

Datentechnische Integration und Visualisierung von Anforderungen innerhalb von CAD-Systemen

Tim Sadek, Daniel Meuris

Lehrstuhl für Maschinenelemente und Konstruktionslehre
Ruhr-Universität Bochum
Universitätsstr. 150
44801 Bochum
sadek@lmk.rub.de
meuris@lmk.rub.de

Abstract: Ein wesentliches Ziel der Produktentwicklungsprozesse ist die fortlaufende Reduzierung der Komplexität ohne wichtige Merkmale aus den Augen zu verlieren. Weiterhin zeichnet sich die virtuelle Produktentwicklung durch eine durchgängige Integration verschiedener Produktinformationen und Produktmodelle aus, wobei dem Gestaltmodell in Form von CAD-Daten eine zentrale Bedeutung zukommt. Dabei basieren diese CAD-Systeme jedoch nicht oder nur zum Teil auf den Grundprinzipien der Konstruktionsmethodik, so dass die vor dem Entwurf entstandenen Informationen nur schwer mit dem Gestaltmodell verknüpft werden können. Am Beispiel der Anforderungsentwicklung wird in dieser Arbeit ein CAD-integriertes Werkzeug vorgestellt, das es dem Konstrukteur ermöglicht, Anforderungen direkt im CAD-System zu erfassen und mit dem Geometrie-Modell zu verknüpfen. Neben der datentechnischen Integration wird auch eine Visualisierung von Anforderungen im CAD-System erstellt. Diese unterstützt den Konstrukteur, um die Vielzahl von abstrakten Informationen zu Baugruppen, Bauteilen und Features zuordnen zu können, was die Qualität und Konsistenz der erzeugten Modelle erhöht.

1 Einleitung

Die Art und Weise, wie eine technische Problemstellung in eine Lösung überführt werden kann, wird mit den Hilfsmitteln der Konstruktionsmethodik beschrieben. Kerngedanke ist dabei die Gliederung des gesamten Entwicklungsprozesses in mehrere Teilphasen, deren Arbeitsergebnisse zum Ziel haben, das zukünftige Produkt bei steigendem Konkretisierungsgrad zu beschreiben. Die *Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte* VDI 2221 definiert diesen Vorgang wie folgt:

„Die Produktentwicklung (Entwickeln und Konstruieren) umfasst die Gesamtheit aller Tätigkeiten mit denen – ausgehend von einer Aufgabenstellung – eine Lösung und die zur Herstellung und Nutzung benötigten Informationen erarbeitet werden. [...] Sie

schließen die gedankliche Zusammensetzung der einzelnen Funktionen und Teile eines Produktes, den Aufbau zu einem Ganzen und das Festlegen aller Eigenschaften ein.“ [VDI93]

Am Ende dieser Prozesse steht ein realer oder virtueller Prototyp¹, der verschiedene Rückschlüsse auf das Verhalten des realen Produktes zulässt. Insbesondere die virtuelle Entwicklung von Produktmodellen wurde bedingt die großen Fortschritte der Informations- und Kommunikationstechnik in den letzten Jahrzehnten massiv weiterentwickelt. Heute existiert unter dem Sammelbegriff der CAx-Technologie² eine Vielzahl von problem- und lösungsspezifischer Entwurfswerkzeuge, die den Entwickler bei Analyse- und Synthesaufgaben unterstützen. Problematisch ist aber, dass sich der Einsatz dieser Werkzeuge bislang auf die späten Entwicklungsphasen, die durch die Gestalt eines Produktes in ihren verschiedenen Ausprägungen dominiert werden, erstreckt [Con08]. WEBER formuliert diesen Umstand wie folgt: „Existing TTS concepts have not shown any impact on CAx-Development“ [Web09] Demzufolge ist es bis heute ein Problem abstrakte Informationen aus den frühen Phasen sowie die Entwicklungsprozesse in einem einheitlichen, rechnerbasierten Systemmodell zu integrieren, weil die verfügbaren Softwarewerkzeuge bislang nur auf bauteil- und geometrieorientierten Modellen und nicht explizit auf den Grundlagen einer Konstruktionsmethodik basieren. Dabei sollte gerade die Gestaltung eines Produktes permanent an Informationen aus den frühen Phasen gespiegelt werden können, um einerseits grundlegende Fehler zu vermeiden und andererseits den Kommunikationsaufwand, der durch ein fortwährendes, manuelles Abgleichen von Geometriemerkmalen und technischen Anforderungen entsteht, zu verringern. Die mitunter wichtigsten Informationen, die in frühen Entwicklungsphasen gesammelt werden, sind in den Produkthanforderungen zu sehen, da sie alle Rahmenbedingungen, Restriktionen und geforderten Eigenschaften des zukünftigen Produktes beschreiben [Ehr06]. Dieser enormen Bedeutung werden jedoch die momentan verfügbaren kommerziellen CAD-Systeme nicht gerecht, da sie keine Werkzeuge bereitstellen, die eine zentrale Verwaltung von Produkthanforderungen erlauben. Deshalb soll in dieser Arbeit eine prototypische, werkzeuggestützte Integration der Abbildung von technischen Anforderungen in CAD-Systemen vorgestellt werden. Dies soll zum Einen als Kommunikationsbasis für den Konstrukteur dienen und zum Anderen helfen, die Schnittstellenprobleme zwischen den einzelnen Entwicklungsphasen zu beherrschen.

Dabei sollen folgende Fragestellungen beantwortet werden:

- Wie können die produktweiten Abhängigkeiten von Anforderungen in einem CAD-System verwaltet werden?
- Wie kann das produktspezifische Wissen aus den frühen Entwicklungsphasen im CAD-System modelliert und visualisiert werden?

¹ Diese Arbeit befasst sich vordergründig mit virtuellen Prototypen, die im Folgenden mit dem Begriff „Produktmodell“ bezeichnet werden.

² Beispiele solcher Systeme: CAM – Computer Aided Manufacturing, CAE – Computer Aided Engineering, CASE – Computer Aided Software Engineering und CAD – Computer Aided Design

Nach den einleitenden Bemerkungen in diesem Kapitel werden zunächst die Begriffe der virtuellen Produktentwicklung und der technischen Anforderungsentwicklung geklärt. Aufbauend auf diesen Erläuterungen wird dann der Integrationsansatz vorgestellt, der neben der informationstechnischen Grundlage auch das methodische Fundament für eine Integration von Anforderungen in ein CAD-System legt. Schließlich wird der „EPI-Editor“³ als prototypisches Werkzeug vorgestellt und am Beispiel der Konstruktion eines Planetengetriebes evaluiert.

2 Virtuelle Produktentwicklung

Die klassische Produktentwicklung wird durch die zunehmende Unterstützung von Software-Systemen im Produktentwicklungsprozess immer mehr durch die virtuelle Produktentwicklung ersetzt [Gho08]. Virtuelle Produktentwicklung umfasst somit „*im weitesten Sinne alle rechnerunterstützten Entwicklungsprozesse, einschließlich der Konzipierung, Konstruktion, sowie Simulation der Entstehung und des Verhaltens zukünftiger Produkte*“ [Abr07]. Ein wichtiger Begriff im Zusammenhang der virtuellen Produktentwicklung ist der Modellbegriff. Ein Modell ist ein Abbild realer oder künstlicher Objekte, das die für einen Anwendungsfall relevanten Informationen beschreibt. Diese Zuordnung ist disjunkt und „*wird nur für die modellbenutzenden Subjekte beschränkt auf festgelegte Zeitintervalle im Kontext gedanklicher oder tatsächlicher Operationen hergestellt*“ [Sta73]. Demzufolge ist das Produktmodell eine schematische Sammlung von Informationen, die das reale Produkt so beschreiben, dass Rückschlüsse auf das spätere Verhalten möglich werden [Suh93].

2.1 Partialmodelle der virtuellen Produktentwicklung

Bedingt durch die Komplexität heutiger Produkte und ihrer Entwicklungsprozesse kann ein Produktmodell nicht als ganzheitliche Einheit betrachtet werden. Vielmehr zerfällt es in Partialmodelle, die sich an den einzelnen Entwicklungsphasen orientieren [Lin08]. Abbildung 1 stellt diese Partialmodelle und die Tätigkeiten, die zu ihrer Erstellung führen, da.

³ Early-Phase-Integration Editor

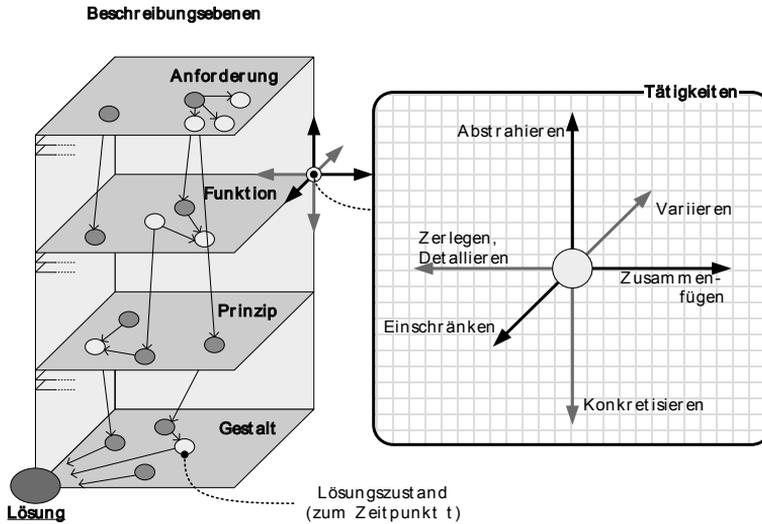


Abbildung 1: Die vier wesentlichen Partialmodelle der Produktentwicklung [Lin08]

Das **Anforderungsmodell** enthält die Anforderungen an das zukünftige Produkt, sowie alle Rahmenbedingungen der Entwicklung. Die Modellelemente des Anforderungsmodells sind Texte, Skizzen und technische Daten, sogenannte Spezifikationen. Aufgrund der Heterogenität der Informationen besitzt das Anforderungsmodell meistens einen informellen Charakter, was dessen Integration in die übrigen Partialmodelle erschwert. Das **Funktionsmodell** beschreibt die Art und Weise des funktionalen Systemzusammenhangs in Form von funktionalen Strukturen. Die Modellelemente des Funktionsmodells sind Funktionen, als Subjekt-Verb-Kombinationen und Relationen, die in Form von zwei-dimensionalen Graphen dargestellt werden. Das **Prinzipmodell** bildet die physikalische Wirkstruktur eines Produktes ab. Dieses Modell wird durch Texte, Skizzen und technischen Daten beschrieben, wobei die Skizzen oftmals einer formalen Notation unterliegen. Beispielsweise gibt es allgemein akzeptierte Notationen für die Skizzierung der Wirkungsweise von getriebetechnischen Systemen oder elektrotechnischen Wirkzusammenhängen. Das **Gestaltmodell** enthält die Geometrie und Struktur des Produktes. Hier werden Bauteile und Baugruppen bzw. deren Beziehungen und Bestandteile durch geometrische Körper beschrieben. Weiterhin bildet das Gestaltmodell die Informationsbasis für nachfolgende Fertigungsprozesse, da aus der Gestalt eines Produktes die notwendigen Fertigungsmittel und -prozesse abgeleitet werden können.

2.2 CAx-Technologie

Aufgabe der virtuellen Produktentwicklung ist nun die Bereitstellung von rechnerinternen Beschreibungen und methodischer Unterstützung bei der Entwicklung dieser Partialmodelle. Insbesondere die zunehmende Komplexität von Produkten erfordert den intensiven Einsatz von CAx-Technologien, da sich dadurch „*signifikant erweiterte Möglichkeiten zur Produktmodellierung, die damit auch zu neuen Formen*

methodischer Vorgehensmodelle in der Produktentwicklung führen“[Vaj08], ergeben. Dabei ist gerade im Maschinenbau die CAD-Methodik ein zentraler Baustein und Treiber für neue Innovationen [Szi01].

Heutige Systeme bilden die Geometrie parametrisch ab. Unter Verwendung von parametrischen Beschreibungen der Elemente wird es möglich, im Nachhinein das Modell durch Variation der Parameter und der damit verknüpften mathematischen Zusammenhänge zu verändern bzw. zu steuern. Die Verwendung von sogenannten Features erlaubt neben der Integration von Modellparametern auch die Einbringung von semantischen Informationen im Geometriemodell, da sie im Zusammenhang mit der CAD-Technologie als Aggregation von Geometrie und Semantik gesehen werden kann [Spu97]. WEBER definiert Features als „informationstechnische Elemente, die Bereiche von besonderem (technischem) Interesse von einzelnen oder mehreren Produkten darstellen“[Web96]. Abbildung 2 zeigt den Aufbau von CAD-Modellen am Beispiel eines handelsüblichen PCs. Neben der Struktur, die sich in Gesamtsystem, Modul und Bauteil untergliedert; sind auch verschiedene Features aufgeführt. So können auf der Gesamtsystemebene verschiedene Konfigurationsmöglichkeiten des Produkts durch Features ausgedrückt werden. Auf der Modulebene werden spezifische elektronische Schnittstellen durch Features repräsentiert. Weiterhin werden geometrische Operationen durch Features abgebildet. So entspricht das „Entfernen“ eines zylinderartigen Volumens einer metrischen Bohrung. Diese Information kann dann von Benutzern während der Produktion genutzt werden, um die Teile zu fertigen.

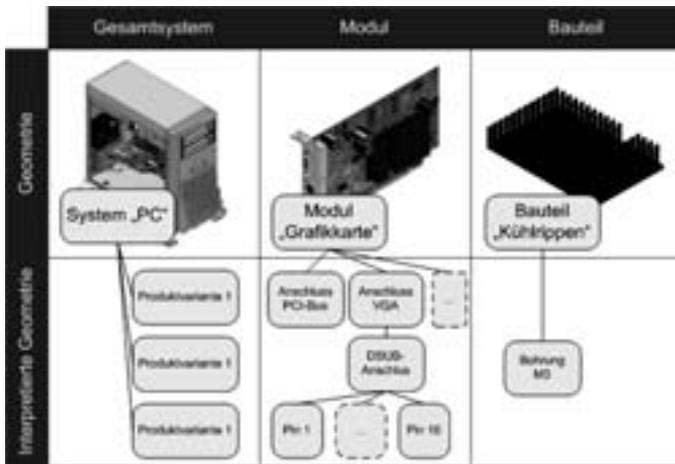


Abbildung 2: Features als Aggregation von Geometrie und Semantik auf verschiedenen Detaillierungsebenen

Zusammenfassend lässt sich sagen, dass sich die Funktionalität von CAD-Systemen nicht nur über die reine Erstellung von Geometrie erstreckt. Vielmehr ist unter diesem Begriff die „umfassende Unterstützung aller Konstruktionstätigkeiten durch Rechnerhilfsmittel zu verstehen, d.h. es gehören außer der Geometrieerzeugung auch alle anderen Tätigkeiten des Berechnens des Simulierens und der

Informationsgewinnung zum Zwecke der Erstellung eines Produktmodells dazu“
[Vaj08].

3 Technische Anforderungsentwicklung

Die Überführung der technischen Problemstellung in ein Problemmodell ist Aufgabe der Planungsphase bzw. der Anforderungsentwicklung⁴. Dabei kann zwischen Kunden- und Systemanforderungen [Ehr06] unterschieden werden. Die Kundenanforderungen werden in einem Lastenheft aufgeführt und beinhalten die Beschreibung des technischen Problems. Das Lastenheft ist die Vertragsgrundlage zwischen dem Kunden und dem Lieferanten und das entwickelte Produkt muss diese Anforderungen erfüllen. Mit Hilfe der Kundenanforderungen werden dann die Systemanforderungen entwickelt. In einem sogenannten Pflichtenheft zusammengestellt, beantworten sie die Frage, *wie* und *womit* ein Element des Lastenhefts zu lösen ist. Sie stellen somit eine Konkretisierung der Kundenanforderungen dar. Diese Transformation muss immer eine Prüfung auf Realisierbarkeit und Widerspruchsfreiheit beinhalten, denn Mängel in den Anforderungen sind hauptverantwortlich für das Scheitern von Entwicklungsprojekten [Ehr07].

Weiterhin haben die Anforderungen nicht nur eine technische Dimension, sondern sie legen vielmehr auch die Rahmenbedingungen fest, in denen die Entwicklung ablaufen soll. Dazu gehören Kosten- und Zeitvereinbarungen, juristische Aspekte in Form von zu beachtenden Normen und Gesetzen, oder auch die verfügbaren Fertigungs- und Produktionsverfahren. Somit bestimmen die Anforderungen die Vorgehensweise in allen weiteren Entwicklungsphasen und sind erheblich für die entstehenden Kosten verantwortlich [Pah07, Ehr06].

3.1 Klassifizierung von Anforderungen

Eine wichtige Aufgabe während der Entwicklung von Anforderungen ist die Klassifizierung bzw. Strukturierung und ihre Spezifizierung. Es gibt zwei Typen von Anforderungen. Die qualitativen Anforderungen beschreiben Produkteigenschaften, die oftmals aus Kundenwünschen abgeleitet werden, in rein textueller Form. Diese Formulierungen können implizit („Die Pumpe ist geräuscharm auszuführen.“) oder explizit („Das Pumpe darf nicht lauter als der Motor sein.“) sein. Daneben sind es die quantitativen Anforderungen, die als Punkt-, Minimal-/Maximal- oder Bereichsforderungen unter allen Umständen erfüllt werden müssen. Dabei haben Punktforderungen keinen Toleranzbereich („Übersetzung Getriebe $i=10$ “). Bereichsforderungen geben ein zulässiges Intervall an („Bauhöhe 300mm \pm 2mm“), innerhalb dessen die Forderung als erfüllt anzusehen ist. Minimal-/Maximalforderungen geben ein festgelegten Wert an, ab dem die Forderung erfüllt ist („Wirkungsgrad $\geq 95\%$ “).

⁴ Die Planungsphase umfasst noch weitere Tätigkeiten, die jedoch eine übergeordnete Rolle spielen. Dazu gehört z.B. die Ideenfindung.

Diese Forderungen können mehr oder weniger gut erfüllt werden. Ein Lösungsvorschlag, der diesen Forderungen nicht genügt, ist inakzeptabel.

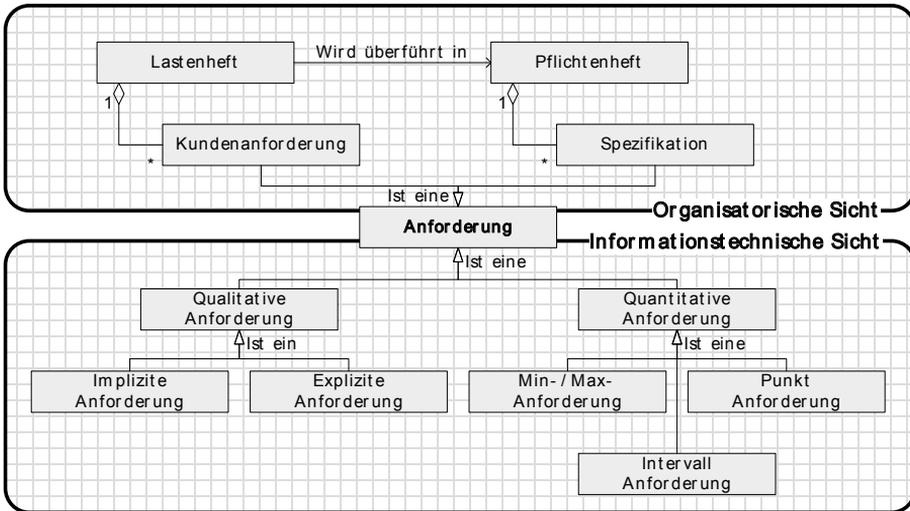


Abbildung 3: Organisatorische und datentechnische Sicht auf Anforderungen (in Anlehnung an [Lin08])

Abbildung 3 bringt die erwähnten Begriffe in einer objektorientierten Struktur zusammen. Gezeigt werden die organisatorische Sicht auf die Anforderungen und die damit verbundene Unterteilung in Lasten- und Pflichtenhefte sowie die informationstechnische Sicht, die die Anforderungen nach ihrem Informationsgehalt unterteilt.

3.2 Beziehungen zwischen Anforderungen

Die Klassifizierung von Anforderungen reicht aber nicht aus, um schon in dieser frühen Entwicklungsphase mögliche Fehlerquellen zu entdecken. Vielmehr müssen die einzelnen Anforderungen in ihrer Gesamtheit betrachtet und ihre Beziehungen untereinander ermittelt werden. Dabei gibt es verschiedene Typen von Beziehungen, die Anforderungen miteinander verknüpfen. Abbildung 4 zeigt diese Relationen und belegt sie mit Beispielen.

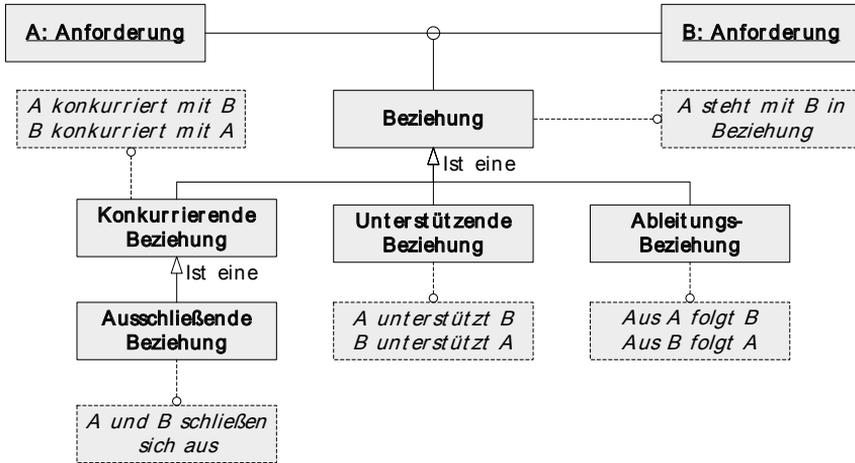


Abbildung 4: Mögliche Beziehungen von Anforderungen (in Anlehnung an [Lin08])

Wie die Abbildung zeigt können zwei⁵ Anforderungen miteinander konkurrieren und sich im Extremfall sogar ausschließen. Dies ist z.B. bei Automobilen in Bezug auf den Kraftstoffverbrauch und die Stabilität der Fall. Um den Verbrauch zu minimieren, muss eine Leichtbauweise angestrebt werden. Eine stabile Karosserie sollte allerdings eine hohe Steifigkeit besitzen, die wiederum bei Leichtbauweise nur sehr schwer realisiert werden kann. Im Gegensatz dazu können Anforderungen sich gegenseitig auch positiv beeinflussen. Um beim Beispiel des Automobils zu bleiben, könnten das z.B. Anforderungen bezüglich der Gewichtsreduktion und der Cabriolet-Bauweise sein.

Zuletzt gibt es noch eine weitere Anforderungsbeziehung, die die Nachverfolgbarkeit von Änderungen abbilden kann. Die Ableitungsbeziehung zeigt an, dass eine Anforderung notwendig wurde, weil sie aus einer ursprünglichen Menge von Anforderungen resultiert. Ein Beispiel für solch eine Beziehung sind beispielsweise Anforderungen, die nötig werden, wenn der Kunde ein Produkt in ein bestimmtes Land liefern möchte.

4 Integrationsansatz

Da der zentralen Bedeutung der Anforderungen (s. Kap. 3) auch in der rechnergestützten Konstruktion Rechnung getragen werden muss, stellt sich die Frage, wie die objektorientierte Beschreibungsmöglichkeiten von Anforderungen und deren Beziehungen (s. Kap. 3) in CAD-Werkzeuge integriert werden können. Abbildung 6 zeigt die relevanten Partialmodelle für diese Integration.

⁵ Es handelt sich eigentlich um n:m Beziehungen. Der Übersichtlichkeit halber sollen hier jeweils nur zwei Anforderungen miteinander verknüpft werden.

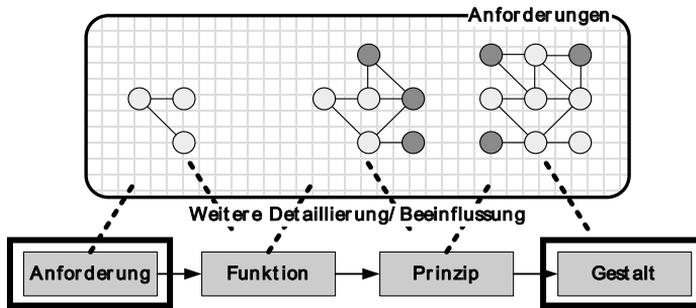


Abbildung 5: Bedeutung der Anforderungen (in Anlehnung an [Lin08])

Demzufolge muss ein CAD-integriertes Werkzeug implementiert werden, das in der Lage ist, dem Konstrukteur das Produktwissen, das in den frühen Entwicklungsphasen aufgebaut wird, verfügbar zu machen bzw. es zu visualisieren. Dabei sollen insbesondere Anforderungen bzw. deren Strukturen fokussiert werden. Der Benutzer soll erkennen, welche Features und Parameter des CAD-Modells in welcher Art und Weise durch eine Anforderung hervorgerufen wurden. Außerdem sollen die externen Einflussfaktoren, die in Form von Normen, Richtlinien, Gesetzen und anderen Dokumenten abgelegt sind, in der CAD-Umgebung im besten Falle kontextsensitiv bereitgestellt werden. Methodische Grundlage für die Integration bildet die Produktmodellierung auf Basis von Produktmerkmalen und Eigenschaften nach *WEBER* [Web03]. Dieser integriert die Nutzung von rechnerbasierten Hilfsmitteln in die Produktentwicklungsprozesse und ist zudem gut auf die CAD-Technologie anwendbar [Vaj08]. Als Bezeichnung für diesen Ansatz haben sich die englischen Bezeichnungen *Characteristic-Property-Modelling (CPM)* für den Produktmodell-Teil und *Process-Driven-Development (PDD)* für den prozessmodellierenden Anteil durchgesetzt. Zentrale Annahme dieses Ansatzes ist die Unterteilung eines Produktmodells in Produktmerkmale und Produkteigenschaften. Produktmerkmale sind alle Elemente des Produktmodells, die vom Entwickler direkt beeinflussbar sind, wie z.B. die Geometrie und Parameter innerhalb der CAD-Modelle. Produkteigenschaften sind die Daten, die das zukünftige Verhalten des Produkts beschreiben. Dazu können Eigenschaften, wie Festigkeit, Kinematik usw. gehören. Sie können nicht direkt manipuliert werden, sondern ergeben sich aus den verschiedenen Produktmerkmalen. Zwar werden diese Attribute des Produktmodells auch schon in den klassischen Vorgehensmodellen verwendet, aber bei CPM/PDD rücken diese in den Mittelpunkt der Betrachtung. Abbildung 6 zeigt den für die Integration von Anforderungen in ein CAD-System relevanten Regelkreis, der aus Anwendung des CPM/PDD Ansatzes für das hier vorgestellte Problem hervorgeht.

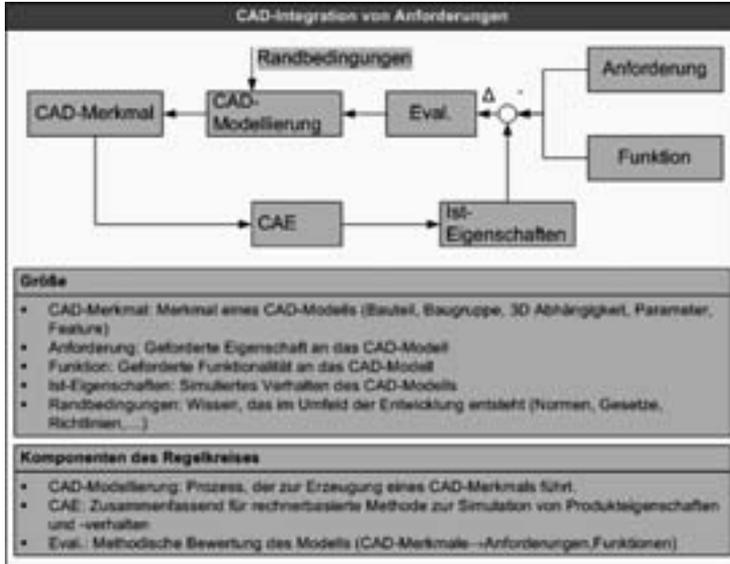


Abbildung 6: Grundlage der Anforderungsintegration (in Anlehnung an [Web03])

Ausgehend von Produktanforderungen und -funktionen erfolgt nach einer methodischen Bewertung die CAD-Modellierung. Ziel ist es CAD-Merkmale abzuleiten, die aus den Anforderungen entnommen werden können. Zwar können nie die Anforderungen an ein Produkt in ihrer Gesamtheit direkt auf CAD-Merkmale abgebildet werden, aber die Komplexität der Konstruktion und des Konstruktionsprozesses kann allein durch die Bereitstellung der Anforderungen im CAD-Modell reduziert werden und somit eine effizientere Arbeitsweise unterstützen. Die Gesamtheit des Produktwissens, das sich aus den Randbedingungen der Entwicklung ergibt, wird während dieses Vorgangs ebenfalls zur Verfügung gestellt, um dem Konstrukteur unnötige Rechercharbeiten zu ersparen. Die Rückführung erfolgt über computerbasierte Analyseverfahren (z.B. FEM – Finite Elemente Methode), die die verschiedenen CAD-Merkmale wieder auf Anforderungen abbilden können.

5 Entwurf und Implementierung des CAD-Anforderungswerkzeugs

Für die Implementierung der Schnittstelle wurde das CAD-System „Autodesk Inventor 2009“ (AI) des Anbieters „Autodesk“⁶ gewählt. Es bietet dem Software-Entwickler vielfältige Möglichkeiten zur Erweiterung der Grundfunktionalität durch Anbindung eigenen Codes. Neben der Makrosprache *Visual Basic for Applications (VBA)* die von Microsoft vertrieben wird, können über das *Component Object Model (COM)* beliebige

⁶ <http://www.autodesk.de>

COM-fähige⁷ Codebibliotheken entweder integriert als *AddIn*- oder unabhängig als *Standalone*-Systeme, die verfügbaren Objekte von AI nutzen.

Das CAD-System bietet die Möglichkeit der Modellierung von einzelnen Bauteilen und von ganzen Baugruppen. Die Bereitstellung der verfügbaren Befehle zur Manipulation des CAD-Modells wird dabei über die Verwendung von sog. Umgebungen gesteuert. So gibt es eine Umgebung für die Bauteilbearbeitung, den Zusammenbau oder für die Zeichnungsableitung. Desweiteren besteht die Möglichkeit der Definition eigener Umgebungen, die die vorhandenen erweitern. So gibt es beispielsweise eine Umgebung für die Festigkeitsanalyse von Bauteilen, die die Befehle der Bauteilumgebung um die Aufprägung von Lasten oder Lagerungen, Analyseverfahren oder Berichterstellung erweitert. Für die CAD-Integration werden zwei neue Umgebungen definiert, die jeweils für Bauteile und Baugruppen die neue Funktionalität der Anforderungs- und Funktionsbearbeitung bereitstellen.

AI bildet die CAD-Modelle in der API als featurebasierte Modelle ab. Über die entsprechenden Features können dann die Geometrieelemente ermittelt werden, die durch bestimmte Features erzeugt, gelöscht oder manipuliert wurden. Eine Stärke von AI ist Verwendung des Attribut-Konzepts. Dies erlaubt eine interne⁸ Attributierung beliebiger Datenobjekte des CAD-Modells. Die Organisation dieser Attribute erfolgt dabei über Attributsätze, die über einen modellweiten, eindeutigen Bezeichner angelegt werden können. Jeder dieser Attributsätze kann wiederum beliebig viele Attribute aufnehmen. Zulässige Datenwerte für die Attribute sind Zahlen (Ganzzahl, Fließkomma), Wahrheitswerte (Boolean), Zeichenketten (String) oder beliebige Binärdaten (Binary). Mit Hilfe der Attribute können somit in AI beliebige Objekte des CAD-Modells mit externen Informationen angereichert werden.

Die Implementierung dieser CAD-Integration wird als Add-In ausgeführt. Gegenüber VBA bringt das den Vorteil der Möglichkeit der Verwendung einer objektorientierten Programmiersprache. Die Wahl der Programmiersprache fällt auf C#. Diese Sprache ist Bestandteil der .NET Framework Plattform von *Microsoft* und weist in der grammatikalischen Definition große Ähnlichkeit zu *JAVA* auf.

Zusammenfassend kann gesagt werden, dass die Softwarelösung als COM-AddIn für die CAD-Software Autodesk Inventor 2009 implementiert wird. Dabei kommt die Programmiersprache C# zum Einsatz, die auf dem .NET Framework 2.0 basiert. Es werden zwei Benutzerumgebungen erstellt, die jeweils für Bauteile und Baugruppen eine Betrachtung von Anforderungen erlauben.

Abbildung 9 zeigt die Architektur des *Early Phase Integration Editors* (EPI). Dargestellt werden die Befehlsumgebungen für Bauteile und Baugruppen, die über ein COM-Addin mit dem EPI Editor kommunizieren. Zusätzlich erfolgt die Bearbeitung der Datendefinitionen in einem Verwaltungsmodul. Über ein Datenzugriffsmodul kann der

⁷ Z.B. Clients, die mit C++, .Net Sprachen oder mit Visual Basic implementiert wurden.

⁸ Attribute können nur über die API und nicht über die herkömmliche Oberfläche gesteuert werden.

EPI-Editor mit drei verschiedenen Datenbasen interagieren. Die Datenbank *Partialmodell-Definitionen* beherbergt die datenbezogenen Definitionen der Modellelemente und deren Verknüpfungen. Hier werden die Konzepte für Anforderungen (vgl. Kap. 3 → Abbildung 3, Abbildung 5) in einer relationalen Struktur auf Basis von SQL abgelegt. Hier kommt der SQL Server Express zum Einsatz. Die instanziierten Modellelemente werden in der Datenbank für Partialmodell-Instanzen verwaltet. Dabei entsprechen jeweils ein Modellelement und eine Relation einem Datensatz. Im dritten Datenteilbereich wird das verknüpfte Wissen in Form von Dokumenten oder Internet-Verknüpfungen abgelegt.

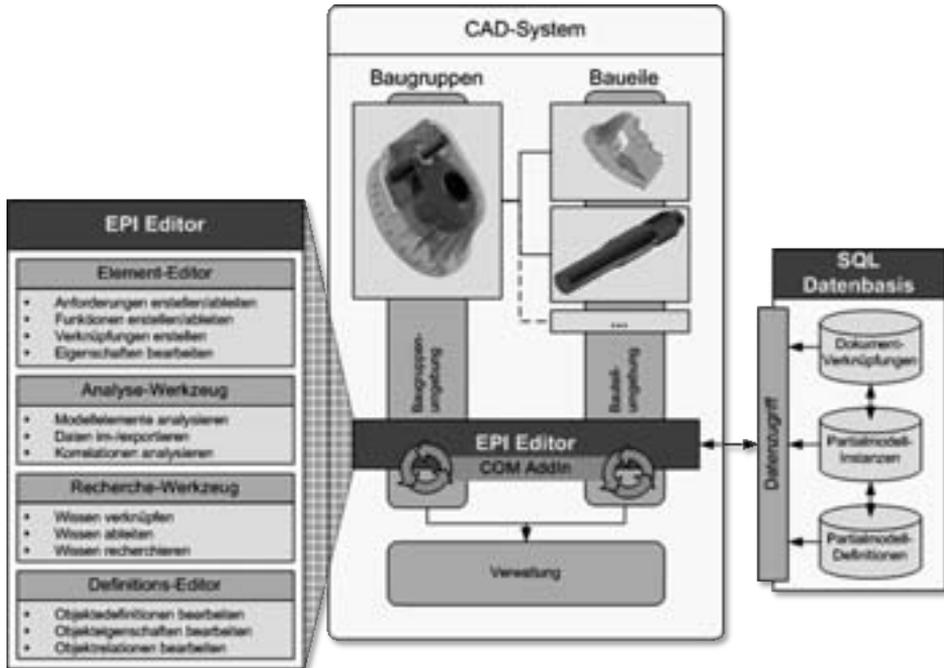


Abbildung 7: Architektur des EPI-Editors

Der EPI-Editor selber erlaubt das Editieren und Analysieren der Partialmodelle, die Recherche und Ableitung von Wissen. Weiterhin wird das gesamte Partialmodell mit allen verfügbaren Daten angezeigt. Dadurch kann der Entwickler jederzeit einen Überblick über das Gesamtsystem erhalten, um eventuelle Problemstellen ausfindig zu machen. Der EPI Editor wird als Zusatzpaket zu der normalen Funktionalität von AI konzipiert. Um die zusätzlichen Befehle nutzen zu können, muss die spezielle Umgebung aktiviert werden. In AI gibt es ein Menü, das die verfügbaren Umgebungen auflistet. In dem Hauptformular der Anwendung werden die Bestandteile der Partialmodelle aufgelistet. Zudem erfolgt hier die Neuanlage und Bearbeitung der Modellelemente. Abbildung 9 zeigt das Formular, das sich in zwei Bereiche untergliedert.

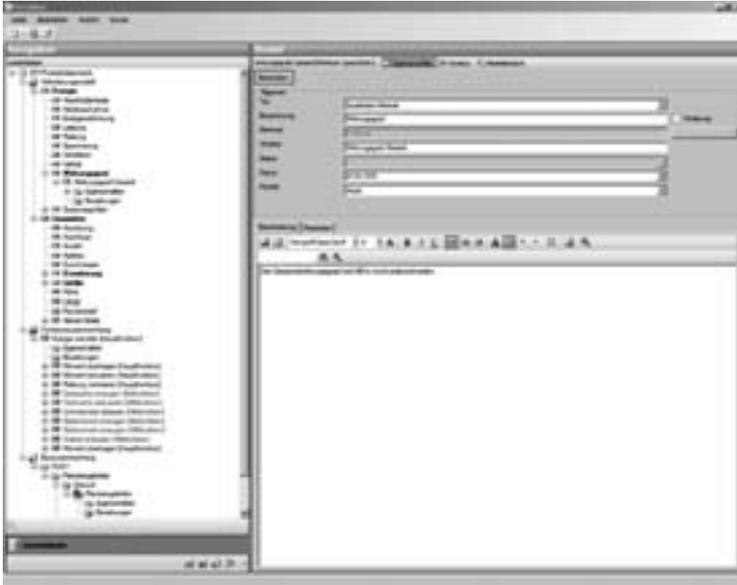


Abbildung 8: Hauptformular des EPI-Editors

Im rechten Teil werden die Bestandteile der Partialmodelle für Anforderungen und Bauzusammenhänge als Baumstruktur abgebildet. Die Verwendung hierarchischer Modellstrukturen bietet für die einzelnen Modelle verschiedene Vorteile. Die Anforderungen können in dieser Darstellung zu ihren Merkmalen zugeordnet werden. Baustrukturen, die sich wiederum in Bauteile und Baugruppe untergliedern, bilden die Grundlage für die Verbindung zu konkreten CAD-Daten. Die fortlaufende Konkretisierung wird durch die Darstellung als Strukturbaum und durch grafische Notationen unterstützt. Der Baum zeigt alle verfügbaren Daten zu einem Modellelement und parallel dazu werden wichtige Informationen, wie verknüpfte Anforderungen oder Parameter in einem Grafikfenster angezeigt (vgl. Abbildung 10).

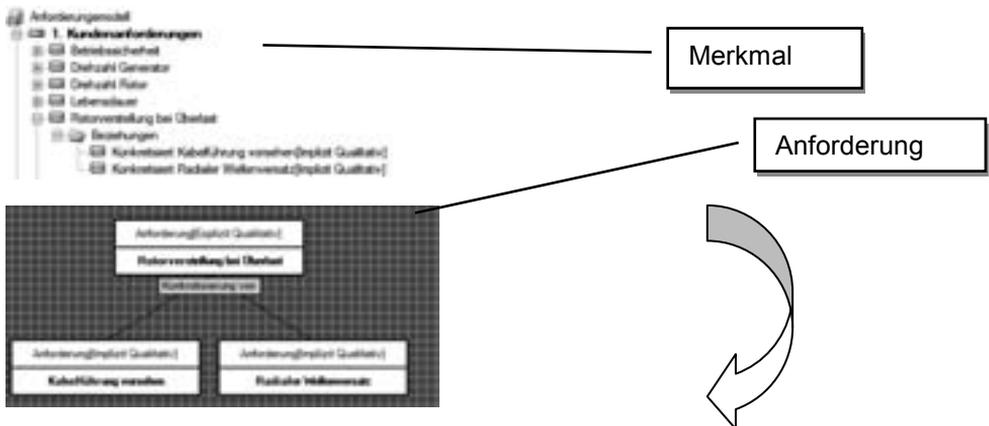


Abbildung 9: Baum- und Grafikdarstellung der Modell-Elemente

Die hier gezeigten Formulare und Oberflächenelemente werden in einer Bibliothek gekapselt. So wird es möglich, sie aus der CAD-Umgebung heraus aufzurufen, um so den Entwickler gezielt auf wesentliche Aspekte des technischen Systems und der Randbedingungen aufmerksam zu machen. Zusätzlich kann durch die Verknüpfung von Parametern und Anforderungen gezeigt werden, welche Parameter durch Anforderungen determiniert sind.

6 Evaluierung des Integrationswerkzeugs am Beispiel eines Planetengetriebes

Der entwickelte Editor soll am Beispiel der Entwicklung eines Planetengetriebes für eine Windkraftanlage evaluiert werden. Dieses technische System bietet sich an, weil die grundlegenden Wirkprinzipien des Antriebsstranges bekannt sind und somit eine Anpassungs- oder Variantenkonstruktion vorliegt. Demzufolge entfällt die Erarbeitung von funktionalen Zusammenhängen oder Wirkprinzipien [Pah07].

6.1 Klärung und Präzisierung der Aufgabenstellung

Die Entwicklung des Getriebes erfordert zunächst die inhaltliche Klärung der Aufgabenstellung. Die Anforderungen des Kunden sind in diesem Fall nur sehr rudimentär und bedürfen der weiteren Konkretisierung. Im günstigsten Fall führt diese Konkretisierung so weit, dass die quantitativen Merkmale in Form von Parametern schon direkt auf bestimmte CAD-Objekte abgebildet werden können. Um die Unterstützung des EPI Editors während der Aufgabenklärung aufzuzeigen, sind die Anforderungen an so ein Getriebe sind nur sehr grob gehalten. Tabelle 1 zeigt die geforderten Eigenschaften.

Bezeichnung	Qualitativ	Quantitativ
Eingangsleistung	Die Gesamtleistung des Getriebes	$\geq 1,5\text{MW}$
Antriebsdrehzahl	Die Drehzahl des Rotors	$= 15\text{ min}^{-1}$
Abtriebsdrehzahl	Die Drehzahl des Generators	$= 1500\text{ min}^{-1}$
Wirkungsgrad	Gesamtwirkungsgrad des Getriebes	$\geq 95\%$
Lebensdauer	Minimale Lebensdauer in Jahren	$\geq 20\text{ Jahre}$

Tabelle 1: Kunden-Anforderungen an das Planetengetriebe

Die Aufgabenklärung wird vom EPI-Editor in zweierlei Hinsicht unterstützt. Zunächst können Checklisten verwaltet werden. Diese Fragenkataloge sollen helfen, auf Probleme bei der Entwicklung explizit hinzuweisen. Dabei unterteilen sich die Fragen in zwei

Kategorien. Fragen an den Kunden sind im Gespräch mit dem Kunden zu klären und interne Frage müssen im Entwicklungsteam geklärt werden.

Die zweite methodische Unterstützung besteht in der Verwaltung von einer hierarchischen Hauptmerkmalsliste. Die Bezeichnung und die Struktur sind dabei frei wählbar. So sind Einteilungen nach technischen Hauptmerkmalen oder nach organisatorischen Prozessmerkmalen denkbar. Abbildung 11 zeigt die Darstellung im Baum des EPI-Editors.

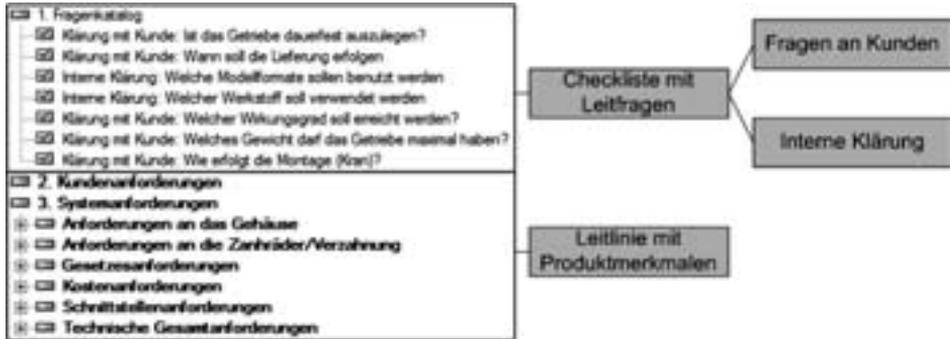


Abbildung 10: Methodische Unterstützung der Aufgabenklärung

Im Folgenden wird nun der Einsatz des EPI-Editors zur weiteren Konkretisierung dieser Entwicklungsaufgabe aufgezeigt

6.2 Definition und Konkretisierung von Anforderungen

Zunächst helfen die Leitfragen weitere Anforderungen zu bestimmen. Die Frage nach der Dauerfestigkeit des Getriebes führt beispielsweise zur Anforderung *Lebensdauer*. Diese Anforderung ist quantifizierbar und erhält somit einen Parameter Lebensdauer der zahlenmäßig mit 20 Jahren belegt wird. Daraus abgeleitet ergibt sich die Anforderung an die Dichtheit des Getriebes und einer hohen Stabilität des Gehäuses. Zur visuellen Unterstützung sind diese Konkretisierungsbeziehungen in dem Grafikenster des EPI-Editors abgebildet. In Abbildung 14 wird diese Abfolge im rechten Teil dargestellt.

Die Frage nach einem zu verwendenden Werkstoff führt zu der Anforderung *Gussgerechte Gestaltung*. Dieser Anforderung sind Dokumente zur Recherche hinterlegt worden, um bei der Konstruktion das gussgerechte Gestalten zu ermöglichen. Der EPI-Editor hat für diese Dokumente einen Viewer, der auf Internet-Browser Technologie basiert, für alle gängigen Dokumentformate integriert. Abbildung 14 stellt diese Anforderung im linken Teil dar. Zusätzlich ist hier auch die Eingabemaske für Anforderungen abgebildet.

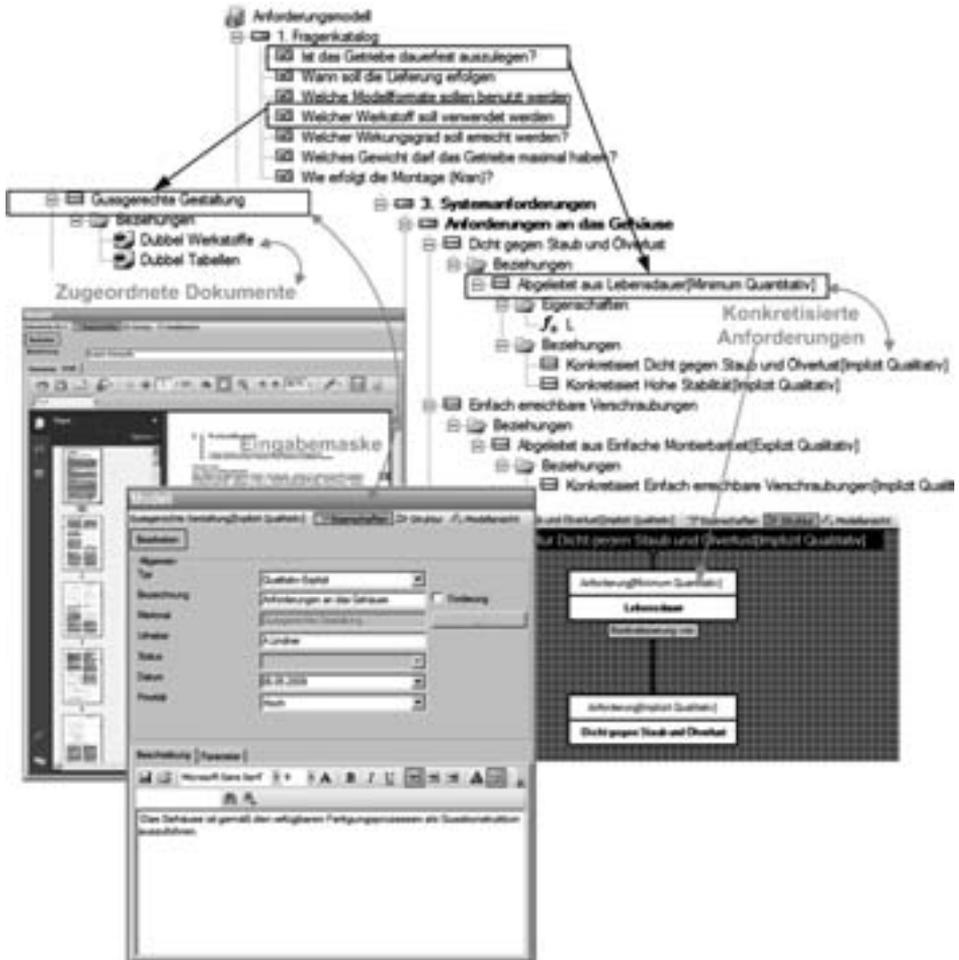


Abbildung 11: Anforderungen im EPI Editor

6.3 Gliederung in realisierbare Module

Die Anlage der Baustuktur erfolgt analog zur Anforderungsdefinition. Es können Baugruppen und Bauteile verwendet werden und zu einer hierarchischen Gesamtstruktur verbunden werden. Wenn dieser Vorgang abgeschlossen ist, erfolgt die Zuordnung von Anforderungen. Beispielsweise kann die Anforderung „Gussgerechtes Gestalten“ direkt mit den Gehäusekomponenten des Getriebes verbunden werden. Durch die applikationsübergreifende Vernetzung der jeweiligen Datensätze werden Anforderungen, die bereits mit anderen Anforderungen korrelieren, auch mit dem Bauelement verknüpft. So entsteht ein durchgängiges Modell. Abbildung 13 zeigt die Baustuktur des gesamten Getriebes und im rechten Teil werden die Komponenten der

Baugruppe „Stirnradstufen“ detailliert aufgelistet. Hier lässt sich erkennen, wie die Anforderungen mit den Komponenten verbunden sind.

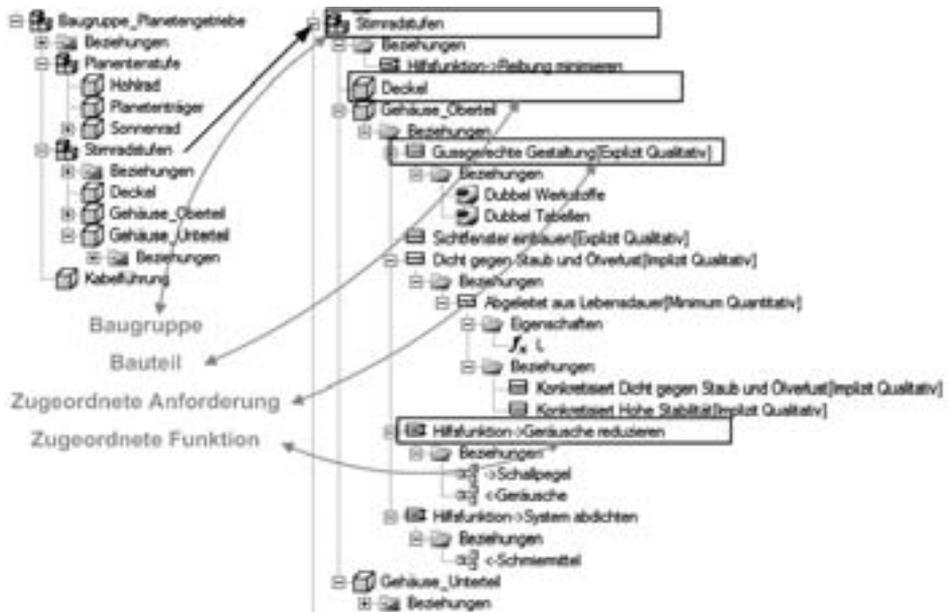


Abbildung 12: Baustrukturen im EPI-Editor

6.4 Verknüpfung mit CAD

Wenn die Baustruktur im EPI-Editor konzipiert und mit verschiedenen Modellelementen verknüpft wurde, kann die gesamte Baugruppe über einen Befehl erzeugt werden. Daneben gibt es auch die Möglichkeit der gezielten Erzeugung einzelner Bauteile oder Baugruppen. Somit kann der Konstrukteur gezielt bestimmte Modellteile konkretisieren. Bauteile, die bereits über ein zugeordnetes Modell verfügen, werden im Strukturbaum unterstrichen dargestellt und zusätzlich kann das Modell über den integrierten Viewer betrachtet werden. Der Befehl zur Modellerzeugung erstellt neben den Bauteildateien selbst, grafische Repräsentationen im Modelfenster des CAD-Systems. Hier werden dem Konstrukteur dann gezielt Informationen zu Anforderungen und verknüpften Dokumenten angezeigt. Abbildung 14 zeigt diese Visualisierung im Strukturbaum und im Grafikfenster. Hier sind drei verschiedene Anforderungen mit dem CAD-Modell verknüpft worden.

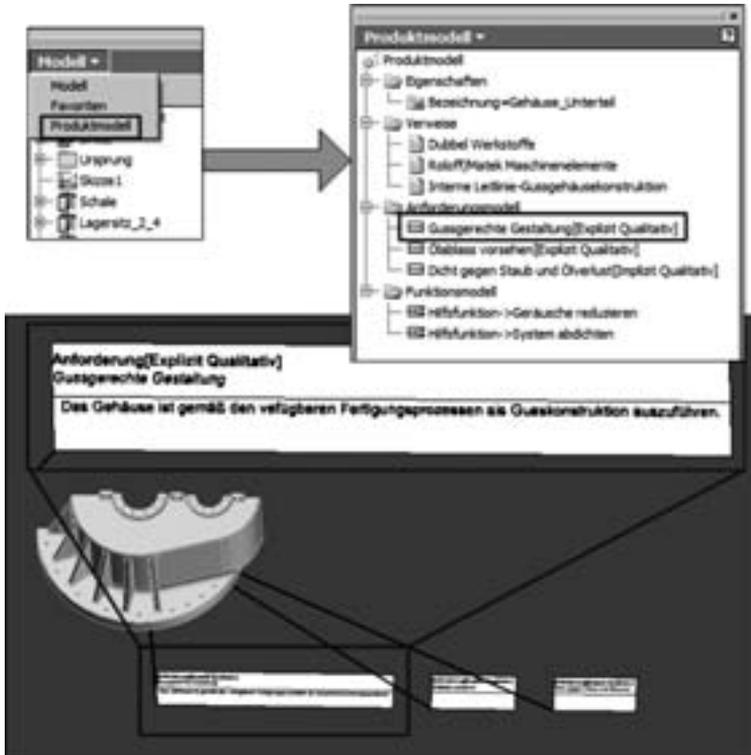


Abbildung 13: Anforderungsvisualisierung im CAD-System

Die Verbindung zwischen CAD-Dokument und EPI-Datenbank ist bidirektional, so dass der Konstrukteur direkt aus dem CAD-System einzelne Datensätze aufrufen kann und somit Anforderungen und Dokumente recherchieren kann. Weiterhin besteht die Möglichkeit der Ableitung von Parametern, die sich aus EPI-Objekten ergeben. Demzufolge können automatisiert Parametertabellen angelegt werden. Abbildung 15 zeigt die grundsätzliche Abfolge des Verknüpfens von CAD- und EPI-Daten. Zunächst wird über den Befehl „CAD-Verknüpfung ableiten“ ein Dialog gestartet, der die Parameter der EPI-Datenbank denen des CAD-Modells gegenüberstellt. Nach der Auswahl einer EPI-CAD Kombination kann dieses Paar dann verbunden werden. Diese neu angelegte Beziehung wird dann sowohl im CAD-Dokument als auch in der EPI Datenbank abgelegt. In dem hier gezeigten Beispiel wird eine Bereichsanforderung bezüglich der Wanddicke des Getriebegehäuses definiert. Daraufhin führt die EPI-Datenbank zwei Parameter, die mit dieser Anforderung verknüpft sind. Über den Auswahldialog kann dann einer dieser Parameter mit einem Modellparameter verknüpft werden.



Abbildung 14: Verknüpfen von CAD und EPI

Der Benutzer kann dann später in der Parametertabelle von AI erkennen, dass ein Parameter mit Anforderungen verknüpft wurde. Über den EPI-Hauptdialog können dann wiederum die hinterlegten Informationen abgefragt werden.

7 Zusammenfassung

Die virtuelle Produktentwicklung zeichnet sich durch eine durchgehende Integration verschiedener Produktinformationen und -modelle aus, wobei dem Gestaltmodell in Form von CAD-Daten eine zentrale Bedeutung zukommt. Ein wesentliches Ziel der Produktentwicklungsprozesse ist die fortlaufende Reduzierung der Komplexität ohne wichtige Merkmale aus den Augen zu verlieren. Um dieses Ziel zu erreichen wird das Produktmodell in viele Partialmodelle zerlegt, die getrennt voneinander betrachtet werden. Diese Arbeit beschäftigt sich mit der Fragestellung in welcher Art und Weise das Anforderungsmodell in ein CAD-System implementiert werden kann, um so das Produktwissen, das in diesen Modellen abgebildet wird, den Konstrukteuren vollständig zugänglich zu machen.

Zu diesem Zweck wurde ein Werkzeug entwickelt, das die Funktionalität eines herkömmlichen CAD-Systems in der Art erweitert, dass eine Modellierung und Visualisierung von Anforderungen und deren Beziehungen möglich wird. Weiterhin

können Parameter des CAD-Systems mit technischen Werten, die aus Anforderungen entnommen werden können, verknüpft werden. Durch diese neuen Funktionen kann der Konstrukteur jederzeit aus dem CAD-System heraus, wesentliche Anforderungen überprüfen, was zu einem qualitativ höherwertigen Modell und somit zu einem effizienterem Konstruktionsprozess führt.

Literaturverzeichnis

- [Abr07] Abramovici, M.; Meimann, V.: *Ein Ansatz zur Erhöhung der Stabilität der CAD-Modelle für die Verbesserung der Modell-Wiederverwendung in der Virtuellen Produktentwicklung*, In: Tagungsband zum 5. Kolloquium Konstruktionstechnik, Eigendruck, Dresden, 2007
- [Cha06] Chasiotis, C.: *Prozessbegleitende Wissensdokumentation und integrierte Wissensvisualisierung in der Digitalen Produktentwicklung*, Dissertation Shaker, Aachen, 2006
- [Con08] Conrad, K.-J.: *Taschenbuch der Konstruktionstechnik*, Carl Hanser Verlag, München Wien, 2008
- [Ehr06] Ehrlenspiel, K.: *Integrierte Produktentwicklung*, Carl Hanser Verlag, München Wien, 2006
- [Ehr07] Ehrlenspiel, K.; Kiewert, A.; Lindemann, U.: *Kostengünstig Entwickeln und Konstruieren*, Axel Springer Verlag, Berlin Heidelberg, 2006
- [Gho08] Ghoffrani, M.: *Entwicklung und Einführung eines flexiblen Softwaresystems zur Konfigurierung virtueller Produkte*, Dissertation, Shaker, Aachen, 2008
- [Lin08] Lindemann, U.: *Konzeptentwicklung und Gestaltung technischer Produkte*, Axel Springer Verlag, Berlin, 2008
- [Pah07] Pahl, G.; Beitz, W.; Feldhusen, J.; Grote, K.-H.: *Konstruktionstechnik*, Axel Springer Verlag, Berlin, 2007
- [Spu97] Spur, G.; Krause, F.-L.: *Das virtuelle Produkt*, Carl Hansen Verlag, München, 1997
- [Sta73] Stachowiak, F.: *Allgemeine Modelltheorie*, Axel Springer Verlag, Berlin, 1973
- [Suh93] Suhm, A.: *Produktmodellierung in wissensbasierten Konstruktionssystemen auf der Basis von Lösungsmustern*, Dissertation, Shaker, Karlsruhe, 1993
- [Szy01] Szykman, S.; Sriram, R.D.; Regli, W.C.: *The Role of Knowledge in Next-Generation Produkt-Development Systems*, In: Journal of Computing and Information Science in Engineering, 1/2001
- [Vaj08] Vajna, S.; Weber, C.; Bley, H.; Zeman, K.: *CAX für Ingenieure*, 2. Aufl., Axel Springer Verlag, Berlin Heidelberg, 2008
- [VDI93] VDI-Richtlinie 2221: *Methodik für das Entwickeln und Konstruieren technischer Systeme und Produkte*, Beuth Verlag, Düsseldorf, 1993
- [Web96] Weber, C.: *What is a Feature and What is its Use?*, In: Results of FEMEX Working Group I. 29th International Symposium on Automotive Technology and Automation, Florenz, 1996
- [Web05] Weber, C.: *CPM/PDD – An Extended Theoretical Approach to Modelling Products and Product Development Processes*, In: Bley, H.; Jansen, H.; Krause, F.-L.; Shpitalni, M. (Hrsg.): Proceedings Of The 2nd German-Israeli Symposium, Stuttgart, 2005
- [Web09] Weber, C.: *THEORY OF TECHNICAL SYSTEMS (TTS) – EXISTING APPROACHES AND CHALLENGES*, In: Bergendahl, M.N.; Grimheden, M.; Leifer, L.: Proceedings of ICED'09, Stanford, 2009

Internetquellen

[ADS09] Offizielle Webseite von *Autodesk Inc.*, URL:<http://www.autodesk.de>, Letzter Zugriff: Dezember 2009