

# Konsistente Anforderungsentwicklung und Systemableitung im mechatronischen Entwicklungsprozess von Brennstoffzellenantrieben

Rodrigo Biurrun<sup>1</sup>, Detlef Stolten<sup>2</sup>

<sup>1</sup> Robert Bosch GmbH, Gasoline Systems, Project House Fuel Cell Technology, Systems Engineering,  
70442 Stuttgart, Rodrigo.Biurrun@de.bosch.com

<sup>2</sup> Forschungszentrum Jülich GmbH, IEK-3: Fuel Cells,  
52425 Jülich, d.stolten@fz-juelich.de

## 1 Einleitung

Die Systementwicklung softwareintensiver Antriebe mit Brennstoffzellen bewegt sich in einem Spannungsfeld. Einerseits ist die zukünftige Kundennutzung in weiten Teilen nicht bekannt, was die Evaluierung technologischer Lösungen stark erschwert. Andererseits werden maximale Anforderungen an Sicherheit, Robustheit und Plattformtauglichkeit gestellt. Eine der Softwareentwicklung ähnliche Vorgehensweise der schrittweisen Modulentwicklung ist in der Regel nicht möglich. Aus diesem Grunde wird angestrebt, die frühen Phasen der Systementwicklung umfassend zu systematisieren. Kundenanforderungen müssen erfasst, strukturiert und bis in den Topologieentwurf des Brennstoffzellensystems und seiner eingebetteten Software verfolgt werden. Dazu müssen Spezifikationsaspekte des physikalischen Systems mit Analyse- und Designmodellen der Softwareanteile integriert werden [1]. Der vorgestellte Ansatz zeigt eine Möglichkeit, die Spezifikation mechatronischer Subsysteme und Komponenten aus einer Betrachtung der Anwendungsfunktionen des Gesamtsystems zu erreichen.

## 2 Spezielle Herausforderungen bei der Entwicklung von Brennstoffzellensystemen

Ein elektrischer Antriebsstrang mit einem Brennstoffzellensystem vereint die Komplexität eines elektrischen Hochspannungs-Bordnetzes im Fahrzeug mit der Verfahrenstechnik und Steuerungssoftware des Brennstoffzellensystems. Sensoren, Aktuatoren, Spannungswandler, Elektromotoren und Softwareanteile müssen nach den Anforderungen des Gesamtsystems ausgelegt oder angepasst werden [2]. Neben dieser systeminhärenten Komplexität gibt es eine Vielzahl von Kundenansprüchen an das Systemverhalten sowie Lebenszyklusanforderungen, welche die Auslegung maßgeblich beeinflussen. Die Spezifikation von Subsystemen und Komponenten kann daher nur mit einem Systemansatz erfolgen, der die unscharfen Kundenansprüche auf Fahrzeugebene systematisiert und die vielen Interaktionen zwischen Komponenten und Funktionssoftware identifiziert.

## 3 Herleitung und Strukturierung von Anforderungen

In einer systematischen Konzeptentwicklung wird die grundlegende Systemstruktur des Antriebssystems, bestehend aus Hardware und eingebetteter Funktionssoftware, konzipiert. Um dieses Ziel zu erreichen, müssen die Kundenanforderungen an das Gesamtfahrzeug erfasst und in Anforderungen an Subsysteme, Komponenten und Softwareanteile heruntergebrochen werden. Die Anforderungen selbst beschreiben in weiten Teilen lösungsabhängige, systemische Eigenschaften. Ihre Identifikation und Allokation ist daher zwangsläufig ein explorativer Prozess. Eine reine Top-Down-Vorgehensweise („Reduktionismus“) eignet sich in der Regel nur für stark modulare Systeme [3]. In der Spezifikation müssen sich daher Designschritte der Systemarchitektur effizient an eine vollständige Modellierung des Problemraums anschließen. Zudem muss das Systemverhalten möglichst vollständig modelliert werden. Dies schließt die Anwendungsfälle des Systems ein, aber auch die Nutzungsumgebung, Regularien und alle sonstigen Kundenziele. Die hier verwendete Modellierung des Problemraums mit der Ziele/Szenario-Methodik berücksichtigt die beschriebenen Kernforderungen:

- Lebenszyklusintegration: Alle Phasen des Produktlebenszyklus von Produktion bis Verwertung werden beachtet
- Modellintegration: Anforderungen und Modelle des Nutzerverhaltens müssen ineinander greifen und in Systemeigenschaften übersetzt werden
- Feed-Forward-Planning: Die Funktionsanalyse muss sich nahtlos an die Modellierung des Problemraums anschließen. Nur so lassen sich Anforderungsänderungen schnell und nachverfolgbar in Designevolutionen übersetzen [3]
- Constraint-Mapping: Randbedingungen müssen identifiziert und Systemfunktionen zugeordnet werden können

In der Anforderungsherleitung wird eine formale Kette verwendet, an deren erster Stelle eine detaillierte Zielmodellierung steht. Das hier verwendete Zielmodell verwendet keine logischen Verknüpfungen (Und/Oder) im Zielmodell, sondern bildet den Produktlebenszyklus nach DIN ISO 15226 ab. Die in einfacher Prosa gehaltenen Zieldefinitionen werden einmal hierarchisch verfeinert und in Zielklassen gruppiert. Durch die Definition der Klasse der Verhaltensziele wird die direkte Einbindung eines Use-Case-Modells ermöglicht. Jedes Verhaltensziel muss in einem Eintrag im Use-Case-Modell mit einer Eventkette beschrieben werden. Die sonstigen Attribute des Zielmodells und des Use-Case-Modells orientieren sich an üblichen Standards.

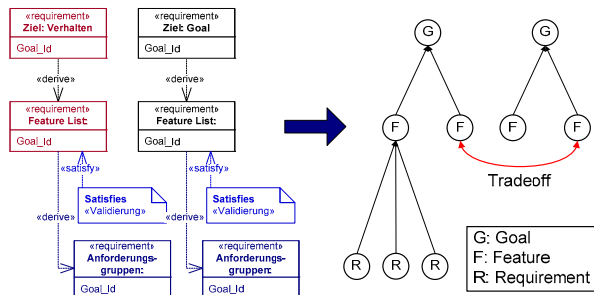


Abbildung 1: Zielmodell und Feature-Mapping

Um die Beschreibung der Ziele und Use-Cases in die Ebene der spezifizierbaren Anforderungen zu überführen, wird ein Feature-Mapping durchgeführt. Dabei werden den Zielen oder Verhaltensbeschreibungen Systemeigenschaften (Features) zugeordnet. Es ergibt sich eine Reihe von Vorteilen. Durch mehrfach zugeordnete Systemeigenschaften lassen sich auslegungsrelevante Tradeoffs schon früh erkennen. Außerdem ergibt sich zur Erstellung des Systemlastenhefts eine formal hergeleitete Struktur, die einen hohen Vollständigkeitsgrad der Anforderungen sichert und die Verfolgbarkeit bis in das Zielmodell gewährleistet. Dadurch lässt sich die Herkunft von Anforderungen auf unteren Ebenen leicht analysieren.

#### 4 Übergang von der Analyse zum Design

Um den Topologieentwurf des Brennstoffzellenantriebssystems und seiner eingebetteten Software nahtlos aus der Anforderungsanalyse abzuleiten, müssen die identifizierten Kundenanforderungen vollständig in Systemfunktionen und einschränkende Randbedingungen überführt werden. Die gesuchte Systemstruktur (funktionale Architektur) ist dann definiert, wenn die folgenden Kriterien erfüllt wurden:

- Systemfunktionen sind vollständig identifiziert
- Die Funktionsvernetzung ist hinreichend bekannt
- Bekannte Randbedingungen sind Funktionen zugeordnet (Constraint Mapping)
- Eine geeignete Funktionsclusterung zu Subsystemen wurde durchgeführt

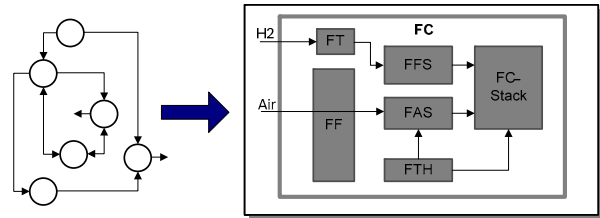


Abbildung 2: Funktionale Architektur

Die physikalischen Systemanteile sind durch ihre Gestalt bestimmten Einschränkungen unterworfen, denen die Softwareanteile oftmals nicht unterliegen [3]. Die implementierten Funktionsabläufe bestimmen jedoch maßgeblich das Systemverhalten. Um den Zusammenhang zwischen Funktion, Verhalten und Struktur herzustellen, werden Spezifikationsaspekte des physikalischen Systems mit Analyse- und Designmodellen der Softwareanteile integriert.

Ausgehend von den Systemschnittstellen wird in einer physikalischen Funktionsmodellierung die Realisierung der relevanten physikalischen Services beschrieben. Die Vorgehensweise orientiert sich dabei an gängigen Techniken der Funktionsmodellierung aus der Ingenieursanwendung. Basierend auf den Eventketten der Use-Cases wird zusätzlich eine funktionale Verhaltensmodellierung aufgebaut. Schlüsselabläufe des Systems werden als UML-Zustandsmaschinen modelliert, um Funktionen, Betriebsmodi und logische Abhängigkeiten zu identifizieren. Die Überlagerung der Informationen aus beiden Modellen liefert ein vollständiges Funktionsnetz. Mit den Werkzeugen der Netzwerkanalyse kann basierend auf diesem Netz eine Clusterung zu sinnvollen Subsystemen durchgeführt werden, die vollständig aus Use-Cases des Gesamtsystems (Fahrzeug) abgeleitet sind.

#### 5 Zusammenfassung

Mit dem beschriebenen Ansatz werden Systemanforderungen systematisch erfasst, strukturiert und in die Design-Ebene überführt. Dazu werden Spezifikationstechniken aus Softwareentwurf und Ingenieursanwendung kombiniert. Mechatronische Subsysteme und Komponenten können auf diese Weise aus einer Betrachtung der Anwendungsfunktionen des Gesamtsystems direkt hergeleitet werden.

#### Referenzen

- [1] Sonntag, G.: DFG-Projekt GEPRIS – Integration von Techniken der Softwarespezifikation für ingenieurwissenschaftliche Anwendungen, 2007
- [2] Wiedemann, G., Gottwick, U., Kropp, M., Kimmich, R.: Suitable Balance-of-Plant Components – A Key to Success of Fuel Cell Vehicles. In: WHEC 2010, 18<sup>th</sup> World Hydrogen Energy Conference
- [3] Crawley, E. et al: The Influence of Architecture in Engineering Systems. In: MIT Engineering Systems Monograph, 2004