicklung geprägte heterogene Systemlandschaft, in der Entwicklerteams parallel und verteilt zusammenarbeiten. Mit der Ausweitung des PLM-Konzepts auf den Prozess und die Prozessobjekte des Design Reviews mit VR verschiebt sich die bestehende Position des Werkzeuges VR im Verlauf des Produktentwicklungsprozesses. Wird die VR-Technologie bisher als Resultat der mangelnden Integration meist am Ende von Entwicklungsetappen verwendet, kann sie mittels im PLM integrierter PES- und SES-Modelle als stetige Basis im Produktentwicklungsprozess verwendet werden (Bild 3-6). Hierdurch lassen sich Entwicklungsarbeiten am virtuellen Prototyp im Rahmen der unternehmensweiten Daten- und Prozessschicht des PLM konsistent bewerten und in iterativen Abläufen organisieren.

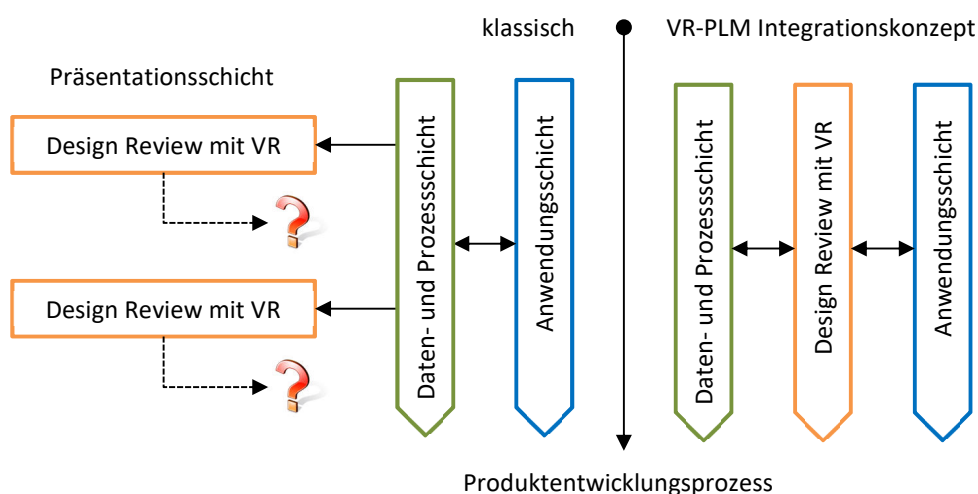


Bild 3-6: Integration des Design Reviews mit VR in den Produktentwicklungsprozess mittels des VR-PLM Integrationskonzeptes

4 Daten- und Prozessmodellierung

4.1 Konventionen zur Beschreibung der Datenmodelle und des Systemverhaltens

Die in diesem Kapitel entwickelten Datenmodelle und Systemabläufe sind mittels der Unified Modeling Language (UML) beschrieben. UML ist eine grafische Modellierungssprache zur Spezifikation, Konstruktion und Dokumentation von Software-Teilen und anderen Systemen [Bal11]. Im Rahmen dieser Arbeit erfolgt die Abbildung von Datenmodellen in Form von Klassendiagrammen; Prozesse sind mittels Aktivitätsdiagrammen beschrieben. Beide Diagrammarten sind unabhängig von der Implementierung.

Im Folgenden sind die in dieser Arbeit verwendeten UML-Notionen für Klassen- und Aktivitätsdiagramme nach [OMG15] erläutert. In der Praxis existieren weitere UML-Notationen, die im Rahmen dieser Arbeit keine Anwendung finden. Für eine weiterführende Einführung in UML wird auf [OMG15] verwiesen.

4.1.1 Klassendiagramm

Die Modellierung von Datenmodellen erfolgt im Rahmen dieser Arbeit in Form von Klassendiagrammen. Eine Klasse repräsentiert in der objektorientierten Programmierung ein Modell für Objekte des abzubildenden Systems. Jede Klasse ist durch Eigenschaften und Methoden beschrieben, die sich entweder in einem privaten oder öffentlichen Zustand befinden. Während der Zugriff auf private Eigenschaften und Methoden nur innerhalb der Klasse möglich ist, kann im öffentlichen Zustand ein Zugriff auch von außerhalb der Klasse erfolgen. Der private Zustand ist durch ein vorangestelltes Minus-Zeichen, der öffentliche durch ein vorangestelltes Plus-Zeichen gekennzeichnet (Bild 4-1). Der Datentyp der Eigenschaft ist dem Namen, durch einen Doppelpunkt abgetrennt, nachgestellt. Für eine weiterführende Einführung in Datentypen wird auf [MuW15] verwiesen.

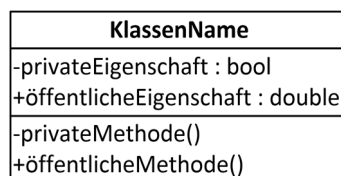


Bild 4-1: UML-Notion einer Klasse

Beziehungen zwischen Klassen sind durch so genannte Assoziationen im UML dargestellt und an einer durchgezogenen Linie zwischen zwei Klassen zu erkennen (Bild 4-2). Die Wirkungsweise der Assoziation ist mittels der Assoziationsenden definiert. Innerhalb dieser

Arbeit sind die Assoziationsenden auf Aussagen zur Kardinalitätsrestriktion und Navigation begrenzt. Die Kardinalitätsrestriktion beschreibt, wie viele Objekte einer Klasse mit den Objekten einer anderen Klasse in Beziehung stehen können oder müssen. Das PES-Produktmodell ist auf Kardinalitätsrestriktionen vom Typ 1:1 und 1:* (1:n) beschränkt. Über die Navigation ist beschrieben welche Klasse Wissen über die jeweils andere besitzt. Notationen zur Navigation sind durch eine offene Pfeilspitze dargestellt. Im Beispiel in Bild 4-2 kennt *Klasse1* die *Klasse2*.

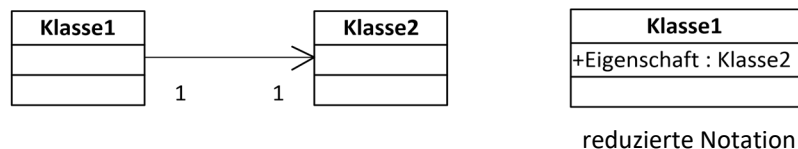


Bild 4-2: UML-Notion einer 1:1 Assoziation

Eine gerichtete Beziehung zwischen einer allgemeinen und einer bzw. mehreren spezialisierten Klassen wird als Generalisierung bezeichnet (Bild 4-3). Die allgemeine Klasse dient dazu, gleiche Eigenschaften und Methoden der spezialisierten Klassen in einem gemeinsamen Objekt zu kapseln. In den spezialisierten Klassen verbleiben nur ungleiche Eigenschaften und Methoden. Wird von einer spezialisierten Klasse eine Instanz bzw. ein Objekt erstellt, beinhaltet diese sowohl allgemeine als auch spezialisierte Eigenschaften und Methoden. Die Generalisierung ist durch eine Assoziation gekennzeichnet, deren Assoziationsende durch eine geschlossene, nicht ausgefüllte Pfeilspitze gekennzeichnet ist. Die Pfeilspitze zeigt stets auf die allgemeine Klasse.

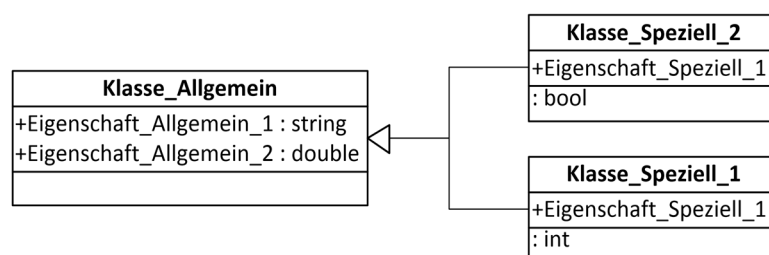


Bild 4-3: UML-Notion Generalisierung

Eine Enumeration ist ein Datentyp für Variablen mit einer endlichen Wertemenge. Sie wird zum Deklarieren einer Gruppe benannter Konstanten verwendet, welche bei der Deklaration mit einem eindeutigen Namen definiert werden. Im PES-Produktmodell dienen Enumerationen zur Definition von Zuständen der VR-Szene selbst bzw. von darin enthaltenen Objekten. Bild 4-4 zeigt die UML-Notation einer Enumeration sowie deren Verwendung innerhalb einer Klasse.

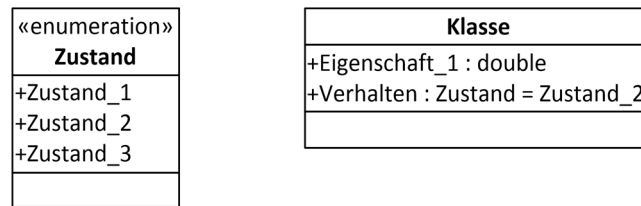


Bild 4-4: UML-Notion Enumeration

Wenn es darum geht, eine bestimmte Anzahl von Methoden und Eigenschaften einer Klasse objektorientiert zu kapseln, definiert man so genannte Klassenschnittstellen bzw. Interfaces (Bild 4-5). Man spricht von angebotenen Schnittstellen, wenn die Klasse gewährleistet, dass die in der Schnittstelle definierten Methoden implementiert sind. Angebotene Schnittstellen sind in UML entweder durch eine gestrichelte Assoziation gekennzeichnet, deren Assoziationsende durch eine geschlossene, nicht ausgefüllte Pfeilspitze, oder durch ein Ballsymbol gekennzeichnet ist. Dem gegenüber steht die Verwendung einer Schnittstelle als Anforderung an die Implementierung einer Klasse. Man spricht dann von einer benötigten Klassenschnittstelle. In diesem Fall wird von der abonnierenden Klasse erwartet, dass sie die definierten Methoden und Eigenschaften umsetzt. Benötigte Schnittstellen sind in UML entweder mittels einer gestrichelten Assoziation gekennzeichnet, deren Assoziationsende durch eine offene Pfeilspitze oder durch ein Sockelsymbol gekennzeichnet ist.

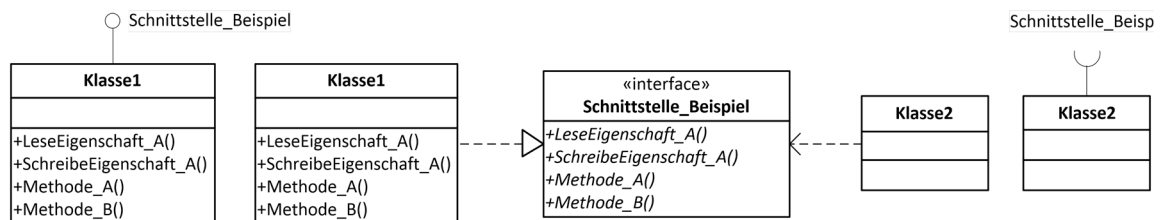


Bild 4-5: UML-Notion Klassenschnittstelle

Neben Schnittstellen können in der objektorientierten Programmierung auch abstrakte Klassen verwendet, um Methoden und Eigenschaften zu kapseln. Als abstrakt gekennzeichnete Methoden müssen in einer Klasse, die von einer abstrakten Klasse ableitet, überschrieben bzw. neu implementiert werden. Methoden, die als nicht abstrakt gekennzeichnet sind, werden regulär vererbt. Abstrakte Klassen sind in der UML-Notation an einem kursiv dargestellten Klassennamen zu erkennen.

4.1.2 Aktivitätsdiagramm

UML-Aktivitätsdiagramme beschreiben die prozedurale Verarbeitung von Arbeitsschritten zur Realisierung eines Verhaltens. Eine Aktivität ist als ein gerichteter Graph zu verstehen, dessen Knoten Arbeitsschritte und dessen verbindende Kanten Kontroll- und Datenflüsse beschreiben. Die in diesem Abschnitt aufgeführten UML-Notationen entsprechen der UML-Version 2.5.

Eine Aktivität ist mittels UML-Aktivitätsdiagramm durch einen viereckigen Rahmen dargestellt, der in der linken oberen Ecke einen abgetrennten Bereich enthält. Dieser enthält den Namen der Aktivität. Innerhalb des Rahmens befinden sich mindestens ein Initial- und Endknoten sowie Elemente, die verschachtelte Aktivitäten, Arbeitsschritte (Aktionen), Datenobjekte, Kontroll- und Datenflüsse modellieren (Bild 4-6). Verschachtelte Aktivitäten sind wie Aktionen durch abgerundete Vierecke dargestellt und um ein Gabelsymbol erweitert. Vierecke ohne abgerundete Ecken entsprechen Datenobjekten. Flüsse sind Kanten zwischen Aktivitäten, Aktionen und Datenobjekten.

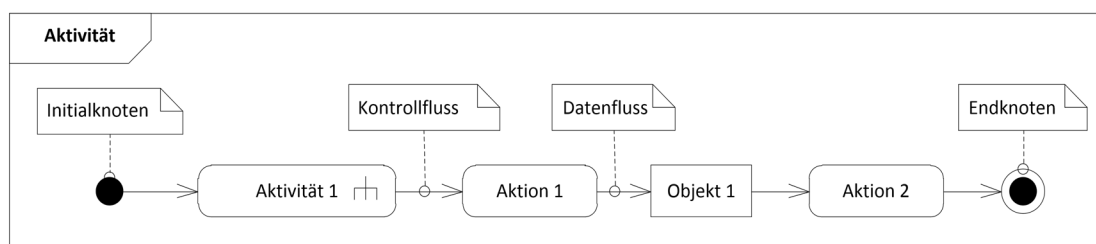


Bild 4-6: Notation Aktivitätsdiagramm I

Kontrollflüsse lassen sich im Verlauf der Aktivität verzweigen und/oder parallelisieren (Bild 4-7). Verzweigungen sind alternative Kontrollflüsse, die einander ausschließen und aus dem Zustand des Systems resultieren. Beginn und Ende einer Verzweigung ist jeweils ein Raute-Symbol. An der einleitenden Raute ist die Bedingung eingetragen nach welcher verzweigt wird. Die von der Raute abgehenden Kontrollflüsse notieren die zugrundeliegende Bedingung. Im Gegensatz zur Verzweigung, wird bei der Parallelisierung der Kontrollfluss unabhängig von einer Bedingung aufgespalten. Die entstehenden Nebenläufe existieren solange nebeneinander bis sie beendet oder synchronisiert werden. Die Synchronisierung erfolgt, wenn alle eingehenden Kontrollflüsse abgearbeitet sind. Parallelisierung und Synchronisierung sind im UML-Aktivitätsdiagramm durch schwarze Balken dargestellt. Ein Kreissymbol mit einem Kreuz symbolisiert das Ende eines verzweigten bzw. parallelisierten Kontrollflusses.

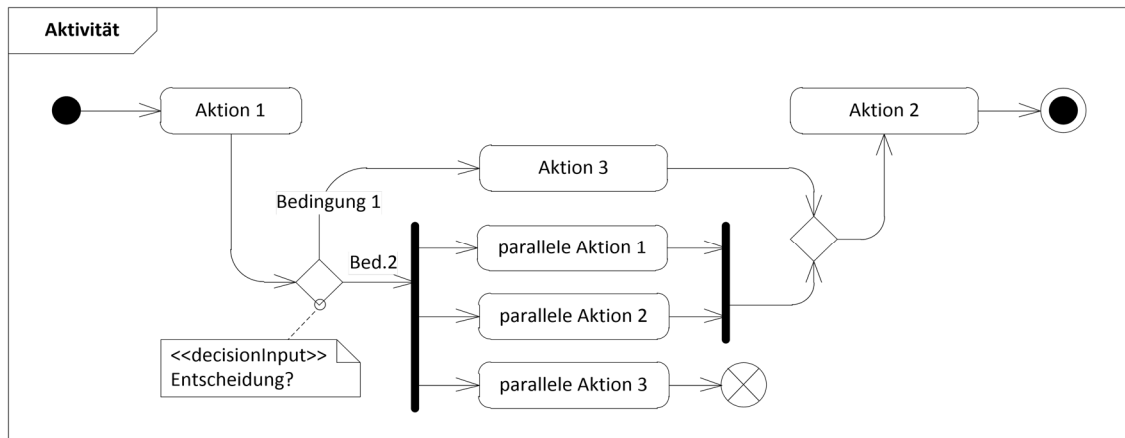


Bild 4-7: Notation Aktivitätsdiagramm II

Die Darstellung von asynchronen Arbeitsschritten erfolgt im UML-Aktivitätsdiagramm durch Sonderformen von Aktionen (Bild 4-8). Hierzu gehören das Senden von Signalen sowie das Empfangen von Ereignissen. Während das Senden von Signalen den Kontrollfluss nicht unterbricht, setzen das Empfangen von Ereignissen und der damit einhergehende Kontrollfluss ein Eingangssignal voraus. Asynchrones Verhalten kann sowohl innerhalb einer Aktivität, als auch Aktivitätsübergreifend definiert sein.

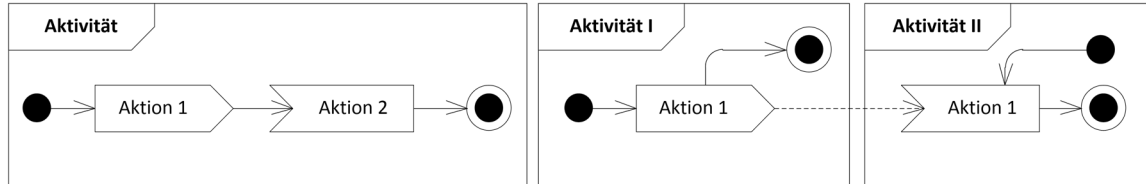


Bild 4-8: Notation Aktivitätsdiagramm III

Der Transport von Daten durch das Netz der Aktivität ist mittels so genannter Objektknoten beschrieben (Bild 4-9). Objektknoten symbolisieren Datenspeicher, die Informationen eines bestimmten Datentyps verwalten. Dies können elementare Datentypen oder Objekte von Klassen sein. Die Erzeugung von Objektknoten erfolgt durch Aktionen bzw. Eingaben. Ist ein Objektknoten Eingangsparameter einer Aktion, kann diese den Wert der Ausprägung verändern oder verarbeiten. Die Relation zwischen Aktion und Objektknoten ist vom Typ 1:n.

Sind zwei Aktionen durch einen Objektknoten verbunden, spricht man von einem Datenfluss. Für die Darstellung des Datenflusses existieren zwei Notationen (Bild 4-9). Zum einen wird ein explizit sichtbarer Objektknoten zwischen zwei Aktionen geschaltet und durch Kanten verbunden. Zum anderen wird der Objektknoten durch so genannte Pins an der Start- und Zielaktion abstrahiert visualisiert. Zwischen den Pins existiert ebenfalls eine Kante, die dann die Ausprägung des Objektes trägt.

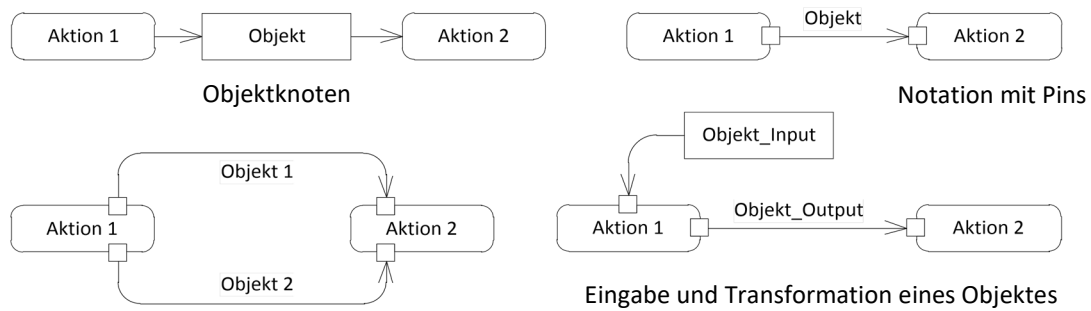


Bild 4-9: Notation Aktivitätsdiagramm IV

Neben der direkten Übermittlung von Daten zwischen Aktionen, kann auch ein so genannter Datastore zum Einsatz kommen (Bild 4-10). Ein Datastore ist ein Pufferknoten, der Daten eines bestimmten Typs aus Objektflüssen persistent und konsistent verwaltet. Die Verwendung eines Datastores ist dann von Vorteil, wenn multiple Aktionen einer Aktivität mit einem gemeinsamen Datenbestand arbeiten. Die Visualisierung erfordert analog zum Objektknoten, dass der Datastore und die zugehörige Aktion durch eine Kante verbunden sind.

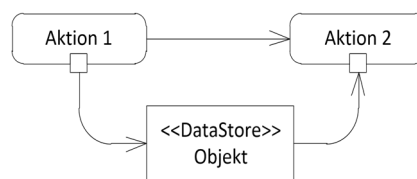


Bild 4-10: Notation Aktivitätsdiagramm V

Mit steigender Anzahl von Aktionen und zugehörigen Kontroll- und Datenflüssen sinkt die Übersicht im UML-Aktivitätsdiagramm. Um dem entgegenzuwirken, empfiehlt sich die Verwendung von so genannten Konnektoren. Konnektoren brechen Kontroll- und Datenflüsse auf und ermöglichen es, diese an einer anderen Stelle im Diagramm fortzuführen (Bild 4-11). An der Trennstelle wird hierzu ein Verbindungspunkt gesetzt, der eine eindeutige Bezeichnung aufweist. Die Fortführung des Flusses beginnt dann am zweiten gleichnamigen Verbindungspunkt im UML-Aktivitätsdiagramm.

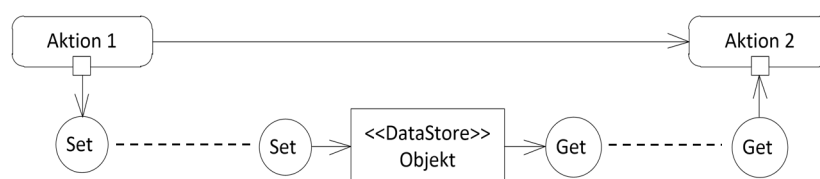


Bild 4-11: Notation Aktivitätsdiagramm VI

Aktionen, die mit einer bestimmten Anzahl von Wiederholungen auszuführen sind, werden als Loop Knoten bezeichnet und mittels so genannter strukturierter Aktivitätsknoten dargestellt (Bild 4-12). Loop Knoten bestehen aus drei Bereichen. Der Setup-Bereich wird beim Einstieg in die Schleife ausgeführt und initiiert Variablen, die in der Abbruchbedingung der Schleife anzuwenden sind. Der Test-Bereich definiert die Abbruchbedingung. Der Body-Bereich enthält Aktionen, die wiederholt ausgeführt werden, bis die Abbruchbedingung eintritt.

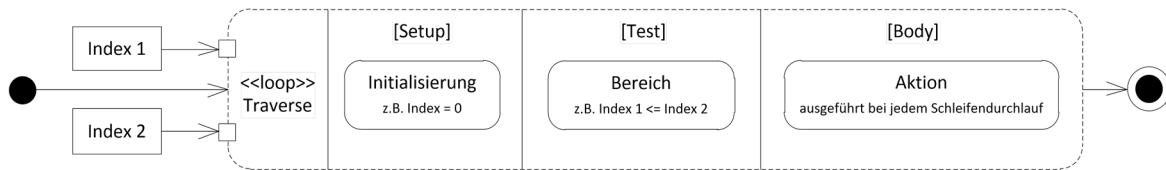


Bild 4-12: Notation Aktivitätsdiagramm VII

4.2 PES-Produktmodell

Die Summe der mit VR dargestellten Daten wird allgemein als VR-Szene bezeichnet. Der Begriff leitet sich aus dem Bereich der Computergrafik ab und beschreibt nach [NFH12] die Synthese von einfachen grafischen Objekten zu einer natürlich wirkenden Szene bzw. eines Szenarios.

Im Rahmen des VR-PLM Integrationskonzepts werden Produktgestalt, Funktion und Präsentationsumgebung als die relevanten Elemente der VR-Szene im Design Review aufgefasst (Bild 4-13) und im PES-Produktmodell verwaltet. Der Begriff Produktgestalt definiert hierbei die visuelle Präsentation des virtuellen Prototyps. Sie resultiert aus der Anwendung von Informationen zur Lage und Orientierung (Transformation) sowie zum Erscheinungsbild auf die VR-konformen Geometriebeschreibungen der Komponenten entlang der Struktur der VR-Szene.

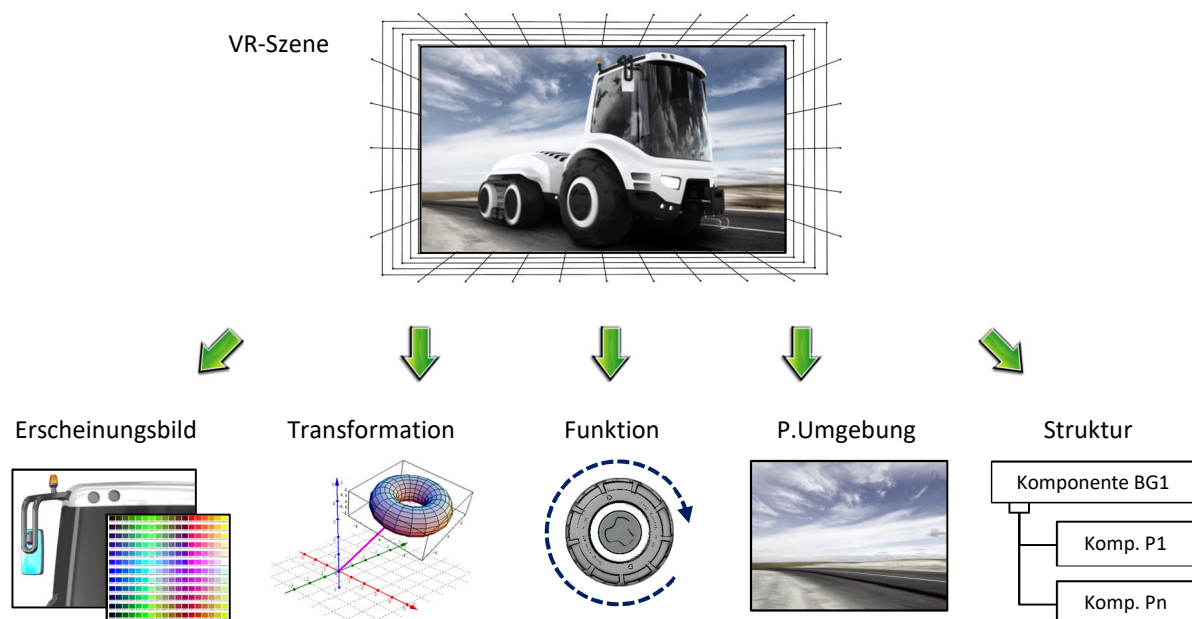


Bild 4-13: Elemente des PES-Produktmodells

Das in VR-Szenen ebenfalls erforderliche Kameramodell wird entsprechend den Untersuchungen in [Woe09] vernachlässigt, wonach von Benutzern in kooperativen Design Reviews gefordert wird, einen individuellen Fokus einstellen zu können und zu ändern, ohne dass dies Auswirkungen auf andere Teilnehmer hat. Damit ist das Kameramodell nicht Teil des PES-Produktmodells und muss nicht persistent gespeichert werden.

4.2.1 Abbildung der Produktstruktur

Die Objekte einer VR-Szene werden im Allgemeinen als gerichtete objektorientierte Datenstruktur bzw. gewurzelte Baumstruktur verwaltet, die als Szenegraph bezeichnet wird. Gerichtete Baumstrukturen bestehen nach [ESB16] aus einer beliebigen Anzahl von Knoten, die so verbunden sind, dass keine Kreisverweise auftreten. Der oberste Knoten des Baumes wird als Wurzel bezeichnet. Von der Wurzel aus ergeben sich alle weiteren Beziehungen zu untergeordneten Knoten. Der Wurzel untergeordnete Knoten die keine weiteren untergeordneten Knoten besitzen werden als Blatt, alle anderen als innere Knoten bezeichnet. Die gerichtete Darstellung definiert, dass nur der jeweils übergeordnete Knoten seine Nachfolger kennt.

Gewurzelte Baumstrukturen finden ebenfalls in den etablierten PDM-Lösungen Anwendung, indem mittels Strukturstücklisten und zweistufigen Baukastenstücklisten die Produktstruktur virtueller Prototypen beschrieben wird. Speziell die zweistufige Baukastenstückliste (ZBS), wird nach [EiS09] als grundlegende Strukturbeschreibung für IT-Anwendungen verstanden. Zweistufige Baukastenstücklisten enthalten einen Knoten der Produktstruktur und referenzieren ausschließlich Komponenten der hierarchisch niedrigeren Strukturebene (Bild 4-14). Die vollständige Produktstruktur resultiert aus der Verknüpfung aller zum Produkt gehörigen zweistufigen Baukastenstücklisten und der darin referenzierten Komponenten.

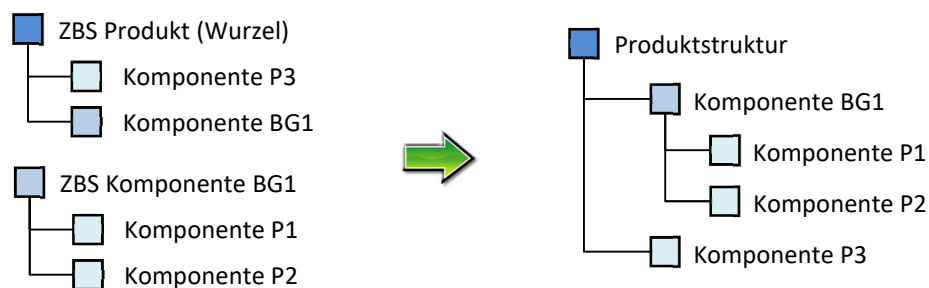


Bild 4-14: Ableitung der Produktstruktur aus zweistufigen Baukastenstücklisten

Die Produktstruktur eines virtuellen Prototyps ist im PES-Produktmodell ebenfalls nach dem Konzept der zweistufigen Baukastenstücklisten beschrieben und existiert parallel zur im PDM verwalteten Produktstruktur. Die im PES-Produktmodell abgebildete Produktstruktur des virtuellen Prototyps wird nachfolgend als PES-Produktstruktur bezeichnet.

Die separate Verwaltung der Produktstruktur im PES-Produktmodell gründet auf CAx-spezifischen Informationen von einzeln bzw. mehrfach verwendeten Komponenten, die für die Visualisierung des Produktes im Design Review mit VR benötigt werden. Im grundlegenden PDM-Konzept bestehend aus der Verwaltung von Dokumenten und Artikeln sind diese jedoch nicht berücksichtigt. Im VR-PLM Integrationskonzept gehören hierzu Informationen zur

Position und Orientierung sowie zum Erscheinungsbild. Anders als in den meisten PDM-Lösungen, sind im VR-PLM Integrationskonzept daher alle Objektinstanzen des Produktmodells innerhalb der PES-Produktstruktur abgebildet. Bild 4-15 verdeutlicht den Unterschied zwischen PDM- und PES-Produktstruktur am Beispiel einer Grundplatte mit zwei Sechskantschrauben.

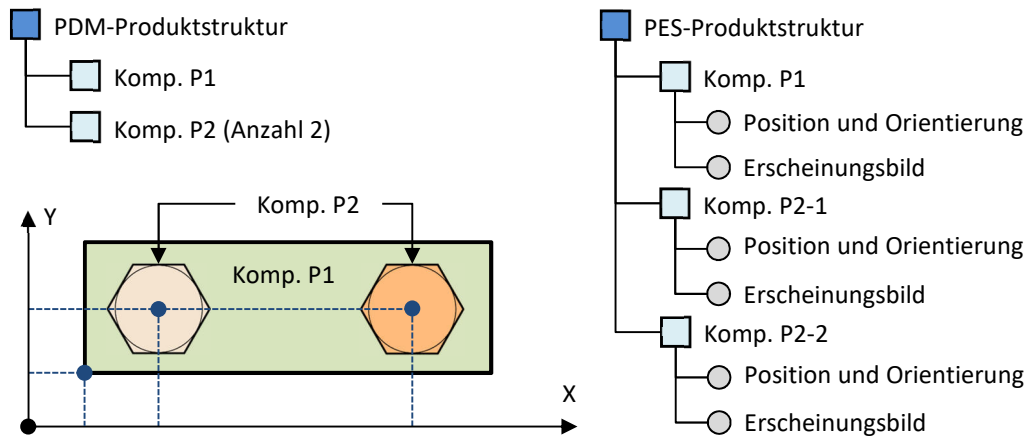


Bild 4-15: Unterschiede zwischen PDM- und PES-Produktstruktur

Gewurzelte Baumstrukturen werden in UML-basierten Datenmodellen mit dem Composite Pattern dargestellt. Das Composite Pattern ist ein Entwurfsmuster der Softwaretechnik, das zur Kategorie der Strukturmuster gehört. Es wird angewendet, um Teil-Ganzes-Hierarchien zu repräsentieren [Gei15]. Die Grundidee hierbei ist, Objekte vom Typ Wurzel, Blatt und innerer Knoten in Form einer Basisklasse zu generalisieren (Bild 4-16). Die Klasse *Komponente* steht stellvertretend für das PES-Produktmodell und kapselt Methoden sowie Attribute die allgemein für Objekte vom Typ Wurzel, Blatt und innerer Knoten gültig sind. In der Klasse *Knoten*, die Objekte vom Typ Wurzel und innerer Knoten beschreibt, erfolgt die Abbildung der Produktstruktur. Dies wird durch eine 1:n Assoziation mit der Klasse *Komponente* ausgedrückt. Das bedeutet, untergeordnete Objekte können vom Typ Knoten und Blatt sein.

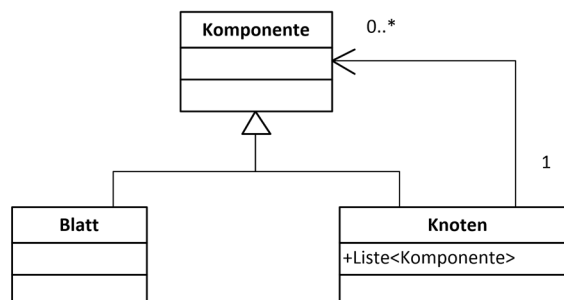


Bild 4-16: Abbildung von Strukturen in UML-basierten Datenmodellen mittels des Composite Patterns

Um zu gewährleisten, dass der Nutzer im Design Review mit VR zwischen den Objekten der VR-Szene und den zugrundeliegenden Objekten im CAx- bzw. PDM-System assoziieren kann, muss die Klasse *Komponente* ein so genanntes sprechendes Merkmal aufweisen. Dieses Merkmal ist im PES-Produktmodell der *KomponentenName*, der Attribut der Klasse *Komponente* ist. Im Rahmen dieser Arbeit wird vorgeschlagen, die Leserlichkeit der Struktur im Kontext des CAx- und PDM-Systems zu unterstützen. Das bedeutet, der *KomponentenName* setzt sich aus identifizierenden Metadaten des CAx- und PDM-Systems zusammen. Für den Fall, dass mehrere Objekte desselben Typs innerhalb einer Strukturebene existieren, wird empfohlen den *KomponentenName* um einen Index zu erweitern. Dieser inkrementiert mit jeder Referenz um eins. Hierdurch wird gewährleistet, dass der *KomponentenName* in jeder Strukturebene eindeutig bleibt. Fügt man ausgehend von der Wurzel bis hin zur zu bezeichnenden Komponente alle *KomponentenNamen* zusammen, resultiert eine eineindeutige Bezeichnung. Diese wird im PES-Produktmodell als *StrukturID* bezeichnet (Bild 4-17). Da jede Komponente durch genau eine *StrukturID* beschrieben ist und jede *StrukturID* genau einer Komponente zugeordnet ist, können alle Objekte des PES-Produktmodells durch ein Attribut konsistent verwaltet werden.

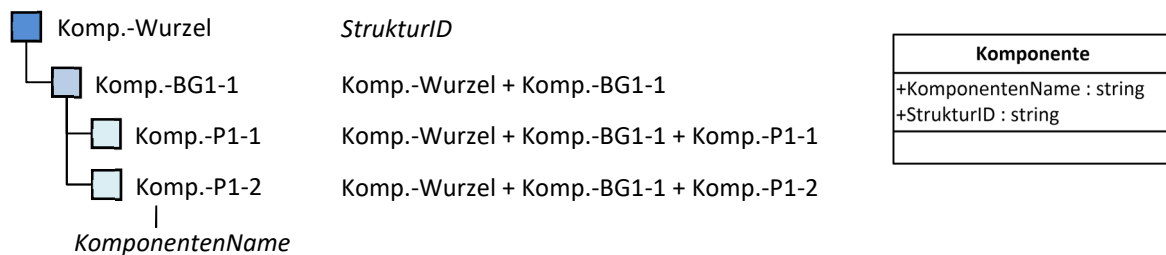


Bild 4-17: Entwicklung der StrukturID

4.2.2 Identifizierende Metadaten

Erfolgreiche Iterationsschleifen im Design Review erfordern, dass die gewonnenen Ergebnisse dem im PLM verwalteten Produktmodell zugeordnet und zurückgeführt werden können. Dieser Zustand ist gegeben, wenn ein Objekt der VR-Szene ein Dokument der PDM-Lösung referenziert. Das bedeutet, die Bewertungstätigkeit im Design Review erfolgt konsistent zu den strukturellen Beschreibungen des Produktes innerhalb der unternehmensweiten Daten- und Prozessschicht des PLM.

Innerhalb der PDM-Lösung erfolgt die Identifizierung eines Dokumentes mittels der Metadaten Dokument- und Versionsnummer. Diese Informationen sind im PES-Produktmodell als Eigenschaften in der Klasse *PDMReferenz* verwaltet. Während des Design Reviews lässt sich somit für jedes Objekt der Kontext zum im PDM verwalteten Produktmodell herstellen. Die Klasse *PDMReferenz* muss immer dann assoziiert werden, wenn ein Objekt des PES-Produktmodells bzw. der VR-Szene im PDM verwaltet ist. Bild 4-18 zeigt die Assoziation der Klasse *PDMReferenz* bezüglich der Klasse *Komponente* im PES-Produktmodell.

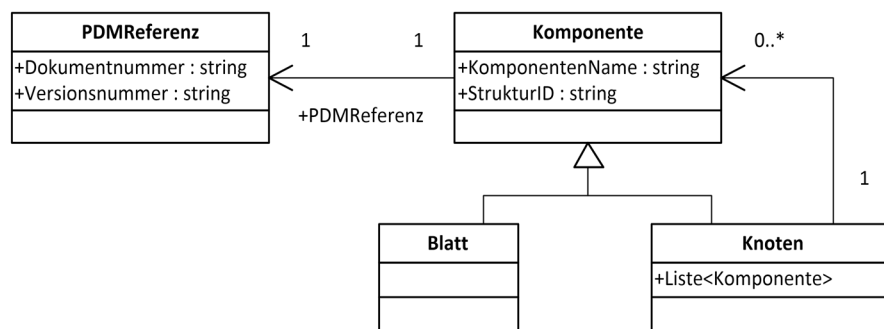


Bild 4-18: Verwaltung der VR-Produktstruktur mit PLM-Metadaten

Die Verwaltung von identifizierenden PDM-Metadaten bildet die Grundlage für die bidirektionale Kopplung der Präsentations- sowie Daten- und Prozessschicht im VR-PLM Integrationskonzept.

4.2.3 Transformation

Mit den strukturellen Zusammenhängen von Komponenten gehen Informationen zur räumlichen Lage einher. Man spricht hierbei von so genannten Transformationen. Im orthogonalen Koordinatensystem sind Verschiebung (Translation), Drehung (Rotation) und Veränderung des Maßstabs (Skalierung) möglich. Bild 4-19 zeigt die genannten Transformationen am Beispiel eines Quaders.

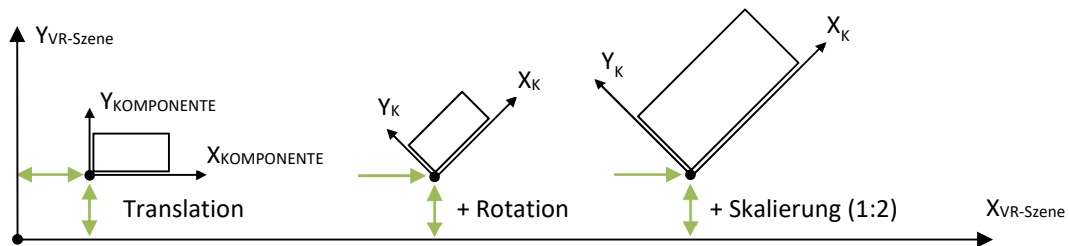


Bild 4-19: Im PES-Produktmodell verwaltete Transformationen

Bei der Translation wird jeder Punkt (K) der Komponente, um einen festen Betrag entlang einer der Raumachsen des Koordinatensystems der CAx-Komponente verschoben. Im dreidimensionalen Raum ergeben sich somit Translationen in X-, Y- und Z-Richtung (BG). Die Beträge der Translationen resultieren aus dem Ortsvektor T des Koordinatensystems der Komponente in Bezug zum übergeordneten Koordinatensystem. Der Ortsvektor T des Koordinatensystems der Komponente ist Parameter im PES-Produktmodell. Er ist informationstechnisch durch ein Feld der Größe drei beschrieben, das reelle Zahlenwerte annimmt. Die Translation eines beliebigen Punktes errechnet sich aus Formel 4-1:

$$\begin{bmatrix} K'_{XT} \\ K'_{YT} \\ K'_{ZT} \end{bmatrix} = \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix} + \begin{bmatrix} K_X \\ K_Y \\ K_Z \end{bmatrix} \quad \text{Formel 4-1}$$

Bei der Rotation wird eine Komponente relativ zum Koordinatensystem der Baugruppe gedreht. Die Verdrehung von Koordinatensystemen wird in der Mathematik mittels Drehmatrizen beschrieben. Bild 4-20 zeigt eine beispielhafte Drehung einer Komponente um die Z-Achse und die dazugehörige Drehmatrix M_{ROT_Z} .

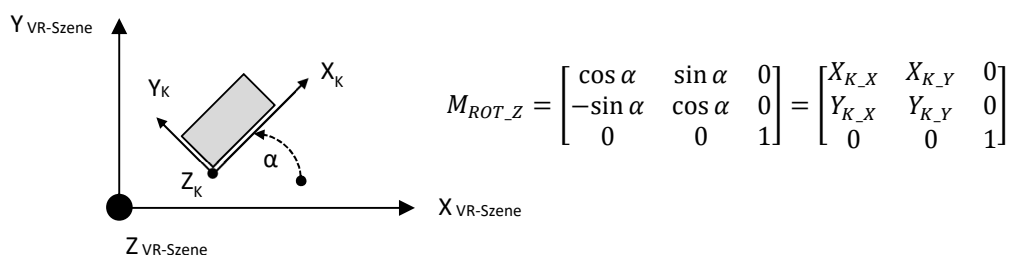
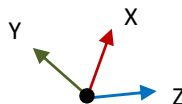


Bild 4-20: Drehung um die Z-Achse I

Nach [DIN9300] bzw. [DIN8855] resultiert die Drehmatrix bei Drehungen um alle drei Raumachsen aus der Drehreihenfolge Z-Y-X. Diese Notation übernimmt das VR-PLM Integrationskonzept. Die Koordinaten eines beliebigen Punktes K nach den Rotationen um die Achsen Z, Y und X errechnet sich aus Formel 4-2 (Bild 4-21).



$$\begin{bmatrix} K'_{XR} \\ K'_{YR} \\ K'_{ZR} \end{bmatrix} = \begin{bmatrix} R_{XX} & R_{XY} & R_{XZ} \\ R_{YX} & R_{YY} & R_{YZ} \\ R_{ZX} & R_{ZY} & R_{ZZ} \end{bmatrix} * \begin{bmatrix} K_X \\ K_Y \\ K_Z \end{bmatrix} \quad \text{Formel 4-2}$$

Bild 4-21: Drehung um die Z-Achse II

Das PES-Produktmodell bildet die Rotation informationstechnisch durch ein Feld der Größe neun ab, welches reelle Zahlenwerte annimmt.

Bei der Skalierung wird jeder Punkt (K) der Komponente mit einem Skalierungsfaktor S multipliziert. Das PES-Produktmodell bildet die Skalierung mittels der Skalare für X (S_X), Y (S_Y) und Z (S_Z) ab. Sie ist informationstechnisch durch ein Feld der Größe drei beschrieben, das reelle Zahlenwerte annimmt. Die Skalierung eines beliebigen Punktes (K) errechnet sich aus Formel 4-3:

$$\begin{bmatrix} K'_{XS} \\ K'_{YS} \\ K'_{ZS} \end{bmatrix} = \begin{bmatrix} S_X & 0 & 0 \\ 0 & S_Y & 0 \\ 0 & 0 & S_Z \end{bmatrix} * \begin{bmatrix} K_X \\ K_Y \\ K_Z \end{bmatrix} \quad \text{Formel 4-3}$$

Transformationsangaben zu Translation und Rotation können absolut oder relativ erfolgen. Während sich absolute Angaben stets auf die Wurzelkomponente beziehen, verweisen relative Angaben auf den in der PES-Produktstruktur übergeordneten Knoten (Bild 4-22). Die Aufsummierung von relativen Transformationsangaben bis hin zur Wurzel, resultiert in einer absoluten Transformationsangabe.

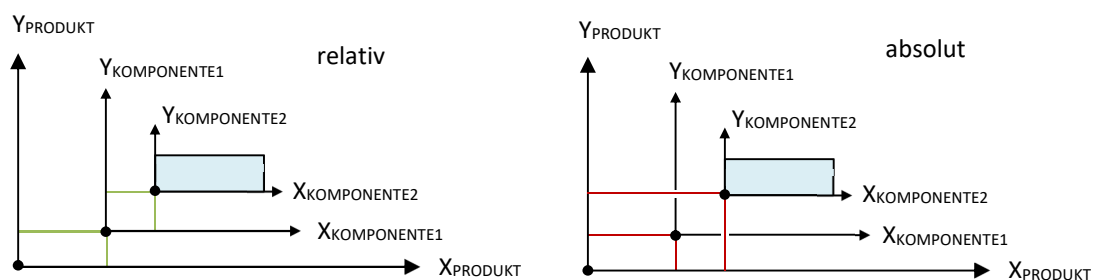


Bild 4-22: Arten der Transformationsangabe

In nativen CAx-Komponenten erfolgt die Angabe von Transformationen meist relativ. Diese Verfahrensweise ist von Vorteil, wenn im Rahmen parametrischer Abhängigkeiten Änderungen entlang der Produktstruktur vererbt werden sollen. D.h. jede Transformation

wirkt sich auf alle in der Produktstruktur nachfolgenden Knoten aus. Die Auswertung der Transformationsangaben erfolgt im VR-PLM Integrationskonzept relativ.

Wie einfühend dargestellt, gehen Transformationsangaben mit den strukturellen Zusammenhängen von Komponenten im Produktmodell einher. Da eine Komponente Bestandteil mehrerer Produktmodelle sein kann und dementsprechend unterschiedliche Transformationen aufweisen kann, sind Transformationsangaben im PES-Produktmodell unabhängig vom Objekt der *Komponente* gespeichert. Die Transformation ist im PES-Produktmodell daher in der Klasse *StrukturKontext* verwaltet, dessen Aufgabe die Verwaltung der Relationen zur übergeordneten Komponente ist (Bild 4-23). Mit anderen Worten, die Klasse *StrukturKontext* ist ein Container der alle Informationen aggregiert, die sich aus der Verwendung einer Komponente in einem spezifischen Produktmodell ergeben. Die Aggregation dient hierbei ausschließlich der Objektorientierung, um die Informationen des PES-Produktmodells zu strukturieren. Die dem *StrukturKontext* assoziierte Klasse *Transformation* enthält die Angaben zur Position, Orientierung und Skalierung in Form der Klassen *D3_Feld* und *D9_Feld*.

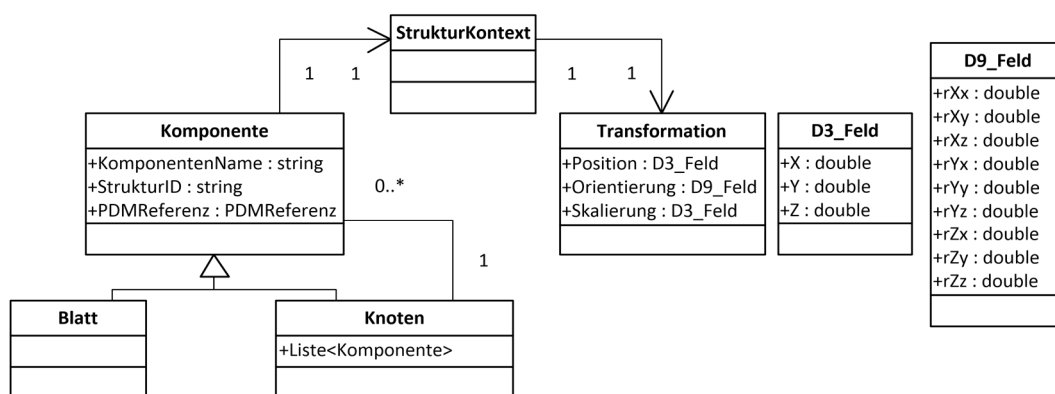


Bild 4-23: Verwaltung kontextabhängiger Transformationen von Komponenten

Das VR-PLM Integrationskonzept setzt voraus, dass die zu integrierenden Erzeugersysteme API-Schnittstellen zum Auslesen der Transformation besitzen.

4.2.4 Erscheinungsbild

Bezüglich des Erscheinungsbildes eines virtuellen Prototyps existieren unterschiedliche Benutzeranforderungen im Design Review mit VR. So werden zum Beispiel zur Bewertung der Produktgestalt und des Projektstands Visualisierungen mit funktionaler oder technologischer Bedeutung verwendet. Bei der Evaluierung von Design und Materialien liegt die Priorität hingegen auf einer fotorealistischen Visualisierung [PSS13]. Im verteilt stattfindenden Design Review ist zu gewährleisten, dass das Erscheinungsbild des virtuellen Prototyps in allen VR-Systemen möglichst gleich ist.

Nach [NFH13] erfolgt in der Computergrafik die Bestimmung des Erscheinungsbildes eines virtuellen Prototyps nicht nach der mathematisch exakten Beschreibung der Natur des Lichtes nach Max Planck und Albert Einstein. Stattdessen wird das Erscheinungsbild näherungsweise bestimmt. Dieser Prozess ist innerhalb der so genannten Visualisierungspipeline beschrieben, die aus den Teilprozessen:

- Einlesen, Filtern und Aufbereiten der Rohdaten zu Visualisierungsdaten,
- Mapping der Visualisierungsdaten in eine Modellbeschreibung (VR-Szene) sowie
- Rendering der Modellbeschreibung

besteht [Sch14], [ScM13]. Der Prozess des Renderings ist wiederum durch die Rendering-Pipeline beschrieben, die so genannte Beleuchtungsmodelle implementiert. Grundlage der meisten Beleuchtungsmodelle ist die bidirektionale Reflektanzverteilungsfunktion (BRDF). Die BRDF beschreibt die Reflexion eines Lichtstrahls in einem Punkt der Oberfläche bei gegebenem Lichteinfall- und Betrachtungswinkel. Hierbei wird die ausgehende Strahldichte ins Verhältnis zur eingehenden Bestrahlungsstärke gesetzt [DBB16]. Bzw. vereinfacht ausgedrückt: Die durch den Beobachter wahrnehmbare Helligkeit der Reflexion steht im Verhältnis zur einfallenden Strahlungsenergie der Lichtquelle. Mit der BRDF werden streuende (Diffuse), optimal (Mirror) und nicht optimal spiegelnde (Glossy) Reflexionen beschrieben (Bild 4-24).

$$BRDF_{\lambda}(\theta_i, \varphi_i, \theta_e, \varphi_e) = \frac{dL_e(\theta_e, \varphi_e)}{dE_i(\theta_i, \varphi_i)}$$

Bild 4-24: BRDF nach [CoW93]

Die aktuelle VR-Systemlandschaft ist durch spezialisierte Lösungen für unterschiedliche Einsatzszenarien geprägt. Dementsprechend differieren die Beleuchtungsmodelle. Zur Gewährleistung möglichst identischer Erscheinungsbilder des virtuellen Prototyps in

verschiedenen VR-Systemen sind die maßgeblichen Eingangsgrößen der Beleuchtungsmodelle im PES-Produktmodell gespeichert. Entsprechend der BRDF zählen hierzu Eigenschaften des Materials sowie der Lichtquelle.

Farben

Eine gemeinsame Eigenschaft von Lichtquellen und Materialien ist, dass von ihnen Licht abgegeben wird (Emission bzw. Reflexion). Je nachdem welche Wellenlänge das Licht besitzt, erscheint die Lichtquelle bzw. der virtuelle Prototyp für den Menschen in einem spezifischen Farbton. Das in der Computergrafik am häufigsten verwendete Modell zur Beschreibung von Farben ist das RGB-Modell [NFH13]. Im RGB-Modell (Bild 4-25) setzt sich jede Farbe additiv aus den drei Grundfarben Rot, Grün und Blau zusammen. Die Intensität der Grundfarben ist durch den Wertebereich $[0,1]$ definiert. D.h. null ist die Minimal- und eins die Maximalintensität einer Grundfarbe. Da in der Computergrafik häufig jeweils ein Byte zur Kodierung einer Grundfarbe verwendet wird, wird eine RGB-Farbe auch oft mittels drei Integer-Werte zwischen 0 und 255 angegeben. Farben sind im PES-Produktmodell mittels des RGB-Schemas angegeben. Hierfür existiert die Klasse *Farbe*, die die drei Eigenschaften *Rot*, *Grün* und *Blau* enthält.

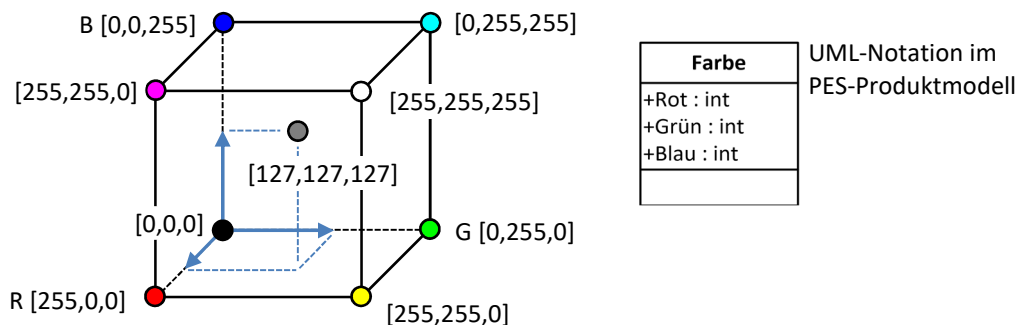


Bild 4-25: Veranschaulichung des RGB-Modells als Farbwürfel

Lichtquellen

Für die Erzeugung virtueller Szenen stehen in modernen CAD- und VR-Systemen unterschiedliche Arten von Lichtquellen zur Verfügung. Nach [Kla10] ist das Streulicht (Ambient Light) die einfachste Form einer Lichtquelle. Streulicht wird als Sammelterm für Beiträge von Reflexionen durch diverse Flächen im Raum, die eine Grundhelligkeit erzeugen verstanden. Die Beleuchtungsintensität bzw. Helligkeit ist unabhängig von einer speziellen Lichtquelle oder von der Lage der Oberfläche. Grundlegende Parameter einer Streulichtquelle

sind Lichtfarbe und -intensität. Bild 4-26 zeigt einen Würfel der mittels grünem Streulicht in unterschiedlichen Intensitäten beleuchtet ist.



Bild 4-26: Zusammenhang der Parameter Lichtfarbe und -intensität bei Streulichtquellen

Weit entfernte Lichtquellen deren Strahlen parallel verlaufen, wie beispielsweise die der Sonne, werden in der Computergrafik als gerichtetes Licht (Directional Light) verstanden. Gegenüber Streulicht lassen sich mit gerichtetem Licht parallel verlaufende Schatten erzeugen. Gerichtetes Licht ist grundlegend durch die Parameter Lichtfarbe, -intensität und die Richtung aus der es eintrifft beschrieben. Die Richtung ist durch den Polarwinkel θ und den Azimutwinkel ϕ definiert (Bild 4-27). Flächen mit einer zum gerichtetem Licht parallelen Orientierung erscheinen schwarz, da kein Lichteintrag erfolgt.

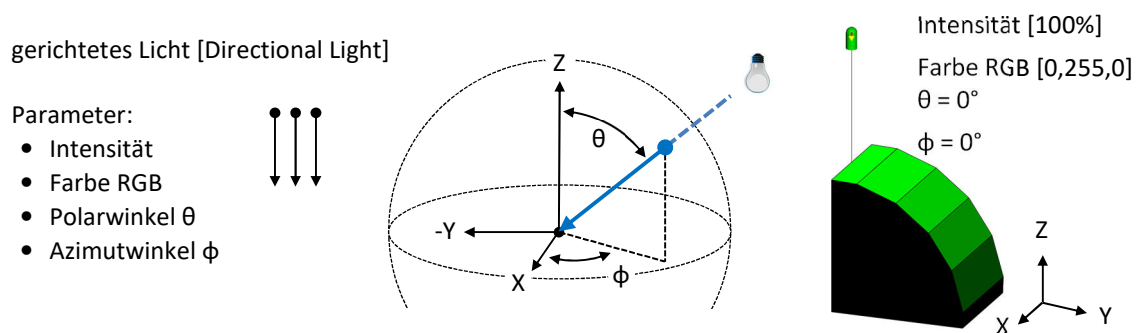


Bild 4-27: Wirkungsweise von gerichtetem Licht

Eine weitere grundlegende Lichtquellenart ist die Punktlichtquelle (Bild 4-28). Punktlichtquellen zeichnen sich dadurch aus, dass sie Licht in alle Richtungen des Raumes abstrahlen und die Intensität des Lichts mit der Entfernung von der Lichtquelle abnimmt. Eine Punktlichtquelle ist durch die Parameter Lichtfarbe, Lichtintensität, Position im Raum und Dämpfung (Attenuation) beschrieben. Während Lichtfarbe, Lichtintensität, Position im Raum über alle CAD- und VR-System einheitlich definiert sind, unterliegt die Beschreibung der Dämpfung keiner exakten Definition. Allgemein betrachtet besitzt die Dämpfung einen Wertebereich von null bis eins der sich aus der Entfernung zwischen Punktlichtquelle und zu bestrahlender Fläche ergibt. Ein Dämpfungswert von null bedeutet, dass die jeweilige Fläche keinen Lichteintrag besitzt und daher schwarz erscheint. Während physikalisch betrachtet die

Intensität einer punktförmigen Lichtquelle mit der Entfernung quadratisch abnimmt, errechnet sich die Dämpfung in der Computergrafik häufig aus einer Kombination von linearer und quadratischer Abnahme mit der Entfernung (siehe Formel 4-4). Um die Dämpfung weiter zu kontrollieren, sind die Konstanten K_{Linear} und $K_{\text{Quadratisch}}$ eingebunden. Mit ihnen lassen sich die Anteile der linearen bzw. quadratischen Abnahme der Lichtintensität gewichten.

$$\text{Dämpfung} = \frac{1}{1 + D * K_{\text{Linear}} + D^2 * K_{\text{Quadratisch}}} \quad \text{Formel 4-4}$$

D = Entfernung zw. Lichtquelle und Objekt

Im PES-Produktmodell ist die Dämpfung mittels der Konstanten K_{Linear} und $K_{\text{Quadratisch}}$ abgebildet. Sie besitzen den Standardwert null, sodass die Punktlichtquelle nicht gedämpft ist. In Abhängigkeit von der Konfigurierbarkeit der Punktlichtquelle im Erzeugersystem des virtuellen Prototyps ergeben sich folgende Zustände im PES Modell:

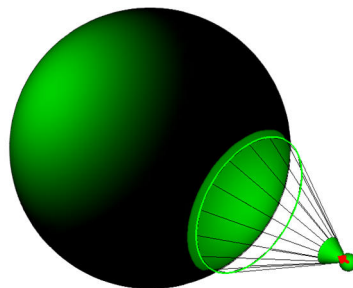
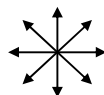
1. Das Erzeugersystem unterscheidet nur zwischen aktivierter und deaktivierter Dämpfung. Bei aktivierter Dämpfung nimmt K_{Linear} den Wert null und $K_{\text{Quadratisch}}$ den Wert eins im PES-Produktmodell an.
2. Das Erzeugersystem unterstützt zusätzlich eine allgemeine Dämpfungskonstante K . Diese entspricht im PES-Produktmodell $K_{\text{Quadratisch}}$.
3. Das Erzeugersystem unterstützt explizit eine lineare und quadratische Dämpfungskonstante. Diese entsprechen im PES-Produktmodell K_{Linear} und $K_{\text{Quadratisch}}$.

Eine ebenfalls häufig verwendete Lichtquellenart ist der Scheinwerfer (Bild 4-28). Scheinwerfer sind Punktlichtquellen, die das Licht kegelförmig in eine vorgegebene Richtung abstrahlen. Scheinwerfer definieren sich über die gleichen Parameter wie Punktlichtquellen zuzüglich einer Abstrahlrichtung, eines Öffnungswinkels und eines Leuchtdichtenexponenten. Der Leuchtdichtenexponent beschreibt, wie die Intensität des Lichts zum Rand des Kegels abnimmt.

Punktlicht [Point Light]

Parameter:

- Intensität
- Farbe RGB
- Position
- linearer Dämpfungsfaktor
- quadratischer Dämpfungsfaktor



Scheinwerfer [Spot Light]

Parameter:

- Intensität
- Farbe RGB
- Position
- linearer Dämpfungsfaktor
- quadratischer Dämpfungsfaktor
- Öffnungswinkel
- Abstrahlrichtung
- Leuchtdichtenexponent

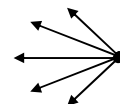


Bild 4-28: Wirkungsweise von Punkt- und Scheinwerferlichtquellen

Nach [Sch12] werden mehrere Lichtquellen zusammengefasst, indem ihre jeweiligen Beiträge zur Beleuchtung addiert werden. Dieser Prozess ist systemspezifisch innerhalb der Visualisierungspipeline der VR-Systeme implementiert.

In modernen Grafikschnittstellen, wie beispielsweise OpenGL 4.5 [SeA16], gibt es keine vordefinierten Lichtquellen mit denen sich VR-Szenen illuminieren lassen. Die Nutzer sind dazu angehalten Konzepte wie Streulicht, Punktlicht und gerichtetes Licht mittels der zur Verfügung stehenden Werkzeuge manuell zu implementieren.

Das VR-PLM Integrationskonzept berücksichtigt die bisher beschriebenen Konzepte für Lichtquellen. Bild 4-29 zeigt deren Verwaltung im PES-Produktmodell. Alle Lichtquellen sind in der Klasse *Erscheinungsbild* aggregiert, die der Objektorientierung dient und der Klasse *Komponente* zugehörig ist. Jede Lichtquellenart leitet sich vom Umgebungslicht ab (Klasse *Licht_Umgebung*), die somit die Generalisierung aller Lichtquellen darstellt. Zu deren Eigenschaften gehören die *LichtID*, *Farbe* und *Intensität*. Die Klassen *Licht_Gerichtet*, *Licht_Punkt* und *Licht_Scheinwerfer* sind Spezialisierungen der Klasse *Licht_Umgebung* und erweitern diese um die eingangs aufgeführten spezifischen Eigenschaften. Das PES-Produktmodell ist somit derart gestaltet, dass weitere spezialisierte Lichtquellen hinzugefügt werden können.

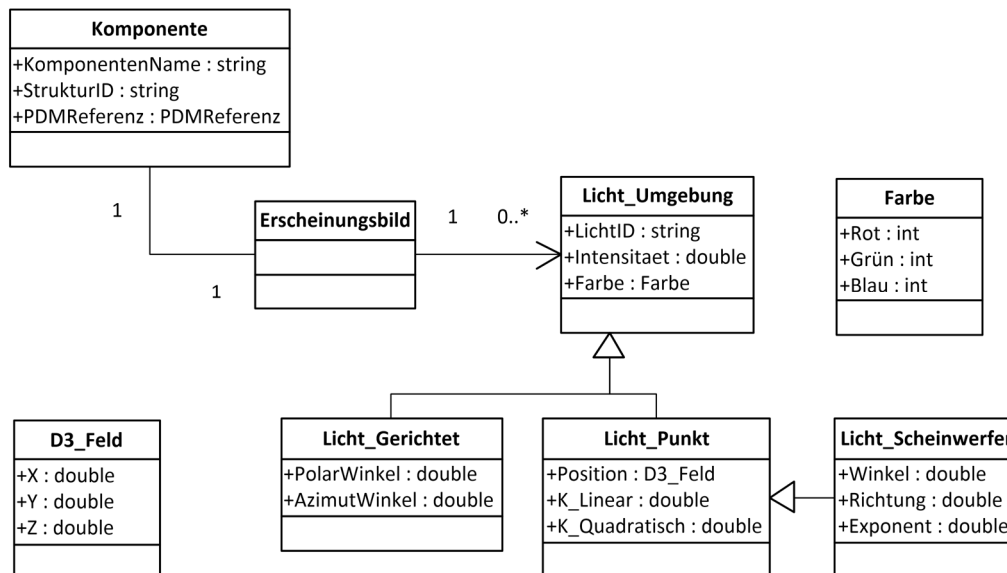


Bild 4-29: Verwaltung von Lichtquellen im PES-Produktmodell

Material

In der Natur resultiert die Farbe eines Objektes aus dessen optischen Materialeigenschaften. Sie bestimmen welche Wellenlängen des einfallenden Lichtes absorbiert, reflektiert bzw. transmittiert werden. In der Computergrafik wird zwischen verschiedenen Arten der Reflexion unterschieden, die zusammen das physikalische Reflexionsverhalten eines Objektes

approximieren. Entsprechend der BRDF gehören hierzu die diffuse, optimal spiegelnde und nicht optimal spiegelnde Reflexion. Darüber hinaus wird in der Computergrafik auch meist die Reflexion des Streulichtes berücksichtigt. Hinsichtlich der verschiedenen Arten der Reflexion sind Objekte der VR-Szene mittels der Parameter:

- Material Ambient Color (MAC),
- Material Diffuse Color (MDC) und
- Material Specular Color (MSC)

definiert. Diese beschreiben allgemein betrachtet Wellenlängen im Spektrum des sichtbaren Lichtes, die von einem Objekt der VR-Szene reflektiert werden können. Bild 4-30 zeigt den Einfluss der verschiedenen Materialfarben am Beispiel einer Kugel, die mit weißem gerichtetem Licht sowie Streulicht beleuchtet ist. Im Fall der Kombination aller Farbwerte überlagern sich die grüne MAC, MDC und die rote MSC im Punkt der Reflexion auf der Oberfläche der Kugel. Entsprechend der additiven Farbmischung des RGB-Modells resultiert daraus ein gelber Farbton.

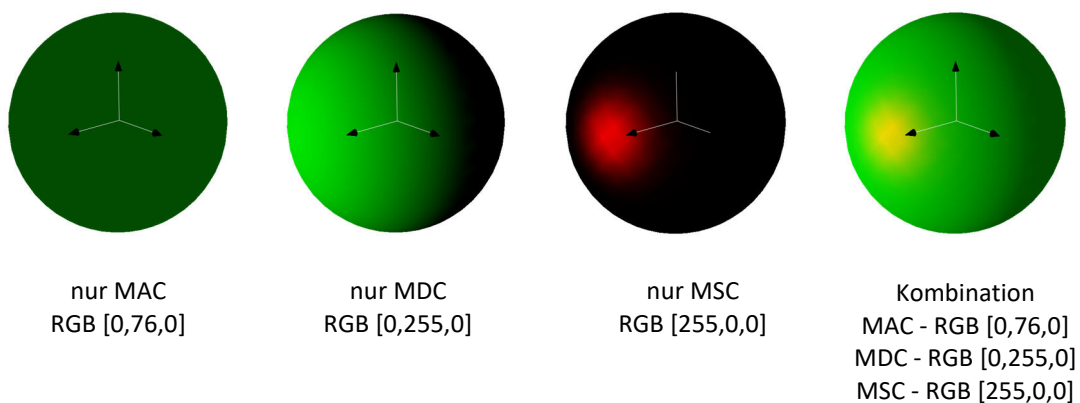


Bild 4-30: Wirkungsweise von Materialfarben in einer virtuellen VR-Szene

Material Ambient Color beschreibt die Wechselwirkung mit dem von Streulichtquellen abgegeben Licht. Da sich Streulicht überall im Raum gleichmäßig verteilt, ist diese Art der Reflexion nur abhängig von der Materialkonstante K_{Ambient} , welche die Intensität steuert. Material Diffuse Color und Material Specular Color beziehen sich auf die Wechselwirkung mit gerichtetem Licht, Punktlichtquellen und Scheinwerfern. Während Material Diffuse Color Parameter für die diffuse Reflexion ist, bezieht sich Material Specular Color auf die optimal spiegelnde Reflexion. Die Intensität der Reflexion wird über die Parameter K_{Diffuse} und K_{Specular} gesteuert. Die Parameter K_{Ambient} , K_{Diffuse} und K_{Specular} besitzen den Wertebereich $[0,1]$ im Raum der reellen Zahlen.

Neben den bisher dargestellten Materialfarben, die auf der Wechselwirkung des einfallenden Lichtes basieren, gibt es in der Computergrafik die sogenannte Material Emissive Color. Hiermit werden Körper simuliert, die selbst Licht emittieren. Die Intensität ist mittels des Parameters K_{Emissive} beschrieben.

Die Beschreibung der nicht optimal spiegelnden Reflexion erfolgt mittels des Parameters Shininess. Dieser Parameter beschreibt die mikroskopische Rauheit der Oberfläche. Je rauer eine Oberfläche beschaffen ist, desto mehr weicht die Reflexion von der ideal spiegelnden ab (Bild 4-31). In den anwendungsspezifischen Implementierungen der Beleuchtungsmodelle erfolgt die Definition des Parameters Shininess meist normalisiert. Die untere Grenze des Wertebereichs steht für eine sehr raue Oberfläche, die obere für eine sehr glatte.

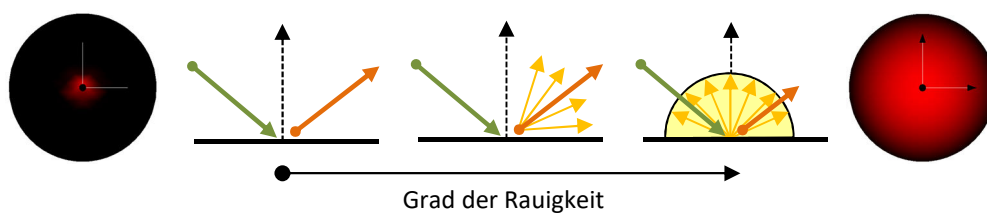


Bild 4-31: Simulation der strukturellen Oberflächenbeschaffenheit

Ursachen der Normalisierung liegen im Konzept der Benutzersteuerung sowie zur Absicherung von numerischen Instabilitäten, infolge zu kleiner bzw. zu großer Werte. Da die Basis der Normalisierung anwendungsspezifisch ist, muss im VR-PLM Integrationskonzept ein anwendungsspezifisches Mapping des Shininess Parameters beim Erzeugen der VR-Szene erfolgen.

Im Design Review mit VR besteht oft die Forderung nach transparenten Darstellungen, wenn zum Beispiel Komponenten bewertet werden sollen, die durch andere Komponenten verdeckt sind. Transparente Darstellungen sind analog der Farbe in der Computergrafik approximiert und somit unabhängig von optischen Materialeigenschaften. Transparente Erscheinungen resultieren aus der Überlagerung zweier Farben im Verhältnis ihrer Opazität, dem so genannten Alpha-Blending [NFH13]. Die Opazität ist der Kehrwert der Transparenz und beschreibt das Maß der Lichtundurchlässigkeit. Bild 4-32 zeigt eine häufig verwendete Art des Alpha-Blendings bezüglich eines undurchsichtigen Hintergrunds und einer teiltransparenten Fläche. Zur Abbildung der Transparenz ist die Klasse *Material* im PES-Produktmodell um die Eigenschaft Transparenz erweitert, die den Wertebereich $[0,1]$ im Raum der reellen Zahlen besitzt. Darüber hinaus besitzt die Klasse *Material* die Eigenschaft Sichtbarkeit, mittels derer Komponenten im VR-PLM Integrationskonzept vom Rendering ausgeschlossen werden können. Die Sichtbarkeit ist vom Wertetyp Boolean.

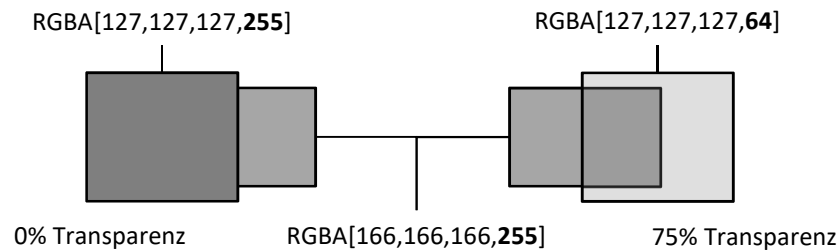


Bild 4-32: Alpha-Blending

In modernen CAD-Systemen können Materialdefinitionen grundlegend auf Struktur- und Komponentenebene erfolgen. Darüber hinaus können innerhalb einer Komponente auch die enthaltenen Volumenkörper sowie deren Flächen Materialdefinitionen annehmen. Auf Strukturebene erfolgt die Auswertung der Materialdefinition hierarchisch entlang der Struktur des Szenegraphen nach der Top-Down Methode. Das heißt, Materialien, die auf Strukturknoten angewendet sind, überschreiben alle darunterliegenden Knoten im Szenegraph. Innerhalb der Komponente erfolgt die Auswertung nach der Bottom-Up Methode. Das bedeutet, dass Materialdefinitionen von übergeordneten Elementen nicht überschrieben werden (Bild 4-33). Das VR-PLM Integrationskonzept muss gewährleisten, dass die im PES-Produktmodell verwalteten Materialdefinitionen entsprechend der Funktionsweise der modernen CAD-Systeme im VR-System korrekt rekonstruiert werden.

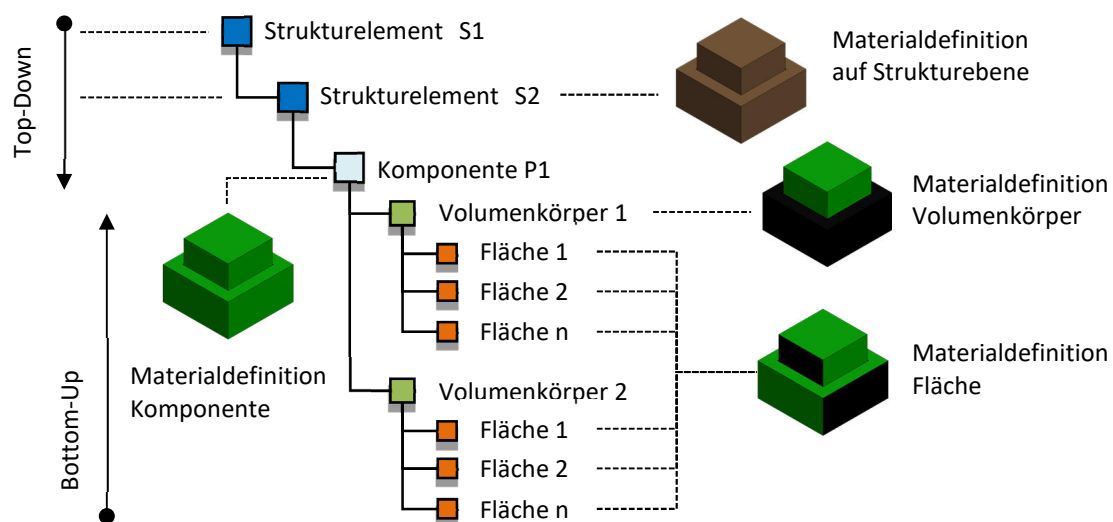


Bild 4-33: Materialdefinitionen im CAD- und VR-System

Bild 4-34 zeigt die Verwaltung der Materialdefinition im PES-Produktmodell. Das PES-Produktmodell berücksichtigt, dass sich Materialfarben mit der Verwendung in einem spezifischen strukturellen Zusammenhang unterscheiden können. Das bedeutet, bei der Visualisierung einer Komponente im Kontext einer Struktur wird beispielsweise Materialdefinition A angewendet. Wird die Komponente alleinstehend visualisiert, weist sie

die Materialdefinition B auf. Aus diesem Grund ist jeweils eine Materialdefinition in den Klassen *Erscheinungsbild* und *Erscheinungsbild_StrukturKontext* gekapselt. Je nach Verwendungszweck der Komponente, muss das VR-PLM Integrationskonzept die entsprechende Materialdefinition zur Visualisierung im VR-System nutzen.

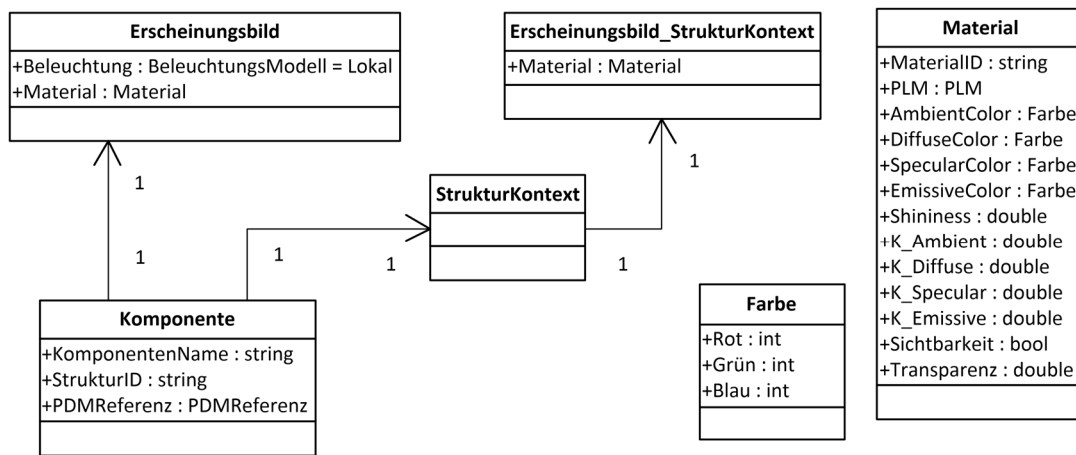


Bild 4-34: Verwaltung der Materialdefinition im PES-Produktmodell

Die Materialparameter MAC, MDC und MSC sind hervorragend dazu geeignet, einfarbige Objekte innerhalb einer VR-Szene darzustellen. Einfarbige Objekte bedeuten, dass die gesamte Oberfläche des Objekts eine Wellenlänge aus dem Spektrum des sichtbaren Lichtes reflektiert. Sollen einzelne Abschnitte der Oberfläche unterschiedliche Wellenlängen reflektieren wird in der Computergrafik auf das so genannte Texture-Mapping zurückgegriffen. Beim Texture-Mapping werden den Oberflächen eines Objektes mittels pixelbasierter Bilder multiple Werte für die Materialparameter vom Typ MAC, MDC und MSC zugeordnet (Bild 4-35). Diese erweiterte Art der Materialdefinition ist sehr performant, da sie in den meinsten Grafiksystemen hardware-beschleunigt ist [NFH12].

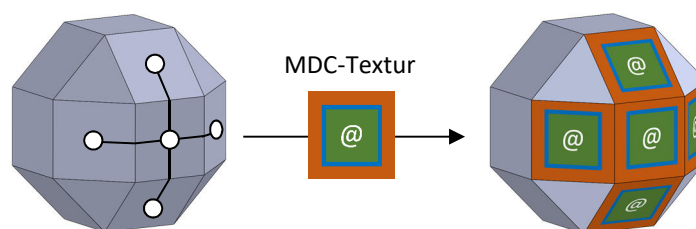


Bild 4-35: Partielles Texture-Mapping

Analog zu den Farbparametern MAC, MDC und MSC können auch die Parameter Transparenz und Shininess auf einer Oberfläche des Objektes mittels Texturen partiell definiert werden. Darüber hinaus haben sich in der Computergrafik Texturen etabliert, die den Detailgrad des 3D-Objektes visuell erhöhen, ohne dabei die Geometrie zu beeinflussen. Dies hat den Vorteil, dass die Performanz der Visualisierung kaum beeinträchtigt wird. Man spricht hierbei von so

genannten Bump-Maps und Normal-Maps. Diese Texturen simulieren ein Höhenprofil, das mit Hilfe von Schattierungen auf eine Oberfläche gezeichnet wird (Bild 4-36). Diese Art der visuellen Täuschung ist jedoch winkelabhängig. Bei einem Betrachtungswinkel von null Grad erfolgt keine Schattierung und der Effekt ist nicht wahrnehmbar. Soll dennoch die Oberfläche eines Objektes winkelunabhängig verändert werden, ohne dabei die native Geometriebeschreibung zu beeinflussen, wird auf so genannte Displacement-Maps zurückgegriffen (Bild 4-36). Diese Texturen verändern die Geometrie auf Kosten der Performanz innerhalb der Rendering-Pipeline.



Bild 4-36: Wirkungsweise von Bump- und Displacement-Maps am Beispiel einer Kugel

Texturen können in modernen CAD- und VR-Systemen beeinflusst werden. Sie lassen sich bezüglich der Abbildungsfläche skalieren, verschieben und drehen. Ist die Textur kleiner als die Abbildungsfläche, kann sie in horizontale und vertikale Richtung gekachelt werden bis die Abbildungsfläche homogen ausgefüllt ist (Bild 4-37). Mittels dieser Parameter lassen sich anhand derselben Textur abweichende Erscheinungsbilder Erzeugen.

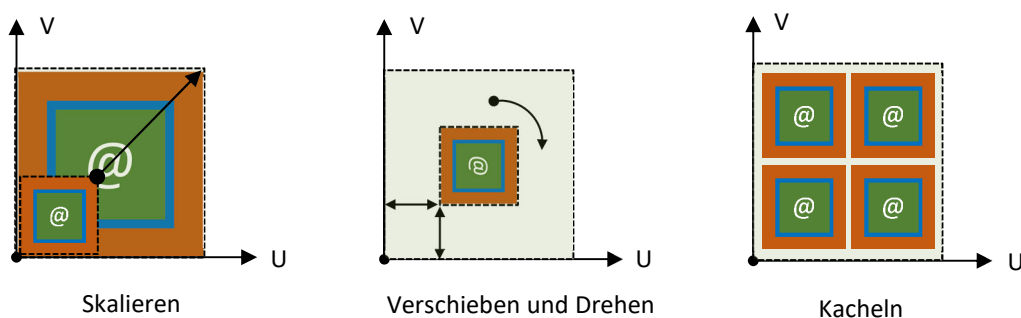


Bild 4-37: Texturoptionen

Da die Renderingpipeline der CAx- und VR-Systeme systemspezifisch ist, ist die Anzahl möglicher Texturen und Texturparameter, um ein Material zu beschreiben, theoretisch unbeschränkt. Die genannten Texturen beschreiben den Stand der Technik hinsichtlich des Texture-Mappings von grundlegenden Materialparametern sowie von performanten Verbesserungen der Visualisierung, die unabhängig von der nativen Geometriebeschreibung sind. Im PES-Produktmodell bilden Texturen eine separate Klasse, die in Assoziation mit der

Klasse *Material* steht (Bild 4-38). Texturen sind im PES-Produktmodell von den strukturellen Zusammenhängen von Komponenten im Produktmodell unabhängig.

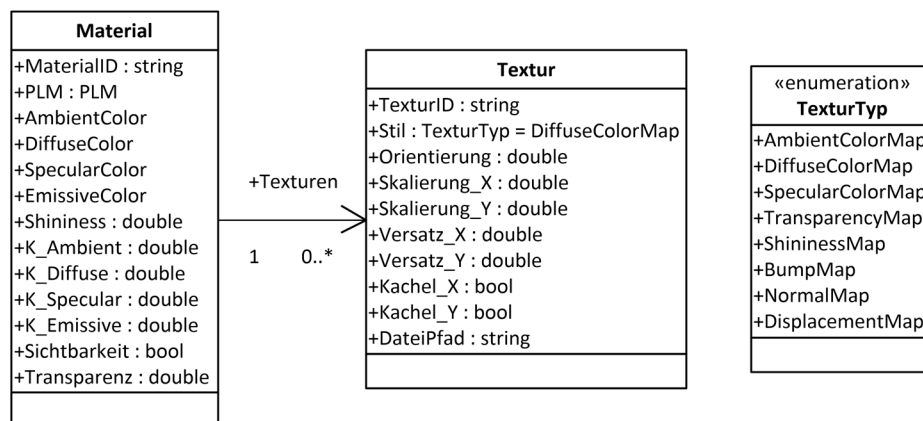


Bild 4-38: Verwaltung von Texturen im PES-Produktmodell

Wie in Kapitel 3 beschrieben trennt das VR-PLM Integrationskonzept Informationen zur Gestalt vom Erscheinungsbild, um heterogene Produktbeschreibungen bestehend aus mehreren neutralen Datenformaten zu gewährleisten. Der mit den neutralen Datenformaten einhergehende differierende Abbildungsraum resultiert in unterschiedlich strukturierten Daten. So speichert VRML zum Beispiel keine Flächen. Hinsichtlich der Gestalt sind Volumenkörper bei den etablierten neutralen Datenformaten der kleinste gemeinsame Nenner. Dementsprechend speichert das PES-Produktmodell Materialdefinition für Strukturen, Komponenten und Volumenkörper. Die Materialdefinition für Volumenkörper ist von den strukturellen Zusammenhängen von Komponenten im Produktmodell unabhängig. Das Speichern von Materialdefinitionen auf Volumenkörperebene ist durch die Ausweitung der in Kapitel 4.2.1 beschriebenen PES-Produktstruktur gegeben (Bild 4-39). Hierzu Assoziiert die Klasse *Blatt* die Klasse *Volumenkörper*, die die Eigenschaften *VolumenID* und *Material* besitzt.

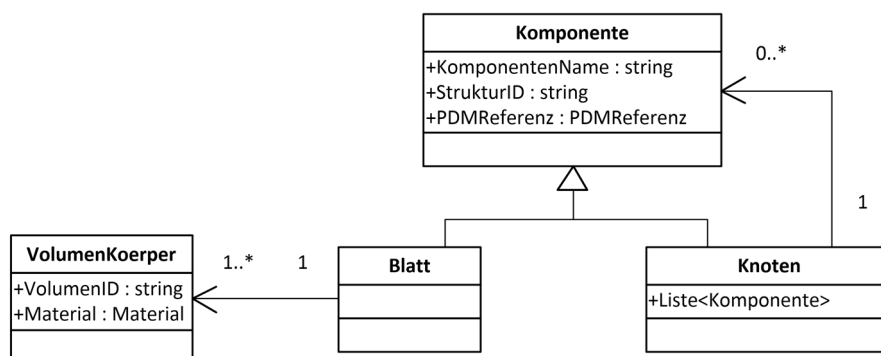


Bild 4-39: Erweiterung der Struktur des PES-Produktmodells um Volumenkörper

4.2.5 Simulation und Animation

Im Design Review werden zunehmend Animationen verwendet, um komplexe funktionale Zusammenhänge am Produkt darzustellen. Dies dient zum einen der Bewertung (Kenntnisvermittlung), zum anderen der Maßnahmengenerierung (Erkenntnisgewinnung) im Design Review [BGW11].

Die VDI-Richtlinie 3633 beschreibt eine Animation als: *„Erzeugung und Präsentation von Bildfolgen, in denen Änderungen [...] einen visuellen Effekt bedingen.“* Unter einem visuellen Effekt versteht die VDI-Richtlinie 3633 eine: *„Über die Zeit variierende Position, die Änderung von Form, Farbe, Transparenz, Struktur und Musterung eines Objektes, die Änderung der Beleuchtung sowie der Position, Orientierung und Brennweite der Kamera.“* [VDI3633].

In der Produktentwicklung haben sich zwei grundlegende Arten der Animation etabliert. Zum einen existieren Animationen im CAx-System, die in Echtzeit anhand der Ergebnisse einer Simulation¹ generiert werden. Zum anderen existieren Animationen im VR-System, die auf Basis von manuell vordefinierten Zustandsgrößen, den so genannten Keyframes, erfolgen.

Die in der VDI-Richtlinie 3633 genannten visuellen Effekte entsprechen den Zustandsgrößen von Objekten einer VR-Szene. Im PES-Produktmodell sind die genannten visuellen Effekte durch Objekte vom Typ Transformation und Material beschrieben. Da im VR-PLM Integrationskonzept die VR-Szene anhand der im PES-Produktmodell gespeicherten Informationen automatisch generiert wird, können als PES-Produktmodell beschriebene Simulationsergebnisse bzw. Animationsschritte zur Aktualisierung der VR-Szene verwendet werden. Erfolgen mehrere Aktualisierungen pro Sekunde, entsteht die geforderte Präsentation von Bildfolgen. Die genannten Animationsarten können somit auf einfache Weise im VR-PLM Integrationskonzept Anwendung finden. Es sind lediglich Momentaufnahmen eines Objekts aus CAx- oder VR-System erforderlich, die als temporäre PES-Produktmodelle verwaltet werden. Anhand der temporären PES-Produktmodelle können dann alle im VR-PLM Integrationskonzept integrierten VR-Systeme, die das animierte Objekt in seiner Grundform geladen haben, aktualisiert werden. Allgemein betrachtet, entspricht dieses Konzept der Aufnahme und Wiedergabe von Informationen.

Das beschriebene Konzept eignet sich auch hervorragend dafür, um einmalige Veränderungen an einem virtuellen Prototyp mit anderen VR-Szenen zu teilen. Wird nur eine einzige

¹ *„Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind; insbesondere werden die Prozesse über die Zeit entwickelt.“* [VDI3633]

Momentaufnahme erzeugt und verteilt, ist innerhalb des VR-PLM Integrationskonzepts von einem Szenen-Update die Rede (Bild 4-40).

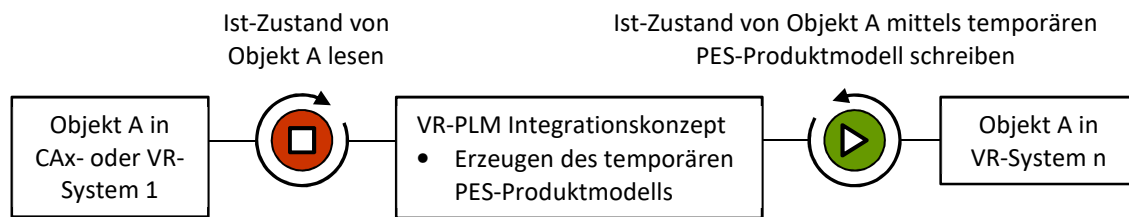


Bild 4-40: Aktualisierung der VR-Szene von integrierten VR-Systemen

Zur Gewährleistung konstanter Bildraten sollten die temporären PES-Produktmodelle im Hauptspeicher des Computers der das VR-System ausführt gepuffert werden. Für das Verbreiten der temporären PES-Animationsmodelle in einer verteilten Umgebung eignen sich die Netzwerkprotokolle TCP und/oder UDP. In gesicherten Netzwerken und wenn der Anspruch nach Performance über dem der Sicherheit steht, ist das Netzwerkprotokoll UDP zu favorisieren [Ant15].

Des Weiteren ist der Umfang der temporären PES-Produktmodelle durch einen Filter einzugrenzen. So ist es zum Beispiel nicht nötig das Material eines Objektes zu aktualisieren bzw. zu speichern, wenn nur dessen Lage im Raum animiert werden soll. Es müssen auch keine temporären PES-Produktmodelle für Komponenten erzeugt werden, die keine Animation erfahren. Durch eine angepasste Konfiguration sinkt der Aufwand zur Bereitstellung der PES-Produktmodelle sowie zur Aktualisierung einer VR-Szene erheblich. Die Folge sind höhere Bildraten.

Eine weitere Anforderung im Design Review ist die Wiederverwendung von Animationen. Hierfür werden bei Bedarf die temporären PES-Animationsmodelle in Abhängigkeit von der Zeit zusammengefasst und im PDM-System gespeichert. Das VR-PLM Integrationskonzept versteht Animationen dabei als entkoppelt vom Produktmodell. D.h. es müssen eigenständige Entitäten im PDM erzeugt und mit den entsprechenden Dokumenten verknüpft werden. Dieser Ansatz gewährleistet, dass sich jeder im PDM verwaltete virtuelle Prototyp mit einer beliebigen Animation kombinieren lässt. Die produktmodellübergreifende Wiederverwendung von Animationen im Design Review mit VR ist somit gegeben (Bild 4-41).

Wiedergabeoptionen von Animationen wie Start, Stopp und Pause können entweder über interne Benutzungsschnittstellen im VR-PLM Integrationskonzept realisiert werden, oder mittels Schnittstellen, die das VR-PLM Integrationskonzept anbietet. Letzteres ermöglicht die Integration von individuellen Benutzungsschnittstellen unabhängig vom VR-PLM Integrationskonzept. Dieser Ansatz entspricht dem Konzept des Model-View-Controllers

(MVC). MVC ist ein Architekturmuster der Informatik und beschreibt die Trennung von Datenmodell, Programmsteuerung und grafischer Benutzungsschnittstelle.

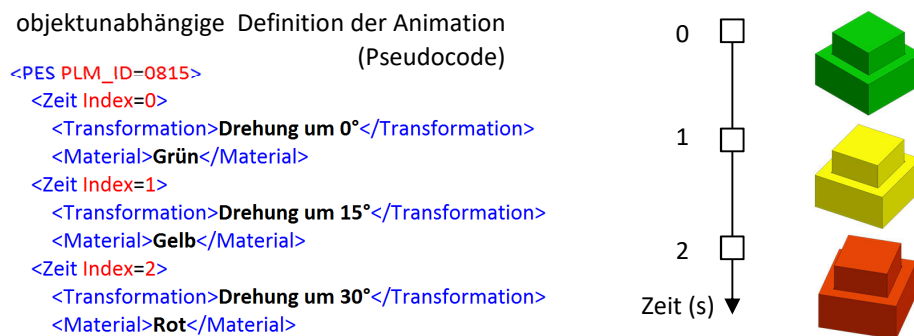


Bild 4-41: Anwendung von im PDM verwalteten Animationen auf einen beliebigen virtuellen Prototyp

Bild 4-42 zeigt die Verwaltung von Animationen im PES-Produktmodell. Die Klasse *Komponente* assoziiert die Klasse *Animation*. Diese ist durch ein Erkennungsmerkmal (*AnimationsID*), eine *PDMReferenz* und einen *Animationstyp* beschrieben. Die Konstante *Animationstyp* dient der Steuerung und beschreibt, ob eine Animation nur einmal (linear) oder fortlaufend (zyklisch) wiedergegeben werden soll. Die Informationen zur Veränderung der Transformation bzw. des Materials sind mit der Klasse *KeyFrame* assoziiert. Diese besitzt die zusätzliche Eigenschaft *Zeitindex*, um die Veränderung zeitlich einzuordnen. Die Verknüpfung der Klassen *Animation* und *KeyFrame* erfolgt mittels der Assoziation *KeyFrames*.

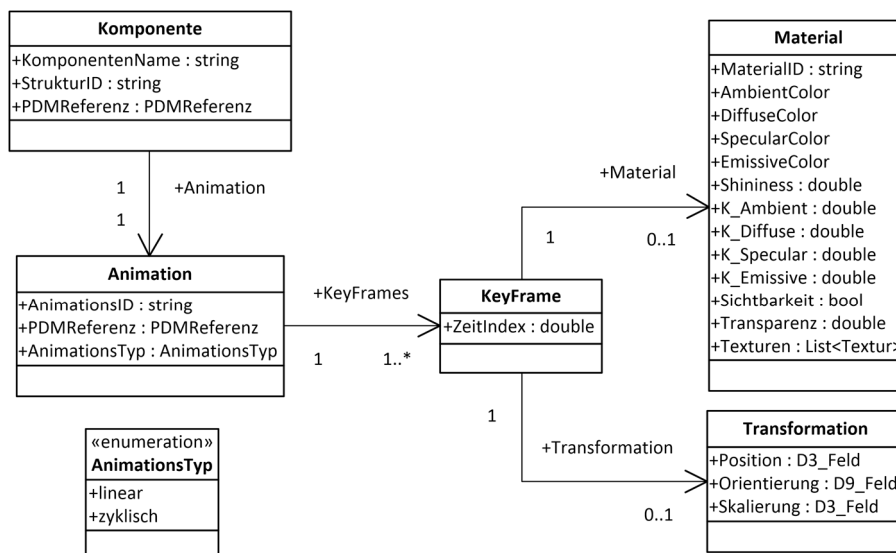


Bild 4-42: Verwaltung von Animationen im PES-Produktmodell

4.2.6 Präsentationsumgebung

Die Evaluierung von Design und Materialien im Design Review erfolgt häufig im Kontext der Umgebung, in der das Produkt später zum Einsatz kommt. Hierdurch lassen sich Wechselwirkungen zwischen der Umgebung und der Gestalt bzw. dem Erscheinungsbild des Produktes visuell wahrnehmen. Im VR-basierten Design Review resultiert die Einbeziehung der Umgebung in einer Erhöhung der Immersion und Imagination.

Nach [DBG14] wird die Umgebung einer VR-Szene durch ein statisches Bild bzw. eine Textur repräsentiert. Diese Umgebungstextur ist auf die Innenfläche bzw. Innenflächen eines dreidimensionalen Körpers projiziert, in dem sowohl der Anwender als auch der zu bewertende virtuelle Prototyp positioniert sind. Verändert der Anwender seine Position und/oder Blickrichtung, entsteht eine differenzierte Sicht auf das Produkt (Bild 4-43). Anstelle einer statischen Textur kommen in modernen VR-Anwendungen auch Filmaufnahmen der realen Welt als Präsentationsumgebung zum Einsatz. Hierdurch werden dynamische Umgebungen realisiert, die beispielsweise den Verlauf der Sonne und die Wolkenbewegung berücksichtigen.

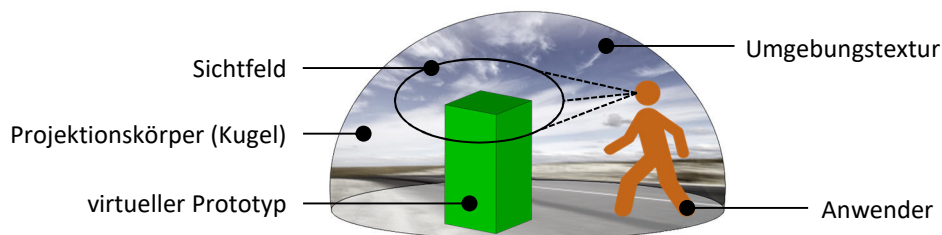


Bild 4-43: Schematische Darstellung der Anwendung einer Umgebungstextur in einer VR-Szene

Bild 4-44 zeigt die Integration der Präsentationsumgebung in das PES-Produktmodell. Die Klasse *PraesentationsUmgebung* ist von der Klasse *Komponente* assoziiert und besitzt eine Kardinalitätsrestriktion vom Typ 1:1. Neben der identifizierenden Eigenschaft *PraesentationsUmgebungsID* ist die Eigenschaft *Transformation* verwaltet, mittels derer die Position und Orientierung des virtuellen Prototyps beschrieben ist. Darüber hinaus versteht das VR-PLM Integrationskonzept Präsentationsumgebungen als im Design Review wiederzuverwendende Objekte, die konsistent in der unternehmensweiten Daten- und Prozessschicht zu verwalten sind. Die Identifizierung der Präsentationsumgebung im PDM erfolgt über die Eigenschaft *PDMReferenz* der Klasse *PraesentationsUmgebung*.

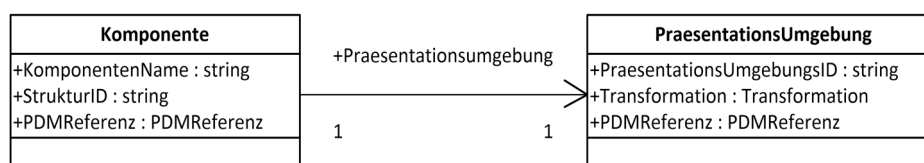


Bild 4-44: Verwaltung der Präsentationsumgebung im PES-Produktmodell

4.3 SES-Modell

Das bisher beschriebene PES-Produktmodell entspricht der in DIN EN 61160 geforderten Vorbereitung eines Unterlagensatzes speziell für virtuelle Prototypen, die mit VR zu visualisieren sind. Hierauf basierend erfolgt die Bewertung des virtuellen Prototyps während der Durchführung des Design Reviews mit VR. Ergebnis der Bewertung ist eine Liste von Empfehlungen und Maßnahmen, die in einem Sitzungsbericht dokumentiert werden und nach dem Design Review umzusetzen sind. Der Sitzungsbericht nach DIN EN 61160, ist im VR-PLM Integrationskonzept durch das SES-Modell abgebildet.

Allgemeine, im Sitzungsbericht zu dokumentierende, Informationen sind der Bewertungsgegenstand, die Motivation, der Status des Design Reviews, der Zeitpunkt der Durchführung sowie eine Beschreibung der bewertenden Tätigkeit. Der Bewertungsgegenstand repräsentiert einen virtuellen Prototyp aus der unternehmensweiten Daten- und Prozessschicht des PLM. Informationen zur Motivation unterstützen die Bestimmung des oder der Zeitpunkte(s) für die Durchführung von Entwicklungsbewertungen. Der Status des Design Reviews dokumentiert den Fortschritt. Dieser kann im VR-PLM Integrationskonzept auch dazu verwendet werden, um Teilprozesse des Design Reviews mit anderen Prozessen des PLM zu verknüpfen. Weiterer Inhalt des SES-Modells ist der zugehörige Personenkreis inklusive der von den Personen eingenommenen Rollen (z. B. Sitzungsleiter) im Design Review.

Die genannten Informationen sind im SES-Modell durch die Klasse Sitzungsbericht verwaltet. Diese assoziiert die Klassen *Komponente* und *Person* sowie die Enumerationen *Motivation* und *DesignReviewStatus* (Bild 4-45). Die Konstanten der Enumerationen sind durch DIN EN 61160 vorgegeben. Die Klasse *Komponente* dient der Identifizierung des bewerteten virtuellen Prototyps. Existiert für den virtuellen Prototyp ein PES-Produktmodell, kann die dem Sitzungsbericht zugrunde liegende VR-Szene ad hoc widerhergestellt werden.

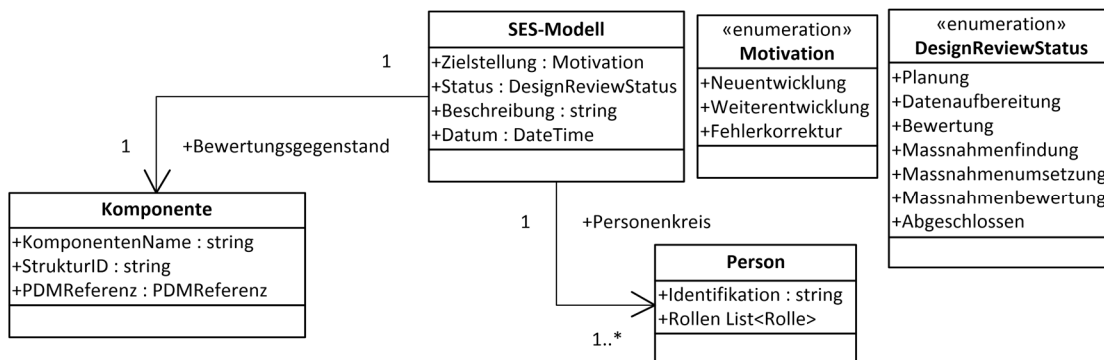


Bild 4-45: Entwicklung des SES-Modells Teil 1

Die im Design Review gewonnenen Ergebnisse sind in Form von Empfehlungen und Maßnahmen im Sitzungsbericht abgebildet. Zu den grundlegenden Eigenschaften einer Empfehlung gehören nach DIN EN 61160 der Bewertungsgegenstand, etwaige Bezugsdokumente, die die Empfehlung inhaltlich verdeutlichen, eine fachliche Beschreibung sowie die Person, welche die Empfehlung ausgesprochen hat. Darüber hinaus sollte hinsichtlich der Realisierung eine Risikoeinschätzung und Betrachtung möglicher Wechselwirkungen erfolgen, um Kosten zu schätzen und die Ausführung planen zu können. Die grundlegenden Eigenschaften einer Maßnahme entsprechen denen einer Empfehlung. Darüber hinaus enthält sie Informationen zur Person, die die Maßnahme initialisiert hat, zur Fälligkeit der Umsetzung, zur Person, die die Maßnahme umzusetzen hat sowie der Maßnahmenstatus, der den Fortschritt der Maßnahmenumsetzung dokumentiert.

Empfehlungen und Maßnahmen sind im SES-Modell durch gleichnamige Klassen abgebildet, wobei die Klasse *Empfehlung* die Klasse *Maßnahme* generalisiert (Bild 4-46). Das bedeutet, die Eigenschaften der Klasse *Maßnahme* sind um die Eigenschaften der Klasse *Empfehlung* erweitert. Mögliche Risiken und Wechselwirkungen sind im SES-Modell durch Enumerationen abgebildet, die mit der Klasse *Empfehlung* assoziiert sind. Der Status der Maßnahme ist ebenfalls durch eine Enumeration abgebildet und mit der Klasse *Maßnahme* assoziiert. Die Konstanten der genannten Enumerationen sind durch DIN EN 61160 vorgegeben und können im VR-PLM Integrationskonzept dazu verwendet werden, Prozesse des PLM an die Zustände von Empfehlungen und Maßnahmen zu binden.

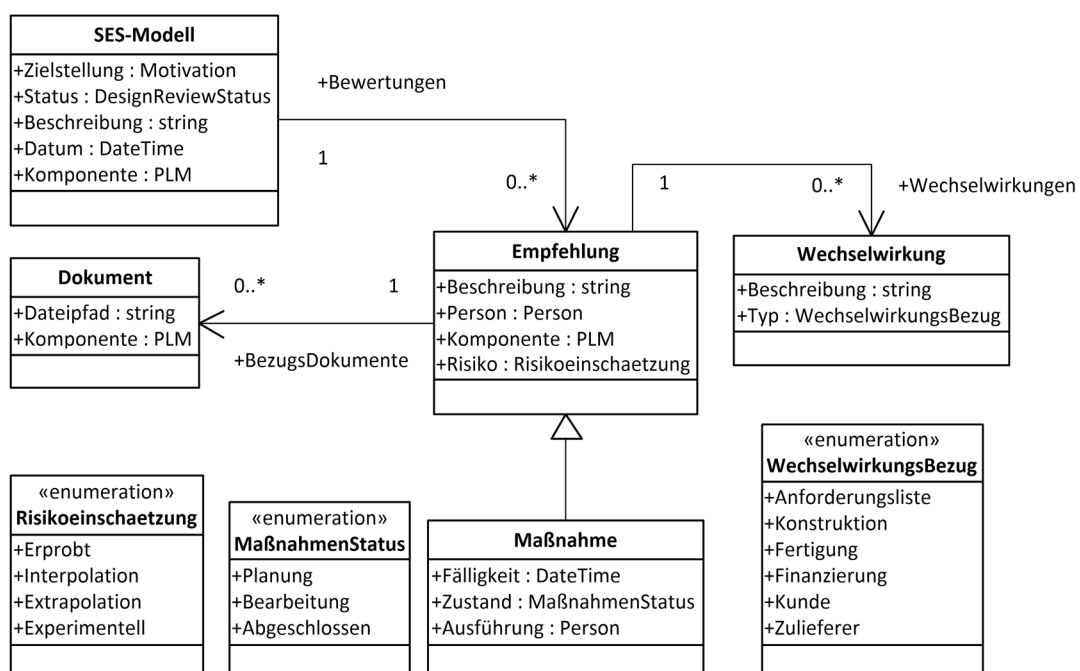


Bild 4-46: Entwicklung des SES-Modells Teil 2

Einer der zentralen Bestandteile des Bewertungsprozesses sind konstruktive Fragen und Antworten. Sie dienen zum einen dazu potentielle Probleme und deren Tragweite zu erkennen, zum anderen, um Empfehlungen zu bewerten und mögliche Maßnahmen zu finden. Darüber hinaus bilden sie Wissen ab, das für zukünftige Entwicklungsprojekte von Bedeutung sein kann.

Im SES-Modell sind Fragen durch die gleichnamige Klasse abgebildet (Bild 4-47). Die Klasse *Frage* wird zum einen von der Klasse *SES-Modell* assoziiert, um allgemeine Fragen zum Design Review abzubilden, zum anderen wird die Klasse *Frage* von der Klasse *Empfehlung* assoziiert, um spezifische Fragen zu dokumentieren. Da die Klasse *Empfehlung* die Klasse *Maßnahme* generalisiert, sind im PES-Produktmodell spezifische Fragen für beide Klassen getrennt beschrieben. Zu jedem Objekt der Klasse *Frage* müssen im SES-Modell mehrere Antworten verwaltet werden können. Antworten sind durch Objekte der Klasse *Stellungnahme* abgebildet und mit dem Objekt der jeweiligen Frage assoziiert. Die Klassen *Frage* und *Stellungnahme* enthalten jeweils eine Freitexteigenschaft, um die Frage bzw. Antwort darzustellen, einen Zeitstempel sowie einen Verweis auf die Person, die die Frage gestellt bzw. beantwortet hat.

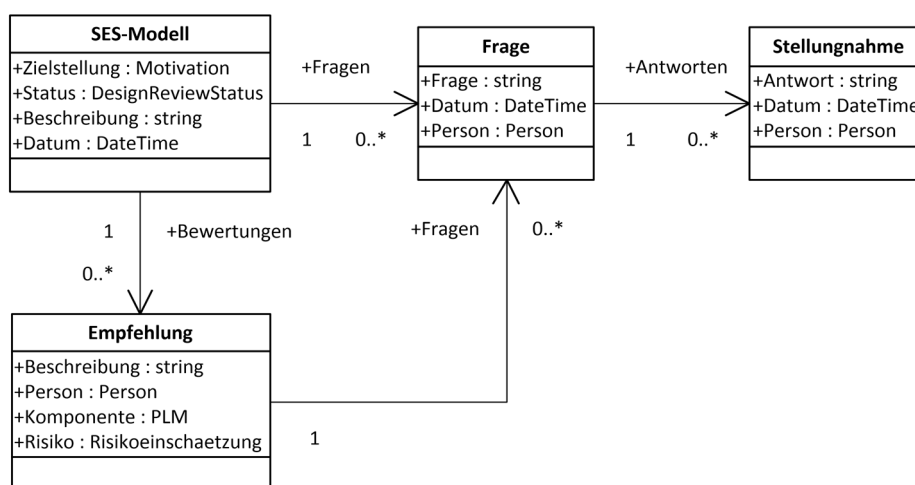


Bild 4-47: Entwicklung des SES-Modells Teil 3

Mittels des erarbeiteten SES-Modells ist es möglich den Status des Design Reviews und den von einzelnen Maßnahmen im VR-PLM Integrationskonzept informationstechnisch auszuwerten und nachzuverfolgen. Darüber hinaus kann die zum Sitzungsbericht zugehörige VR-Szene ad hoc im VR-PLM Integrationskonzept geladen werden, um mittels der bidirektionalen Verbindung zur Anwendungsschicht Empfehlungen zur Laufzeit zu validieren oder Maßnahmen anzustoßen.

Die Funktionen zum Erstellen, Bearbeiten und Verwalten des Sitzungsberichtes muss das VR-PLM Integrationskonzept als Benutzungsschnittstelle (*SESInterface*) zur Verfügung stellen. Hierdurch wird gewährleistet, dass die manuelle Aufbereitung schriftlich getätigter Informationen im Nachgang des Design Reviews entfällt. Das *SESInterface* muss Methoden zum Instanzieren, Verknüpfen und Löschen von Objekten der Klassen *SES-Modell*, *Person*, *Empfehlung*, *Maßnahme*, *Frage*, *Stellungnahme*, *Wechselwirkung* und *Dokument* beschreiben. Darüber müssen Methoden enthalten sein, mittels derer die Eigenschaften der genannten Klassen bearbeitet werden können.

4.4 PLM-Integration des PES-Produktmodells und SES-Modells

Die Integration des PES-Produktmodells und SES-Modells in die unternehmensweite Daten- und Prozessschicht des PLM steht synonym für die Integration des Design Reviews mit VR in den Produktentwicklungsprozess. Entsprechend den Konfigurationsmöglichkeiten der PLM-Lösungen existieren hierfür verschiedene systemspezifische Ansätze. PLM-spezifische Integrationen bedeuten einen enormen Implementierungsaufwand, da für jede PLM-Lösung ein spezifischer Adapter mit differierenden Schnittstellen im VR-PLM Integrationskonzept erforderlich ist.

Ein geringer Schnittstellenumfang ergibt sich aus der Idee, PES- und SES-Modelle dateibasiert in der PLM-Lösung zu verwalten (Bild 4-48). Die PLM-Schnittstelle im VR-PLM Integrationskonzept begrenzt sich somit auf das Suchen von Dokumenten sowie das Laden und Speichern von Dateien. Da diese grundlegenden Schnittstellen von den meisten PLM-Lösungen unterstützt werden, ist ein weites Einsatzgebiet des VR-PLM Integrationskonzepts gewährleistet. Für die Beschreibung der PES- und SES-Modelle eignet sich die plattformunabhängige Auszeichnungssprache XML. XML bietet den Vorteil, dass Daten hierarchisch strukturiert im Klartext beschrieben sind und alle populären Programmiersprachen Frameworks zur Verarbeitung und dateibasierten Serialisierung anbieten.

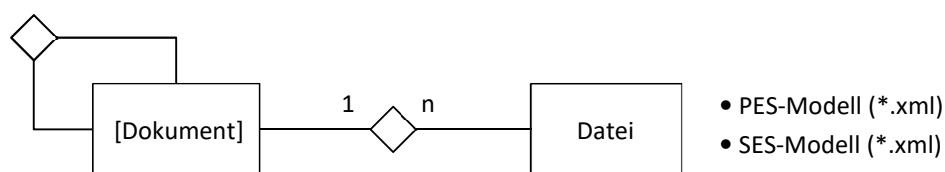


Bild 4-48: Dateibasierte Verwaltung des PES-Produktmodells im PLM (ERM-Notation)

Der beschriebene Integrationsansatz führt dazu, dass die PLM-Lösung kein Wissen über den Inhalt der Modelle hat. Dieser Zustand sichert die Performanz der Datenbank des PLM, denn es werden nur Metadaten bezüglich der Dateien erzeugt. Beispielsweise Abfragen über Informationen zu Materialien oder Transformationen gehen zwar mit einer erhöhten Latenz einher, da zunächst die entsprechende Datei geladen und das Modell geparkt werden muss, jedoch ist diese konstant. Das fortlaufende Speichern der enorm hohen Anzahl von Information der PES- und SES-Modelle in Form von Metadaten führt hingegen zu einem stetigen Anstieg der Latenz bei Datenbankabfragen.

Im PES-Produktmodell referenzierte Informationen zu Animationen, Materialien und Präsentationsumgebungen bildet das VR-PLM Integrationskonzept innerhalb der PLM-Lösung durch eigenständige Entitäten in Form von Dokumenten ab. Den Dokumenten sind neben

etwaigen Dateien wie z. B. Texturen auch die partiellen Informationen des PES-Produktmodells dateibasiert zugeordnet. Das PES-Produktmodell wird somit im PLM verteilt verwaltet (Bild 4-49). Das Verknüpfen von Animationen, Materialien und Präsentationsumgebungen mit einer Komponente erfolgt mittels der Funktionen des PLM, indem Relationen zwischen den Entitäten erstellt werden. Informationen, die den Zustand zwischen zwei Entitäten beschreiben, wie z. B. die Position und Orientierung einer Komponente in einer Präsentationsumgebung, werden der Relation zugeordnet. Wenn Relationen keine zusätzlichen Informationen transportieren können, müssen die verteilt verwalteten Dateien des PES-Produktmodell um die benötigten Relationen erweitert werden. Die Verwaltung von Animationen, Materialien und Präsentationsumgebungen erweitert die PLM-Schnittstelle des VR-PLM Integrationskonzepts um das Erstellen und Verknüpfen von Dokumenten.

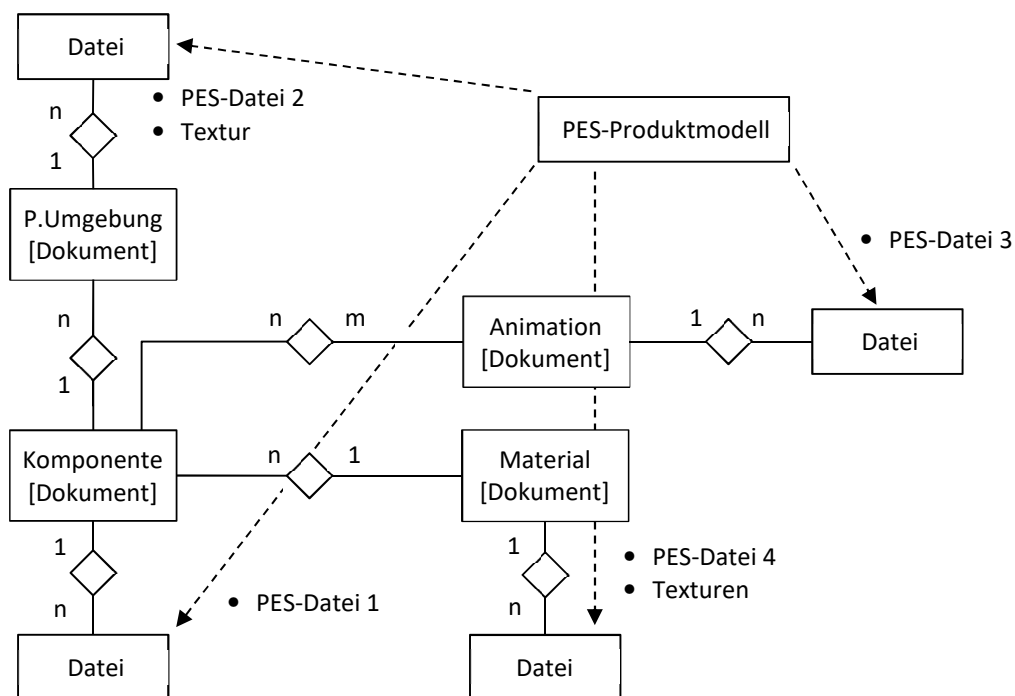


Bild 4-49: Verwaltung von Animationen, Materialien und Präsentationsumgebungen im PLM (ERM-Notation)

Das SES-Modell bzw. der Sitzungsbericht wird innerhalb der PLM-Lösung ebenfalls durch eine eigenständige Entität in Form eines Dokuments integriert (Bild 4-50). Dies gewährleistet das ein Sitzungsbericht mittels performanterer Metadatenabfragen identifiziert werden kann. Analog des PES-Produktmodells sind die Informationsgehalte des SES-Modells dateibasiert mittels der Auszeichnungssprache XML abgebildet. Die Integration der SES-Modelle kann somit über dieselben grundlegenden PLM-Schnittstellen, die auch für die Verwaltung des PES-Produktmodells verwendet werden, erfolgen. Das Speichern der Relation zwischen Sitzungsbericht und Komponente wird mittels der Funktionen des PLM umgesetzt.

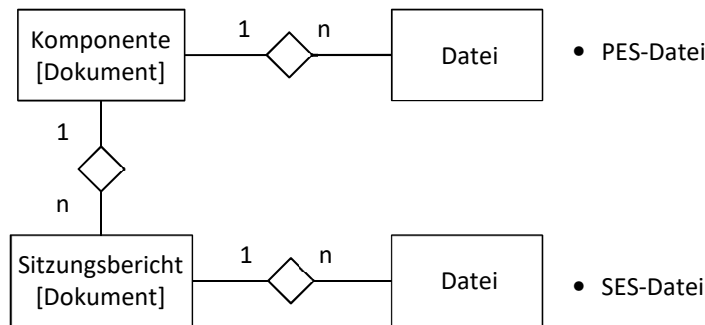


Bild 4-50: Dokumentbasierte Verwaltung der Informationen des SES-Modells im PLM (ERM-Notation)

Damit sich etwaige Wechselwirkungen zwischen PES-Produktmodell bzw. SES-Modell und PLM-Lösung prozesssicher aktivieren lassen, müssen die durch Enumerationen beschriebenen Konstanten ebenfalls innerhalb der PLM-Lösung verwaltet werden. Die Verwaltung muss derart erfolgen, dass die Konstanten mittels der Programmierschnittstellen der PLM-Lösungen ausgelesen werden können. Somit wird gewährleistet, dass die durch Enumerationen beschriebenen variablen Zustandsbeschreibungen des PES-Produktmodells und SES-Modells konsistent in den Benutzungsschnittstellen des VR-PLM Integrationskonzeptes zur Auswahl gestellt werden können.

Die Abbildung der in Kapitel 4.2.4 beschriebenen Ausweitung der PES-Produktstruktur bis auf Volumenkörperebene ist mit dem klassischen dokumentbasierten Verwaltungsansatz im PLM nicht möglich. Das VR-PLM Integrationskonzept erweitert daher komponentenabbildende Dokumente um eine dateibasierte externe Datenstruktur [PSS13] (Bild 4-51). Hierdurch ist es möglich die PES-Produktstruktur, von der Einzelteilebene ausgehend, abwärts beliebig zu erweitern. Inhalt der externen Datenstruktur sind benannte Geometrieobjekte wie zum Beispiel Volumenkörper, die sowohl im CAx-File als auch in der VR-konformen Austauschgeometrie existieren. Diese können entweder Entitäten des PLM referenzieren, um n:m Beziehungen zu realisieren, oder numerische Werte wie zum Beispiel RGBA-Farbangaben, um das Erscheinungsbild zu definieren. Die Generierung und Interpretation der zusätzlich erzeugten Datenstruktur muss durch die Implementierung des VR-PLM Integrationskonzeptes erfolgen.

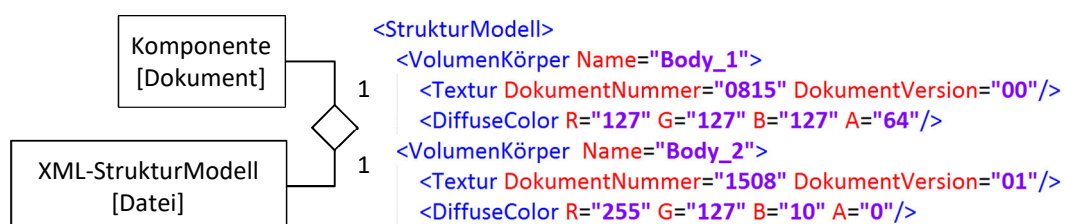


Bild 4-51: Dateibasierte Erweiterung der PES-Produktstruktur um Volumenkörper (ERM-Notation)

4.4.1 Generierung

Das Erzeugen des PES-Produktmodells ist durch die Aktivität PES-Generierung beschrieben (Bild 4-52). Bezüglich der Übersichtlichkeit sind informationstechnische Datenflüsse in der Aktivität zum Teil vernachlässigt. Voraussetzung für die PES-Generierung ist ein im CAx-System geladener virtueller Prototyp, der im PDM verwaltet wird. Dieser Zustand muss über die Programmierschnittstellen des CAx- validiert werden. Liefert die Validierung ein negatives Ergebnis, wird der zu analysierende virtuelle Prototyp mittels der Eingabe von identifizierenden PDM-Metadaten bestimmt. Die eingegebenen PDM-Metadaten bzw. Suchparameter werden dann an die Aktion PDM-Datei-Download übergeben, worauf Kopien des im PDM verwalteten virtuellen Prototyps lokal gespeichert werden. Im Anschluss erfolgt das Laden des virtuellen Prototyps im CAx-System durch die Aktion Laden-CAx. Das Speichern und Laden erfolgt jeweils über die Programmierschnittstellen des angesprochenen Systems.

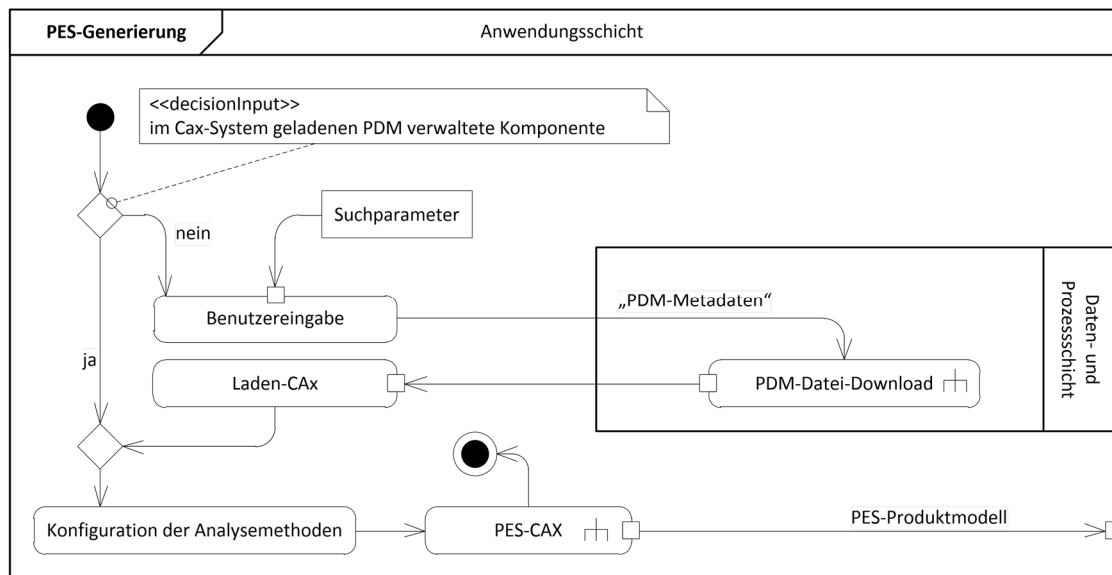


Bild 4-52: Aktivität PES-Generierung

Wenn der virtuelle Prototyp im CAx-System geladen ist, müssen die Analysemethoden konfiguriert werden. Die Praxis zeigt, dass der Aufbau virtueller Prototypen in Abhängigkeit des CAx-Systems, der verwendeten CAx-Integration sowie der Unternehmensrichtlinien differenziert. Beispiele im Umfeld von CAD-Systemen sind:

- virtuelle Baugruppen, bei denen die untergeordneten Komponenten nicht separat verwaltet sind,
- Baugruppen, die aus verschiedenen Versionen nativer virtueller Prototypen bestehen und unterschiedliche Objektmodelle nutzen,
- Produktstrukturen, die aus der Nutzung von Konfigurationen auf Einzelteil- und Baugruppenebene resultieren sowie
- unterschiedlich verwaltete PDM-Metadaten.

Die im Rahmen dieser Arbeit prototypisch implementierten CAD-Analysemethoden basieren auf dem klassischen Baugruppe/Einzelteil Ansatz, bei dem die PDM-Metadaten Bestandteil des Komponentennamens bzw. Dateinamens sind.

Nach der Konfiguration der Analysemethoden, wird mittels der Aktivität PES-CAx der virtuelle Prototyp untersucht (Bild 4-53). Dies erfolgt über die Programmierschnittstellen des CAx-Systems. Handelt es sich beim virtuellen Prototyp um eine Teil-Ganzes-Hierarchie, erfolgt die Analyse rekursiv entlang der Struktur, beginnend mit dem obersten Element. Während der Analyse wird für jede CAx-Komponente ein PES-Objekt vom Typ Komponente erstellt (siehe Kapitel 4.2.1). Dieses transportiert identifizierende PDM-Metadaten, Informationen zur Transformation sowie zum Erscheinungsbild. Existieren untergeordnete CAx-Komponenten, wird das PES-Objekt um eine Produktstruktur sowie die beschriebenen strukturspezifischen Eigenschaften erweitert. Inhalt der Produktstruktur sind identifizierende PDM-Metadaten, die die untergeordneten CAx-Komponenten im PDM beschreiben. Im Anschluss wird jede untergeordnete CAx-Komponente geladen und die Aktivität PES-CAx neu gestartet.

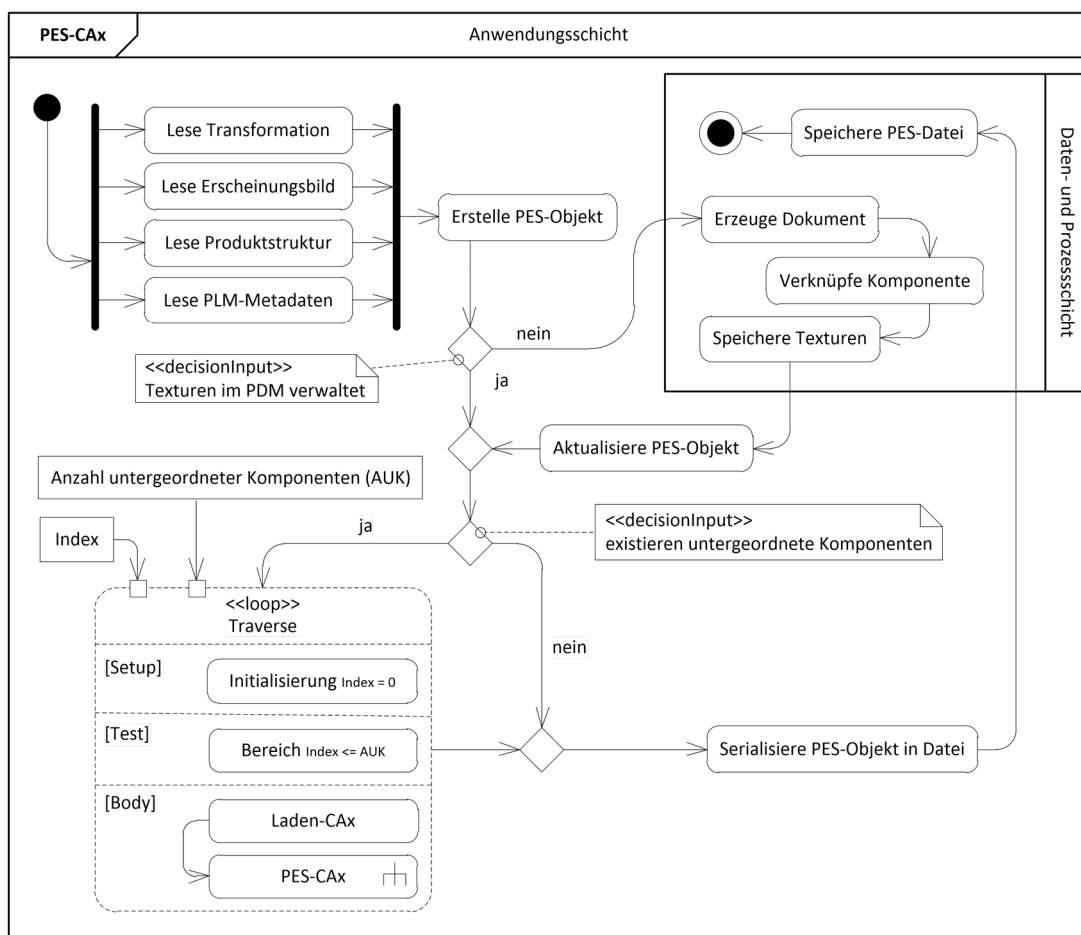


Bild 4-53: Komponentenbasiertes Generieren und Speichern von PES-Produktmodellen

Sind der CAx-Komponente Texturen zugeordnet, die keine PDM-Metadaten aufweisen, erzeugt die Aktivität PES-CAx ein entsprechendes Dokument einschließlich der Texturdateien

im PDM und verknüpft dieses mit dem ursprünglichen Dokument der CAx-Komponente. Die identifizierenden PDM-Metadaten des neu erstellten Dokumentes werden anschließend in der Texturbeschreibung des PES-Objekts ergänzt.

Ist die Analyse der CAx-Komponente abgeschlossen, serialisiert die Aktivität PES-CAx das PES-Objekt in eine Datei und speichert diese zum Dokument der CAx-Komponente im PDM. Für eine etwaige anschließende Visualisierung innerhalb der Präsentationsschicht, gibt die aufrufende Aktivität PES-Generierung die generierten PES-Objekte, in Form des PES-Produktmodells, zurück (Bild 4-52).

Animationen und Präsentationsumgebungen beschreiben gleichwertige, alternierende Zustände des PES-Produktmodells. Das bedeutet, bei der Verwaltung mit dem klassischen dokumentbasierten Ansatz im PDM entsteht für jede Kombination aus CAx-Komponente, Präsentationsumgebung und Animation eine neue Dokumentversion oder ein weiteres Dokument. Die Folge ist ein Anstieg der Komplexität des Produktmodells. In Anbetracht der Tatsache, dass Animationen und Präsentationsumgebungen im Produktlebenszyklus vorrangig in der Produktentwicklung Anwendung finden und optionale Bestandteile des klassischen virtuellen Prototyps sind, bleiben sie in der Aktivität „PES-Generierung“ unberücksichtigt. Das Speichern und Verknüpfen von Animationen und Präsentationsumgebungen im PDM erfolgt im VR-PLM Integrationskonzept nach Bedarf.

4.4.2 Rekonstruktion

Rekonstruktion beschreibt die Generierung des PES-Produktmodells auf Basis von im PDM verwalteten PES-Dateien. Dieser Prozess ist im VR-PLM Integrationskonzept durch die Aktivität PES-Rekonstruktion beschrieben (Bild 4-54). Bezüglich der Übersichtlichkeit sind informationstechnische Datenflüsse zum Teil vernachlässigt.

Die Bestimmung des zu rekonstruierenden PES-Produktmodells erfolgt mit Beginn der Aktivität durch die Eingabe von identifizierenden PDM-Metadaten. Die eingegebenen PDM-Metadaten bzw. Suchparameter werden dann an die Aktivität PDM-Datei-Download übergeben, worauf lokale Kopien von allen im Rekonstruktionsprozess benötigten Dateien gespeichert werden. Hierzu gehören etwaige Texturen, diskrete Geometriebeschreibungen sowie die PES-Dateien. Der Umfang der Dateien resultiert aus der rekursiven Analyse der Relationen des eingangs definierten PDM-Dokuments.

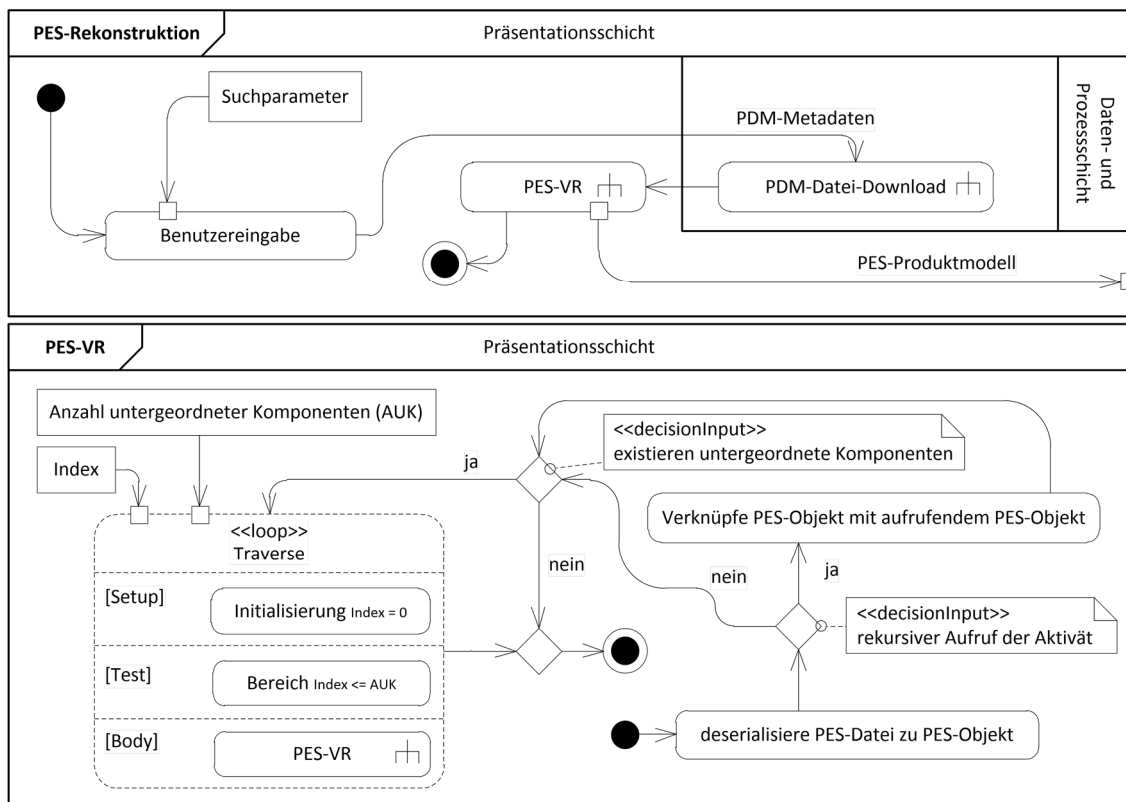


Bild 4-54: Rekonstruktion des PES-Produktmodells

Nach der Bereitstellung der Dateien, wird die Aktivität PES-VR aufgerufen und die Suchparameter der Aktivität PDM-Datei-Download übergeben (Bild 4-54). Mittels der Suchparameter besitzt die Aktivität PES-VR das Wissen über die PES-Datei, welche die Wurzel des PES-Produktmodells darstellt.

Beginnend mit der PES-Datei die den Suchparametern entspricht, wird durch die Aktivität PES-VR ein Objekt vom Typ Komponente deserialisiert (siehe Kapitel 4.2.1). Stellt das deserialisierte Objekt eine Generalisierung der Klasse Knoten dar, erfolgt ein rekursiver Aufruf der Aktivität PES-VR für alle untergeordneten Komponenten bzw. Blätter. Mit der ersten Rekursion wird das jeweils aktive PES-Objekt in die Liste der untergeordneten Komponenten des aufrufenden PES-Objekts eingetragen.

Sind alle rekursiven Aufrufe von PES-VR abgeschlossen, gibt die Aktivität PES-Rekonstruktion das PES-Produktmodell zurück. Dieses bildet die Grundlage für die Szenengenerierung innerhalb der Präsentationsschicht.

Von der Rekonstruktion des PES-Produktmodells ausgenommen sind Informationen des PDM, die mit Relationen des Typs 1:n einhergehen, die VR-Szene jedoch nur Relationen vom Typ 1:1 zulässt. Als Beispiel ist die Verwaltung von Präsentationsumgebungen zu nennen. Im PDM ist der Zustand abgebildet, dass ein virtueller Prototyp in verschiedenen Umgebungen dargestellt werden kann. Dem widerspricht jedoch die VR-Szene, in der jeweils nur eine Präsentationsumgebung aktiv sein kann. Gleiches gilt für Animationen. Informationen des PDM, die nicht eindeutig auf das Datenmodell der VR-Szene übertragen werden können, müssen zur Laufzeit durch den Nutzer eingepflegt werden. Die Realisierung des VR-PLM Integrationskonzepts muss hierfür entsprechende Benutzungsschnittstellen vorsehen.

4.5 Klassifizierende Metadaten

Neben der visuellen Beschreibung des virtuellen Prototyps sind prozessrelevante PLM-Metadaten wie beispielsweise der Freigabestatus eines Dokuments im Design Review von essentieller Bedeutung. Sie geben Auskunft über die Umsetzbarkeit von Empfehlungen bzw. Maßnahmen im Rahmen der Unternehmensprozesse.

Im klassischen Sinne erfolgt die Bestimmung von PLM-Metadaten unabhängig von der Präsentationsschicht durch einen separaten Zugriff auf die Daten- und Prozessschicht des PLM. Entsprechend der durch Listen geprägten Benutzerschnittstellen der verschiedenen PLM-Lösungen ist es Aufgabe des Anwenders, die gewonnenen Informationen auf die Objekte der VR-Szene zu übertragen.

Ein besserer Zugang zu PLM-Metadaten ist gegeben, wenn diese innerhalb der Präsentationsschicht abgerufen und dargestellt werden können. Mittels der im PES-Produktmodell verwalteten identifizierenden PLM-Metadaten können für jedes Objekt der VR-Szene die klassifizierenden PLM-Metadaten aus der PLM-Lösung anfragt werden. Die Anfrage erfolgt über die Programmierschnittstellen der PLM-Lösung.

Klassifizierende PLM-Metadaten, wie zum Beispiel der Freigabestatus, sind alphanumerischer Natur (Bild 4-55). Deren Abbildung unterliegt innerhalb der VR-Szene einer perspektivischen Darstellung, die sich mit jeder Bewegung des Anwenders ändert. Untersuchungen in [GWB07] zeigen, dass perspektivische Darstellungen von Text zu einem Anstieg der Lesezeit führen. Dem widerspricht der Wunsch, Informationen intuitiv aufzunehmen. In der Computergrafik existieren zwar spezielle Renderroutinen zur Kompensation von perspektivischen Darstellungen, diese können jedoch in einer heterogenen VR-Systemlandschaft nicht als Grundfunktion vorausgesetzt werden. Die alternative Nutzung der stets verzerrungsfreien Bildebene kann aufgrund möglicher Verdeckungen nur begrenzt Anwendung finden.



Bild 4-55: Perspektivische Verzerrungen und Verdeckungen bei der Visualisierung alphanumerischer Informationen

Darüber hinaus besteht bei der Visualisierung von alphanumerischen Informationen die generelle Herausforderung, die Zuordnung zum jeweiligen Objekt der VR-Szene herzustellen. Visuelle Verbindungselemente, wie zum Beispiel Linien, sind bei einer Vielzahl von Metadaten nur schwer zu verfolgen und können ebenfalls nicht als Grundfunktion vorausgesetzt werden.

Das VR-PLM Integrationskonzept bietet einen strukturierten Zugang zu Metadaten, indem die verschiedenen Ausprägungen des gewählten Metadatum zunächst durch Farben substituiert werden und diese dann auf die Objekte der VR-Szene angewendet werden. Das bedeutet, die Objekte der VR-Szene sind entsprechend der Wertigkeit des zugehörigen Metadatum eingefärbt (Bild 4-56). Man spricht von so genannten Fehlfarbdarstellungen, die zum Beispiel im Bereich der CAE-Analysen weit verbreitet sind. Fehlfarbdarstellungen sind hinsichtlich der Informationsvermittlung besonders effektiv, wenn die Anzahl der Ausprägungen eines Metadatum gering und deren semantische Bedeutung wenig komplex ist.

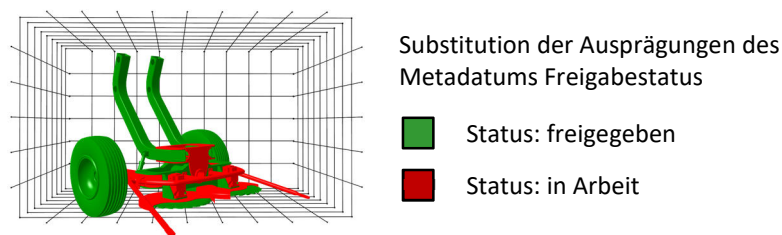


Bild 4-56: Visualisierung von PLM-Metadaten mittels Fehlfarbdarstellung

Fehlfarben ersetzen temporär die in Kapitel 4.2.4 beschriebenen Materialfarben der VR-Objekte. Bis auf die Material Diffuse Color (MDC) werden dabei alle Eigenschaften deaktiviert und eine automatisch generierte MDC neu zugewiesen. Für die Umsetzung der Farbgebung können dieselben Schnittstellen genutzt werden, die auch für die Szenengenerierung verwendet werden. Um die Symbolwirkung von Farben ausnutzen zu können, sollte die Visualisierung der klassifizierenden PLM-Metadaten um Schnittstellen erweitert sein, die eine manuelle Justierung der Fehlfarben erlauben.

Neben den klassifizierenden PLM-Metadaten können Fehlfarbdarstellungen auf alle Eigenschaften des PES- und SES-Modells angewendet werden. Hier bieten vor allem die durch Enumerationen beschriebenen Konstanten des SES-Modells einen Mehrwert im Design Review. So lassen sich zum Beispiel Risiken, Wechselwirkungen sowie der Zustand von Maßnahmen ad hoc am virtuellen Prototyp visualisieren.

Bild 4-57 zeigt die Verwaltung von PLM-Metadaten im PES-Produktmodell. Die Klasse *Komponente* assoziiert zu diesem Zweck die Klasse *MetaDatum*. Jedes Objekt vom Typ *MetaDatum* ist durch die Eigenschaften *ID*, *Wert* und *Fehlfarbe* charakterisiert. *ID* beschreibt die Art des Metadatum, *Wert* die Ausprägung bzw. den Zustand und *Fehlfarbe* die automatisch ermittelte MDC. Die Anzahl der möglichen Objekte vom Typ *MetaDatum* wird durch die Konstanten der Enumeration *MetaDatumTyp* beschrieben. Diese variiert mit den von der PLM-Lösung unterstützten klassifizierenden Metadaten.

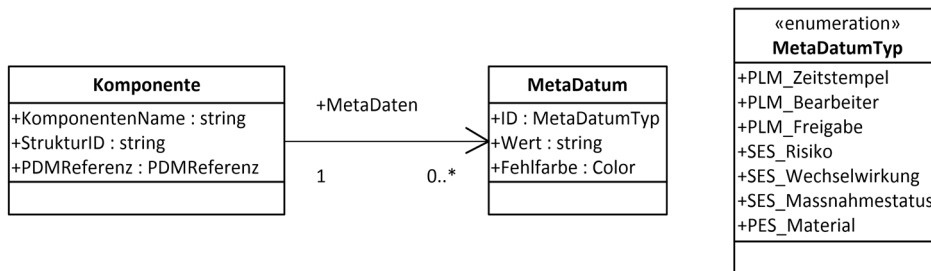


Bild 4-57: Verwaltung von PLM-Metadaten im PES-Produktmodell

Die Ausprägung von klassifizierenden PLM-Metadaten ist eine variable Größe, die anhand von identifizierenden PLM-Metadaten bestimmt wird. Objekte vom Typ *MetaDatum* sind deswegen von der XML-Serialisierung ausgeschlossen. Das bedeutet, die in der PLM-Lösung verwalteten PES-Produktmodelle enthalten keine Informationen zu klassifizierenden Metadaten.

Der Prozess zur Visualisierung von Metadaten ist durch die Aktivität *Metadaten-Visualisierung* beschrieben (Bild 4-58). Initial müssen mittels einer Benutzereingabe innerhalb Präsentations- oder Anwendungsschicht der gewünschte Typ des Metadatums sowie die zugehörige(n) Komponente(n) bestimmt werden. Anschließend wird pro angegebener Komponente das gewünschte Metadatum im PLM gelesen, ein Objekt vom Typ *MetaDatum* erzeugt, die Fehlfarbe generiert und das PES-Produktmodell aktualisiert. Abschließend werden die Fehlfarben auf die Objekte der VR-Szene angewendet und die Ergebnisse der Abfrage den anderen Teilnehmern des Design Reviews zur Verfügung gestellt.

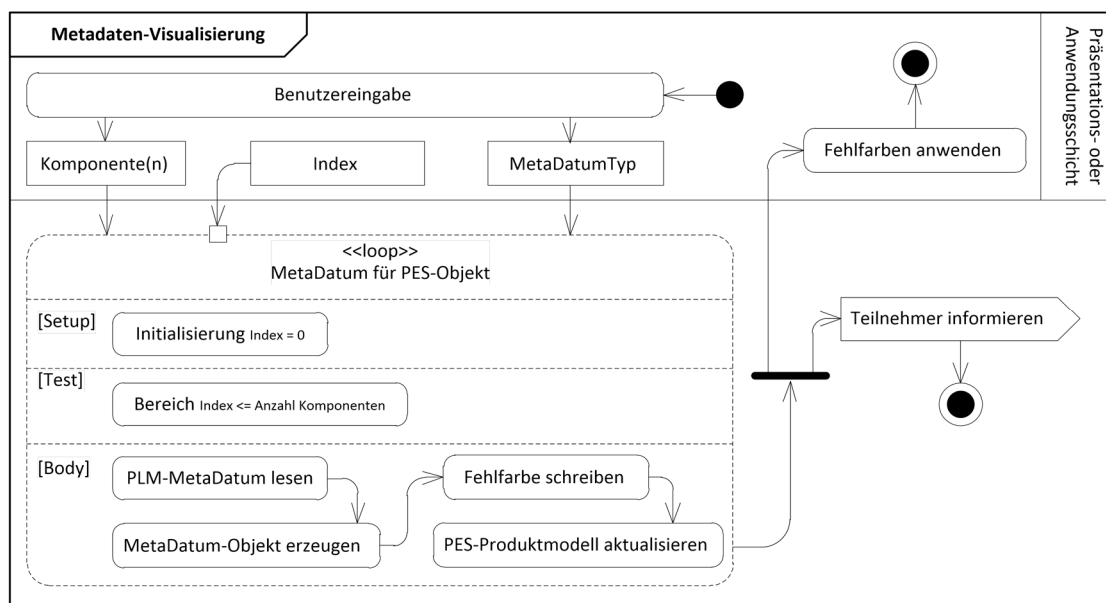


Bild 4-58: Prozessbeschreibung der Visualisierung von Metadaten

Die Generierung der Fehlfarben erfolgt während der Erzeugung des Objekts vom Typ *MetaDatum*. Hierzu kann zum Beispiel ein Farbkreis, wie er im HSV-Farbraum verwendet wird, entsprechend der Anzahl von differierenden Ausprägungen des gewählten Metadatum in Abschnitte unterteilt werden. Die Mittelwerte der Abschnitte entsprechen dann den Fehlfarben. Parallel zur Visualisierung der PLM-Metadaten muss eine geeignete Benutzungsschnittstelle bzw. Legende angezeigt werden, die die Fehlfarben und Ausprägungen des Metadatum in Relation setzt.

4.6 Bidirektionale Kopplung

Mit der Verwaltung von identifizierenden PLM-Metadaten im PES-Produktmodell existiert jedes Objekt der VR-Szene bzw. Präsentationsschicht im Kontext der Daten- und Prozessschicht. Dieser Zustand gleicht der Verwaltung nativer virtueller Prototypen mit klassischen CAx-Integrationslösungen, wie sie vom VR-PLM Integrationskonzept vorausgesetzt wird. Infolge der Synchronisation der Datenbestände von Präsentations- und Anwendungsschicht mit der Daten- und Prozessschicht ist eine wechselseitige Nutzung beider Schichten mit demselben Produktmodell gegeben. Präsentations- und Anwendungsschicht sind im VR-PLM Integrationskonzept somit bidirektional gekoppelt.

Das VR-PLM Integrationskonzept nutzt die bidirektionale Kopplung von Präsentations- und Anwendungsschicht, um die Bewertungstätigkeit im Design Review mittels etwaiger paralleler Bearbeitungsaufgaben am nativen virtuellen Prototyp zu unterstützen. Grundgedanke hierbei ist, die Bewertungstätigkeit als einen iterativen Prozess anzusehen, der auf Optimierungen des Produktmodells basiert. Diese sind innerhalb der Anwendungsschicht umzusetzen.

Die Kopplung erfolgt benutzergesteuert aus einer der Anwendungen der Präsentationsschicht heraus und folgt der Aktivität Kopplung-I (Bild 4-59). Während der Ausführung von Aktivität Kopplung-I werden initial die identifizierenden PDM-Metadaten des zu koppelnden virtuellen Prototyps sowie ein Zielsystem der Anwendungsschicht bestimmt. Danach wird innerhalb der Aktivität Kopplung-I ein Signal an das Zielsystem gesendet, das über die eingegeben identifizierenden PDM-Metadaten in Form eines VRPLM-Befehls (siehe Kapitel 5.5) informiert.

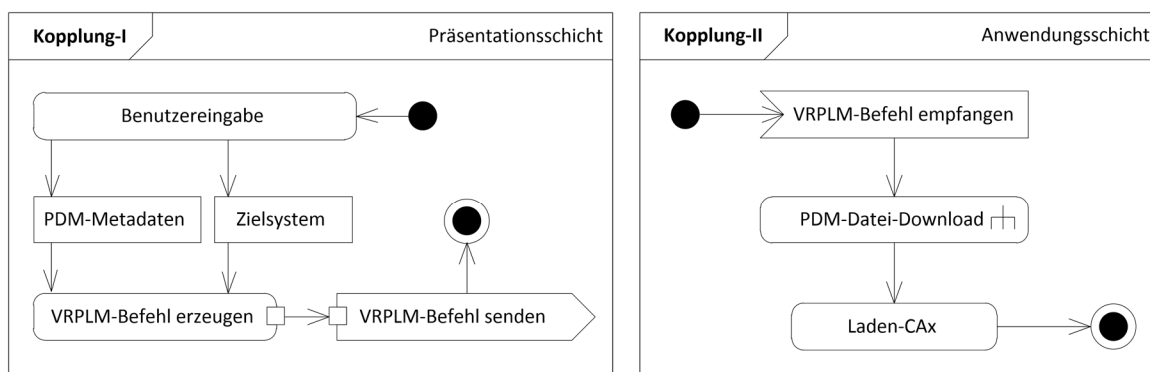


Bild 4-59: Kopplung von Präsentations- und Anwendungsschicht

Im Zielsystem wird innerhalb der Aktivität Kopplung-II ein Ereignis empfangen. Daraufhin wird mittels der zuvor bestimmten PDM-Metadaten die Aktivität PDM-Datei-Download gestartet. Diese erzeugt mittels der vom VR-PLM Integrationskonzept vorausgesetzten CAx-Integrationslösung eine lokale Kopie des im PDM verwalteten virtuellen Prototyps. Im Abschluss wird die lokale Kopie des virtuellen Prototyps im CAx-System geladen (Bild 4-59).

Die Aktualisierung der Präsentationsschicht infolge der Bearbeitung des virtuellen Prototyps erfolgt mittels der Aktivität Kopplung-III und kann jederzeit in einem der Erzeugersysteme der Anwendungsschicht initiiert werden (Bild 4-60). Mit der Initiierung wird die Aktivität PES-Generierung-2 gestartet und ein temporäres PES-Produktmodell erzeugt. Dieses repräsentiert den im Erzeugersystem geladenen virtuellen Prototyp. Die Aktivität PES-Generierung-2 unterscheidet sich derart von der Aktivität PES-Generierung, dass hinsichtlich des PES-Produktmodells keine schreibenden Zugriffe auf das PDM erfolgen. Des Weiteren wird parallel zur Aktivität PES-Generierung-2 eine diskrete Geometriebeschreibung erzeugt, die etwaige geometrische Änderungen abbildet. Im Anschluss wird ein VRPLM-Befehl an alle anderen Teilnehmer des Design Review gesendet, der über das temporäre PES-Produktmodell informiert. Da das temporäre PES-Produktmodell inkonsistent zum PDM ist, muss das ursprüngliche, zum PDM konsistente PES-Produktmodell weiter verwaltet werden. Die Teilnehmer können das temporäre PES-Produktmodell durch benutzergesteuertes initiieren der Aktivität Kopplung-IV nachladen, worauf die VR-Szene aktualisiert wird (Bild 4-60). Diese Aktualisierung kann beliebig oft erfolgen. Das Verwerfen der temporären Änderungen erfolgt durch Reaktivieren des zum PDM konsistenten PES-Produktmodells.

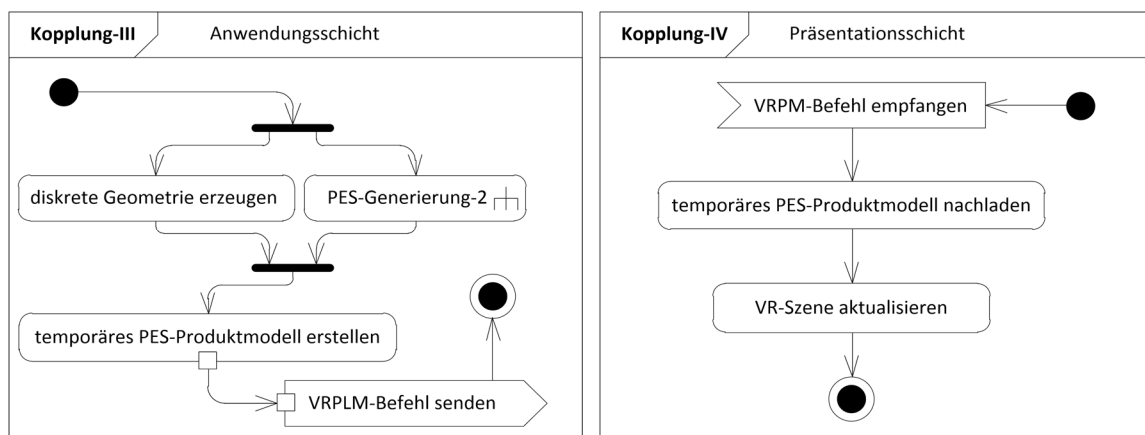


Bild 4-60: Aktualisierung der Präsentationsschicht infolge von Änderungen am nativen virtuellen Prototyp im Erzeugersystem

Der bisher beschriebene Ansatz der bidirektionalen Kopplung geht von der Verwendung nativer virtueller Prototypen innerhalb der Anwendungsschicht aus. Bezüglich konstruktiver Veränderungen am nativen virtuellen Prototyp, die auf Wechselwirkungen mit Komponenten weiterer Erzeugersysteme basieren, erzeugt die Aktivität Kopplung-II bei Bedarf eine heterogene Baugruppenbeschreibung innerhalb der Anwendungsschicht (Bild 4-61). Hierzu müssen in Aktivität Kopplung-I Komponenten des PES-Produktmodells bestimmt werden, die die konstruktive Tätigkeit innerhalb der Anwendungsschicht unterstützen sollen. Während Aktivität Kopplung-II werden dann neben dem zu bearbeitenden nativen virtuellen Prototyp auch die diskreten Geometriebeschreibungen der unterstützenden Komponenten aus dem

PDM geladen. Bei den diskreten Geometriebeschreibungen handelt es sich um dieselben, die auch für die Visualisierung innerhalb der Präsentationsschicht verwendet werden. Im Erzeugersystem wird abschließend eine heterogene Baugruppe erzeugt, die den nativen virtuellen Prototyp und die diskreten Geometriebeschreibungen entsprechend der im PES-Produktmodell gespeicherten Strukturen, Transformationen und Erscheinungsbilder zusammenfasst. Wird die Aktivität Kopplung-III im Kontext der heterogenen Baugruppe ausgeführt, muss nur der native, virtuelle Prototyp analysiert werden.

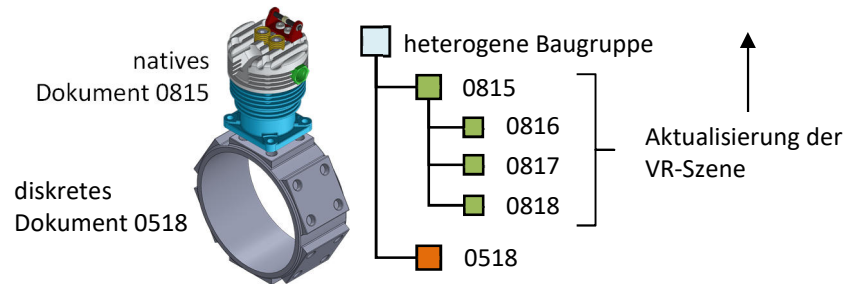


Bild 4-61: Bidirektionale Kopplung mit unterstützenden diskreten Produktdaten

Neben dem Testen temporärer Produktmodelloptimierungen während der Durchführung des Design Reviews kann die bidirektionale Kopplung von Präsentations- und Anwendungsschicht auch für Walkthroughs und zur Initiierung von Maßnahmen verwendet werden. Dazu wird, wie bisher beschrieben, ein virtueller Prototyp an das Erzeugersystem dirigiert. Nach Abschluss der Maßnahme wird aus der Anwendungsschicht heraus das Produktmodell der Daten- und Prozessschicht aktualisiert. Das bedeutet, die CAx-Integrationslösung speichert native und diskrete Geometrieinformationen und das VR-PLM Integrationskonzept aktualisiert das PES-Produktmodell. Nach der Aktualisierung der Daten- und Prozessschicht werden die Datenbestände der Teilnehmer des Design Reviews erneuert, sodass alle Informationen im VR-PLM Integrationskonzept wieder konsistent zur Daten- und Prozessschicht sind. Diese Verfahrensweise ermöglicht es, den Fortschritt der Maßnahmenumsetzung innerhalb der Präsentationsschicht in Echtzeit verfolgen zu können.

4.7 Wechsel des Erscheinungsbildes

Bezüglich des Erscheinungsbildes von Komponenten im VR-System existieren unterschiedliche Benutzeranforderungen. Zur Beurteilung der Produktgestalt und des Projektstands werden Farben mit funktionaler oder technologischer Bedeutung verwendet. Bei der Evaluierung von Design und Materialien liegt die Priorität auf einer fotorealistischen Visualisierung [PSS13].

Das VR-PLM Integrationskonzept unterstützt den wahlweisen Wechsel zwischen funktionalen und fotorealistischen Erscheinungsbildern, indem die im PES-Produktmodell verwalteten Materialeigenschaften unterschiedlich berücksichtigt werden (Bild 4-62). Während für fotorealistische Darstellungen alle Materialeigenschaften auf die Objekte der VR-Szene angewendet werden, sind es bei funktionalen Darstellungen nur die Material Ambient Color und die Material Diffuse Color. Die Reduzierung des Erscheinungsbildes erfolgt im VR-PLM Integrationskonzept nur innerhalb der datenkonsumierenden Präsentationsschicht.

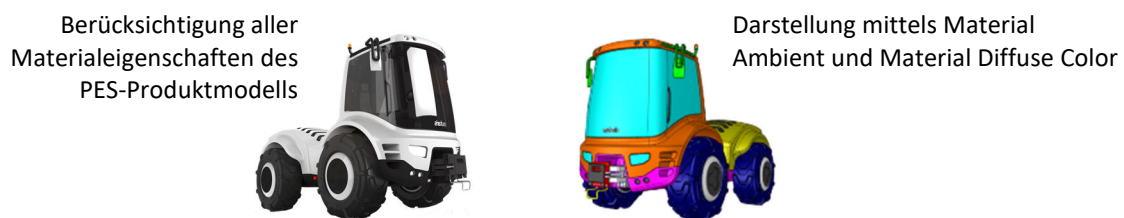


Bild 4-62: Wechselweise Darstellung des virtuellen Prototyps

Neben den im PES-Produktmodell verwalteten Materialeigenschaften bestimmt das im VR-System verwendete Beleuchtungsmodell maßgeblich die Erscheinung fotorealistischer Darstellungen. Wird ausschließlich das von der Lichtquelle emittierte Licht betrachtet, spricht man von einem lokalen Beleuchtungsmodell. Wechselwirkungen zwischen einzelnen Objekten der VR-Szene werden vernachlässigt. Lokale Beleuchtungsmodelle begünstigen das Rendering in Echtzeit und sind hervorragend für Bewertungen geeignet, die ein hohes Maß an Modellinteraktion erfordern. Dem gegenüber stehen so genannte globale Beleuchtungsmodelle. Sie berücksichtigen optische Wechselwirkungen zwischen den Objekten der VR-Szene und zeichnen sich durch ein erhöhtes Maß an Realismus aus. Weit verbreitet sind das Ray-Tracing und Radiosity Verfahren [Suf16]. Die Berechnung globaler Beleuchtungsmodelle ist mit einem erhöhten Aufwand verbunden. Sie werden daher vorrangig für Bewertungen eingesetzt, die anhand statischer Ansichten erfolgen. Gegenwärtig finden sie primär im Bereich des Designs Anwendung. Beleuchtungsmodelle sind systemspezifisch und nicht im PES-Produktmodell verwaltet. Die Aktivierung erfolgt benutzergesteuert über die Funktionen des VR-Systems.

4.8 Lokalisierung von Komponenten

Hinsichtlich der verteilten Bewertungstätigkeit im Design Review muss für jeden Teilnehmer ersichtlich sein, welche Komponenten des Produktes aktueller Gegenstand der Bewertung sind. Gerade bei großen und komplexen Produktmodellen ist es essentiell, den Blick der Teilnehmer auf eine bestimmte Region bzw. ein bestimmtes Objekt der VR-Szene lenken zu können. In Abhängigkeit der Lage der zu bewertenden Komponenten können Verdeckungen durch anliegende Komponenten auftreten, sodass die Sicht auf die zu bewertenden Komponenten blockiert ist. Bei der Arbeit mit virtuellen Prototypen ist es Stand der Technik, verdeckende Komponenten teilweise oder vollständig temporär transparent zu schalten. Diese Funktionalität beherrschen die meisten kommerziellen VR- und CAx-Systeme. Das VR-PLM Integrationskonzept muss Hilfsmittel berücksichtigen, die die Teilnehmer des Design Reviews dabei unterstützen, Komponenten bzw. Objekte der VR-Szene sicher lokalisieren zu können.

Bei der Lokalisierung von Komponenten folgt das VR-PLM Integrationskonzept der in [Lin11] durchgeführten Studie, die zeigt, dass sich der Mensch seiner visuellen Wahrnehmung am stärksten bewusst ist. Das VR-PLM Integrationskonzept berücksichtigt die Lokalisierung von Komponenten derart, dass Objekte der VR-Szene durch einen beliebigen Teilnehmer des Design Reviews markiert und im Wert ihrer Transparenz geändert werden können. Das bedeutet, die Anwendungen der Präsentationsschicht können sich gegenseitig fremdsteuern. Die Fremdsteuerung ist möglich, da die Objekte der VR-Szene in allen Anwendungen der Präsentationsschicht auf demselben PES-Produktmodell, mit identischen Metadaten, basieren. Darüber hinaus sind die Wechselwirkungen mit der VR-Szene auf ein Minimum begrenzt, sodass die Änderungen nahezu in Echtzeit erfolgen. Das Markieren von Objekten sowie die Änderung der Transparenz muss unabhängig vom Kameramodell erfolgen, dessen fremdgesteuerte Manipulation zu Unbehagen führen kann.

Die Markierung von Objekten kann durch zwei Ansätze auf einfache Weise unterstützt werden. Der erste Ansatz nutzt die Funktion der VR-Systeme, die selektierte Komponenten in der Regel visuell hervorgehoben darstellt. In diesem Fall selektiert ein Teilnehmer des Design Reviews eine Komponente und weist die VR-Systeme der anderen Teilnehmer dazu an, diese Selektion ebenfalls auszuführen. Ist eine Komponente im VR-System fremdselektiert, ist es die Aufgabe des Nutzers, das Sichtfeld solange zu verändern, bis die Komponente erscheint. Viele Anwendungen unterstützen auch die Funktion, selektierte Komponenten automatisch in das Sichtfeld des Nutzers zu bringen.

Für den Fall, dass die Programmierschnittstelle eines VR-Systems die Selektion von Komponenten nicht unterstützt, können die Material Diffuse Color der zu selektierenden

Komponente durch eine Signalfarbe ersetzt und alle weiteren Materialeigenschaften deaktiviert werden. Hierfür eignen sich dieselben Schnittstellen, die auch für die Szenengenerierung verwendet werden. Mit der Manipulation der Material Diffuse Color muss das VR-PLM Integrationskonzept eine Benutzungsschnittstelle vorsehen, mittels der die Wiederherstellung des ursprünglichen Erscheinungsbildes von Komponenten aktiviert werden kann.

Die Veränderung der Transparenz folgt dem zweiten Ansatz zur Markierung von Komponenten. Ein Teilnehmer des Design Reviews muss zunächst die Transparenz einer Komponente mittels der Funktionen seines VR-Systems verändern. Daraufhin wird die veränderte Komponente hinsichtlich der Transparenz analysiert und die anderen VR-Systeme über den ermittelten Wert der Transparenz informiert. In den Zielsystemen wird dann über die Schnittstellen zur Szenengenerierung die Transparenz der betroffenen Komponente neu geschrieben. Zur Aktivierung der Wiederherstellung der ursprünglichen Transparenz muss das VR-PLM Integrationskonzept ebenfalls eine Benutzungsschnittstelle vorsehen.

4.9 Import strukturfremder virtueller Prototypen

In Anbetracht der Lösungsfindung im Design Review besteht generell die Anforderung, bereits bestehende Komponenten bzw. Teilefamilien zu berücksichtigen. Da im VR-PLM Integrationskonzept die Geometrie von Komponenten unabhängig von der Struktur des PES-Produktmodells verwaltet wird, kann auf einfache Weise eine strukturfremde Komponente temporär in die VR-Szene integriert werden.

Die temporäre Integration einer strukturfremden Komponente setzt voraus, dass deren geometrische Beschreibung in einem VR-konformen Format vorliegt. Die Praxis zeigt, dass in vielen Unternehmen die Verwaltung von Austauschformaten im PLM zum Stand der Technik gehört. Sollten diese nicht verwaltet werden, können zum Beispiel die Funktionen der Erzeugersysteme verwendet werden, um derartige Beschreibungen zu generieren.

Die Integration erfolgt manuell, indem ein Objekt des PES-Produktmodells ausgewählt und die darin referenzierte VR-konforme Geometriebeschreibung ausgetauscht wird. Gleiches gilt für die Transformation und das Erscheinungsbild. Das PES-Produktmodell der manuell veränderten Komponente wird separat zum ursprünglichen PES-Produktmodell durch das VR-PLM Integrationskonzept verwaltet, sodass die Änderung jederzeit aufgehoben werden kann. Nach der manuellen Integration der strukturfremden Komponente wird das zugehörige Objekt der VR-Szene neu geladen. In der Folge erscheint anstelle der ursprünglichen die strukturfremde Komponente.

Neben der Berücksichtigung von bereits bestehende Komponenten bzw. Teilefamilien im Design Review ermöglicht der beschriebene Ansatz die Integration von physischen Prototypen, die in der Entwicklungsphase keine virtuelle Entsprechung besitzen (Bild 4-63). So können beispielsweise Designmodelle aus Styropor oder Knetmasse zur Laufzeit des Design Reviews wiederholt mittels der Methoden des Reverse Engineering eingescannt und in eine VR-konforme diskrete Geometriebeschreibung konvertiert werden. Im Anschluss erfolgt die Integration ins VR-PLM Integrationskonzept. Ist in der Produktstruktur ein Platzhalter vorgesehen, kann die eingescannte Komponente visualisiert werden, ohne dass eine andere Komponente überschrieben wird.

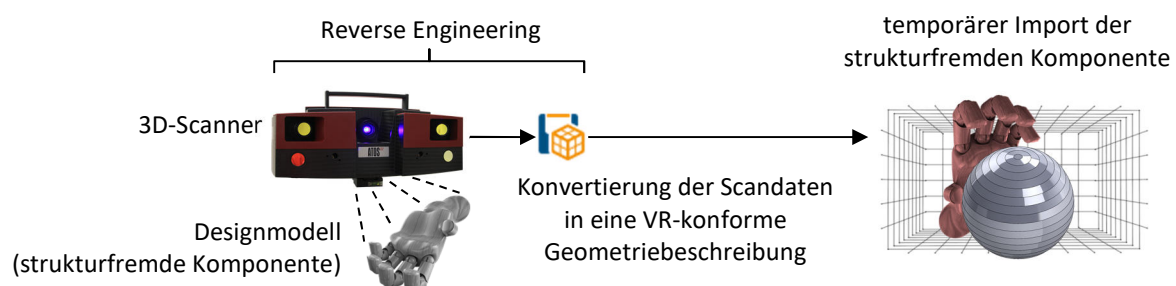


Bild 4-63: Integration von Designmodellen mittels der Methoden des Reverse Engineering

4.10 Handhabung und Organisation

4.10.1 Sitzungskonzept

Ein Design Review zeichnet sich durch die Zusammenarbeit mehrerer Personen im Kontext eines spezifischen Produktmodells aus. Diesen Zustand bildet das VR-PLM Integrationskonzept mittels eines Sitzungskonzeptes ab. Zu den Aufgaben dieses Konzeptes gehören die Verwaltung der Teilnehmer, die Wahrung konsistenter Datenmodelle, sowie die Steuerung von Benutzerinteraktionen. Das Sitzungskonzept unterscheidet zwischen den Modellen Session und Sessionmanager.

Die grundlegende Funktionsweise des Sitzungskonzeptes besteht darin, dass sich ein Teilnehmer beim Sessionmanager registriert und eine Session bzw. ein Design Review eröffnet. Weitere beim Sessionmanager registrierte Teilnehmer können dann der Session beitreten. Alternativ kann jeder Teilnehmer eine weitere Session eröffnen. Das Sitzungskonzept ermöglicht somit die Existenz parallel stattfindender Design Reviews.

Das Sessionmodell repräsentiert das Design Review aus Datensicht und verwaltet das PES-Produktmodell sowie das SES-Modell. Darüber hinaus dient das Sessionmodell als Speicherbereich für temporäre Informationen, die zwischen den Teilnehmern des Design Reviews angesichts der Unterstützung der Bewertungstätigkeit ausgetauscht werden können. Hierzu gehören zum Beispiel die mit der Kopplung von Präsentations- und Anwendungsschicht einhergehenden temporären PES-Produktmodelle und diskreten Geometriebeschreibungen.

Das Modell des Sessionmanagers verwaltet die parallelen Design Reviews. Es bildet Relationen zwischen Teilnehmern und Sessionmodellen ab und kontrolliert Benutzerinteraktionen mittels eines Berechtigungskonzeptes. Das Berechtigungskonzept basiert auf den in DIN EN 61160 beschriebenen personenspezifischen Rollen. Das Modell des Sessionmanagers kann im Sitzungskonzept somit als Kontrollschicht verstanden werden.

Bild 4-64 zeigt das Modell des Sessionmanagers, das das Sessionmodell integriert. Die Klasse *SessionManager* assoziiert die Klassen *Session*, *Teilnehmer* sowie *SessionBerechtigung* und stellt alle Methoden für Benutzerinteraktion mit dem Sitzungskonzept bereit. Die Klasse *Session* besitzt die Eigenschaften *PES*, *SES*, *SessionID* und *VRPLMBefehle*. Mittels der Eigenschaften *PES* und *SES* werden das PES-Produktmodell sowie das SES-Modell repräsentiert. Die *SessionID* stellt einen Bezeichner dar, mittels der jedes Objekt vom Typ *Session* eindeutig identifiziert werden kann. Temporäre Informationen, die zwischen den Teilnehmern des Design Reviews ausgetauscht werden, sind durch die Eigenschaft *VRPLMBefehle* beschrieben. Der Klasse *Teilnehmer* sind die Eigenschaften *Authentifizierung*, *Autorisierung* und *SessionID* zugeordnet. Mittels der *SessionID* wird die Relation zwischen

Teilnehmer und Session abgebildet. Die Eigenschaft *Authentifizierung* dient der eindeutigen Identifizierung eines Teilnehmers. Erlaubte Benutzerinteraktionen des Teilnehmers werden durch Konstanten der Enumeration *Rolle* beschrieben, die der Eigenschaft *Autorisierung* entsprechen. Mit der Klasse *SessionBerechtigung* wird das Berechtigungskonzept verwaltet, indem Konstanten der Enumerationen *Rolle* und *SessionFunktion* in Relation gesetzt werden.

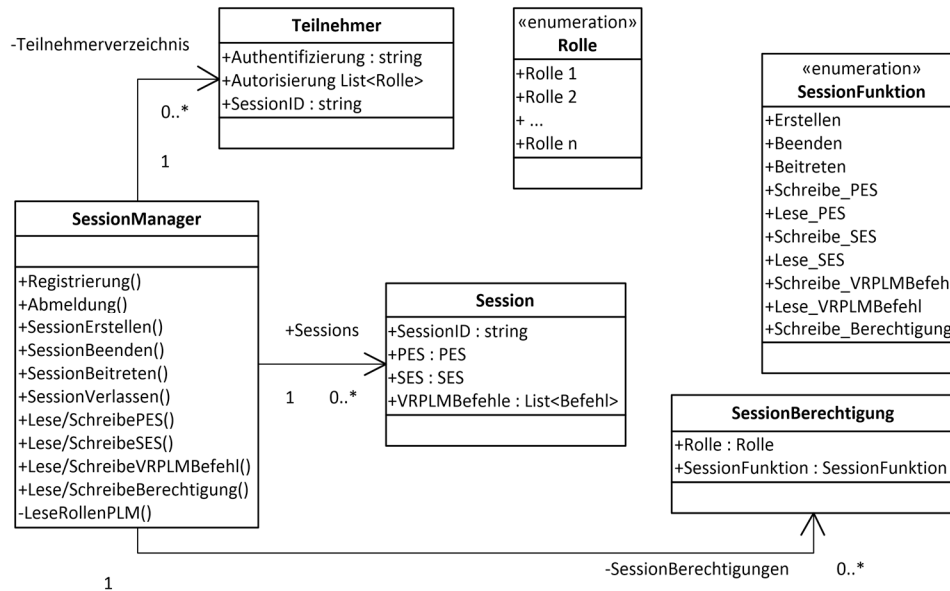


Bild 4-64: Sessionmodell und Modell des Sessionmanagers im Kontext des Sitzungskonzeptes

4.10.2 Autorisierung

Die mit den Berechtigungskonzepten eingeführten Rollen dienen als Kontrollmittel für die Nutzung von Ressourcen und Benutzerinteraktion im VR-PLM Integrationskonzept. Da die meisten PLM-Lösungen bereits über ausgereifte, personenspezifische Rollenkonzepte verfügen, wird eine Übernahme und Erweiterung der bestehenden Konzepte angestrebt. Damit folgt das VR-PLM Integrationskonzept dem Anspruch, die am Design Review beteiligten Anwendungen anforderungskonform einzusetzen.

Die im VR-PLM Integrationskonzept anzuwendenden Rollen entsprechen denen der DIN EN 61160. Hierzu gehören die Rollen Sitzungsleiter, Schriftführer, Entwickler und Fachkraft. Die Integration des PLM-Rollenkonzepts basiert darauf, dass zu jeder der genannten Rollen ein Äquivalent im PLM existiert und die Registrierung eines Teilnehmers im Sitzungskonzept mit der Authentifikation im PLM einhergeht. Letzteres wird in der Softwaretechnik als Single Sign-On (SSO) Konzept bezeichnet. Weitere Voraussetzungen für die Übernahme des PLM-Rollenkonzepts sind zum einen, dass Programmierschnittstellen zur PLM-Benutzerverwaltung existieren. Zum anderen muss das VR-PLM Integrationskonzept als Nutzer des PLM verstanden werden, der Rechte zum Lesen der Benutzerverwaltung besitzt. So ist es möglich die im Sitzungskonzept vorgesehenen Relationen zwischen Teilnehmer und Rollen abzuleiten.

Die mit den Rollen einhergehenden Berechtigungen werden im VR-PLM Integrationskonzept auf Benutzerinteraktionen mit dem Sitzungskonzept, PES-Produktmodell, Sitzungsbericht sowie auf VRPLM-Befehle angewendet. Tabelle 2 zeigt eine mögliche Konfiguration auf Grundlage von DIN EN 61160.

Tabelle 2: Relationen zwischen Rollen und Tätigkeiten im VR-PLM Integrationskonzept

	Session erstellen	Session beenden	Session beitreten	PES-Produktmodell schreiben	PES-Produktmodell lesen	SES-Modell schreiben	SES-Modell lesen	VRPLM-Befehl schreiben	VRPLM-Befehl lesen	VRPLM-Befehl löschen	Berechtigungen anpassen
Sitzungsleiter	•	•	•	•	•		•		•	•	•
Schriftführer			•		•	•	•		•		
Entwickler			•	•	•		•	•	•		
Fachkraft			•		•		•		•		

Der Prozess der Autorisierung folgt im VR-PLM Integrationskonzept der gleichnamigen Aktivität (Bild 4-65). Für den Fall, dass erforderliche Teilnehmer des Design Reviews nicht im PLM registriert (z. B. Kunden) sind, sollte eine Rolle Gast berücksichtigt werden, die sich durch ein passives Verhalten im Design Review auszeichnet. In Bezug auf Tabelle 2 könnten zum Beispiel nur die Berechtigung zum Beitritt der Session sowie zum Lesen des PES-Produktmodells gegeben sein. Des Weiteren ist zu berücksichtigen, dass abweichende Rollenbezeichnungen zwischen PLM und VR-PLM Integrationskonzept existieren. In diesem Fall können die verschiedenen Rollenbezeichnungen durch eine Mapping in Relation gesetzt werden. Im Ergebnis der Aktivität Autorisierung entsteht ein Objekt vom Typ Teilnehmer, das im Sitzungskonzept verwaltet werden kann.

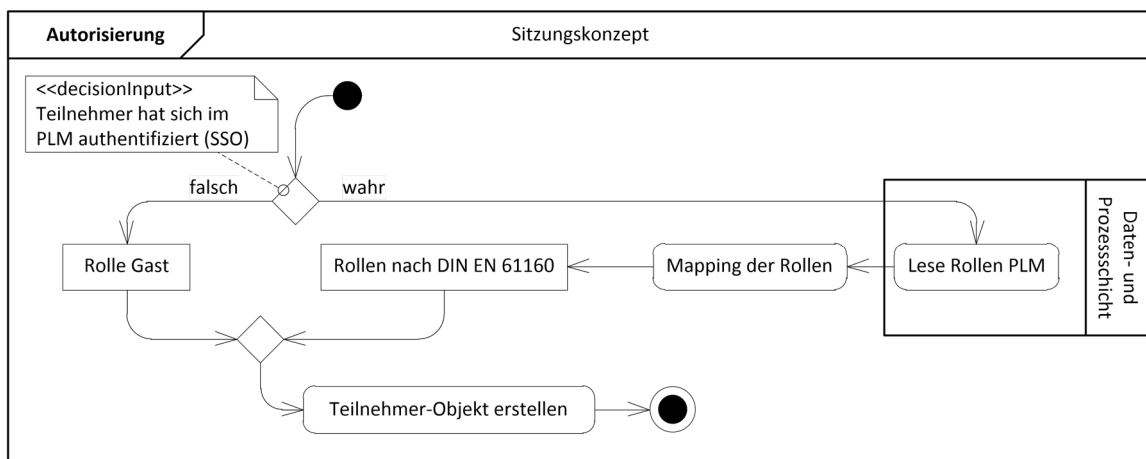


Bild 4-65: Aktivität Autorisierung

5 Realisierung der Middleware

5.1 VR-PLM System

Das VR-PLM Integrationskonzept beschreibt im Allgemeinen, dass entkoppelte Anwendungen der Präsentationsschicht, Daten- und Prozessschicht sowie Anwendungsschicht im Design Review miteinander wechselwirken. Hinsichtlich der Vermittlung von Funktionsaufrufen zwischen entkoppelten Softwarekomponenten muss die Realisierung des VR-PLM Integrationskonzeptes die Heterogenität der zugrunde liegenden verteilten Umgebung abstrahieren. Man spricht hierbei von einem so genannten Middleware-Konzept [CrD16].

Middleware ist eine Softwareschicht, die zwischen Anwendung und zugrundeliegendem Betriebssystem arbeitet. Diese wird im Rahmen der Realisierung des VR-PLM Integrationskonzeptes als VR-PLM System bezeichnet (Bild 5-1). Anwendungsseitig adaptiert die Middleware die API-Schnittstellen, mittels derer das VR-PLM System Parameter der Anwendung ermittelt und manipuliert bzw. die Anwendung selbst steuert. Auf der Seite des Betriebssystems adaptiert die Middleware so genannte Socket-Schnittstellen, mittels derer das VR-PLM System sowohl Daten an eine andere Anwendung sendet als auch von einer anderen Anwendung empfängt.

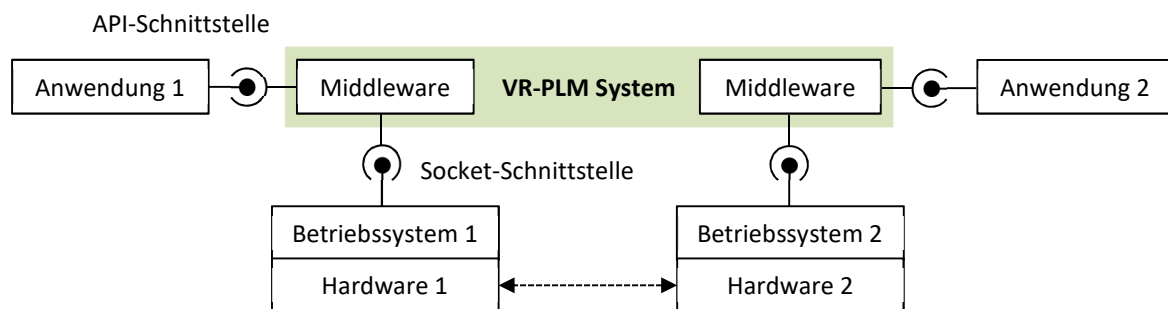


Bild 5-1: Verbinden entkoppelter Softwarekomponenten mittels Middleware

5.2 Client-Server Modell

Die Integration und Organisation von Anwendungen über Systemgrenzen hinweg erfolgt im VR-PLM System mittels des Client-Server Modells. Client-Server Systeme zeichnen sich durch einen zentralen Datenserver aus, der über informationstechnische Infrastrukturen wie das Internet oder lokale Netzwerke Informationen konsistent für eine beliebige Anzahl von Clients verwaltet [Abt15].

So genannte dezentrale Peer-to-Peer-Netzwerkstrukturen, bei denen jeder Client auch als Server fungiert, werden im Rahmen dieser Arbeit für die Integration und Organisation von Anwendungen im Design Review als ungeeignet angesehen. Ursache sind die redundanten Informationsbestände und der damit einhergehende erhöhte Aufwand zur Verwaltung konsistenter Datenmodelle. Zudem sind die Aufgaben des Servers im VR-PLM System auf die Datenverwaltung, Clientorganisation und Weiterleitung von anwendungsübergreifenden Funktionsaufrufen reduziert. Die ressourcenintensiven Methodenaufrufe der Programmierschnittstellen der integrierten Anwendungen erfolgen in den Clients, sodass eine mit Peer-to-Peer-Netzwerken einhergehende skalierende Verfügbarkeit von Ressourcen zu vernachlässigen ist.

Der Server realisiert im VR-PLM System das Sitzungskonzept und wird im weiteren Verlauf als Sessionmanager (SM) bezeichnet. Die Schnittstelle des Sessionmanagers beschreibt Methoden zum Verbindungsauf- und -abbau, zum Interagieren mit der Session und zum Lesen und Schreiben des PES-Produktmodells, des Sitzungsberichtes sowie der VRPLM-Befehle. Clients beschreiben die im VR-PLM System zu integrierenden Systeme und stehen sinnbildlich für die Teilnehmer im verteilt stattfindenden Design Review. Jeder Client implementiert eine Benutzungsschnittstelle, die Programmierschnittstellen der integrierten Anwendung und PLM-Lösung sowie die Schnittstelle zum Sessionmanager (Bild 5-2).

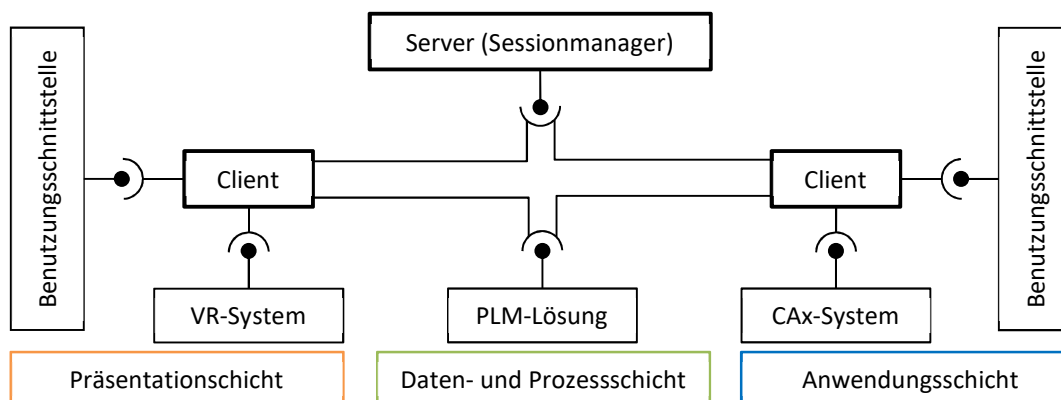


Bild 5-2: Anwendung des Server Client Systems im VR-PLM System

Die Kommunikation erfolgt im VR-PLM System mittels asynchroner Aufrufe. Das bedeutet, das Senden und Empfangen von Daten ist unabhängig voneinander. Dies hat zum einen den Vorteil, dass ein Client nicht blockiert, wenn die Anfrage nicht ad hoc verarbeitet werden kann. Zum anderen sind die Teilnehmer im Design Review flexibel, wenn es um die Abarbeitung von Anfragen geht. Die asynchrone Kommunikation zwischen Clients wird durch das Zwischenspeichern der Anfragen bzw. VRPLM-Befehle im Sessionmanager erreicht. Nach dem Eingang einer Anfrage informiert der Sessionmanager den entsprechenden Client, worauf dieser freigestellt ist die Anfrage zu verarbeiten.

5.3 Session

Die Verwaltung von Clients und Daten erfolgt im VR-PLM System durch den Sessionmanager. Hierbei besteht die Anforderung, dass jeder Client Daten verändern kann und dass die übrigen Clients über diese Änderung informiert werden. Ein hierfür anerkanntes Entwurfsmuster der Softwareentwicklung ist das so genannte Observer-Pattern [BrB16]. Das Observer-Pattern beschreibt im Allgemeinen, dass sich beobachtende Objekte bei einem zu beobachtenden Objekt registrieren und fortan von diesem informiert werden, sobald es sich ändert.

Bild 5-3 zeigt die Abbildung des Observer-Patterns, ausgehend von dem in Kapitel 4.10.1 herausgearbeiteten Sitzungskonzept. Das zu beobachtende Objekt ist der Sessionmanager, Beobachter sind die Clients. Die Klasse *Client* assoziiert die Klasse *SessionManager*. Um zu vermeiden, dass sich Beobachter und zu beobachtendes Objekt gegenseitig aufrufen können (Kreisreferenz), ist die abstrakte Klasse *ClientCallback* eingefügt. Die abstrakte Klasse *ClientCallback* leitet die Klasse *Client* ab. Durch dieses Konstrukt kann der Client während der Anmeldung am Sessionmanager als Objekt vom Typ *ClientCallback* gespeichert werden. In der Folge kommuniziert der Typ *Client* mit dem Typ *SessionManager* und der Typ *SessionManager* mit dem Typ *ClientCallback*. Die Klasse *SessionManager* stellt Methoden zur Registrierung der Beobachter und zur Interaktion mit der *Session* bereit. Der Beobachter wiederum stellt Methoden bereit, mittels der er über Veränderungen im PES-Produktmodell, Sitzungsbericht und den VRPLM-Befehlen informiert werden kann. Zudem stellt er Eigenschaften zur Verwaltung seiner Rollen und Relation zur Session bereit.

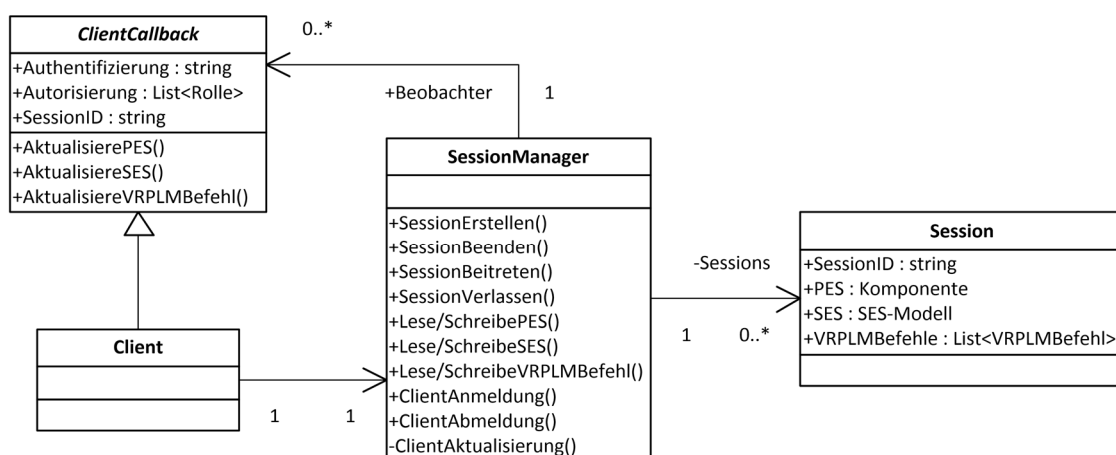


Bild 5-3: Anwendung des Observer-Patterns

Das Observer-Pattern findet Anwendung, wenn sich ein Client am Sessionmanager registriert und einer Session beitrete. Neben diesem Zustand existieren zwei weitere, in denen der Client unabhängig von der Session am Sessionmanager angemeldet bzw. abgemeldet ist. Je nach Zustand des Clients resultiert daraus ein spezifisches Verhalten, so kann sich zum Beispiel ein

Client nicht abmelden, wenn er nicht registriert ist. Das Gleiche gilt für Interaktionen mit der Session: Das Lesen oder Schreiben des PES-Produktmodells setzt voraus, dass der Client einer Session beigetreten ist.

Ein anerkanntes Entwurfsmuster der Softwareentwicklung zur Beschreibung eines laufzeitabhängigen Verhaltens von Objekten, ist das so genannte State-Pattern [BrB16]. Beim State-Pattern werden das Verhalten bzw. werden die Zustände durch separate Objekte beschrieben, die durch eine gemeinsame Schnittstelle beschrieben sind. Der Client kennt in diesem Fall nur ein Objekt, das hinsichtlich der Zustände *InSession*, *Angemeldet* oder *Abgemeldet* die Schnittstelle *ISessionManager* unterschiedlich implementiert (Bild 5-4). Mit der voneinander abweichenden Implementierung in den Zustandsobjekten, kann das geforderte angepasste Verhalten umgesetzt werden. Der Wechsel des Verhaltens wird innerhalb der Zustandsobjekte initialisiert, indem zum Beispiel die Methode *ClientAnmeldung* des Zustandes *Abgemeldet* den Zustand *Angemeldet* im *Client* instanziiert. Dieser Mechanismus erfordert, dass jedes Zustandsobjekt den Client assoziiert. Die Verknüpfung von State- und Observer-Pattern erfolgt, indem die Schnittstelle *ISessionManager* zum einen von den Zustandsobjekten, zum anderen vom *SessionManager* implementiert wird.

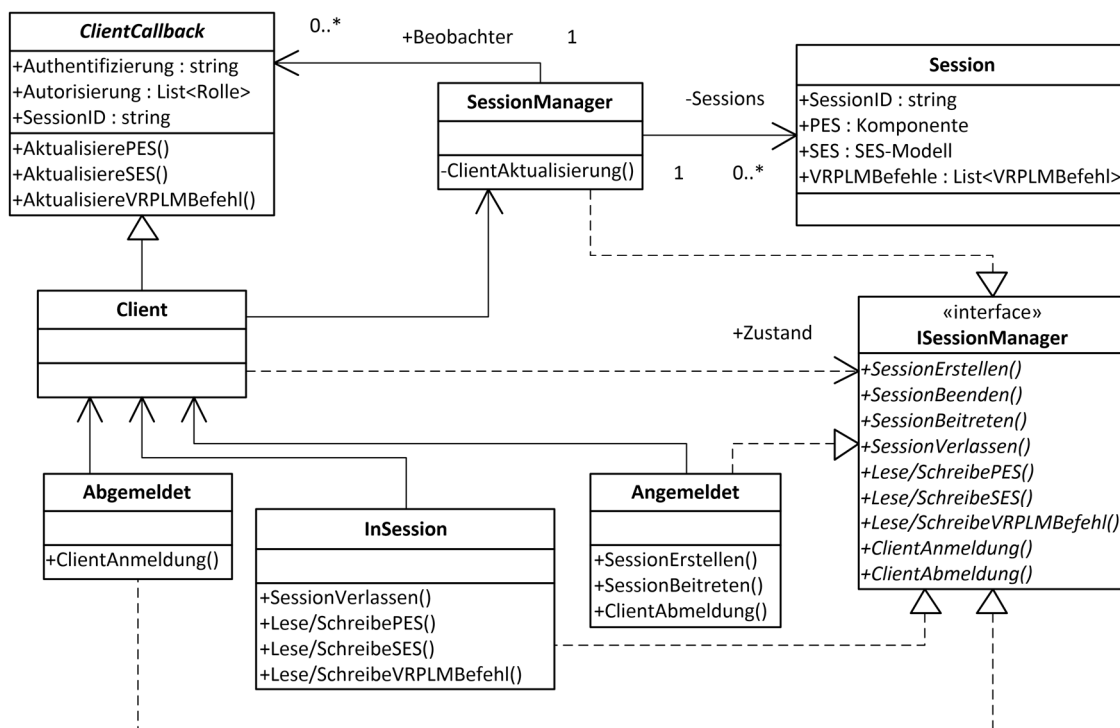


Bild 5-4: Kombination von State- und Observer-Pattern (in den Zustandsklassen sind zur Übersichtlichkeit nur quellcodeführende Methoden der Schnittstelle *ISessionManager* aufgeführt)

5.4 Adapter

In der Praxis sind die Objektmodelle und API-Schnittstellen von Programmen aus der Präsentations-, Anwendungs- sowie Daten- und Prozessschicht systemspezifisch. Das bedeutet, Schnittstellenumfang, Methodenbezeichnungen und verwendete Datentypen unterscheiden sich. Für die Vermittlung von Funktionsaufrufen zwischen den Softwarekomponenten ist daher meist eine API-Schnittstellenübersetzung erforderlich. Ein hierfür anerkanntes Entwurfsmuster ist das so genannte Adapter-Pattern [BrB16].

Die Umsetzung des Adapter-Patterns erfordert die Implementierung von Adaptern, die zum einen die systemspezifischen API-Schnittstellen der Anwendungen, zum anderen die Schnittstellen des VR-PLM Systems implementieren. Bild 5-5 zeigt beispielhaft die Funktionsweise des Adapterkonzepts beim Lesen von Farbinformationen im CAx-System und Schreiben im VR-System. Die allgemeinen Funktionen zum Lesen und Schreiben von Farbinformationen im VR-PLM System sind die Methoden *LeseFarbe* und *SchreibeFarbe*. Der CAx-Adapter implementiert die Methode *LeseFarbe*, die intern die Methoden *LeseRot*, *LeseGrün* und *LeseBlau* der API-Schnittstelle des CAx-System aufruft. Analog hierzu implementiert der VR-Adapter die Methode *SchreibeFarbe*, die intern die API-Schnittstelle *SchreibeRGB* des VR-Systems aufruft.

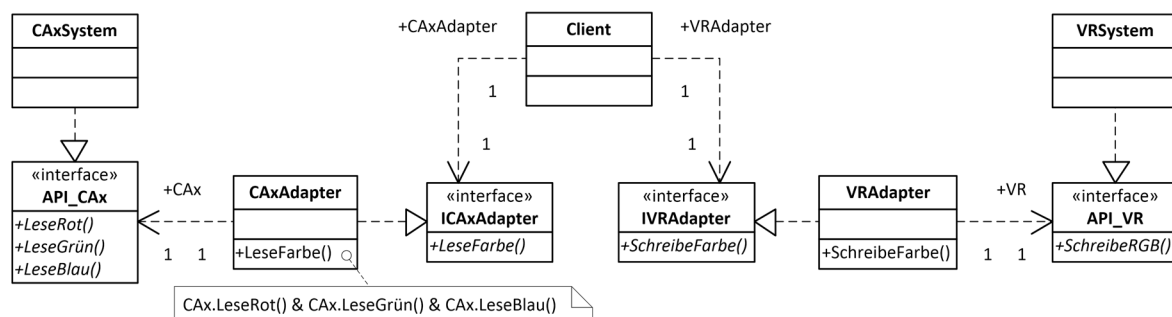


Bild 5-5: Anwendung des Adapter Patterns

Das Adaptern-Pattern bietet den Vorteil, dass die jeweiligen API-Schnittstellen vor dem VR-PLM System bzw. Client verborgen sind. Das bedeutet, es existieren keine Abhängigkeiten zwischen dem VR-PLM System und den Anwendungen der Präsentations-, Anwendungs- sowie Daten- und Prozessschicht. Soll ein weiteres Anwendungsprogramm in das VR-PLM System integriert werden, ist lediglich ein neuer Adapter zu entwickeln.

Adapter für Anwendungen der Präsentations-, Anwendungs- sowie Daten- und Prozessschicht implementieren jeweils eine spezifische Adapterschnittstelle (Bild 5-6). So ist im VR-PLM System gewährleistet, dass alle Adapter einer spezifischen Schicht den gleichen Funktionsumfang besitzen. Adapterschnittstellen der Präsentations- und Anwendungsschicht leiten zusätzlich die Schnittstelle *ICommonAdapter* ab. Diese beschreibt identische

Methodenaufrufe für Anwendungen der Präsentations- und Anwendungsschicht. Der grundlegende Funktionsumfang der Schnittstellen gestaltet wie folgt:

IVRAdapter (Präsentationsschicht)

- *VisualisierePES* (VR-Szene auf Grundlage des PES-Produktmodells generieren)

ICAxAdapter (Anwendungsschicht)

- *GenerierePES* (PES-Produktmodell des im CAx-System geladenen nativen virtuellen Prototyps generieren)
- *Laden* (einen nativen virtuellen Prototyp im CAx-System laden)
- *LadenHeterogen* (einen heterogenen, virtuellen Prototyp im CAx-System laden)

IPLMAdapter (Daten- und Prozessschicht)

- *ErstelleDokument* (Dokument im PDM anlegen)
- *ErstelleRelation* (Dokumente im PDM verknüpfen)
- *SucheDokument* (Dokument im PDM suchen)
- *LeseMetaDatum* (Metadatum eines Dokuments im PDM auslesen)
- *LeseRollen* (Rollen bzw. Berechtigungen eines Teilnehmers auslesen)
- *LadeDatei* (Datei eines Dokumentes laden)
- *SchreibeDatei* (Datei eines Dokumentes schreiben)

ICommonAdapter (Präsentations- und Anwendungsschicht)

- *LeseAnimation* (Animation im VR- oder CAx-System aufzeichnen)
- *Aktualisieren* (Objekte im VR- oder CAx-System aktualisieren)
- *LeseSelektierteKomponente* (selektierte Objekte im VR- oder CAx-System ermitteln)

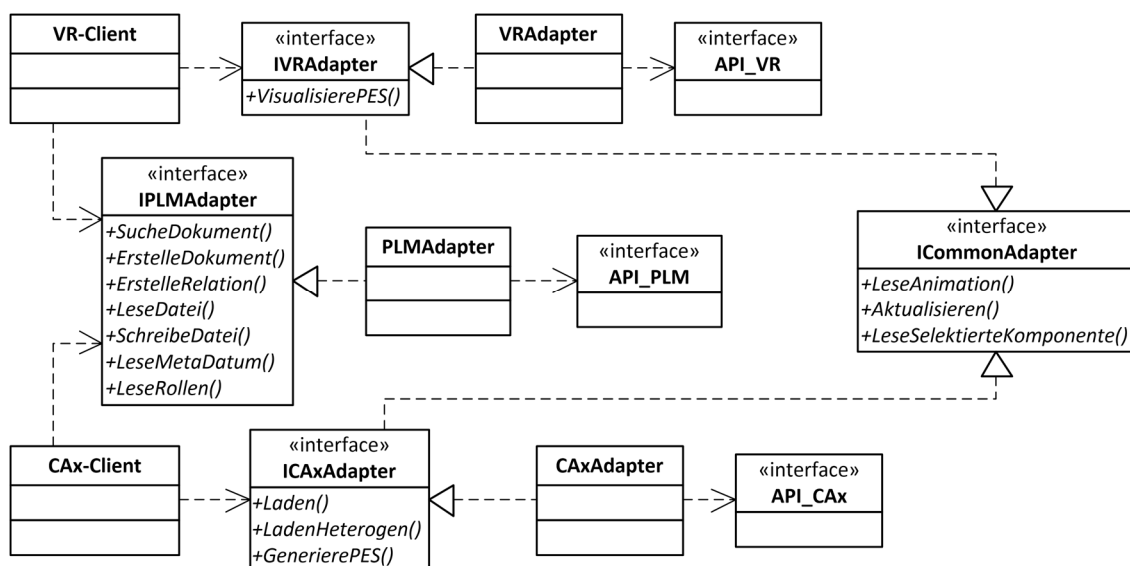


Bild 5-6: Schnittstellenbeschreibung der VR-PLM Adapter

Die anwendungsseitig implementierten API-Schnittstellen bleiben an dieser Stelle unbeachtet. Aufgrund systemspezifischer Objektmodelle und API-Schnittstellen kann eine Verallgemeinerung nicht stattfinden. Bei der Implementierung von Adaptern ist zu recherchieren, welche API-Schnittstellen der zu integrierenden Anwendung für die Umsetzung der jeweiligen Adaptermethode geeignet sind. Eine Orientierung hierfür bieten die Eigenschaften des PES-Produktmodells, die innerhalb der Anwendungsschicht zu lesen und in der Präsentationsschicht zu schreiben sind.

Mit der Existenz mehrerer Adapter für Programme der Präsentations-, Anwendungs- sowie Daten- und Prozessschicht ergeben sich unterschiedliche Konfigurationen des Clients. Um zu vermeiden, dass jede Konfiguration eine spezifische Implementierung erfordert, ist das Adaptern-Pattern um das Strategie-Pattern erweitert. Das Strategie-Pattern beschreibt im Allgemeinen die Austauschbarkeit von Unterklassen. Je nach Konfiguration bestimmt der Client somit selbständig (*DefiniereAdapter*), welcher Adapter zu instanziiieren ist. Bild 5-7 zeigt die Anwendung des Strategie-Patterns für Adapter der Präsentationsschicht.

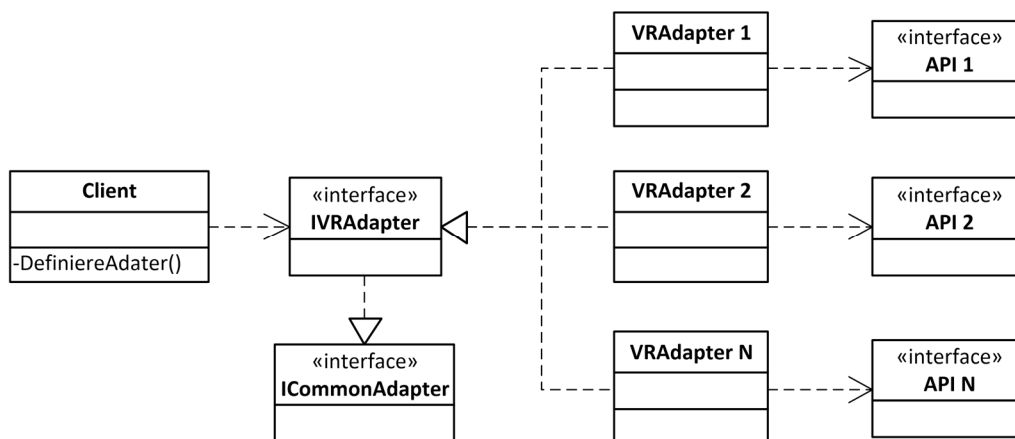


Bild 5-7: Kombination von Adapter- und Strategie-Pattern

5.5 VRPLM-Befehle

Interaktionen zwischen Anwendungen der Präsentations- und/oder Anwendungsschicht bedürfen im VR-PLM System einer Zustimmung des Nutzers des Zielsystems. Diese Einschränkung stellt sicher, dass laufende Bewertungsprozesse nicht durch andere Teilnehmer des Design Reviews unterbrochen werden. Zudem ist gewährleistet, dass Teilnehmer eingehende Informationen entsprechend ihren Bedürfnissen filtern können.

Die asynchrone Ausführung von Interaktionen basiert auf den so genannten VRPLM-Befehlen. Allgemein betrachtet, setzt sich ein VRPLM-Befehl aus drei theoretischen Abschnitten zusammen. Der erste Abschnitt enthält Metadaten, die den Absender, Empfänger, den Zeitpunkt des Versendens sowie die Priorität beschreiben. Der zweite Abschnitt stellt eine Kennung dar, anhand derer das VR-PLM System die Ausführung des VRPLM-Befehls im Zielsystem steuert. Der letzte Abschnitt beinhaltet die zu transportierenden Daten, die sich auf ein spezifisches Objekt im VR- oder CAx-System beziehen. Die ersten zwei Abschnitte sind im VR-PLM System durch die Klasse *VRPLMBefehl* und die Enumeration *Kennung* implementiert. Den dritten Abschnitt stellt die Klasse *Modifikation* dar (Bild 5-8).

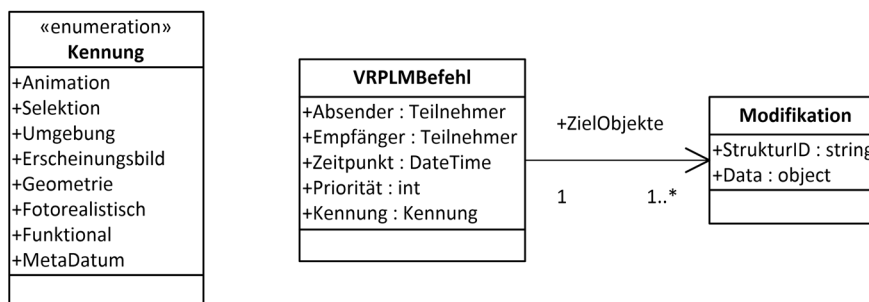


Bild 5-8: Datenmodell des VRPLM-Befehls

Die Generierung der VRPLM-Befehle erfolgt durch Benutzungsschnittstellen der Anwendungs- und Präsentationsschicht. Entsprechend der in Bild 5-8 dargestellten Enumeration *Kennung* existieren VRPLM-Befehle für:

- das Verteilen einer aufgezeichneten oder geladenen Animation,
- das Versenden einer Aufforderung zur Selektion,
- das Nachladen einer Präsentationsumgebung,
- das Versenden von Fehlerdarstellungen hinsichtlich klassifizierender Metadaten,
- die Aufforderung zur Anzeige des funktionalen Erscheinungsbildes,
- die Aufforderung zur Anzeige des fotorealistischen Erscheinungsbildes,
- das Verteilen einer im CAx-System veränderten Geometriebeschreibung sowie
- eines im CAx- oder VR-System veränderten Erscheinungsbildes.

Die Benutzungsschnittstellen der integrierten Anwendungen von Präsentations- und Anwendungsschicht müssen Methoden enthalten, mit denen die im Sessionmodell gespeicherten VRPLM-Befehle angezeigt, sortiert und ausgeführt werden können. VRPLM-Befehle sind solange im VR-PLM System zu verwalten, bis sie durch einen autorisierten Nutzer gelöscht oder nach Ablauf eines vordefinierten Zeitintervalls automatisch entfernt werden.

Einige VRPLM-Befehle stellen temporäre Änderungen am geladenen PES-Produktmodell dar. Hinsichtlich der Wiederherstellung des ursprünglichen Zustandes erzeugt das VR-PLM System vor der Ausführung des VRPLM-Befehls eine lokale Kopie des PES-Produktmodells. Auf dieses wird solange zurückgegriffen, bis sich während der Ausführung des VRPLM-Befehls das im PLM verwaltete PES-Produktmodell ändert.

5.6 Benutzungsschnittstellen

Das Konzept des VR-PLM Integrationskonzepts beschreibt im Allgemeinen, dass entkoppelte Anwendungen der Präsentationsschicht, Daten- und Prozessschicht sowie Anwendungsschicht im Design Review miteinander wechselwirken. Dies geschieht unter der Randbedingung, dass die Daten- und Prozessschicht von den anderen Schichten integriert wird.

Wechselwirkungen zwischen den Schichten basieren auf Mensch-Computer-Interaktionen, die mittels Benutzungsschnittstellen (UI, User-Interface) initiiert werden (Bild 5-9). Der Einsatz von Benutzungsschnittstellen erfolgt pro Client bzw. integrierter Anwendung der Präsentations- und Anwendungsschicht. Darüber hinaus existiert eine weitere Benutzungsschnittstelle für den Sessionmanager.

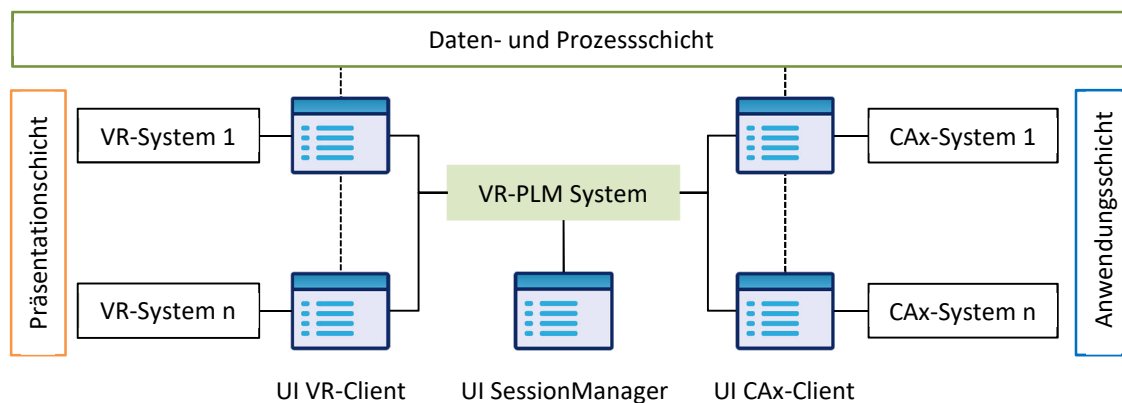


Bild 5-9: Benutzungsschnittstellen im VR-PLM-System

Für die Ausgestaltung von Benutzungsschnittstellen existieren mehrere Lösungsansätze bzw. Technologien. Weit verbreitet sind zum Beispiel textbasierte Konsolenaufrufe, grafische Bedienoberflächen sowie Sprach- und Gestensteuerungen. Die Qualität eines spezifischen Lösungsansatzes ist maßgeblich von den herrschenden Umwelteinflüssen (z. B. Umgebungsgeräusche und Licht), dem Einsatzszenario sowie den Fähigkeiten der bedienenden Person abhängig.

Um eine flexible Technologieauswahl zu gewährleisten, ist im VR-PLM System die Funktionalität der Benutzungsschnittstelle von der Technologie entkoppelt. Das bedeutet, die Benutzungsschnittstelle ist durch logische Berührungspunkte (Programmierschnittstelle) definiert, die Regeln zum Austausch von Informationen beschreiben. Die technologiebezogene Ausgestaltung der Benutzungsschnittstelle erfolgt dann in einer vom VR-PLM System unabhängigen Anwendung, die die definierten Regeln implementiert (Bild

5-10). Richtlinien zur Ausgestaltung von Benutzungsschnittstellen sind in EN ISO 9241-110 gegeben.

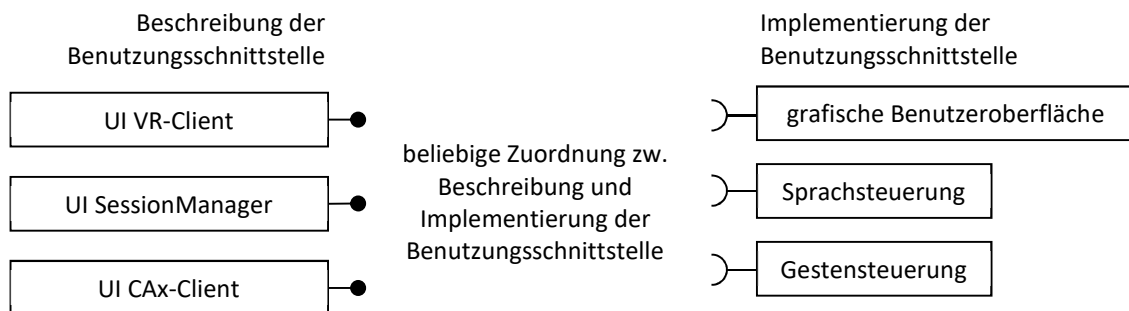


Bild 5-10: Entkopplung der Technologie von Benutzungsschnittstellen

Die Benutzungsschnittstellen für Anwendungen der Präsentations- und Anwendungsschicht sind im VR-PLM System durch die Programmierschnittstellen *IVRUI* und *ICAxUI* beschrieben (Bild 5-11). *ICAxUI* steht sinnbildlich für die Generierung VR-konformer Daten im CAx-System und *IVRUI* für deren Konsumierung im VR-System. Beide Programmierschnittstellen assoziieren die allgemeine Schnittstelle *ICCommonUI*. Diese assoziiert weitere Programmierschnittstellen für Benutzungsschnittstellen, die unabhängig von der Präsentations- und Anwendungsschicht sind. Hierzu gehören Benutzungsschnittstellen zum Umgang mit Animationen (*IAnimationUI*), Präsentationsumgebungen (*IPUmgebungUI*), wechselnden Erscheinungsbildern von Komponenten (*IErscheinungsbildUI*), dem Sitzungskonzept (*ISessionUI*, *IVerbindungUI*), dem Sitzungsbericht (*ISESUI*) und der Umgang mit PLM-Metaddaten (*IMetaDatumUI*). Die Programmierschnittstelle des Sessionmanagers ist durch die Schnittstelle *ISMUI* beschrieben.

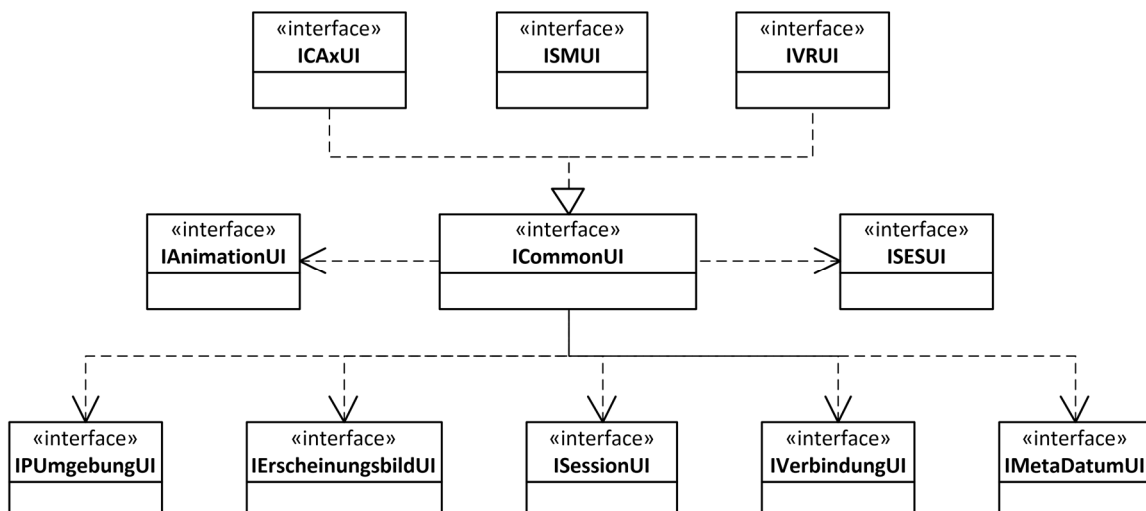


Bild 5-11: Programmierschnittstellen für Benutzungsschnittstellen

Die in Bild 5-11 dargestellten Programmierschnittstellen setzen im VR-PLM System die folgenden, grundlegenden, nach Kategorien geordneten, Benutzerinteraktionen um.

SessionManager (ISMUI)

- Konfiguration der Verbindungsparameter
- Sessionmanager starten
- Sessionmanager schließen

Sitzungskonzept (IVerbindungUI)

- Registrierung und Abmeldung beim Sessionmanager
- Erstellen, Beitreten und Verlassen einer Session
- Administration des Berechtigungskonzeptes
- Adapterauswahl für PLM-Lösung und CAx- oder VR-System

CAx-System (ICAxUI)

- Konfiguration der Analyseinstellungen
- PES-Produktmodell erstellen und im PDM speichern
- PES-Produktmodell aus dem PDM laden (generalisiert von ICommonUI)
- PES-Produktmodell der Session schreiben (generalisiert von ICommonUI)
- PES-Produktmodell der Session lesen (generalisiert von ICommonUI)
- Im PES-Produktmodell referenzierten virtuellen Prototyp laden

VR-System (IVRUI)

- VR-Szene aus PES-Produktmodell erstellen
- Bidirektionale Kopplung mit CAx-System herstellen
- PES-Produktmodell aus dem PDM laden (generalisiert von ICommonUI)
- PES-Produktmodell der Session schreiben (generalisiert von ICommonUI)
- PES-Produktmodell der Session lesen (generalisiert von ICommonUI)

Sitzungsbericht (ISESUI)

- Sitzungsbericht neu erstellen
- Sitzungsbericht aus dem PDM laden
- Sitzungsbericht mit der Session synchronisieren
- Sitzungsbericht der Session bearbeiten
- Sitzungsbericht der Session lesen
- Sitzungsbericht der Session im PDM speichern

- Den im Sitzungsbericht referenzierten virtuellen Prototyp innerhalb der Präsentations- bzw. Anwendungsschicht laden

PLM-Metadaten (IMetaDatumUI)

- Metadaten aus PLM lesen
- Metadaten im VR- bzw. CAx-System mittels Fehlfarbdarstellung visualisieren
- Metadaten als VRPLM-Befehl mit anderen Mitgliedern der Session teilen
- Fehlfarbdarstellung der Metadaten aufheben

Erscheinungsbild (IErscheinungsbildUI)

- Wechsel zwischen funktionaler und fotorealistischer Art der Darstellung im VR- bzw. CAx-System
- Darstellungsart mit anderen Mitgliedern der Session teilen
- Ein im VR- bzw. CAx-System lokal verändertes Erscheinungsbild mit anderen Mitgliedern der Session teilen
- Eine im VR- bzw. CAx-System lokal veränderte Transformation mit anderen Mitgliedern der Session teilen
- Eine im CAx-System lokal veränderte Geometrie mit anderen Mitgliedern der Session teilen
- Die Darstellung lokaler Änderungen von Erscheinungsbild, Transformation und Geometrie aufheben

PES-Animation (IAnimationUI)

- Animation im VR- bzw. CAx-System aufzeichnen
- Die aufgezeichnete Animation im PDM speichern
- Animation aus dem PDM laden
- Animation als VRPLM-Befehl mit anderen Mitgliedern der Session teilen
- Animation im VR- bzw. CAx-System wiedergeben
- Animation pausieren
- Animation aufheben

Präsentationsumgebung (IPUmgebungUI)

- Verwendung einer lokal gespeicherten Textur als Präsentationsumgebung im VR- bzw. CAx-System
- Verwendung einer im PDM gespeicherten Textur als Präsentationsumgebung im VR- bzw. CAx-System
- Präsentationsumgebung als VRPLM-Befehl mit anderen Mitgliedern der Session teilen

- Präsentationsumgebung und Transformation des virtuellen Prototyps im PDM speichern
- Präsentationsumgebung aus der VR-Szene entfernen

Hinsichtlich der in EN ISO 9241-110 angeführten Aufgabenangemessenheit und Selbstbeschreibungsfähigkeit wird empfohlen, Benutzerinteraktionen innerhalb der Präsentations- und Anwendungsschicht entsprechend den genannten Kategorien zu implementieren. Für die Selektion von Komponenten des PES-Produktmodells oder von Teilnehmern des Design Reviews eignen sich visualisierte Strukturen.

Die Konfiguration der Benutzungsschnittstelle eines Clients folgt im VR-PLM System dem Strategie-Pattern (Bild 5-12). Entsprechend des im Client konfigurierten Adapters für eine Anwendung der Präsentations- oder Anwendungsschicht, generalisiert die allgemeine Schnittstelle *IVRPLMUI* eine Schnittstelle vom Typ *ICAxUI* oder *IVRUI*.

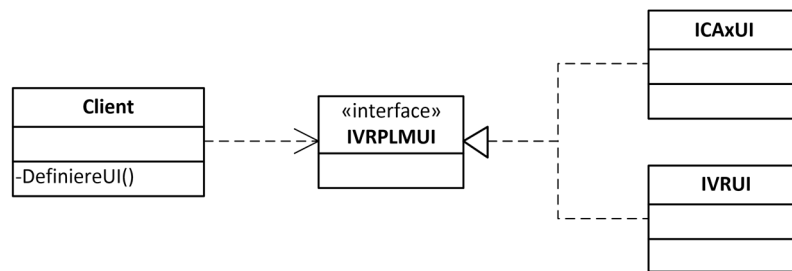


Bild 5-12: Anwendung des Strategie-Patterns zur Konfiguration der Benutzungsschnittstelle

6 Anwendungsszenario

Das in dieser Arbeit entwickelte VR-PLM Integrationskonzept ist durch die VR-PLM System Middleware prototypisch implementiert. Nachfolgend ist an einem Beispiel die Umsetzung der in Kapitel 2.3 herausgearbeiteten Prozesse Modellaufbereitung, Durchführung und Rückführung der Ergebnisse durch die prototypische Implementierung dargestellt. Im Fokus des Beispiels stehen die bidirektionalen Kopplungen der Präsentations- sowie Anwendungsschicht mit der Daten- und Prozessschicht des PLM.

Das Anwendungsszenario basiert auf einem im PDM verwalteten heterogenen virtuellen Prototyp (10000234726 DRW 000 00). Als Beispiel dient eine vereinfachte Flanschverbindung, deren linker und rechter Flansch mit einem unterschiedlichen CAD-System erstellt wurde (Bild 6-1).

Document				
0000000000000010000234726				
Type / Part / Vers. DRW / 000 / 00				
Description hybrider virtueller Prototyp				
Base qty (ST) 1.000				
L	Item	O.	Component number	Assembly
1	0010		10000234730 DRW 000 00	<input checked="" type="checkbox"/>
1	0020		10000234727 DRW 000 00	<input checked="" type="checkbox"/>
1			10000234730 DRW 000 00	<input type="checkbox"/>
2	0010		10000234731 DRW 000 00	<input type="checkbox"/>
2	0020		10000234732 DRW 000 00	<input type="checkbox"/>
1			10000234727 DRW 000 00	<input type="checkbox"/>
2	0010		10000234728 DRW 000 00	<input type="checkbox"/>
2	0020		10000234729 DRW 000 00	<input type="checkbox"/>

Beschreibung im PLM - 10000234726 DRW 000 00

Bild 6-1: Im PDM verwalteter heterogener virtueller Prototyp

6.1 Modellaufbereitung

Im Rahmen der Datenvorbereitung nach DIN EN 61660 müssen zunächst die PES-Produktmodelle der Teilbaugruppen erzeugt werden. Der Prozess der PES-Generierung kann entweder durch einen Batchaufruf oder durch eine Benutzungsschnittstelle initiiert werden. Im Rahmen dieses Beispiels wird der zuletzt genannte Ansatz demonstriert.

Die angesprochene Benutzungsschnittstelle ist in den Clients des VR-PLM Systems integriert (Bild 6-2). Da die PES-Generierung mit der Analyse der CAD-Daten im Erzeugersystem sowie dem Erstellen von Dokumenten im PLM einhergeht, müssen vor der Initiierung die Schnittstellenadapter für die jeweiligen Systeme ausgewählt werden. Der Client bietet hierfür Auswahlelemente, die die implementierten Adapter auflisten.

Nach der Auswahl der Adapter wird der Prozess der PES-Produktmodellgenerierung manuell durch die Schaltfläche *natives PES-Produktmodell generieren (ohne SM)* gestartet. Der


```

<m_CADData>
  <EntityName>Flansch_OEM_10000234731_DRW_000_00[1]</EntityName>
  <ComponentName>Flansch_OEM_10000234731_DRW_000_00</ComponentName>
  <ComponentID>1</ComponentID>
  <ConfigurationID>STD</ConfigurationID>
  <System>SOLIDWORKS</System>
  <FileName>Flansch_OEM_10000234731_DRW_000_00.SLDPRT</FileName>
  <AssemblyContext>
    <Appearance>
      <Material>nothing</Material>
      <RGBColor Array="Double">
        <Double>211</Double>
        <Double>170</Double>
        <Double>33</Double>
      </RGBColor>
      <Visibility>True</Visibility>
      <Opacity>0</Opacity>
    </Appearance>
    <RelativeTransformation>
      <Scale Array="Double">
        <Double>1</Double>
        <Double>1</Double>
        <Double>1</Double>
      </Scale>
    </RelativeTransformation>
  </AssemblyContext>
</m_CADData>
</NodeContainer>
<m_PDMData>
  <PDMSystem>SAP</PDMSystem>
  <DocumentNumber>000000000000010000234730</DocumentNumber>
  <DocumentType>DRW</DocumentType>
  <DocumentPart>000</DocumentPart>
  <DocumentVersion>00</DocumentVersion>
  <Description>adapter_oem.sldasm</Description>
</m_PDMData>
<m_CADData>
  <ComponentName>Adapter_OEM_10000234730_DRW_000_00</ComponentName>
  <System>SOLIDWORKS</System>
  <FileName>Adapter_OEM_10000234730_DRW_000_00.SLDASM</FileName>
  <RegularContext>
  </RegularContext>
</m_CADData>
<Children>
  <NodeContainer>
  <NodeContainer>
  </Children>
</NodeContainer>

```

Bild 6-3: Auszüge PES-Produktmodell

Zur weiteren Vervollständigung des PES-Produktmodells muss der in Bild 6-2 dargestellte Prozess auf alle nativen Komponenten des heterogenen virtuellen Prototyps angewendet werden. Der letzte Schritt der PES-Generierung besteht darin, dass auch für die heterogene PDM-Beschreibung ein PES-Produktmodell generiert wird. Dies erfolgt durch Betätigung der Schaltfläche *hybrides PES-Produktmodell generieren (ohne SM)*. Im Gegensatz zu einem nativen PES-Produktmodell besitzt ein hybrides PES-Produktmodell nur PDM-Referenzen, da keine zugehörige CAx-Komponente existiert. Dementsprechend findet auch keine Analyse eines virtuellen Prototyps statt.

Damit die nativen Komponenten des heterogenen virtuellen Prototyps innerhalb der VR-Szene positioniert werden können, wird das hybride PES-Produktmodell um Transformationsangaben erweitert (Bild 6-4). Die vom VR-PLM System standardmäßig erzeugten Transformationsmatrizen beschreiben Komponenten, die nicht verschoben, gedreht und skaliert sind. Dieser Zustand beschreibt einen heterogenen virtuellen Prototyp, dessen Komponenten gegen einen einheitlichen Koordinatenursprung konstruiert sind. Ist dies nicht der Fall, müssen im Rahmen der prototypischen Implementierung des VR-PLM Systems die Transformationsmatrizen des hybriden PES-Produktmodells manuell angepasst

werden. Ein Ansatz zur programmgesteuerten Anpassung der Transformationsmatrizen ist in [SSS10] dargestellt.



Bild 6-4: PES-Produktmodell einer hybriden Strukturbeschreibung

Mit der Generierung der PES-Produktmodelle für den linken und rechten Flansch sowie für die heterogene Beschreibung der Flanschverbindung ist der Prozess der Modellaufbereitung abgeschlossen.

6.2 Durchführung

Damit die Teilnehmer des Design Reviews miteinander wechselwirken können, muss initial der Sessionmanager gestartet werden (Bild 6-5). Da der Sessionmanager als Server agiert, muss er auf einer Hardware ausgeführt werden, die für alle Teilnehmer über eine informationstechnische Infrastruktur zu erreichen ist. Die mit der Erreichbarkeit einhergehende Konfiguration der Verbindungsparameter beinhaltet das Festlegen einer IP-Adresse, Port-Nummer sowie eines Passworts.

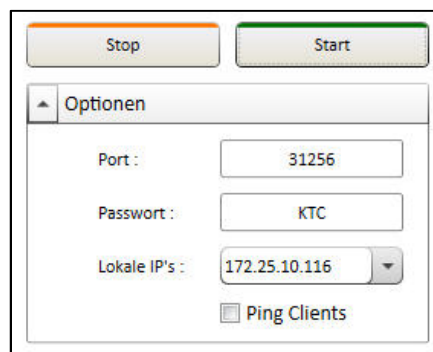


Bild 6-5: Prototypisch implementierte Benutzungsschnittstelle des SessionManagers

Ist der Sessionmanager gestartet können sich eine beliebige Anzahl von Clients verbinden. Die Client-Anmeldung erfordert neben der Adapterkonfiguration die Eingabe der im Sessionmanager definierten Verbindungsparameter. Darüber hinaus werden gültige PLM-Anmeldeinformationen abgefragt, damit sich der Client mit der Daten- und Prozessschicht verbinden kann. Nach der Anmeldung übersendet der Sessionmanager eine Übersicht der registrierten Teilnehmer und aktiven Sessions an den Client zurück (Bild 6-6).

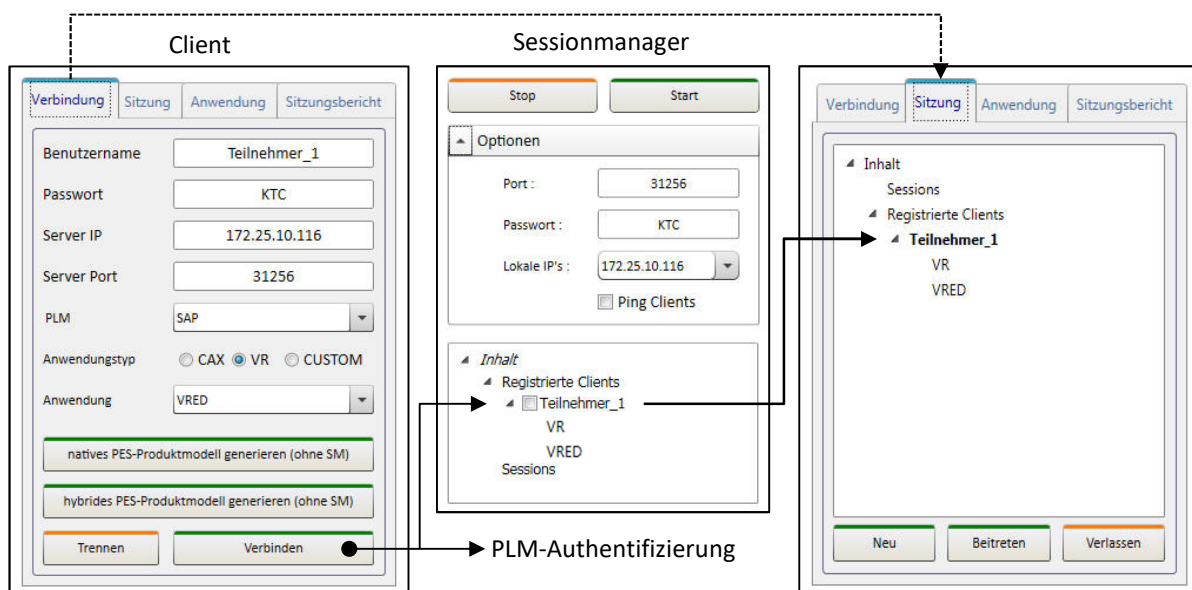


Bild 6-6: Registrierung eines Clients am Sessionmanager

Nach der Registrierung am SessionManager kann der Client eine neue Session erstellen oder einer existierenden Session beitreten (Bild 6-7). Mit Betätigung der Schaltfläche zum Erstellen einer Session wird der Benutzer aufgefordert eine SessionID und ein zugehöriges Passwort zu vergeben. Danach öffnet sich eine PDM-Suchmaske, in der der Benutzer durch Angabe von identifizierenden PDM-Metadaten den im Design Review mit VR zu bewertenden virtuellen Prototyp bestimmt. Daraufhin werden vom VR-PLM System das zuvor erstellte PES-Produktmodell sowie die zugehörigen VR-konformen Geometriebeschreibungen aus dem PDM geladen und analysiert. Im Anschluss startet das VR-PLM System die konfigurierte VR-Anwendung, koppelt diese an den Client und initialisiert das Laden des virtuellen Prototyps.

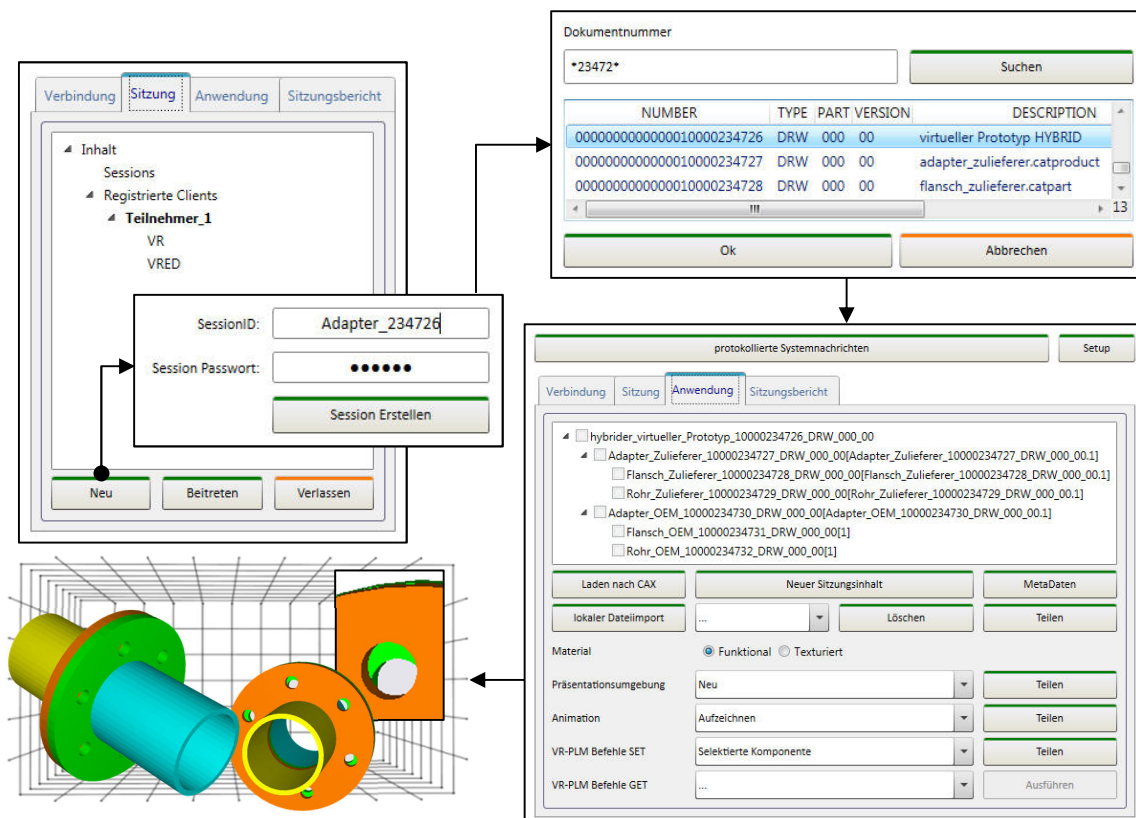


Bild 6-7: Erstellen einer Session

Nachdem die Session erstellt ist, können weitere Teilnehmer beitreten. Hierzu registrieren die Teilnehmer ihren konfigurierten Client zunächst am Sessionmanager. Mit Betätigung der Schaltfläche zum Beitreten einer Session wird der Teilnehmer aufgefordert eine der existierenden Sessions auszuwählen und das zugehörige Passwort einzugeben. Danach lädt der Client das zugehörige PES-Produktmodell aus dem Sessionmanager und die zugehörigen VR-konformen Geometriebeschreibungen aus dem PLM. Im Anschluss startet das VR-PLM System wieder die konfigurierte VR-Anwendung, koppelt diese an den Client und initialisiert das Laden des virtuellen Prototyps. In der Folge sind alle Teilnehmer und Anwendungen synchronisiert und die Bewertung des virtuellen Prototyps kann mittels der Funktionen des VR-PLM Systems durchgeführt werden.

6.3 Rückführung der Ergebnisse

Im Beispiel wird während der Bewertung festgestellt, dass die Teilkreise der Bohrungen im linken und rechten Flansch differieren. Als Folge wird eine Maßnahme zur Anpassung der Teilkreise beschlossen. Hierbei stellen sich zum Beispiel die Fragen, ob einer der Flansche noch nicht gefertigt wird und wenn ja, welcher Konstrukteur die Maßnahme umsetzen kann. Mittels der Visualisierung von PLM-Metadaten (Bild 6-8) wird auf einfache Weise ermittelt, dass die rechte Flanschbaugruppe mit der Dokumentnummer 10000234730 nicht freigegeben ist und derzeit durch die Person Konstrukteur-1 bearbeitet wird.

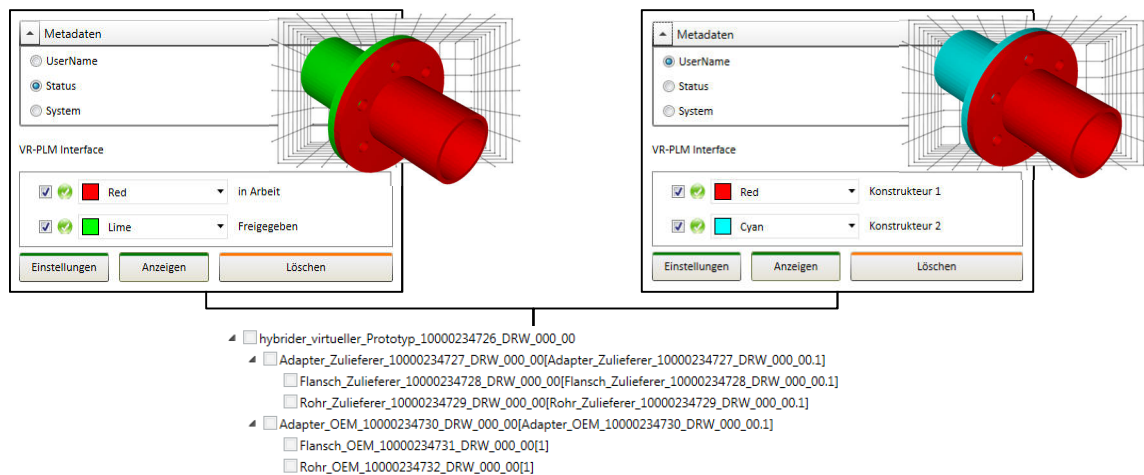


Bild 6-8: Visualisierung von PLM-Metadaten

Die Dokumentation der Maßnahme erfolgt durch den Schriftführer zur Laufzeit des Design Reviews. Der Reiter Sitzungsbericht im Client bildet hierfür das SES-Modell in Form diverser Benutzungsschnittstellen ab (Bild 6-9). Um die Dateneingabe zu beschleunigen und gleichzeitig Fehler zu vermeiden, sind Benutzungsschnittstellen die keinen Freitext im SES-Modell referenzieren durch mögliche Eingaben vorausgefüllt. Diese stammen aus dem PES-Produktmodell, SES-Modell und dem Modell der Session.

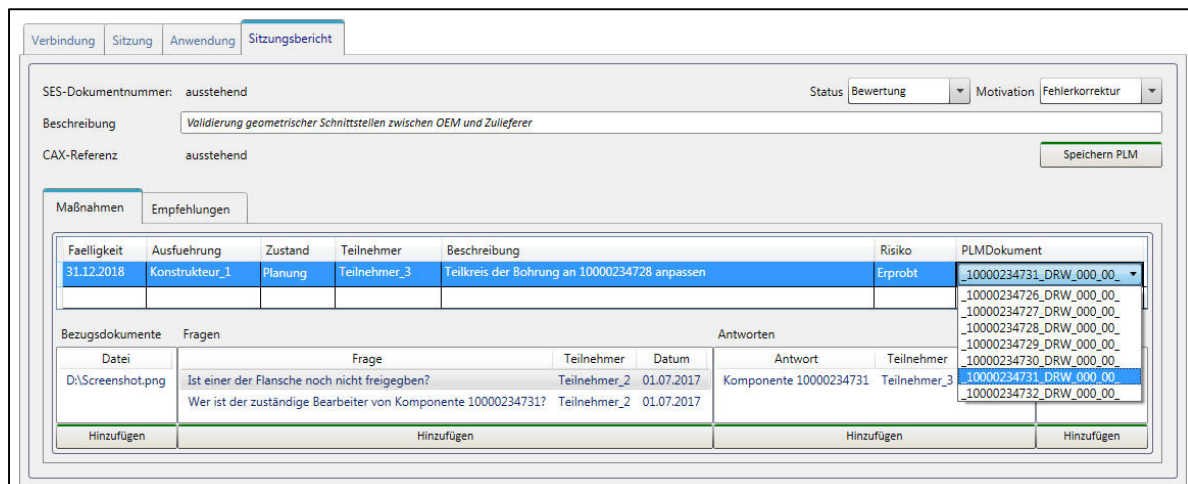


Bild 6-9: Verfassen des Sitzungsberichts

Mit dem Fortschreiten der Dokumentation wird der Sitzungsbericht stetig an alle Teilnehmer der Session verteilt. Diese können den Sitzungsbericht in ihrem Client ebenfalls einsehen, jedoch nicht verändern. Zum Abschluss des Design Reviews ist es die Aufgabe des Schriftführers, den Sitzungsbericht über die Benutzungsschnittstelle des Clients im PLM zu speichern (Bild 6-10). Ist der bewertete virtuelle Prototyp wiederholt Inhalt eines Design Reviews, wird das zuvor gespeicherte SES-Modell durch das VR-PLM System automatisch abgerufen und im Reiter Sitzungsbericht der Clients visualisiert.

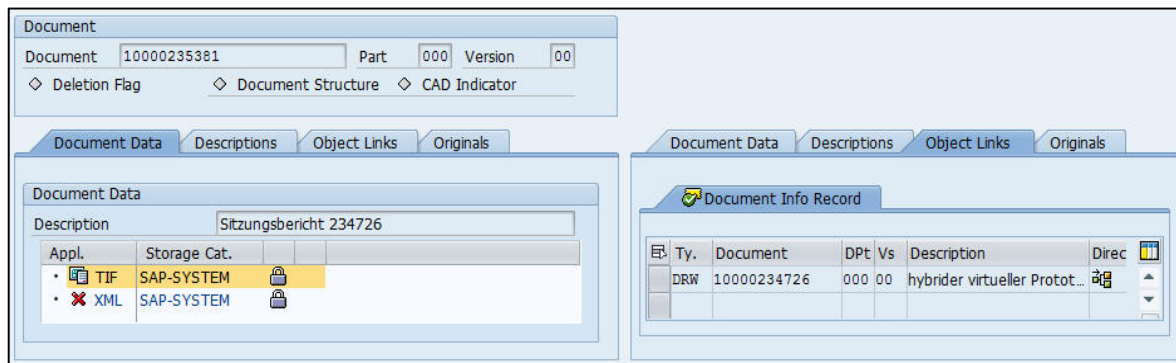


Bild 6-10: Verwaltung des SES-Modells im PLM

6.4 Umsetzung von Maßnahmen

Im Anwendungsszenario soll die Umsetzung der Maßnahme zur Laufzeit der Bewertung demonstriert werden. Hierzu wird zunächst der ermittelte zuständige Bearbeiter Konstrukteur-1 in die Session eingeladen. Er konfiguriert seinen Client derart, dass der CAX-Adapter mit dem Erzeugersystem der zu verändernden Komponente übereinstimmt. Ist der zuständige Bearbeiter der Session beigetreten, kann der Client von Konstrukteur-1 aus einem der Clients mit VR-Adapter heraus angesprochen werden. Im Client mit VR-Adapter wird dazu die ermittelte, zu bearbeitende Geometrie selektiert (10000234731) und per Kontextmenü als native Geometrie markiert. Da laut beschlossener Maßnahme der Teilkreis von Komponente 10000234731 aus CAD-System 1 an den Teilkreis von Komponente 10000234728 aus CAD-System 2 anzupassen ist, wird letztere Komponente zusätzlich als Hilfsgeometrie markiert (Bild 6-11). Durch Betätigen der Schaltfläche *Laden nach CAX* ermittelt das VR-PLM System alle integrierten Clients, die die als nativ markierte Geometrie laden können. In der Folge erscheint ein Auswahldialog, der den zuvor beigetretenen Konstrukteur-1 auflistet. Nach Auswahl und Bestätigung erscheint am Arbeitsplatz des Konstrukteurs ein Dialog, der auf einkommende Daten hinweist. Mit Bestätigung des Konstrukteurs wird im CAD-System 1 durch das VR-PLM System eine heterogene Baugruppe erzeugt, die Komponente 10000234731 nativ und Komponente 10000234728 diskret darstellt.

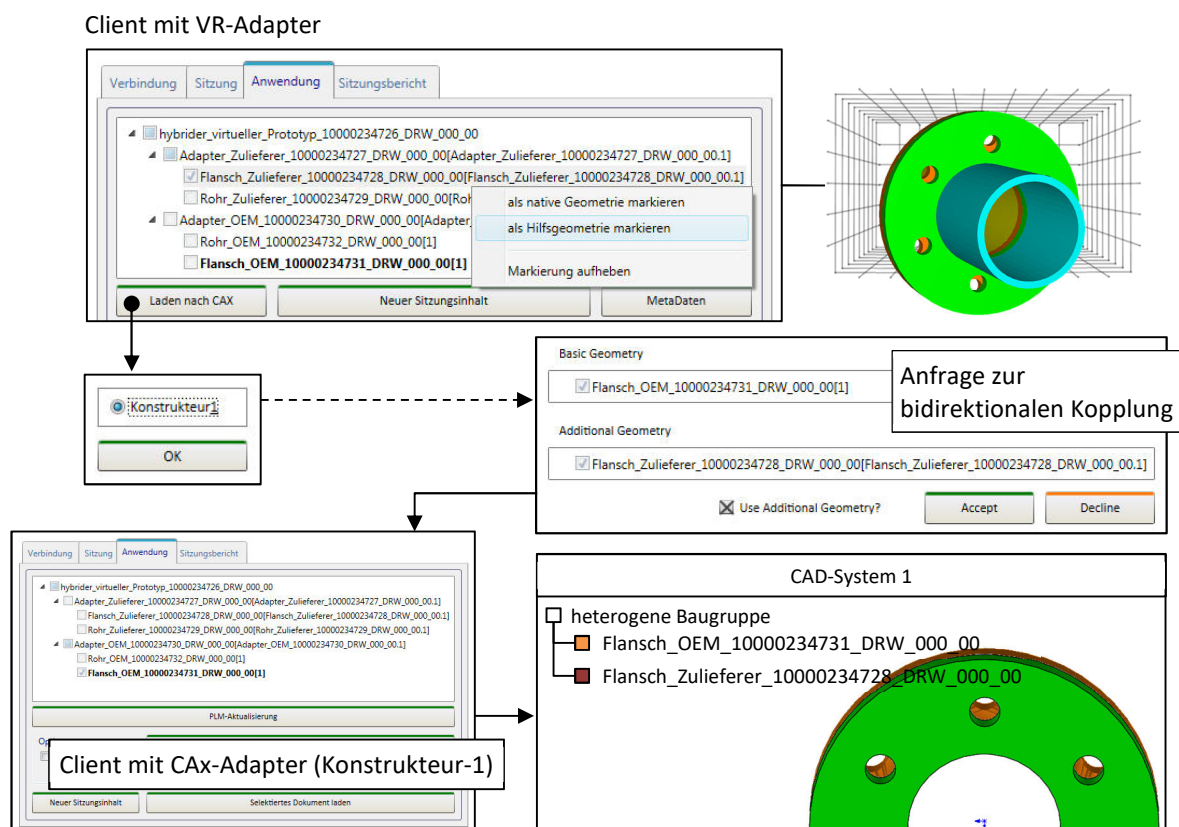


Bild 6-11: Initialisierung des CAD-Systems für Bearbeitungsaufgaben aus dem VR-System heraus

Nach der Initialisierung von CAD-System 1 beginnt der Konstrukteur den Teilkreis der Bohrungen der nativen Komponente anzupassen. Hierfür verwendet er die Funktionen des CAD-Systems. Nach beendeter Arbeit betätigt er die Schaltfläche zum Aktualisieren der VR-Szene (Bild 6-12). Daraufhin analysiert das VR-PLM System die im CAD-System geladene native Komponente, erstellt ein temporäres PES-Produktmodell und erzeugt eine VR-konforme Geometriebeschreibung. Diese ersetzen die ursprüngliche Beschreibung der nativen Komponente im PES-Produktmodell. Anschließend werden die in der Session integrierten VR-Systeme über die temporäre Änderung informiert und die neuen Daten nachgeladen.

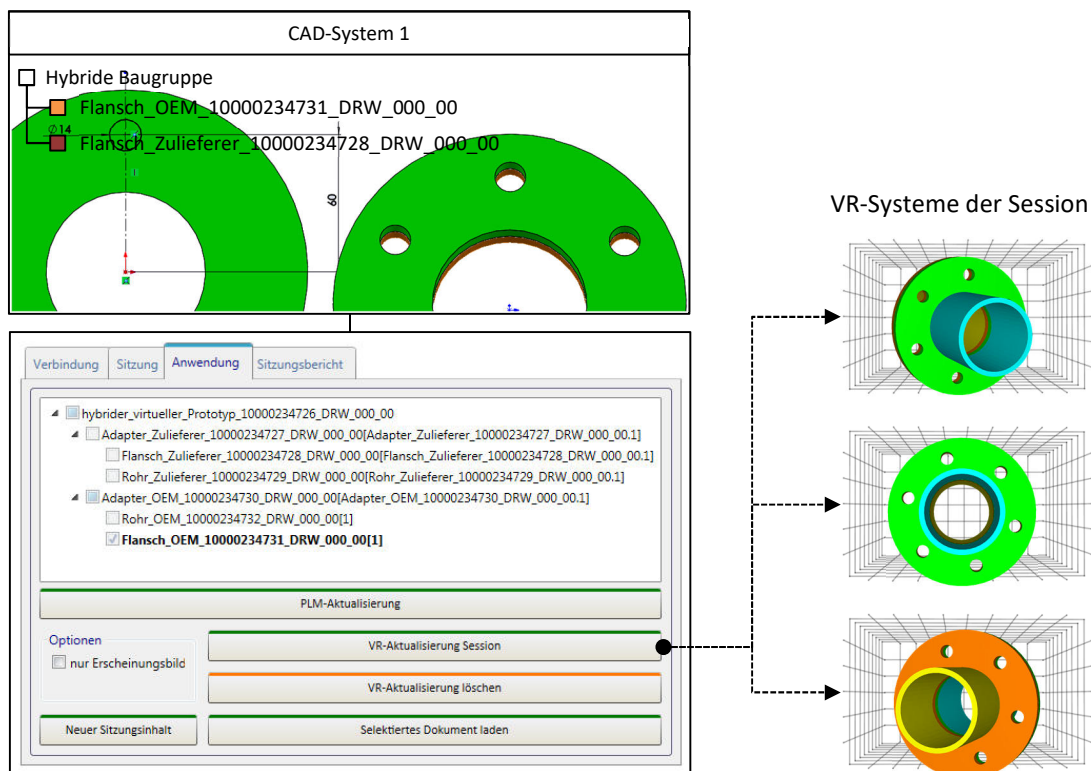


Bild 6-12: Aktualisierung der VR-Systeme nach Änderungen im Erzeugersystem

Ist die Maßnahme infolge der Änderung erfolgreich abgeschlossen, kann der Konstrukteur mittels der Schaltfläche *PLM-Aktualisierung* die bearbeitete native Komponente sowie das zugehörige temporäre PES-Produktmodell und die VR-konforme Geometriebeschreibung im PDM speichern. Der Schriftführer setzt in diesem Fall den Status der Maßnahme auf abgeschlossen.

Ist die Bearbeitung der nativen Komponente nicht zielführend gewesen, kann das PES-Produktmodell durch betätigen der Schaltfläche *VR-Aktualisierung löschen* wieder in den ursprünglichen Zustand zurückgeführt werden. In der Folge visualisieren die in der Session integrierten VR-Systeme den Zustand vor der Bearbeitung der nativen Komponente. Der beschriebene Prozess der Wechselwirkung zwischen Programmen der Anwendungs- und Präsentationsschicht kann beliebig oft erfolgen.

7 Zusammenfassung

Das in dieser Arbeit beschriebene VR-PLM Integrationskonzept stellt einen Ansatz dar, um den Prozess des Design Reviews mittels VR-Technologie zu unterstützen und in ein PLM-Konzept zu integrieren. Das Design Review kann hierdurch als ein andauernder Prozess verstanden werden, der sich entsprechend der Konzepte des Systems Engineering und der Integrierten Produktentwicklung über den gesamten Produktlebenszyklus und alle Entwicklungsdomänen erstreckt.

Die Verwendung der VR-Technologie als Schnittstelle zum PLM ermöglicht es, die verwalteten virtuellen Prototypen in beliebiger Kombination zu visualisieren (Bild 7-1). Diese existieren somit nicht mehr nebeneinander, sondern können in Form eines übergreifenden heterogenen virtuellen Prototyps in allen Fachabteilungen genutzt werden. Dieser Zustand unterstützt das Erkennen von interdisziplinären Wechselwirkungen in hohem Maße. Darüber hinaus gewährleistet die Integration der VR-Technologie in das PLM-Konzept, dass zu jedem Objekt der VR-Szene PLM-Metadaten abgerufen und geeignet visualisiert werden können. In der Folge lassen sich komplexe Modelle, deren Gesamtverhalten innerhalb von PLM-Lösungen durch relationale bzw. objektorientierte Datenbanken abgebildet ist, besser verstehen.

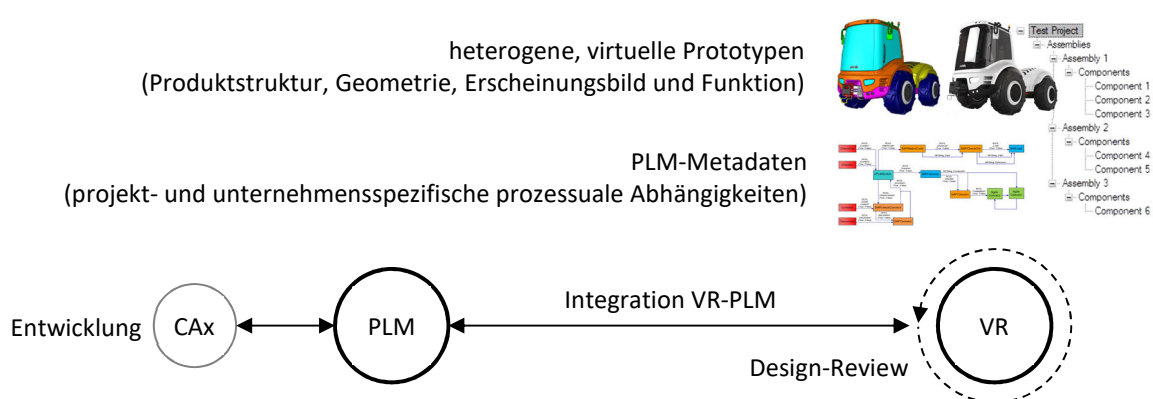


Bild 7-1: Visualisierung des PLM-Produktmodells im Design Review mit VR

Die Integration der VR-Technologie in das PLM-Konzept erfolgt anhand des in dieser Arbeit entwickelten PES-Produktmodells. Das PES-Produktmodell ist ein Datenmodell, das virtuelle Prototypen aus visueller, funktionaler und struktureller Sicht im Kontext des PDM für die Verwendung mit VR abbildet. Das PES-Produktmodell entsteht sukzessive mit der Verwaltung virtueller Prototypen im PDM. In der Folge kann die Visualisierung virtueller Prototypen jederzeit ad hoc ohne manuelle Vorarbeiten im Design Review mit VR erfolgen.

Mit der Verwaltung von identifizierenden PDM-Metadaten im PES-Produktmodell existiert jedes Objekt der VR-Szene bzw. Präsentationsschicht im Kontext der Daten- und Prozessschicht. Unter Voraussetzung der Anwendung klassischer CAx-Integrationslösungen sind somit die Datenbestände von Präsentations- und Anwendungsschicht mit der Daten- und Prozessschicht synchronisiert. Präsentations- und Anwendungsschicht sind im VR-PLM Integrationskonzept somit stets bidirektional gekoppelt. In der Folge ist eine wechselseitige Nutzung von Anwendungen der Präsentations- und Anwendungsschicht mit demselben Produktmodell gegeben. Hierdurch ist die Bewertungstätigkeit im Design Review derart unterstützt, dass Änderungen am virtuellen Prototyp in Echtzeit in die Darstellung mit VR einfließen können.

Mit der Kopplung von Präsentations- und Anwendungsschicht lassen sich die am Design Review beteiligten Anwendungen anforderungskonform einzusetzen. So steht zum Beispiel CAD für die Erstellung und Bearbeitung exakter parametrischer 3D-Geometriemodelle einschließlich funktionaler und fertigungstechnischer Produktmerkmale. VR nutzt diese Informationen, um anhand von Darstellungen in hoher Wahrnehmungsqualität die Beurteilung digitaler Produktmodelle zu verbessern. CAD- und VR-System werden in ihren Domänen wirksam belassen und nicht durch Funktionen erweitert, über die eines der jeweils anderen Systeme ohnehin verfügt.

Die konsistente Rückführung von Maßnahmen und Empfehlungen in den Produktentwicklungsprozess erfolgt anhand des in dieser Arbeit entwickelten SES-Modells. Das SES-Modell ist ein Datenmodell, das den bewerteten nativen virtuellen Prototyp innerhalb der Daten- und Prozessschicht des PLM identifiziert und die Ergebnisse des Design Reviews entsprechend DIN EN 61160 informationstechnisch abbildet. Eine redaktionelle Nachbearbeitung der gewonnenen Ergebnisse entfällt. Mittels des SES-Modells ist es möglich den Status des Design Reviews und den von einzelnen Maßnahmen im VR-PLM Integrationskonzept informationstechnisch auszuwerten und nachzuverfolgen. Durch die Verknüpfung zum PES-Produktmodell kann die dem Sitzungsbericht zugehörige VR-Szene ad hoc rekonstruiert werden.

Darüber hinaus unterstützt das VR-PLM Integrationskonzept die Kollaboration, indem neben der Verwendung heterogener virtueller Prototypen der Einsatz von VR im Sinne einer virtuellen, verteilten Umgebung erfolgt. Das bedeutet, die Durchführung des Design Reviews ist unabhängig vom Aufenthaltsort der Teilnehmer. Die Administration der teambasierten Planungs-, Koordinations- und Beurteilungsaufgaben erfolgt anhand der existierenden Authentifizierungs- und Autorisierungskonzepte der integrierten PLM-Lösung. Hierdurch ist auch die PLM-Lösung anwendungskonform eingesetzt.

Zusammengefasst beschreibt das VR-PLM Integrationskonzept eine VR-basierte Benutzungsschnittstelle zum PLM, mittels der Entwicklungsarbeiten am virtuellen Prototyp im Rahmen der Unternehmensprozesse bewertet und in iterativen Abläufen organisiert werden können. Grundlage hierfür sind das im Rahmen dieser Arbeit entwickelte PES-Produktmodell und SES-Modell, mittels derer die VR-Szene sowie die Ergebnisse des Design Reviews konsistent zum und im PLM beschrieben sind. Infolge der stetigen Verfügbarkeit des PES-Produktmodells entfällt der in DIN EN 61160 beschriebene Teilprozess der Modellaufbereitung. Hierdurch kann das Design Review als andauernder Prozess innerhalb der Produktentwicklung verstanden werden, die bis zur abschließenden Lösung Modelle iterativ generiert, bewertet und bearbeitet (Bild 7-2).

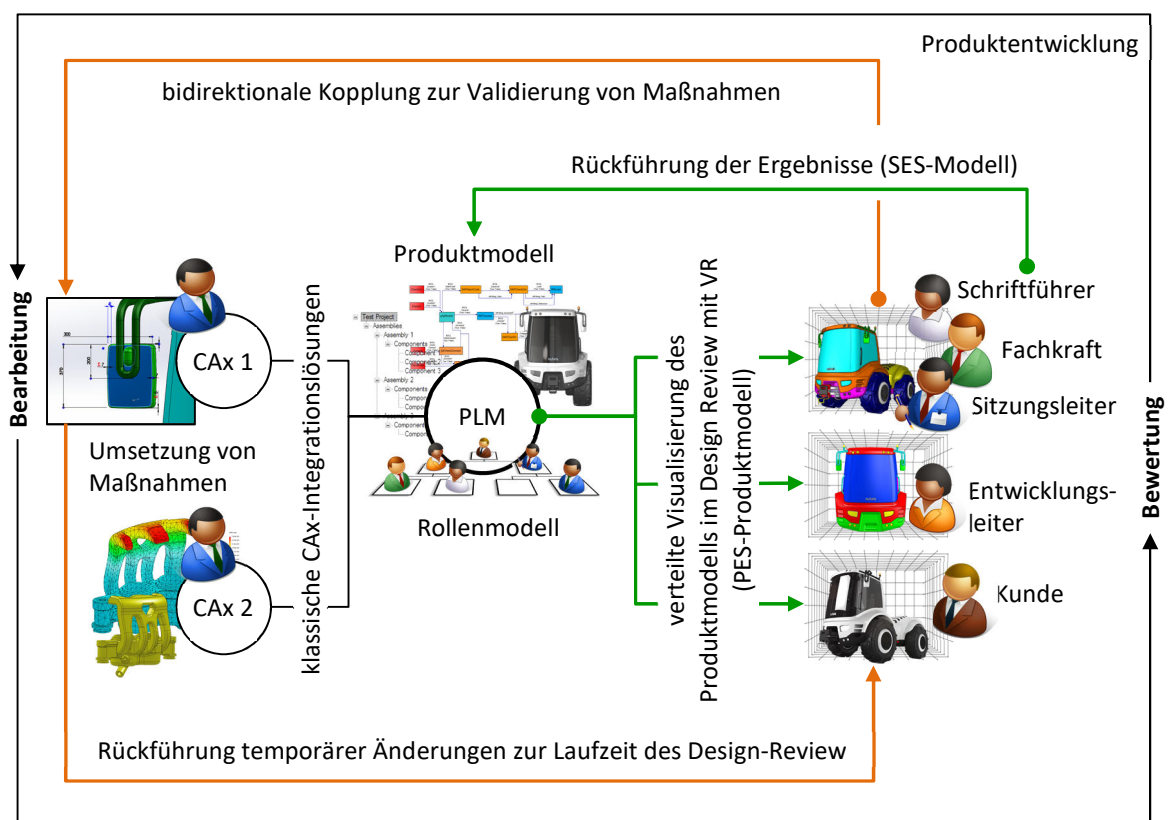


Bild 7-2: Integration des Design Reviews mit VR im Produktentwicklungsprozesses

8 Literaturverzeichnis

- [Abt15] D. Abts : Masterkurs Client/Server-Programmierung mit Java: Anwendungen entwickeln mit Standard-Technologien, Springer-Verlag, 2015
- [AHO06] J. C. Aurich; H. Hagen; D. Ostermayer; M. Bertram : VR-unterstützter KVP-Workshop, Neues Anwendungsfeld des Virtual Engineering - In: wt Werkstattstechnik online Ausgabe 1/2-2006, Springer-VDI-Verlag, 2006
- [Ant15] R. Anthony : Systems Programming: Designing and Developing Distributed Applications, Morgan Kaufmann, 2015
- [Bal11] H. Balzert : Lehrbuch der Objektmodellierung: Analyse und Entwurf mit der UML 2, 2. Auflage, Spektrum Akademischer Verlag, 2011
- [BFS11] R. Beckers; A. Fröhlich; J. Stjepandic : Entwicklungsstand genormter Visualisierungsformate in der Produktentwicklung - In: In: 10. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung, 2011
- [BGW11] U. Bracht; D. Geckler; S. Wenzel : Digitale Fabrik - Methoden und Praxisbeispiele, Springer-Verlag Berlin Heidelberg, 2011
- [BrB16] E. J. Braude; M. E. Bernstein : Software Engineering: Modern Approaches, 2. Auflage, Waveland Press, 2016
- [Bur06] G. C. Burdea : Virtual Reality Technology , 2. Auflage, Wiley India Pvt. Limited, 2006
- [CoW93] M. F. Cohen; J. R. Wallace : Radiosity and Realistic Image Synthesis, Elsevier, 1993
- [CrD16] T. Critchlow; K. K. van Dam : Data-Intensive Science, CRC Press, 2016
- [CuC08] G. Curtis; D. Cobham : Business Information Systems - Analysis, Design and Practice, 6. Auflage, Financial Times Prent.Int, 2008
- [DBB16] P. Dutre; P. Bekaert; K. Bala : Advanced Global Illumination, 2. Auflage, CRC Press, 2016

- [DBG14] R. Dörner; W. Broll; P. Grimm; B. Jung : Virtual und Augmented Reality (VR/AR) - Grundlagen und Methoden der Virtuellen und Augmentierten Realität, Springer-Verlag, 2014
- [DIN16792] DIN ISO 16792 : Technische Produktdokumentation - Verfahren für digitale Produktdefinitionsdaten, Beuth Verlag GmbH, 2008
- [DIN199] DIN 199 : CAD-Modelle, Zeichnungen und Stücklisten, Beuth Verlag GmbH, 2002
- [DIN8855] DIN ISO 8855 : Straßenfahrzeuge - Fahrzeugdynamik und Fahrverhalten , Beuth Verlag GmbH, 2013
- [DIN9300] DIN 9300 : Luft- und Raumfahrt; Begriffe, Größen und Formelzeichen der Flugmechanik; Bewegung des Luftfahrzeuges gegenüber der Luft, Beuth Verlag GmbH, 1990
- [Eck14] C. Eckert : IT-Sicherheit: Konzepte - Verfahren - Protokolle, Walter de Gruyter GmbH & Co KG, 2014
- [Ehr09] K. Ehrlenspiel : Integrierte Produktentwicklung, 4. Auflage, Springer Verlag Berlin Heidelberg, 2009
- [EiS09] M. Eigner; R. Stelzer : Product Lifecycle Management, 2. Auflage, Springer Verlag, 2009
- [ESB16] H. Ernst; J. Schmidt; G. Beneken : Grundkurs Informatik. Grundlagen und Konzepte für die erfolgreiche IT-Praxis - Eine umfassende, praxisorientierte Einführung, 6. Auflage, Springer Vieweg, 2016
- [FLL11] A. Friedewald; H. Lödding; U. Freiherr von Lukas; B. Mesing; M. Roth; S. Schleusener; F. Titov : Benchmark neutraler Formate für den prozessübergreifenden Datenaustausch im Schiffbau, Fraunhofer Berichtsnummer: 11rp001-FIGDR, 2011
- [FuP16] H. Fujita; G. A. Papadopoulos : New Trends in Software Methodologies, Tools and Techniques: Proceedings of the Fifteenth SoMeT 16, IOS Press, 2016
- [GBS02] H. Graf; G. Brunetti; A. Stork : A Methodology Supporting the Preparation of 3D-CAD Data for Design Reviews in VR - In: Proceedings of DESIGN 2002, the 7th International Design Conference, Marjanovic D. (Ed.), 2002

- [Gei15] M. Geirhos : Entwurfsmuster: Das umfassende Handbuch , 1. Auflage, Rheinwerk Computing, 2015
- [GrS05] H. Graf; A. Stork : Bridging the Gap between CAD/CAE and VR and Incorporating Semantics into the Analysis Process - In: ProSTEP iViP Association: ProSTEP iViP Science Days 2005, 2005
- [GWB07] T., Wigdor, D., Balakrishnan, R. Grossman : Exploring and reducing the effects of orientation on text readability in volumetric displays - In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM New York, 2007
- [HeH94] E. Heineken; T. Habermann : Lernpsychologie für den beruflichen Alltag, 3. Auflage, Sauer-Verlag, 1994
- [HeM06] E. Heidling; P. Meil : Entfernung überbrücken - Verteilte Arbeit entlang internationaler Wertschöpfungsketten. Neue Anforderungen in grenzüberschreitenden Arbeitsprozessen - In: Von der Allgegenwart der verschwindenden Arbeit. Herausforderungen für die Arbeitsforschung, Edition Sigma, 2006
- [HSW08] S. Husung; M. Sander; C. Weber; G. Höhne : Optimierung der CAD-VR Datenaufbereitung zur Erzeugung funktionsrelevanter Modellstrukturen - In: Tagungsband zum 7. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung, HNI-Verlagsschriftenreihe Band 232, 2008
- [JeS10] K. Jenewein; M. Schenk : Virtuelle Realität in der technischen Aus- und Weiterbildung - Gegenstandsbestimmung und Umsetzungsbeispiele - IBBP Arbeitsbericht Nr. 74, Hrsg: Institut für Berufs- und Betriebspädagogik (IBBP) der Otto-von-Guerike-Universität Magdeburg Prof.'en Dr. Dietmar Frommberger, Dr. Klaus Jenewein, Dr. Sibylle Peters, 2010
- [KaH08] A. Katzenbach; S. Haasis : Virtual and Mixed Reality in a SOA based engineering environment - In: Design Synthesis, CIRP Design Conference, Enschede, 2008
- [KFG07] F.-L. Krause; H.-J. Franke; J. Gausemeier : Innovationspotenziale in der Produktentwicklung, Hanser München, 2007
- [Kla10] F. Klawonn : Grundkurs Computergrafik mit Java, 3. Auflage, Vieweg+Teubner Verlag, 2010

- [Kun12] S. Kunst : Wirtschaftlichkeit der Virtual Reality Technologie - Grundlagen, Modell, Realisierung, AV Akademikerverlag, 2012
- [Lin11] M. Lindstrom : Brand Sense: Warum wir starke Marken fühlen, riechen, schmecken, hören und sehen können, Campus Verlag, 2011
- [LKM17] J. J. LaViola Jr.; E. Kruijff; R. P. McMahan; D. A. Bowman; I. Poupyrev : 3D User Interfaces: Theory and Practice, 2. Auflage, Addison-Wesley Professional, 2017
- [LLR13] L. Lucas; C. Loscos; Y. Rémon : 3D Video: From Capture to Diffusion, John Wiley & Sons, 2013
- [Mey07] M. Meywerk : CAE-Methoden in der Fahrzeugtechnik, Springer-Verlag Berlin Heidelberg, 2007
- [Mil11] J. L. Miller : Distributed virtual environment scalability and security, University of Cambridge Computer Laboratory, 2011
- [MuW15] H. Müller; F. Weichert : Vorkurs Informatik: Der Einstieg ins Informatikstudium, 4. Auflage, Springer-Verlag, 2015
- [MZF10] L. Ma; W. Zhang; H. Fu; Y. Guo; D. Chablat; F. Bennis; A. Sawanoi; N. Fugiwara : A framework for interactive work design based on motion tracking, simulation, and analysis - In: Human Factors and Ergonomics in Manufacturing & Service Industries Vol20(4), John Wiley & Sons, 2010
- [NFH12] A. Nischwitz; M. Fischer; P. Haberäcker; G. Socher : Computergrafik und Bildverarbeitung: Band I: Computergrafik, 3. Auflage, Springer-Verlag, 2012
- [NFH13] A. Nischwitz; M. Fischer; P. Haberäcker; G. Socher : Computergrafik und Bildverarbeitung - Band I und II, 3. Auflage, Springer Vieweg, 2013
- [NWG10] R. Neugebauer; V. Wittstock; J. Glänzel; M. Schumann; A. Meyer : Adaptive Finite-Element-Methode zur Lösung spezieller Auslegungsprobleme des Maschinebaus und zur Erweiterung von Virtual Reality-Technologien - In: Tagungsband zum 9. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung, HNI-Verlagsschriftenreihe 274, 2010
- [OMG15] Object Management Group. (2015) OMG Unified Modeling Language - Version 2.5. PDF-Dokument [Online]: <http://www.omg.org/spec/UML/2.5/PDF>

- [Opp10] U. Oppolzer : Super lernen: Tipps & Tricks von A-Z Effektiver Lernen Mit vielen Übungen, Schlütersche, 2010
- [Ovt05] J. Ovtcharova : Life cycle Engineering Konzepte, ICIDO Summit "Next Generation VR", 2005
- [PBF13] G. Pahl; W. Beitz; J. Feldhusen; K.-H. Grote : Konstruktionslehre - Methoden und Anwendung erfolgreicher Produktentwicklung, 8. Auflage, Springer Berlin Heidelberg, 2013
- [PSS11] D. Petermann; B. Saske; R. Stelzer : Der virtuelle Integrationsraum - In: Virtuelle Realität, 8. Workshop der GI-Fachgruppe VR/AR, Shaker Verlag, 2011
- [PSS13] D. Petermann; P. Sembdner; W. Steger; R. Stelzer : Fotorealistische Review-Daten im Produktentwicklungsprozess - In: Tagungsband zum 11. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung, HNI-Verlagsschriftenreihe Band 311, 2013
- [RDK13] M. H. Rademacher; C. Dabs; Krömker H. : Fertigungstoleranzen in der Automobilindustrie - In: Tagungsband zum 11. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung, HNI-Verlagsschriftenreihe Band 311, 2013
- [ReP06] P. Rechenberg; Pomberger G. : Informatik-Handbuch , 4. Auflage, Carl Hanser Verlag, 2006
- [ReW12] I. Rehfeld; Wunderlich, J. : Virtual Reality und Product Lifecycle Management – Entwicklung eines durchgängigen Prozesses für die BSH Bosch und Siemens Hausgeräte GmbH - In: Entwerfen Entwickeln Erleben - Methoden und Werkzeuge in der Produktentwicklung, 10. Gemeinsames Kolloquium Konstruktionstechnik (KT2012) Ralph Stelzer, Karl-Heinrich Grote, Klaus Brökel Frank Rieg, Jörg Feldhusen (Hrsg.), 2012
- [Rhe91] H. Rheingold : Virtual Reality, Summit Books, 1991
- [SBD10] D. Spath; R. Blach; M. Danglmaier; G. Varalda : Anforderungen und Spezifikation für ein projektionsbasiertes System für den Mehrbenutzer-Design-Review - In: 9. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung, HNI-Verlagsschriftenreihe 274, 2010

- [Sch12] H.-G. Schiele : Computergrafik für Ingenieure: Eine anwendungsorientierte Einführung, Springer-Verlag, 2012
- [Sch14] T. Schlegel : Multi-Touch: Interaktion durch Berührung, Springer-Verlag, 2014
- [ScM13] H. Schumann; W. Müller : Visualisierung: Grundlagen und allgemeine methoden, Springer-Verlag, 2013
- [ScP15] R. Schmitt; T. Pfeifer : Qualitätsmanagement: Strategien - Methoden - Techniken, 5. Auflage, Carl Hanser Verlag GmbH & Co. KG, 2015
- [SeA16] M. Segal; K. Akeley. (2016) The OpenGL Graphics System: A Specification - Version 4.5 (Core Profile). PDF-Dokument [Online]: <https://www.khronos.org/registry/OpenGL/specs/gl/glspec45.core.pdf>
- [She00] T. B. Sheridan : Interaction, imagination and immersion some research needs - In: Proceedings of the ACM symposium on Virtual reality software and technology, ACM, 2000
- [SKK12] M. Schenk; C. Karthaus; W. Klos; H. Binz; B. Bertsche : Effizienzsteigerung bei der Antriebsstrangerprobung durch Einsatz moderner Erprobungsmethoden und Optimierung der Fehleranalyse - In: VDI-Berichte 2169; SIMVEC Berechnung, Simulation und Erprobung im Fahrzeugbau, VDI Verlag, 2012
- [SKW06] A. Schilling; S. Kim; Weissmann, D. ; Z. Tang; S. Choi : CAD-VR geometry and meta data synchronization - In: Journal of Zhejiang University SCIENCE A, 2006
- [SpK97] G. Spur; F.-L. Krause : Das virtuelle Produkt - Management der CAD-Technik, Carl Hanser Verlag München Wien, 1997
- [SPS09A] R. Stelzer; D. Petermann; B. Saske; W. Steger : Kollaborationsumgebung in einer heterogenen PDM-CAD-VR Systemlandschaft - In: Tagungsband zum 8. Paderborner Workshop AR/VR in der Produktentstehung, 252, HNI-Verlagsschriftenreihe Band, 2009
- [SPS09B] R. Stelzer; D. Petermann; B. Saske; W. Steger : Kollaborationsumgebung für CAD-VR - In: Zeitschrift für wirtschaftlichen Fabrikbetrieb. Ausgabe 11/2009, Carl Hanser Verlag GmbH & Co. KG, 2009

- [SPS10] R. Stelzer; D. Petermann; B. Saske : Dynamische Modellierung unternehmensübergreifender Prozesse - In: PROSTEP Produktdatenjournal Ausgabe 01/2010, 2010
- [SSP11] R. Stelzer; W. Steger; D. Petermann : Hantieren mit realen Objekten in virtuellen Umgebungen - In: Tagungsband zum 10. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung, HNI-Verlagsschriftenreihe Band 295, 2011
- [SSS10] H. R., Sembdner, P., Steger, W. Stelzer : Verknüpfung komplexer und heterogener Baugruppenmodelle in VR - In: Tagungsband zum 9. Paderborner Workshop AR/VR in der Produktentstehung, HNI-Verlagsschriftenreihe Band 274, 2010
- [StW13] R. Stark; L. Wolter : Einfache und sichere AdHoc Kollaboration in Design Reviews - In: Newsletter Wissenschaftliche Gesellschaft für Produktentwicklung WiGeP, Ausgabe 1, 2013
- [Suf16] K. Suffern : Ray Tracing from the Ground up, CRC Press, 2016
- [VDC13] VDC - Virtual Dimension Center. (2013) Whitepaper Collaborative Virtual Engineering – Techniken, Prozessen, Nutzen. PDF-Dokument [Online]: <http://www.vdc-fellbach.de/files/Whitepaper/2013%20VDC-Whitepaper%20Collaborative%20Virtual%20Engineering.pdf>
- [VDI2221] VDI-Richtlinie 2221 : Methodik zum Entwickeln und Konstruieren Technischer Systeme und Produkte, Beuth Verlag GmbH, 1993
- [VDI3633] VDI-Richtlinie 3633 : Simulation von Logistik-, Materialfluss- und Produktionssystemen , Beuth Verlag GmbH, 2013
- [VDI4499] VDI-Richtlinie 4499 : Digitale Fabrik, Beuth Verlag GmbH, 2008
- [VDI5600] VDI-Richtlinie 5600 : Fertigungsmanagementsysteme, Beuth Verlag GmbH, 2007
- [VSG11] S. Vornholt; M. Stoye; I. Geist; V. Köppen; G. Saake : Datenmodell zur flexiblen Verwaltung von Datenaustauschprozessen in der virtuellen Produktentwicklung - In: Roland Kasper u.a., editor, 10. Magdeburger Maschinenbau-Tage 27.-29.09.2011, Magdeburg, September 2011. Otto-von-Guericke-Universität Magdeburg

- [WaB13] H.-J. Warnecke; H.-J. Bullinger : Kunststück Innovation: Praxisbeispiele aus der Fraunhofer-Gesellschaft, Springer-Verlag, 2013
- [WaS07] J. Wagner; K. Schwarzenbacher : Föderative Unternehmensprozesse: Technologien, Standards und Perspektiven für vernetzte Systeme, John Wiley & Sons, 2007
- [WLW10] Q.-H. Wang; J.-R. Li; B.-L. Wu; X.-M. Zhang : Live Parametric Design modifications in CAD-linked virtual environment - In: International Journal of Manufacturing Technology, 2010
- [Woe09] U. Woessner : Virtuelle und hybride Prototypen in kooperativen Arbeitsumgebungen, Höchstleistungsrechenzentrum Universität Stuttgart (HLRS), Fakultät Energie-, Verfahrens- und Biotechnik, Dissertation 2009
- [YGK08] S. E. Yoon; E. Gobbetti; D. Kasik; D. Manocha : Real-time massive model rendering - In: Synthesis Lectures on Computer Graphics and Animation Vol.7, Morgan and Claypool Publishers , 2008
- [Zer00] P. Zerbe. (2000, Juni) DIE WELT [Online]: <https://www.welt.de/print-welt/article520813/Airbus-Jeder-Jet-besteht-aus-ueber-drei-Millionen-Teilen.html>