



□ Prof. Dr. Klaus Pohl

(E-Mail: klaus.pohl@sse.uni-due.de)

ist Leiter der Arbeitsgruppe Software Systems Engineering der Universität Duisburg-Essen. Die Schwerpunkte der Arbeitsgruppe sind Requirements Engineering, Software Services und service-basierte Systeme sowie Software-Produktlinien.



□ Ernst Sikora

(E-Mail: ernst.sikora@sse.uni-due.de)

leitet das Requirements-Engineering-Team der Arbeitsgruppe Software Systems Engineering an der Universität Duisburg-Essen. Er promoviert auf dem Gebiet „Requirements Engineering und Architektur“.

COSMOD-RE: Verzahnung des Architektur-entwurfs mit dem Requirements Engineering

Eine funktionierende Zusammenarbeit zwischen Anforderungsingenieuren und -architekten ist eine wichtige Grundlage für die erfolgreiche Realisierung jedes komplexeren softwareintensiven Systems. Defizite bei der Abstimmung zwischen den Systemanforderungen und der Systemarchitektur sollten daher möglichst früh erkannt und behoben werden. Zum Beispiel sollten alle architekturtreibenden Anforderungen möglichst frühzeitig den Architekten bekannt sein, damit die Architekten eine für das Problem angemessene Architekturlösung bereitstellen können. Funktioniert das Zusammenspiel zwischen Requirements Engineering und Architekturentwurf in frühen Projektphasen nicht wie gewünscht, kann dies zahlreiche gravierende und kostspielige Fehler zur Folge haben. In diesem Beitrag wird die COSMOD-RE-Methode vorgestellt. Diese Methode wurde mit dem Ziel entwickelt, ein erfolgreiches Zusammenspiel zwischen Architekturentwurf und Requirements Engineering zu unterstützen.

Die Bedeutung einer geeigneten methodischen Anleitung

Neben menschlichen und sozialen Faktoren spielen unserer Erfahrung nach die existierenden Entwicklungsmethoden und -prozesse eine zentrale Rolle für ein Gelingen des Zusammenspiels zwischen Requirements Engineering und Architekturentwurf. Viele existierende Methoden und Prozesse propagieren jedoch ein Vorgehen, bei dem die Architekten erst ins Spiel kommen, nachdem eine vollständige Anforderungsspezifikation erstellt wurde. Dies führt in der Praxis zu verschiedenen, teilweise gravierenden Problemen. Wenn die Anforderungen spezifiziert und abgestimmt werden, ohne die Architekten einzubinden, sind die Anforderungen auf Sand gebaut, da sie möglicherweise nicht erfüllt werden können. Ein weiterer, für den Projekt- und Unternehmenserfolg ebenso gravierender Nachteil besteht in der verpassten Chance, die Kreativität und die Ideen der Architekten und Designer bei der Anforderungsdefinition auszuschöpfen.

Erfahrene Projektteams brechen daher (zu Recht) mit der Theorie und verzichten auf die durch die Methoden und Prozesse vorgegebene Trennung von Requirements Engineering und Entwurf. Doch auch dies

ist ein nicht gänzlich risikoloses Unterfangen. Meist führt es dazu, dass sowohl im Prozess als auch in dessen Ergebnissen zwischen der Problemdefinition (den Anforderungen) und der Lösungsbeschreibung (dem Architekturentwurf) keine ausreichende Trennung besteht. Die Vermischung von Problem und Lösung führt zu neuerlichen Schwierigkeiten. Die Projektbeteiligten befassen sich zu früh mit technischen Details der Lösung, während die Anforderungen – also das eigentliche Problem, für das der Auftraggeber oder Nutzer eine Lösung benötigt – nicht herausgearbeitet wird. Die Freiheitsgrade der Entwickler sind durch die detaillierten und teilweise unnötigen Lösungsvorgaben erheblich eingeschränkt. Bessere und kostengünstigere Lösungen werden gar nicht erst in Betracht gezogen, da diese den „Anforderungen“ widersprechen würden.

Um aus dieser Methoden-Zwickmühle zu entkommen, werden Entwicklungsansätze benötigt, die das Zusammenspiel zwischen Requirements Engineering und Architekturentwurf in frühen Projektphasen fördern und dabei gleichzeitig eine klare Trennung von Problemdefinition und Lösungsbeschreibung unterstützen.

Prinzipien der COSMOD-RE-Methode

Die Arbeitsgruppe Software Systems Engineering der Universität Duisburg-Essen [SSE] hat im Rahmen von mehreren Industriekooperationen die Interaktion zwischen Requirements Engineering und Architekturentwurf untersucht und als umfassende Lösung die COSMOD-RE-Methode entwickelt. Ein wesentliches Ziel von COSMOD-RE ist es, das Zusammenspiel zwischen Requirements Engineering und Architekturentwurf systematisch zu unterstützen. COSMOD-RE basiert auf den folgenden Grundprinzipien:

- **Modellbasierte Entwicklung:** Zur Dokumentation von Anforderungen und Architektur greift COSMOD-RE auf etablierte Anforderungs- und Architekturmodelle zurück. Da komplexe Systeme häufig in mehrere Systemebenen strukturiert werden, wie zum Beispiel „Gesamtsystem“, „Subsysteme“ und „Komponenten“, und an den Übergängen zwischen den verschiedenen Ebenen möglichst keine Brüche in der Modellierung auftreten sollten, werden in COSMOD-RE auf jeder Systemebene dieselben grundlegenden Mo-

Modelltyp	Kurzbeschreibung	Modellierungssprachen
Zielmodell	Definiert die Ziele, die das System erfüllen soll, einschließlich der hierarchischen Dekomposition der Ziele, Zielalternativen und Zielkonflikte. Ein Ziel dokumentiert die Intention eines Stakeholders und kann sich sowohl auf gewünschte Eigenschaften der Systemumgebung als auch auf Eigenschaften des Systems beziehen. Ziele werden unabhängig von einer möglichen oder intendierten Systemlösung definiert.	KAOS Goal Model [AvL]
Szenario-modell	Definiert eine Menge von Szenarien und setzt diese zur Systemumgebung in Beziehung. Ein Szenario beschreibt eine Folge von Interaktionen zwischen dem System und dessen Umgebung (d. h. mit externen Systemen, Personen oder Mess- und Steuergrößen), die zur Erfüllung eines bestimmten Ziels führen soll.	UML/SysML Anwendungsfallendiagramm [UML] [SML] UML/SysML Sequenzdiagramm [UML] [SML]
Funktions-modell	Definiert die Funktionen (Dienste), die das geplante System seiner Umgebung bereitstellen soll, die Eingabe- und Ausgabedatenflüsse der einzelnen Funktionen sowie wesentliche Datenspeicher des Systems.	SysML Aktivitätsdiagramm [SML]
Architektur-modell	Definiert wesentliche Strukturen des geplanten Systems. Dies umfasst Schnittstellen zu anderen Systemen, die Dekomposition des Systems in Subsysteme und Komponenten sowie Kommunikationsverbindungen zwischen den Komponenten.	UML Komponentendiagramm [UML] SysML Internal Block Diagram [SML]

Tab. 1: Charakterisierung der Modelltypen

delle eingesetzt. Allerdings variieren von Ebene zu Ebene der Detaillierungs- und der Abstraktionsgrad der Modelle.

- **Klar definierte Abstraktionsstufen:** Der steigenden Systemkomplexität setzt COSMOD-RE ein Abstraktionsstufenmodell entgegen, das die Anforderungs-Ingenieure und Architekten dabei unterstützt, einen für die jeweilige Entwicklungsaufgabe und die jeweilige Systemebene geeigneten Abstraktions- und Detaillierungsgrad ihrer Anforderungs- und Architekturmodelle zu wählen. Die vorgegebenen Abstraktionsstufen sollen zudem verhindern, dass auf einer Systemebene Lösungsdetails betrachtet werden, die für diese Systemebene noch nicht relevant sind.
- **Co-Design-Prozesse:** Anforderungs- und Architekturartefakte werden in sogenannten Co-Design-Prozessen entwickelt und dabei eng aufeinander abgestimmt. In einem Co-Design-Prozess werden Architekten von Anfang an in die Konzeption des Systems einbezogen. Ideen aus dem Architekturentwurf fließen in das Requirements-Engineering ein, und die vorgeschlagenen Architekturentwürfe werden frühzeitig gegen die Anforderungen evaluiert.

Die Umsetzung dieser drei Prinzipien in COSMOD-RE wird im Folgenden erläutert.

Erstes Prinzip: Modellbasierte Dokumentation von Anforderungen und Architektur

Der Einsatz von Modellen im Requirements Engineering und im Architekturentwurf bietet eine Reihe von Vorteilen. Modelle unterstützen die Kommunikation zwischen den beteiligten Stakeholdern wie zum Beispiel Auftraggebern, Anforderungs-Ingenieuren und Systemarchitekten. Dabei leisten sie vor allem einen Beitrag zu einem verbesserten, gemeinsamen Verständnis des Systems, der Aufgaben und Probleme, die das System lösen soll, sowie der Umgebung, in der das System seine Aufgaben erfüllen soll. Jeder verwendete Modelltyp lenkt die Aufmerksamkeit der Stakeholder auf einen anderen, für die Entwicklung zentralen Aspekt des Systems und hilft damit, von Details zu abstrahieren, die für die jeweilige Entwicklungsaufgabe von geringerer Bedeutung sind. Zudem machen Modelle die dokumentierten Sachverhalte einer frühen Konsistenzprüfung zugänglich, da sie Teilaspekte der Systemanforderungen und der Systemarchitektur formalisieren.

Damit die erstellten Modelle gut verständlich und ausreichend gehaltvoll sind, hat jedes Anforderungs- oder Architekturmodell in COSMOD-RE zwei wesentliche Bestandteile. Erstens beinhaltet ein Modell ein oder mehrere Diagramme, die die wesentlichen Modellelemente und deren Beziehungen grafisch definieren. Zweitens beinhaltet jedes Modell textuelle Detailbeschreibungen der einzelnen Modellelemente, sodass diese klar, verständlich und möglichst eindeutig interpretierbar sind. Die textuellen Beschreibungen werden häufig mittels Schablonen strukturiert, die die benötigten Informationen für das jeweilige Modellelement (zum Beispiel eine Funktion) festlegen.

Die COSMOD-RE Methode definiert drei grundlegende Arten von Modellen zur Dokumentation der Anforderungen und der Architektur eines Systems:

- **Lösungsneutrale Anforderungsmodelle** dokumentieren die Anforderungen an das System in einer lösungsneutralen Weise. Sie spiegeln im Wesentlichen die Nutzungs- und Auftraggebersicht auf das System wider. COSMOD-RE verwendet zwei zueinander komplementäre, lösungsneutrale Arten von Anforderungsmodellen: Zielmodelle und Szenariomodell (siehe Tabelle 1).
- **Lösungsorientierte Anforderungsmodelle** bieten die notwendigen Detailinformationen für die Systementwickler und -tester. Ein Beispiel eines lösungsorientierten Modells ist das Funktionsmodell des Systems (siehe Tabelle 1).
- Das **Architekturmodell** beinhaltet verschiedene Architektursichten, wie zum Beispiel eine Struktursicht und eine Prozesssicht. Der Schwerpunkt von COSMOD-RE liegt derzeit auf der Struktursicht der Architektur (siehe Tabelle 1).

Eine wesentliche Eigenschaft komplexer Systeme wie zum Beispiel von Fahrzeugen, Flugzeugen oder Produktionsanlagen ist die Strukturierung dieser Systeme in mehrere Systemebenen. Ein Beispiel einer solchen Strukturierung ist in **Abbildung 1** (links) dargestellt. Da auf jeder Systemebene dieselben prinzipiellen Entwicklungsschritte durchgeführt werden müssen und die Übergänge zwischen den verschiedenen Systemebenen möglichst frei von Brüchen in der Modellierungsmethodik sein sollten, werden in COSMOD-RE auf jeder

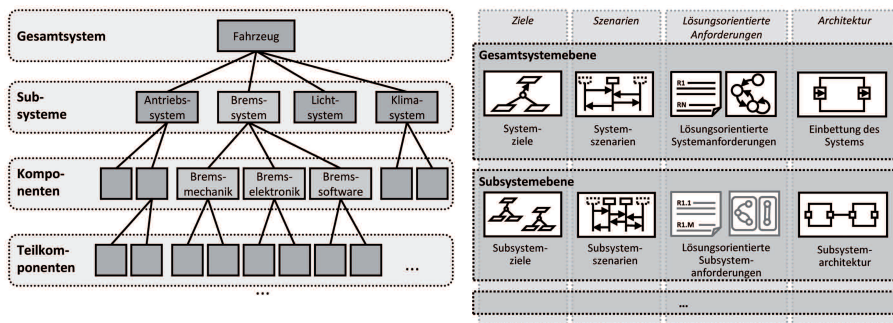


Abb. 1: Systemstruktur und -modelle über mehrere Ebenen hinweg

Systemebene dieselben prinzipiellen Typen von Anforderungs- und Architekturmodellen eingesetzt. Mit anderen Worten werden die oben beschriebenen Anforderungsmodelle sowohl zur Beschreibung der Gesamtsystemanforderungen als auch der Subsystemanforderungen und der Komponentenanforderungen eingesetzt (siehe Abbildung 1, rechts).

Zweites Prinzip: Klar definierte Abstraktionsstufen

Modellierungssprachen wie zum Beispiel die Unified Modeling Language (UML [UML]) oder die Systems Modeling Language (SysML [SML]) bieten keine Hinweise darauf, welchen Detaillierungsgrad ein Modell haben sollte. Nicht selten sehen sich Anforderungs-Ingenieure, die die UML oder SysML einsetzen, dazu verleitet, bereits zu einem frühen Zeitpunkt Lösungsdetails zu modellieren, wie zum Beispiel Klassen, Operationen und deren Aufrufschnittstellen. COSMOD-RE unterstützt Anforderungs-Ingenieure und Architekten daher mittels eines Abstraktionsstufenmodells, den richtigen Detaillierungsgrad für ihre Anforderungs- und Architekturmodelle zu wählen. Da das Abstraktionsstufenmodell die Abstraktion von Lösungsdetails fördert, führt es (auf höheren Abstraktionsstufen) gleichzeitig zur Komplexitätsreduktion der erstellten Modelle. Das Abstraktionsstufenmodell von COSMOD-RE ist in Abbildung 2 dargestellt.

Auf der obersten Abstraktionsstufe, der Stufe „System (Black Box)“, liegt der Schwerpunkt auf der Festlegung und Abstimmung der Systemziele, Systeminteraktionen und Systemfunktionen. Das System wird dabei als eine „schwarze Kiste“ betrachtet, die über eine Menge verschiedenartiger Schnittstellen (zum Beispiel

Mensch-Maschine- und Kommunikationschnittstellen) ihrer Umgebung Funktionen und Dienste bereitstellt. Auf tieferen Abstraktionsstufen wird die Anforderungs- und Architekturspezifikation zunehmend detaillierter und auch zunehmend technischer, bis schließlich ein für die Implementierung und den Test des Systems ausreichender Detaillierungsgrad erzielt wird.

Die höchste Abstraktionsstufe und die niedrigste Abstraktionsstufe zeichnen sich dadurch aus, dass sie jeweils nur einer Systemebene zugewiesen werden (der höchsten und der niedrigsten Ebene). Die Abstraktionsstufen „Funktionale Dekomposition“ und „Hardware-Software-Partitionierung“ können hingegen jeweils einer oder mehreren Systemebenen zugewiesen werden. Mit anderen Worten kann zum Beispiel eine funktionale Dekomposition des Systems über mehrere Systemebenen hinweg erfolgen (zum Beispiel Sub-Systeme und Sub-Sub-Systeme).

Drittes Prinzip: Verzahnte Entwicklung von Anforderungen und Architektur

Die Verzahnung von Requirements Engineering und Architekturentwurf erfolgt in

COSMOD-RE durch eine prozessbasierte Integration von Requirements-Engineering- und Entwurfsaktivitäten. Die COSMOD-RE-Methode stellt somit ein Prozessmodell bereit, das die verzahnte Entwicklung und wechselseitige Abstimmung zwischen Anforderungen und Architektur unterstützt. Dazu definiert COSMOD-RE das Konzept des „Co-Design“-Prozesses. Ein Co-Design-Prozess verfolgt das Ziel, zunächst (lösungsneutrale) Anforderungen aus Sicht des Auftraggebers zu identifizieren und gleichzeitig einen ersten, groben Architekturvorschlag zu erarbeiten. Erst nachdem die Problemstellung in Form von lösungsneutralen Anforderungsmodellen (Zielen und Szenarien) herausgearbeitet, ein Architekturvorschlag definiert und Ziele, Szenarien und Architektur aufeinander abgestimmt wurden, erfolgt die Spezifikation detaillierter, lösungsorientierter Anforderungen.

Lösungsorientierte Anforderungen werden häufig durch Architekturwissen beeinflusst, wie zum Beispiel die Annahme, dass bestimmte Systemschnittstellen oder sogar bestimmte Systemkomponenten vorhanden sind. Die Erstellung der lösungsorientierten Anforderungen im Anschluss an die Abstimmung von Zielen, Szenarien und Architektur hat den Vorteil, dass zu diesem Zeitpunkt das Problem und die architekturelle Lösung bereits gut verstanden und verhältnismäßig stabil sind. Mehrfache Änderungen der bei komplexen Systemen sehr umfangreichen, lösungsorientierten Anforderungen, zum Beispiel aufgrund veränderter Ziele oder aufgrund von Architekturänderungen, können auf diese Weise reduziert werden.

Abbildung 3 zeigt die einzelnen Subprozesse eines Co-Design-Prozesses und die Informationsflüsse zwischen den Subprozessen.

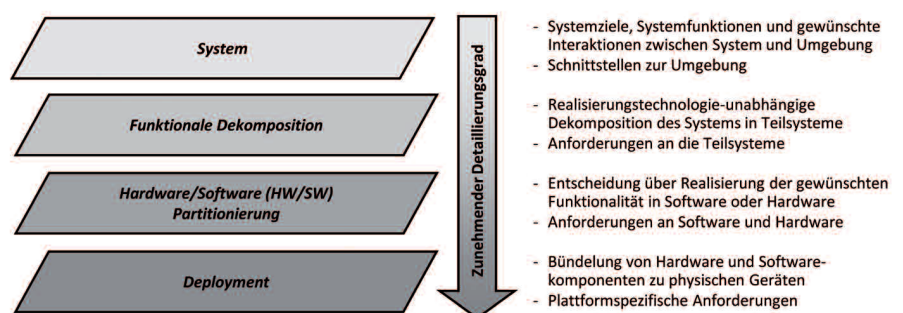


Abb. 2: Abstraktionsstufenmodell der COSMOD-RE-Methode

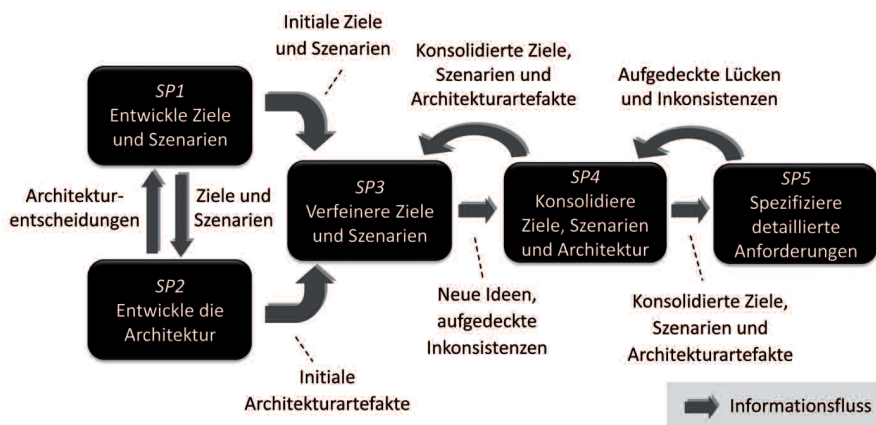


Abb. 3: Struktur eines Co-Design-Prozesses zur verzahnten Entwicklung von Anforderungen und Architektur

Diese fünf Subprozesse und die darin erstellten Artefakte (siehe Tabelle 1) können wie folgt charakterisiert werden:

- Subprozess SP1 ist für die Entwicklung initialer Ziele und Szenarien für das geplante System verantwortlich.
- Subprozess SP2 ist für die Entwicklung einer initialen, groben Systemarchitektur verantwortlich.
- Subprozess SP3 ist für die Zuordnung der Ziele und Szenarien zu Architekturelementen verantwortlich. Basierend auf der Zuordnung werden Diskrepanzen zwischen den (lösungsneutralen) Anforderungen und der Architektur identifiziert, wie zum Beispiel Architekturelemente, denen keine Anforderung zugeordnet ist.
- Subprozess SP4 ist für die Konsolidierung der Ziele, Szenarien und Architekturartefakte zuständig, um die in Subprozess SP3 identifizierten Diskrepanzen aufzulösen.
- Subprozess SP5 ist für die Spezifikation detaillierter, lösungsorientierter Anforderungen (unter anderem des Funktionsmodells) zuständig. Als wesentliche Eingaben fließen in diesen Subprozess die definierten und bereits konsolidierten Ziele, Szenarien und Architekturartefakte ein.

Damit in Subprozess SP3 eine klare Zuordnung von Zielen und Szenarien zu Architekturelementen (Subsystemen) möglich ist, ist es erforderlich, die Ziele und Szenarien zu verfeinern. Das Ziel der Verfeinerung besteht darin, Ziele und

Szenarien soweit zu verfeinern, dass die Teilziele ebenso wie die Interaktionsschritte der Szenarien einzelnen Architekturelementen eindeutig zugeordnet werden können. Dieser Schritt stellt gleichzeitig den Übergang von einer Systemebene zur nächsten dar, zum Beispiel vom Gesamtsystem zu den Subsystemen, da den einzelnen Subsystemen initiale Anforderungen (in Form von Zielen und Szenarien) zugewiesen werden.

Auszüge aus einem Modellierungsbeispiel

Wie aus der obigen Beschreibung deutlich wird, ist die integrierte Entwicklung und Verfeinerung von Zielen, Szenarien und Architekturartefakten der wesentliche Treiber für die Verzahnung von Requirements Engineering und Architektur-entwurf. Dies wird daher im Folgenden an dem Beispiel eines Fahrerassistenzsystems (Adaptive Fahrtgeschwindigkeitsregelung „ACC“) veranschaulicht.

Abbildung 4 zeigt Auszüge aus den initialen, für das ACC-System erstellten Anfor-

derungsmodellen sowie die Definition der Systemschnittstellen (also einen Teil des Architekturmodells). Die in Abbildung 4 gezeigten Ziele, Szenarien und Schnittstellen werden zudem mittels entsprechender Schablonen im Detail definiert. Beispielsweise wird in einer Szenarioschablone der konkrete Ablauf des Szenarios „Vorausfahrendes Fahrzeug folgen“ beschrieben. Basierend auf diesen, initialen Modellen schlagen die Systemarchitekten eine initiale, grobe Architektur vor (siehe Abbildung 5).

Die initiale, grobe Architektur definiert fünf wesentliche Subsysteme, die benötigt werden, um die definierten Ziele und Szenarien zu erfüllen. Darüber hinaus sind die Systemschnittstellen den jeweils zuständigen Subsystemen zugeordnet.

Um die Eignung der initialen Architektur für die definierten, initialen Anforderungen festzustellen sowie gegebenenfalls neue Ideen für weitere Anforderungen zu erhalten, erfolgt nun eine Zuordnung und ein Abgleich zwischen Zielen, Szenarien und Architektur. Damit eine Zuordnung, beispielsweise von Zielen zu Subsystemen, möglich wird, müssen die Ziele zunächst jedoch verfeinert werden. Ein Auszug aus dem verfeinerten Zielmodell des ACC-Systems wird in Abbildung 6 gezeigt. Die Ziele an den Blättern dieses Zielmodells werden einzelnen Subsystemen zugewiesen und definieren somit initiale Anforderungen für diese Subsysteme. Dies stellt, wie oben beschrieben, den Übergang zur nächsten Systemebene, d.h. von der Gesamtsystemebene zur Subsystemebene, dar.

Bei der Verfeinerung der Ziele und Szenarien und der Abbildung in die Systemarchitektur werden zum Beispiel fehlende Ziele, fehlende Szenarien oder fehlende Architekturelemente aufgedeckt. Beispielsweise ist der Umstand, dass einem Subsystem keine Ziele zugeordnet wurden,

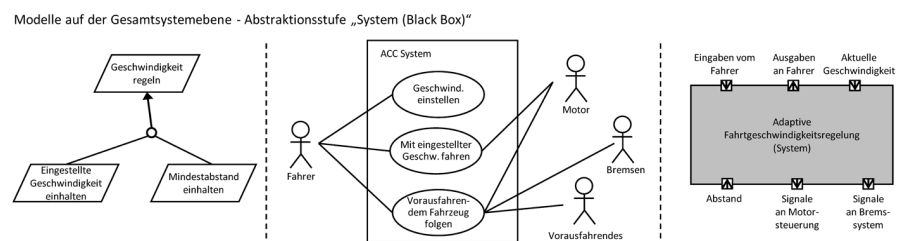


Abb. 4: Auszüge aus dem Zielmodell, Szenariomodell und Architekturmodell auf der Gesamtsystemebene

Grobes Architekturmodell auf der Subsystemebene - Abstraktionsstufe „Funktionale Dekomposition“

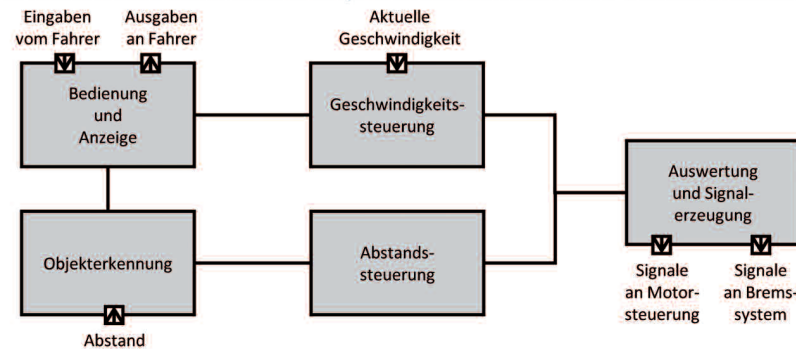


Abb. 5: Beispiel einer initialen, groben Systemarchitektur auf der Subsystemebene

ein Hinweis auf bisher übersehene Systemziele oder auf eine nicht auf die Anforderungen abgestimmte Architektur. Im Anschluss an die Verfeinerung der Ziele und Szenarien und die Abbildung in die Systemarchitektur erfolgt daher eine Überarbeitung und Anpassung der Anforderungs- und Architekturmodelle.

Weitere Informationen und Ausblick

COSMOD-RE wird bereits erfolgreich in der Praxis eingesetzt. Basierend auf den Erfahrungen aus der Anwendung wird die Methode zudem kontinuierlich weiterentwickelt. Gegenwärtig wird COSMOD-RE in einen durchgängigen, modellbasierten Ansatz zur Entwicklung softwareintensiver eingebetteter Systeme, inklusive einer (prototypischen) Werkzeugunterstützung, integriert. Diese Arbeiten werden in der

Innovationsallianz „Software Plattform Embedded Systems 2020“ (SPES2020) durchgeführt und vom Bundesministerium für Bildung und Forschung unter dem Kennzeichen 01IS08045 gefördert. Nähere Informationen zu SPES2020 sind unter [BMBF] zu finden. Die in diesem Artikel vorgestellte COSMOD-RE-Methode erscheint demnächst in einer ausführlicheren Fassung als Teil des Lehrbuchs und Kompendiums „Requirements Engineering – Foundations, Principles, Techniques“. Aktuelle Informationen zu diesem Werk werden unter [Poh] und [SSE] bekannt gegeben.

Wenn Sie Fragen oder Anregungen zu diesem Artikel haben, sollten Sie nicht zögern, die Autoren zu kontaktieren. Eine Gelegenheit, mit den Autoren persönlich zu diskutieren, bietet sich zum Beispiel bei dem Workshop ENVISION2020 [Env] im

Rahmen der deutschen Software-Engineering-Tagung in Paderborn ebenso wie bei der International Working Conference on Requirements Engineering (REFSQ 2010 [Ref]) in Essen.

Referenzen

- [SSE]** Arbeitsgruppe Software Systems Engineering der Universität Duisburg-Essen. <http://www.sse.uni-due.de>
- [UML]** Object Management Group: Unified Modeling Language. <http://www.uml.org>
- [ISML]** Object Management Group: Systems Modeling Language. <http://www.omg.sysml.org>
- [AvL]** Axel van Lamsweerde: Requirements Engineering – From System Goals to UML Models to Software Specifications. Wiley, 2009
- [BMBF]** BMBF Innovationsallianz „Software Plattform Embedded Systems 2020“. <http://www.spes2020.de>
- [Poh]** Klaus Pohl: Requirements Engineering – Grundlagen, Prinzipien, Techniken. Webseite zum Lehrbuch und Kompendium. <http://www.re-buch.de>
- [Env]** Workshop zur Zukunft der Entwicklung softwareintensiver, eingebetteter Systeme. <http://www.envision2020.eu>
- [Ref]** International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2010). <http://www.refsq.org>

Auszug aus dem Zielmodell auf der Subsystemebene - Abstraktionsstufe „Funktionale Dekomposition“

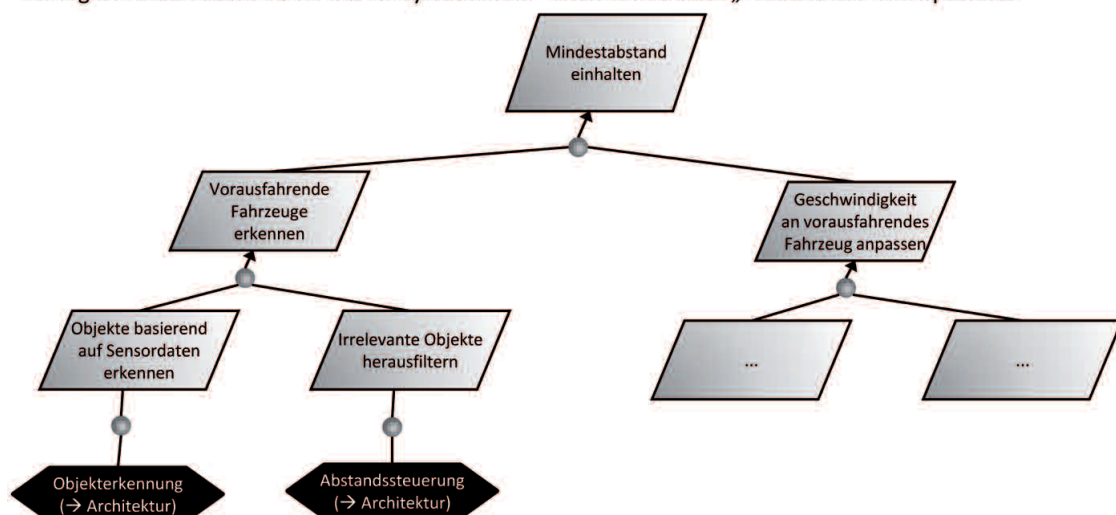


Abb. 6: Beispiel für die Verfeinerung eines Systemziels und Zuordnung der Teilziele zu Subsystemen