

MODELLIERUNG VON KOMPLEXEN ANFORDERUNGEN

Carsten Stechert, Hans-Joachim Franke

Zusammenfassung

Die Entwicklung komplexer Produkte findet immer häufiger in Form einer interdisziplinären Zusammenarbeit in Kooperationsnetzwerken statt. Hierfür wird das Gesamtsystem in kleinere, handhabbare Teilsysteme (oder Teilaufgaben) zerlegt, die im Idealfall unabhängig voneinander bearbeitet werden können. Eine große Herausforderung ist es, diese Teilsysteme im Verlauf einer iterativen und dynamischen Entwicklung widerspruchsfrei zu halten. Der Grund hierfür sind i. Allg. unscharfe Systemgrenzen und Auswirkungen von Änderungen auf mehrere Teilsysteme gleichzeitig.

Es ist unbestritten, dass Anforderungen der Kern einer erfolgreichen Produktentwicklung sind. Es wird ein Vorgehen gezeigt, wie ein Anforderungsmodell auf Basis der Systems Modeling Language (SysML) aufgebaut und genutzt werden kann. Durch systematische Erweiterungen werden unterschiedliche Modellelemente (z.B. Szenarios, Funktionsstrukturen, mechanische Struktur) in eine abstrakte, leicht verständliche Beschreibungssprache integrierbar. Auf der Basis der erarbeiteten Klassifikation von Anforderungen und Beziehungen (qualitativ und quantitativ) erfolgt eine Analyse, die das Auffinden von Zielkonflikten und das Abschätzen von Änderungsauswirkungen unterstützt.

Dieser Ansatz wird auf dem Gebiet hochdynamischer Parallelroboter entwickelt. Parallelroboter sind mechatronische Produkte, die teilweise als Serienprodukte entstehen, aber meist auf die spezifischen Kundenbedürfnisse zugeschnitten werden müssen. Die generelle Akzeptanz des Ansatzes wird mit Studierenden als Testpersonen an einem überschaubaren Projekt aus dem Automodellbau überprüft.

1 Entwicklung komplexer Produkte in Kooperationsnetzwerken

Komplexe Produkte überschneiden verschiedene Fachbereiche. Das sind zum einen die an der Produktentwicklung direkt beteiligten mechatronischen Fachbereiche, wie Informatik, Elektrotechnik und Maschinenbau, zum anderen werden diese Produkte häufig in weiteren (nichttechnischen) Fachbereichen (z. B. Medizin, Lebensmitteltechnik) eingesetzt und teilweise von technisch wenig vorgebildeten Nutzern verwendet (z. B. Kraftfahrzeuge). Ein typischer Vertreter dieser Produkte ist ein Parallelroboter. Im Sonderforschungsbereich SFB 562 „*Robotersystem für Handhabung und Montage – Hochdynamische Parallelstrukturen mit adaptronischen Komponenten*“ werden unter anderem Entwicklungsmethoden und ein Baukastenkonzept für Parallelroboter entwickelt. Diese Roboter sind aus geschlossenen kinematischen Ketten aufgebaut und die Antriebe können gestellnah angeordnet werden. Dadurch werden geringe bewegte Massen, eine hohe Dynamik (d. h. hohe Geschwindigkeiten und Beschleunigungen) und hohe Steifigkeit (d. h. gute Genauigkeiten) erreicht. Zur Entwicklung dieses mechatronischen Produktes sind eine Reihe unterschiedlicher Modelle und Werkzeuge notwendig [1]. Da jeder Roboter auf die spezifischen Kundenwünsche zugeschnitten werden muss, sind außerdem eine Wiederverwendung von Wissen und Modellen sowie ein sinnvolles Änderungsmanagement unerlässlich.

In dieser Arbeit wird unter einem Kooperationsnetzwerk eine Gruppe organisatorisch und örtlich getrennter Unternehmen oder Abteilungen verstanden, die zusammen an der Entwicklung eines bestimmten Produktes arbeiten. Dabei ist das Ziel die spezielle Erfahrung und das spezifische Wissen der verschiedenen Unternehmen aus unterschiedlichen Fachbereichen zur gemeinsamen Produktentwicklung zu nutzen [2]. Dieses Vorgehen berücksichtigt, dass insbesondere kleine Unternehmen nicht in vielen unterschiedlichen Bereichen großes Wissen und Erfahrung besitzen können, aber dennoch Experten und Weltmarktführer in einer Nische sein können. Außerdem sind Kooperationsnetzwerke in der Lage durch die Parallelisierung von Arbeitspaketen die Kosten und die Markteinführungszeit zu senken und gleichzeitig die Qualität zu steigern.

Aus der Literatur sind Schwierigkeiten und Gefahren bei dieser Art von Entwicklungsprojekten bekannt. Sie können unter den folgenden Punkten grob zusammengefasst werden:

- Unzureichende Kommunikation zwischen Entwicklungspartnern.
- Schlechter Austausch von Informationen, z. B. unterschiedliche Datenformate.
- Unüberlegte Arbeitsteilung und unzureichende Überwachung der Schnittstellen.
- Unvollständige und zu späte Zusammenführung und Integration von Ergebnissen.
- Mangelnde Berücksichtigung von menschlichem Verhalten und zwischenmenschlichen Beziehungen.
- Ineffiziente und unmotivierte Anwendung von Methoden und Werkzeugen.
- Schlechte Organisation.

Ein Anforderungsmodell alleine kann zwar keine schlechte Organisation ausgleichen und ändert auch nichts an der Tatsache, dass viele der unterschiedlichen verwendeten Datenformate nicht ohne weiteres kompatibel sind, es schafft aber ein Bewusstsein für diese Probleme und eine Transparenz, die jedem Mitglied der Projektgruppe kommuniziert werden kann. Es bildet die Grundlage für ein besseres gegenseitiges Verständnis und für die Kontrollierbarkeit des Entwicklungsprozesses.

2 Anforderungsmodellierung

Es gibt eine große Anzahl an unterschiedlichen Definitionen für den Begriff „Modell“ (z. B. [3, 4]), die hier nicht wiederholt werden sollen. Wesentliche Punkte in allen Definitionen sind die Abstraktion und Vereinfachung der realen Welt, um Transparenz zu erzeugen und sich auf die wichtigsten Eigenschaften zu konzentrieren. Nur durch Modelle können die realen – oft komplexen – Beziehungen kontrolliert werden.

2.1 Anforderungen an Modelle

Die wichtigsten Anforderungen an Modelle im Allgemeinen sind in [4] zusammengefasst. Ein Modell soll...

- ...den zu modellierenden Gegenstand repräsentieren.
- ...unwichtige Details ignorieren (Abstraktion).
- ...einen unkomplizierten Gebrauch erlauben.

- ...das Verständnis unterstützen und verbessern.
- ...eine gemeinsame Diskussionsgrundlage schaffen.
- ...eine Möglichkeit zum Informationsaustausch bieten.
- ...den Vergleich unterschiedlicher Lösungen erlauben.
- ...eine Analyse und Vorhersage des Systemverhaltens und der Systemeigenschaften ermöglichen.

Um diese Anforderungen zu erfüllen, soll einem Modell eine gewisse Struktur gegeben werden. Außerdem sollen die internen Beziehungen der einzelnen Elemente untereinander und die externen Beziehungen zu Nachbarsystemen berücksichtigt werden.

Nicht in jedem Entwicklungsschritt und nicht für jeden Bearbeiter sind es die gleichen Eigenschaften, auf die es sich zu konzentrieren gilt. Es sind z. B. die Gliedlängen und Gelenkwinkel für ein kinematisches Modell wichtig, Masse und Trägheitsmomente dagegen unwichtig. Letztere Eigenschaften sind für das dynamische Modell hingegen von entscheidender Bedeutung. Während des Produktentwicklungsprozesses werden also eine Reihe unterschiedlicher Partialmodelle verwendet, die eine bestimmte Sichtweise auf das Gesamtsystem abbilden. Dabei ist es wichtig den jeweils richtigen Grad der Formalisierung zu treffen. In [5] wird z. B. ein Ansatz zur Visualisierung rein qualitativer Aussagen vorgestellt und auf den Bereich Parallelroboter angewendet. Je mehr Kreativität gefordert wird, desto hinderlicher ist eine starke Formalisierung. Beispielsweise würde kein guter Entwickler seine ersten Ideen in einem CAD-System modellieren, sondern eine Freihandzeichnung anfertigen. In späteren Phasen sind dann allerdings stark formalisierte Modelle (z. B. Werkstattzeichnungen oder 3D CAD-Modelle) notwendig, damit die erzeugten Ergebnisse in einer allgemein verständlichen Sprache ausgetauscht und ggf. in andere Modelle (z. B. FEM) transferiert werden können. Dass heißt der richtige Formalisierungsgrad folgt dem „Vom Groben zum Konkreten“-Ansatz vieler Vorgehensmodelle.

In [4] und [6] wird der Nutzen der Formalisierung zwischen einer zu hohen und einer zu niedrigen Formalisierung als konkave Funktion angenommen. Die Entscheidung über ein „zu hoch“ oder „zu niedrig“ ist projektabhängig: Je weniger Mitarbeiter und Fachbereiche involviert sind und je geringer der Projektumfang ist, desto geringer sollte die Formalisierung sein. Kann die benötigte Formalisierung vorausgesetzt werden (z. B. das Verstehen einer technischen Zeichnung), ist der Aufwand zur Formalisierung bei gleichem Nutzen geringer als z. B. bei gänzlich neu entwickelten Notationen.

Ingenieure gelten i. Allg. als visuell orientierte Denker. Doch auch neue Studien [7] kommen zu dem Ergebnis, dass eine diagrammatische Visualisierung von qualitativen Informationen in frühen Phasen kaum Verwendung findet. Das ist verwunderlich, da Diagramme alle oben angegebenen Anforderungen unterstützen können.

2.2 Die Modellierungssprache SysML

Die Systems Modeling Language (SysML) ist ein Ansatz, um ein Produkt auf unterschiedlichen Abstraktionsebenen und bezüglich unterschiedlicher Sichtweisen zu modellieren. SysML nutzt Teile der Unified Modeling Language (UML) und einige zusätzliche Erweiterungen zur Systemmodellierung (z. B. Anforderungsdiagramme). Im September 2007 wurde die OMG SysML v1.0 als „Available Specification“ [8] veröffentlicht und bildet eine allgemein bekannte Basis für einen besseren Informationsaustausch. Es können Anforderungen, Strukturen (z. B. mit Blöcken, Packages und Constraints) und das Verhalten (z. B. mit Aktivitäten und Anwendungsfällen) beschrieben werden. SysML / UML ist in den Bereichen Software-

entwicklung, Elektronikentwicklung und Automatisierungstechnik weit verbreitet, findet aber auch in einigen Bereichen des Maschinenbaus Anwendung (z. B. zur Beschreibung von Flugzeugstrukturen [9] und Funktionsstrukturen [10]). Beziehungen werden ebenfalls in Diagrammen für unterschiedliche Sichtweisen dargestellt. Außerdem sind aus dem Modell etablierte Beschreibungsarten, wie z. B. Anforderungslisten oder Design Structure Matrices (DSM) ableitbar. Die SysML kann als Profil bereits in vielen der bekannten Entwicklungswerkzeuge genutzt werden. Es lassen sich darüber hinaus eigene Profile und Erweiterungen erstellen, um projekt- und produktspezifische Eigenheiten besser abbilden zu können. Außerdem unterstützen viele Entwicklungswerkzeuge den Zugriff auf eine Mehrbenutzerdatenbank mit dazugehöriger Rechteverwaltung und Versionierungskonzepten. Das verteilte Arbeiten in Kooperationsnetzwerken wird dadurch erleichtert.

Da es sich bei SysML / UML um eine weit verbreitete Notation handelt, die auch an vielen Universitäten in unterschiedlichen Fachbereichen unterrichtet wird, kann von einer größeren Akzeptanz ausgegangen werden, als es für eine vollkommen neu zu erlernende Notation der Fall wäre.

2.3 Das Anforderungsmodell

In jedem größeren Entwicklungsprozess wird das Gesamtsystem in kleinere, handhabbare und gleichzeitig konsistente Teilsysteme und Teilaufgaben zerlegt. Das Anforderungsmodell ist dabei der Kern, der im weiteren Entwicklungsprozess dazu zwingt, die ursprünglichen Kundenwünsche, die strategischen Unternehmensziele und verschiedene Randbedingungen zu erfüllen. Bild 1 zeigt diesen Denkansatz als einen schematischen Überblick für den Parallelroboter HEXA.

Das Anforderungsmodell ist eines der ersten Produktmodelle und nicht a priori vollständig. Je weiter der Entwicklungsprozess fortschreitet, desto mehr Anforderungen werden notwendig, da auch das Wissen über das Produkt und die Konkretisierung des Produkts zunimmt. Beispielsweise könnte die Entscheidung „Das Robotergestell soll geschweißt

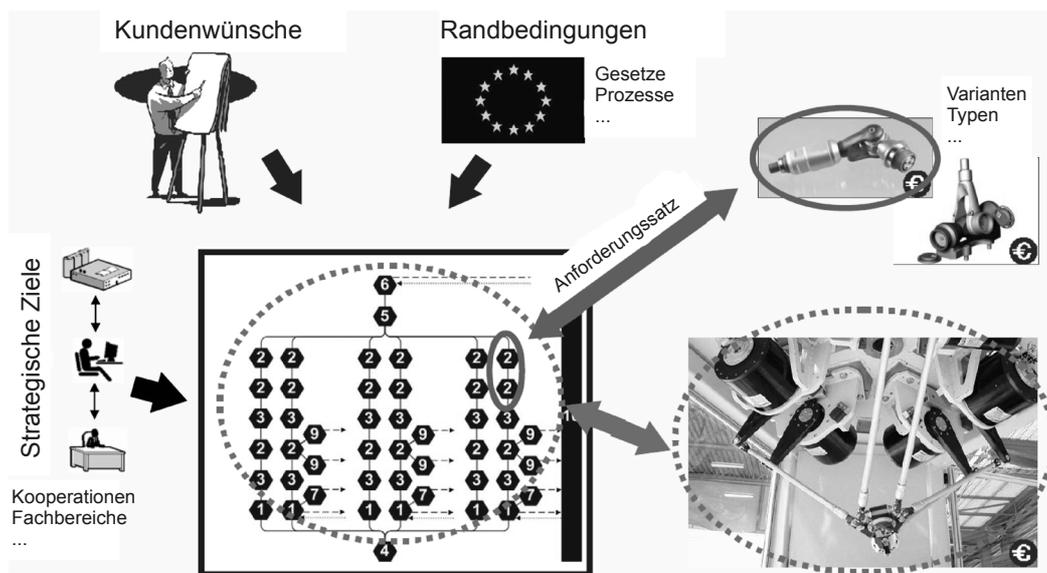


Bild 1: Anforderungen und Bedingungen für den Parallelroboter HEXA (sechs Freiheitsgrade), ein schematischer Überblick.

werden“ zu einer Anforderung „Benutze Profile aus Stahl“ führen. Die Entscheidung „Das Robotergestell soll gegossen werden“ würde zu gänzlich anderen Anforderungen wie z. B.

„Achte auf große Radien“ führen. Weiterhin können die Anforderungen im Verlauf der Produktentwicklung immer konkreter gefasst werden, so dass anfänglich nur qualitativ beschreibbare Anforderungen schließlich mit Werten belegt werden.

Erfahrungswerte besagen, dass ungefähr 50% der Anforderungen erst nach der „Aufgabenklärungsphase“ erfassbar werden. In [11] wird der qualitative Verlauf der Anforderungsklä- rung für ein interdisziplinäres Projekt beschrieben. Die Intensität der Anforderungsklä- rung ist lediglich im ersten Drittel der Projektdauer hoch und sinkt dann schnell auf sehr kleine Werte ab. Zwar werden auch in späteren Phasen einige Anforderungen spezifiziert, aber nur etwa die Hälfte der notwendigen Anforderungen wird dokumentiert. Bei einer mitlaufenden Anfor- derungsmodellierung werden die Anforderungen gleichzeitig auch dokumentiert.

Im Folgenden werden die Aspekte Produktumgebung, Anforderungsstruktur und Beziehun- gen näher betrachtet.

2.3.1 *Produktumgebung*

Einer der ersten Schritte im Produktentwicklungsprozess ist die Analyse der Produktumge- bung (z. B. [12]). Bereits hier sollte der gesamte Produktlebenszyklus betrachtet werden und unterschiedliche Szenarien und Anwendungsfälle mit den involvierten Akteuren, der relevan- ten Umwelt und möglichen Störeffekten untersucht werden. Das bedeutet es müssen die unterschiedlichen Produktsichten der verschiedenen Fachbereiche berücksichtigt werden, genauso wie Anforderungen, die erst durch spätere Entwicklungsschritte entstehen. Falls auf Grund von gesetzlichen Vorschriften z. B. Nachrechnungen nach einem bestimmten Verfah- ren vorgesehen sind, so ist es sinnvoll das Produkt frühzeitig derart zu gestalten, dass die (oftmals rechnerunterstützten) Rechenverfahren einfach angewendet werden können. Eine systematische Dokumentation hilft aus der Produktumgebung die relevanten Anforderungen zu generieren.

Ein Anwendungsfall für einen Parallelroboter während der Lebenslaufphase „Nutzung“ ist z. B. die Handhabung von Muffins. Dieser Anwendungsfall bringt unter anderem eine Verfei- nerung der Anforderungen „Arbeitsraum“ und „Nutzlast“ mit sich.

2.3.2 *Anforderungsstruktur*

Eine Strukturierung der Anforderungen erleichtert den Zugriff und die Analyse [13, 14]. Eine hierarchische Einteilung kann wie folgt erfolgen (vgl. auch Bild 2):

- *«goal»*
Übergeordnete (strategische) Ziele die erreicht werden sollen. Dazu soll das Produkt einen Beitrag leisten, z. B. kurze Zykluszeiten bei der Handhabung eines Objekts.
- *«target»*
Ziele, die direkt durch das Produkt erfüllt werden können und die übergeordneten Zie- le unterstützen, z. B. hohe Dynamik des Robotersystems.
- *«requirements»*
Anforderungen an das Produkt. Diese werden wiederum in System- und Teilsystem- anforderungen unterteilt, z. B. „hohe Beschleunigung“ an der Arbeitsplattform und „niedrige Reibung“ in den Gelenken. Die Tiefe der Unterteilung hängt hier von der Größe der Entwicklungsaufgabe ab.

Zusätzlich werden den Anforderungen gewisse Zugriffsmerkmale zugeordnet, um spezifi- sche Sichten zu generieren. Sinnvoll sind neben der bekannten Einteilung in Fest-, Mindest-

und Wunschforderungen z. B. Fachbereich und Relevanzeintritt. Außerdem kann eine Aussage über die Bestimmtheit der Anforderung bzw. die Wahrscheinlichkeit einer Änderung dieser Anforderung festgehalten werden. Bei einer hohen Änderungswahrscheinlichkeit sollten möglichst lange die entscheidenden konstruktiven Freiheitsgrade erhalten bleiben, d. h. hier gibt es entscheidende Hinweise bzgl. Modulbildung, Integral-/Differenzialbauweise und flexiblen Elementen [15]. Weitere Merkmale sind Funktionalität, Quantifizierbarkeit, Konkretisierungs- und Erfüllungsgrad.

2.3.3 Beziehungen

Modellelemente stehen in Beziehungen zueinander. Nach [7] unterscheidet sich ein System von seiner Umwelt dadurch, dass innerhalb der Systemgrenzen eine größere Anzahl an Beziehungen vorliegt als von Systemelementen nach „außen“. Das systematische Betrachten von Beziehungen zwischen den Systemelementen hilft dem Entwickler den Gesamtzusammenhang der Entwicklung besser zu begreifen und auch Schnittstellen zu anderen Fachbereichen und Systemen zu erkennen. Dadurch werden eine frühzeitige Kommunikation gefördert und fachbereichs- und systemübergreifende Zielkonflikte aufgedeckt. Außerdem können erkannte Beziehungen im Rahmen der Vervollständigung des Anforderungskataloges neue Anforderungen erkennen lassen [11].

Beziehungen können hinsichtlich unterschiedlicher Aspekte gegliedert werden, um das Systemverständnis zu verbessern und eine sinnvolle Analyse zu ermöglichen. Mögliche Aspekte sind die Phasen im Entwicklungsprozess, die Granularität, Wechselwirkung, Richtung, Direktheit und Quantifizierbarkeit [16].

Ein wichtiger Aspekt bei der Entwicklung von Produkten mit komplexen Anforderungen ist das Aufdecken von möglichen Zielkonflikten bereits in frühen Phasen. Je früher ein Bewusstsein für mögliche Probleme geschaffen wird, desto höher wird der spätere Nutzen sein. Das bedeutet nicht nur Lösungskonzepte frühzeitig auszuschließen, sondern insbesondere auch vorbereitet zu sein, falls ein möglicher Zielkonflikt in späteren Phasen tatsächlich eintritt. Zielkonflikte entstehen meist nicht in den frühen abstrakten Phasen, in denen Informationen häufig nur qualitativer Natur sind, sondern auf Grund von konkreten Entscheidungen in späteren, quantitativ beschreibbaren Phasen. Die Aufgabenteilung in unterschiedliche Teilsysteme, die Bearbeitung durch verschiedene Fachbereiche und die daraus resultierenden diffusen Grenzen erschweren die Verfolgbarkeit. Die systematische Modellierung der Beziehungen und die Rechnerunterstützung erlaubt eine Zurückverfolgung der Beziehungen auch über unterschiedliche Teilsysteme, so dass Zielkonflikte erkannt werden können sobald sie entstehen.

Bild 2 zeigt auf der rechten Seite den Aufbau einer kinematischen Kette des HEXA Parallelroboters aus seinen Strukturelementen. Auf der linken Seite soll das Gelenk näher betrachtet werden. Innerhalb des Gelenks sind die Anforderungen „geringe Reibung“ und „geringes Spiel“ dargestellt. Jede dieser Teilsystemanforderungen unterstützt die Systemanforderungen und damit schließlich die übergeordneten Ziele „hohe Qualität des Handhabungsobjektes“ bzw. „kurze Zykluszeit“. Zwischen den beiden Teilsystemanforderungen kann eine quantifizierbare Beziehung auf Basis der Coulomb'schen Reibung dargestellt werden. Die Anforderungen stehen in einem Konflikt, der sich ebenfalls auf die genannten Ziele überträgt. Es können nun die Auswirkungen genauer analysiert werden und ggf. ein Kompromiss erzwungen werden [17]. Es kann aber auch ein neues Konzept entwickelt werden, das den Zielkonflikt umgeht. Falls die beiden Anforderungen nicht zeitgleich erfüllt werden müssen, sind adaptive Gelenke [18] eine viel versprechende Lösung. In jeden Fall muss die Gültigkeit der Annahmen im konkreten Fall überprüft werden.

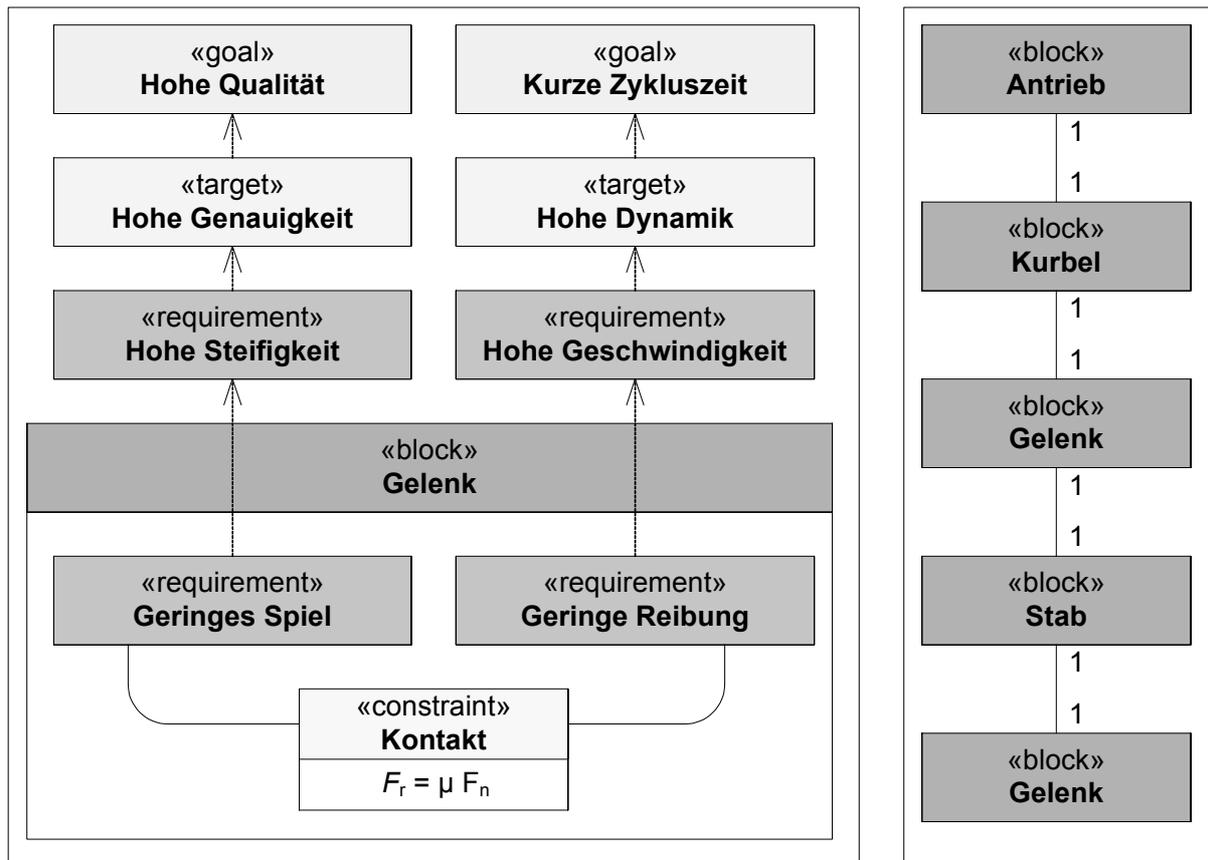


Bild 2: Vereinfachtes Beispiel des hierarchischen Aufbaus und der Beziehungen (links) und der strukturelle Aufbau einer kinematischen Kette des HEXA Parallelroboters (rechts).

Das Vorgehen hilft auch, um die Auswirkungen einer Änderung (z. B. der Ziele oder Randbedingungen) zu verfolgen und die Auswirkungen auf das Gesamtsystem zu beurteilen. Beispielsweise könnte ein Kunde entgegen der ursprünglichen Kostenziele einen Roboter zu einem niedrigeren Preis fordern. Durch Zurückverfolgung der Beziehungen zeigt sich, dass ein Austausch der Antriebe den größten Spielraum brächte. Da die preislich niedrigeren Antriebe ebenfalls geringere Leistungen erzielen, würde das übergeordnete Ziel der Zykluszeit dadurch beeinträchtigt. Gleichzeitig ändert sich z. B. der Energieverbrauch des Systems und ggf. Wartungsintervalle, d. h. die Lebenszykluskosten werden sowohl positiv als auch negativ beeinflusst. Diese Beziehungen können transparent gemacht und dem Kunden kommuniziert werden. So kann letztendlich der Kunde selbst entscheiden, ob ein kostengünstigerer Roboter eine längere Zykluszeit rechtfertigt.

3 Auslegung und Optimierung des Antriebsstrangs eines Modellautos

Eine Methoden, Methodiken oder Werkzeuge können nur dann als gut bezeichnet werden, wenn sie von den Nutzern akzeptiert werden. Aus diesem Grund wird die gezeigte Anforderungsmodellierung im Rahmen einer Vorlesung mit Studierenden (den möglichen Nutzern von morgen) überprüft. Das Projekt stammt zwar aus einem anderen Themenbereich ist aber hinsichtlich seiner Komplexität hinreichend einfach, um im Rahmen einer Vorlesung von Studierenden bearbeitet zu werden.

Die Teilnehmer sind Studierende des Maschinenbaus im Hauptdiplom und besitzen ein konstruktionsmethodisches Hintergrundwissen, d. h. die gebräuchlichen Vorgehensweisen und Hilfsmittel zur Anforderungsklä rung und Dokumentation sind bekannt und wurden im Verlauf des Studiums mehrfach in verschiedenen Projektarbeiten angewendet. Anwenderkenntnisse über die verwendete Notation SysML / UML waren bei keinem der Teilnehmer vorhanden. Die Aufgabe wurde nach einer kurzen Einweisung in die Notation und die Bedienung der Software in Gruppenarbeit von jeweils vier Studierenden gemeinsam gelöst.

Eine erste Überprüfung zielte auf die generelle Akzeptanz des Vorgehens, der Notation und der Verwendung einer kommerziellen Modellierungssoftware. Weitere vertiefende Überprüfungen folgten zum Jahresende.

3.1 Die Aufgabe

Das in Bild 3 a) abgebildete Modellauto ist mit einem 2-Takt-Motor ($0,88 \text{ kW}$ bei 19000 min^{-1}) ausgestattet und erreicht eine Spitzengeschwindigkeit von ca. 45 km/h . Das zu entwickelnde Derivat soll sich durch besseres Abgasverhalten auszeichnen, gleichzeitig aber eine akzeptable Fahrdynamik bereitstellen. Dazu soll der Antriebsstrang inkl. Motor durch eine Neukonstruktion ersetzt werden. Als Austauschmotor dient ein verbrauchsärmerer 4-Takt-Motor, der allerdings auch eine wesentlich geringere Leistung bei gleichzeitig geringeren Drehzahlen aufweist ($0,29 \text{ kW}$ bei 9500 min^{-1}).

Zur Bewältigung der Aufgabe werden alle Phasen der Produktentwicklung durchlaufen. Im ersten Schritt wird ein Anforderungsmodell in SysML mit einer kommerziellen Modellierungssoftware erstellt. Dazu müssen die Ziele, Randbedingungen und schließlich die Anforderungen selbst erarbeitet werden. Anschließend wird mit Hilfe von mathematischen Modellen in Computer Algebra Systemen (CAS) eine Übersetzungsstrategie festgelegt. Hier werden die Motorcharakteristiken und Fahrzeugwiderstände berücksichtigt. Mit einem Programm zur Berechnung von Maschinenelementen werden die Zahnräder ausgelegt und schließlich im CAD-System modelliert. Auf dem 3D-Drucker des Instituts werden ABS-Prototypen hergestellt (vgl. Bild 3 b). Das Derivat wird anschließend getestet.

3.2 Die Ergebnisse

Bei der Befragung der Teilnehmer konnte eine gute Akzeptanz der Notation festgestellt werden. Die Grundzüge der Notation wurden innerhalb kurzer Zeit verstanden und die Bedienung der Software wurde als in angemessener Zeit erlernbar bezeichnet. Lediglich

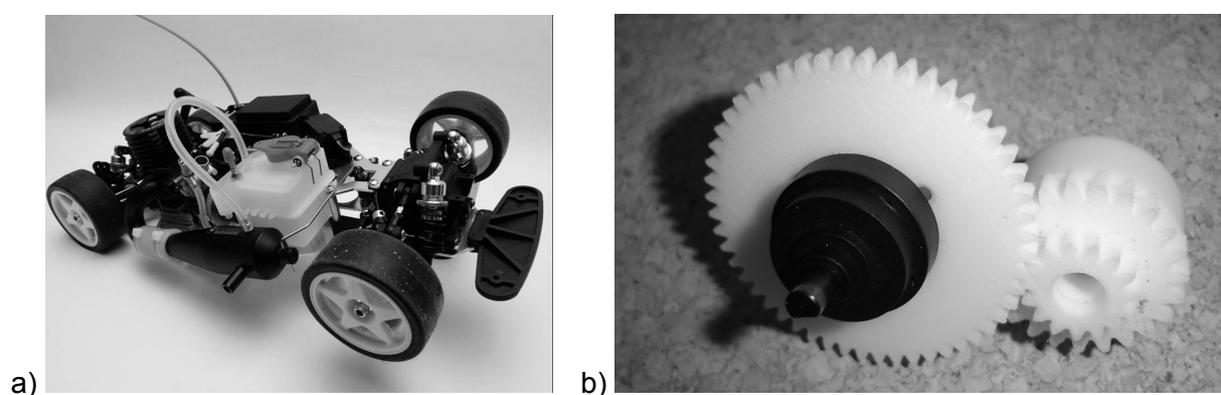


Bild 3: a) Das Basismodell des Modellautos mit eingebautem 2-Takt-Motor und 1-Gang-Getriebe, b) ABS-Prototyp des entwickelten 2-Gang-Getriebes für den 4-Takt-Motor.

einige Besonderheiten in der Benutzerführung sorgten anfänglich für Probleme. Eine etwas ausführlichere Einweisung in die Software kann diese Probleme beheben. Die Anforderungsmodellierung erschien den Teilnehmern wichtig und sinnvoll. Insbesondere mussten sie sich klar mit ihren Zielen auseinandersetzen und wurden sich über wichtige Zusammenhänge klar. Im Laufe des Projekts entschieden sich die Teilnehmer von dem 1-Gang-Getriebe des Basismodells abzurücken und ein 2-Gang-Getriebe zu entwickeln. Die Kenntnis der Zusammenhänge war insbesondere für die hierfür notwendigen Iterationen nützlich und kürzte den Entwicklungsprozess ab.

Die Hälfte der Teilnehmer war der Meinung, dass sie bereits durch ein intuitives Vorgehen ein sinnvolles Anforderungssystem erstellen konnten. Ein Drittel meinte, dass ihnen durch das Anforderungsmodell wichtige Zusammenhänge klar wurden, während zwei Drittel äußerten, dass ein Anforderungsmodell keine Zusatzinformation zur Anforderungsliste bringen würde. Grund für die letzte Aussage war sicherlich der begrenzte Rahmen des Projekts. Es handelt sich weder um eine sehr komplexe noch interdisziplinäre Problemstellung und bei einer Gruppengröße von vier Personen ist i. Allg. auch eine ausreichende Kommunikation gewährleistet.

4 Zusammenfassung

Die Entwicklung von Produkten in Kooperationsnetzwerken ist eine typische Arbeitsbedingung, die komplexe Anforderungsstrukturen hervorbringt. Es wird kurz auf die Verwendung von Anforderungen im Produktentwicklungsprozess eingegangen und die Schwierigkeiten und Gefahren, die sich bei der verteilten Entwicklung ergeben vorgestellt. Anschließend wird ein Ansatz zur Anforderungsmodellierung dargestellt, der basierend auf der bekannten Notation SysML die Aspekte Produktumgebung, Struktur und Beziehungen von Anforderungen berücksichtigt. Anschließend wird über erste Erfahrungen hinsichtlich der Akzeptanz des Ansatzes berichtet.

Es wird gezeigt, dass für eine Produktentwicklung mit komplexen Anforderungen eine sinnvolle Modellierung der Anforderungen zu mehr Transparenz und einem besseren Problemverständnis bei allen Beteiligten führt. Zielkonflikte und Schnittstellenprobleme können frühzeitig erkannt und die Auswirkungen von Änderungen zurückverfolgt werden.

5 Danksagung

Die Autoren danken der Deutschen Forschungsgemeinschaft (DFG) für die Unterstützung des Sonderforschungsbereichs SFB 562 "Robotersysteme für Handhabung und Montage – Hochdynamische Parallelstrukturen mit adaptiven Komponenten". Außerdem gilt der Dank allen Studierenden, die sich im Rahmen der Vorlesung als Testpersonen zur Verfügung gestellt haben.

6 Literatur

- [1] Stechert, C. Alexandrescu, I. Franke, H.-J.: Modelling of Inter-Model Relations for a Customer Oriented Development of Complex Products. 16th International Conference on Engineering Design ICED 07 (Paris, Frankreich, 28.-30. August), 2007
- [2] Franke, H.-J. Huch, B. Herrmann, C. Löffler, S. (Hrsg.): Ganzheitliche Innovationsprozesse in modularen Unternehmensnetzwerken. Logos Verlag, Berlin, 2005
- [3] Ort, A.: Entwicklungsbegleitende Kalkulation mit Teilebibliotheken. Papierflieger, Clausthal-Zellerfeld, 1998

- [4] Avgoustinov, N.: Modelling in Mechanical Engineering and Mechatronics - Towards Autonomous Intelligent Software Models, Springer Verlag, London, 2007
- [5] Stechert, C. Bauer, S. Franke, H.-J. Meerkamm, H.: Requirements Management in Early Stages of Mechatronic Design by Visualisation of Interdependencies. 10th International Design Conference DESIGN 2008 (Dubrovnik, Kroatien, 19.-22. Mai), 2008
- [6] Daenzer, W.F. Huber, F. (Hrsg.): Systems Engineering - Methodik und Praxis. Verlag industrielle Organisation, Zürich, 2002
- [7] Salustri, F.A. Eng, N.L. Weerasinghe, J.S.: Visualizing Information in the Early Stages of Engineering Design. In Computer-Aided Design & Applications (5: 1-4), 2008
- [8] The Official OMG Systems Modelling Language (SysML) site: <http://www.omg.sysml.org/>. 2007
- [9] La Rocca, G. van Tooren, M.J.L.: A modular reconfigurable software tool to support distributed multidisciplinary design and optimisation of complex products. 16th CIRP International Design Seminar (Kananaskis, Kanada, 16.-19. Juli), 2006
- [10] Johar, A. Stetter, R.: A Proposal for the Use of Diagrams of UML for Mechatronics Engineering, 10th International Design Conference DESIGN 2008 (Dubrovnik, Kroatien, 19.-22. Mai), 2008
- [11] Jung, C.: Anforderungsklärun in interdisziplinärer Entwicklungsumgebung. Verlag Dr. Hut, München, 2006
- [12] Franke, H.-J.: Untersuchungen zur Algorithmisierbarkeit des Konstruktionsprozesses. VDI Verlag GmbH, Düsseldorf, 1976
- [13] Stechert, C. Franke, H.-J.: Requirement-Oriented Configuration of Parallel Robotic Systems. 17th CIRP Design Conference - The Future of Product Development (Berlin, Deutschland, 26.-28. März), 2007
- [14] Krusche, T.: Strukturierung von Anforderungen für eine effiziente und effektive Produktentwicklung. Verlag Mainz, Wissenschaftsverlag, Aachen, 2000
- [15] Bischof, A. Blessing, L.: Gestaltungsrichtlinien zur Entwicklung flexibler Produkte. 18. Symposium „Design-for-X“ (Neukirchen, Deutschland, 11.-12. Oktober), 2007
- [16] Stechert, C. Franke, H.-J.: Managing Requirements as the Core of Multi-Disciplinary Product Development. 18th CIRP Design Conference - Design Synthesis (Enschede, Niederlande, 7.-9. April), 2008
- [17] Otremba, R.: Systematische Entwicklung von Gelenken für Parallelroboter, Logos Verlag, Berlin, 2005
- [18] Stechert, C. Pavlovic, N. Franke, H.-J.: Parallel Robots with Adaptronic Components - Design Through Different Knowledge Domains. 12th IFToMM World Congress (Besançon, Frankreich, 17.-21. Juni), 2007

Dipl.-Ing. Carsten Stechert
Prof. Dr.-Ing. Hans-Joachim Franke
Institut für Konstruktionstechnik
Technische Universität Braunschweig
Langer Kamp 8, D-38106 Braunschweig
Tel: +49-531-391-3349
Fax: +49-531-391-4572
Email: stechert@ikt.tu-bs.de
URL: <http://www.ikt.tu-bs.de>