

BEITRÄGE ZUR ERHÖHUNG DER QUALITÄT KOMPLEXER AVIONIK-SYSTEME DURCH MODELLGETRIEBENE ANFORDERUNGSBASIERTE ENTWICKLUNGSMETHODEN

T. Jungebloud (tino.jungebloud@tu-ilmenau.de)

J. Werner (johannes.werner@tu-ilmenau.de)

V. Zerbe (volker.zerbe@tu-ilmenau.de)

Technische Universität Ilmenau, Postfach 10 05 65, 98684 Ilmenau, Deutschland

Zusammenfassung

Die rein textuelle Erhebung und Dekomposition von Anforderungen (*Requirements Based Engineering*, RBE) birgt das Risiko von Ungenauigkeiten und Widersprüchen innerhalb der Spezifikationsdokumente. Unzulänglichkeiten führen zur Realisierung eines fehlerbehafteten Systemverhaltens. Dies wird jedoch häufig erst während der Integrationsphase erkannt und erfordert kostenintensive Gegenmaßnahmen. Die rein modellbasierte Systementwicklung (*Model Based Engineering*, MBE) begünstigt die frühzeitige Validierung und Verifikation des Entwurfes gegen das erwünschte Verhalten. Die Wahl eines passenden Abstraktionsgrades erfordert allerdings ein hohes Maß an Erfahrung im Bereich der Modellbildung. Darüber hinaus besteht das Problem, dass der deutlich erhöhte Mehraufwand zu Beginn der Entwicklung nicht die nötige Akzeptanz findet. Die Kombination beider Ansätze in einer neuen Entwicklungsmethodik hat das Potenzial, bewährte und akzeptierte Prozesse beizubehalten. Gleichzeitig können die Vorteile der modellbasierten Entwicklung zur Stabilisierung des Systementwicklungsprozesses genutzt werden.

1 EINLEITUNG

Die Planung und Realisierung komplexer Luft- und Raumfahrt-Systeme stellt höchste Anforderungen an die beteiligten Entwickler und den zugrunde liegenden Entwicklungsprozess.

Die derzeit dominierende Vorgehensweise in der Entwicklung umfangreicher Hardware-/Software-Systeme ist rein anforderungsbasiert. Allerdings deuten beispielsweise die Verzögerungen des Airbus A380 darauf hin, dass mit den etablierten anforderungsbasierten Entwicklungsmethoden derart komplexe verteilte Projekte kaum noch handhabbar sind (z.B. [1], [2]). Auch die jüngste Berichterstattung bezüglich diverser Probleme und Fehleinschätzungen im Zusammenhang mit dem Airbus A400M legt ähnliche Schlüsse nahe (z.B. [3]).

Häufig wird zur Stabilisierung der Entwicklungsprozesse die Ablösung der anforderungsbasierten Vorgehensweise durch modellbasierte Methoden postuliert (z.B. [4]).

Im Gegensatz dazu verfolgt das vorliegende Papier den Grundansatz, die Vorzüge beider Verfahrensweisen zu kombinieren und somit einen modellgetriebenen anforderungsbasierten Prozess zu etablieren.

Um an die Thematik heranzuführen, werden die beiden grundlegenden Verfahrensausprägungen im Folgenden zunächst skizziert und bewertet.

Der Hauptteil beschreibt die Kombination beider Methoden. Dabei stehen insbesondere die

Maßnahmen zur Stabilisierung der anforderungsbasierten Verfahrensweisen im Vordergrund.

Im letzten Teil fasst ein Fazit die Inhalte zusammen und bewertet die entwickelte Methodik. Ein Ausblick schließt das Papier ab.

2 ANFORDERUNGSBASIERTE ENTWICKLUNGSMETHODIK

Im Zentrum der anforderungsbasierten Entwicklung steht die Erstellung und Dokumentation der Anforderungen an ein System in Form textueller Spezifikationen. Zu diesem Zweck erfolgt zu Beginn eines Projektes die Erfassung der erwünschten grundlegenden Fähigkeiten eines Systems (Top-Level-Anforderungen, TL-Anforderungen).

Auf Basis dieser TL-Anforderungen wird das System in kleinere, besser beherrschbare Teile dekomponiert. Den einzelnen Teilsystemen werden dabei die jeweils zugehörigen Anforderungen zugewiesen. Falls aufgrund der strukturellen Zerlegung des Gesamtsystems keine Zuweisung zu einem einzelnen Teilsystem erfolgen kann, so wird für die entsprechende Anforderung ebenfalls eine Dekomposition durchgeführt. Hierbei wird die TL-Anforderung so verfeinert, dass im Anschluss eine klare Trennung der Teilsysteme möglich ist. Um die Nachvollziehbarkeit der Anforderungsdekomposition zu wahren, muss jeder Verfeinerungsschritt

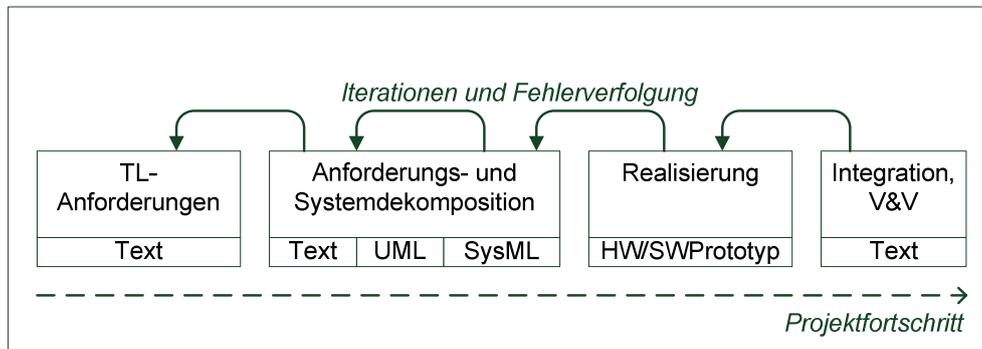


BILD 1: Schema anforderungsbasierte Entwicklung

dokumentiert und dadurch die Verknüpfung zur ursprünglichen Anforderung hergestellt werden (Traceability).

Falls nach einer Anzahl von Dekompositionsschritten die Teilsysteme einen handhabbaren Umfang erreicht haben, können diese an geeignete Expertenteams zur weiteren Bearbeitung vergeben werden.

In weiteren Verfeinerungsschritten erfolgt hier unter Umständen eine fortschreitende Zerlegung komplexer Anforderungen.

Nach dem Erreichen einer geeigneten Anforderungsgranularität kann die technische Umsetzung des beschriebenen Verhaltens erfolgen (Realisierung/Implementierung).

Die vollständig realisierten Komponenten können daraufhin einzeln auf Fehlerfreiheit (Verifizierung) und die vollständig sowie korrekte Umsetzung des spezifizierten Verhaltens (Validierung) überprüft werden (V&V auf Komponentenebene). Anschließend erfolgt ein Zusammenfügen der Komponenten zu komplexeren Teilsystemen (Integration) sowie das Überprüfen des resultierenden Systemverhaltens (Integrationstest, V&V auf Systemebene) ([5]).

Es sei angemerkt, dass der Begriff der textuellen Anforderungen einen Einsatz unterstützender und klärender Grafiken nicht ausschließt. So kommen beispielsweise häufig Diagrammtypen der UML (Unified Modeling Language, [6]) oder der SysML (System Modeling Language, [7]) zum Einsatz, um die innerhalb einer Anforderung formulierten Zusammenhänge zu präzisieren. Da in der Reinform der anforderungsbasierten Entwicklung Diagramme jedoch in der Regel einen illustrierenden Charakter aufweisen, handelt es sich bei besagten UML-/SysML-Diagrammen nicht um Modellelemente im Sinne des vorliegenden Papiers.

BILD 1 fasst die wichtigsten Prozessschritte der anforderungsbasierten Entwicklungsmethodik nochmals kompakt zusammen.

Wie bereits erwähnt wurde, stellt die anforderungsbasierte Entwicklung die derzeit dominierende Prozessausprägung dar. Die Methodik ist vielfach erprobt und gilt daher als anerkannt.

Neben den vorhandenen Erfahrungswerten können weitere Vorzüge der anforderungsbasierten Entwicklung angeführt werden. Zum einen ist der

Einsatz spezieller Werkzeuge nicht zwingend erforderlich, jedoch angeraten, insbesondere um in komplexen Projekten die Traceability der Anforderungen zu gewährleisten. Des Weiteren ist in frühen Phasen noch keine Kenntnis über später eingesetzte formale Entwurfsmethoden notwendig, was eine interdisziplinäre Zusammenarbeit zu Beginn auf den ersten Blick fördert.

Allerdings liegt in dieser wenig formalisierten Form der Zusammenarbeit auch der zentrale Schwachpunkt dieses Entwicklungsansatzes. Aufgrund unterschiedlicher fachlicher Kenntnisse und Erfahrungen beinhalten Anforderungen häufig Annahmen und implizite Informationen, die sich fachfremden Personen nicht erschließen. Darüber hinaus ist natürlich-sprachlicher Text in der Regel nicht präzise genug, insbesondere um Unvollständigkeiten, Inkonsistenzen und Widersprüche in komplexen Systemen zu erkennen oder zu vermeiden.

Die so entstehenden Fehler innerhalb der Spezifikation führen im weiteren Verlauf zu nicht spezifiziertem oder fehlerhaftem Verhalten. Da diese Unzulänglichkeiten allerdings erst erkannt werden können, nachdem bereits eine Realisierung stattgefunden hat, stellt sich deren Beseitigung häufig als sehr kosten- und zeitintensiv heraus.

3 MODELLBASIERTE ENTWICKLUNGSMETHODIK

Zur Beseitigung der Nachteile der anforderungsbasierten Entwicklungsmethodik wird häufig der Einsatz von Modellen postuliert (z.B. [4]).

Der Modellbegriff findet allerdings in diversen Themengebieten wie beispielsweise der Mathematik oder Physik (mathematische Modelle), der Psychologie (Verhaltensmodelle) oder der Informatik bzw. Ingenieurwissenschaften (Software- oder Systemmodelle) Anwendung. Aus diesem Grund soll eine kurze Begriffsbestimmung den Sprachgebrauch innerhalb des vorliegenden Papiers festlegen.

Im Allgemeinen besitzen Modelle drei zentrale Eigenschaften:

1. **Abbildung:** Ein Modell ist die Repräsentation eines Originals, das selbst wieder ein Modell sein kann.

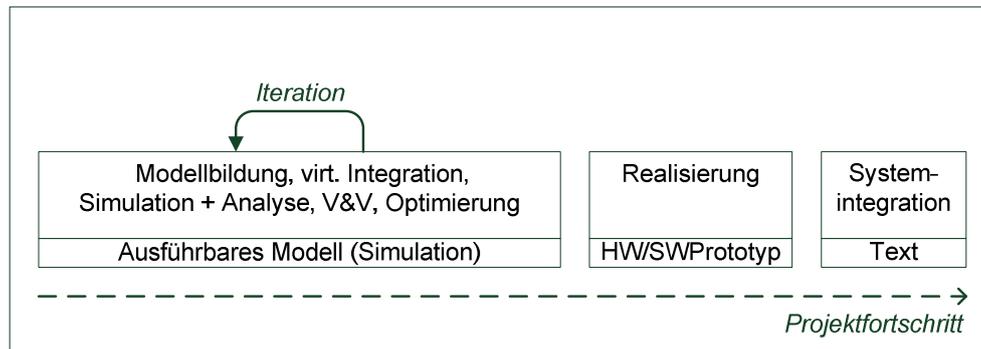


BILD 2: Schema modellbasierte Entwicklung

2. **Verkürzung:** Modelle geben nicht alle Eigenschaften des Originals wieder, sondern nur die, welche dem Zweck der Systemanalyse dienen.
3. **Pragmatik:** Das Modell und dessen Detailgrad werden vom angestrebten Erkenntnisgewinn bestimmt. Hier ist beispielsweise eine Unterscheidung in statische Modelle (nicht simulierbar) und dynamische Modelle (simulierbar) möglich.

Ein Modell ist somit die abstrakte Beschreibung eines Systems für einen bestimmten Zweck der Analyse. Für verschiedene Analyseziele können daher die Modelle ein und desselben Systems vollkommen unterschiedlich sein ([8]).

Wie bereits angedeutet, werden innerhalb der anforderungsbasierten Entwicklung häufig Diagramme der UML genutzt. Diese stellen per Definition bereits statische Modelle dar. Da hier jedoch die Möglichkeit der Simulation nicht gegeben ist, wird im Rahmen dieses Papieres der Einsatz dynamischer Modelle bevorzugt und mit dem Begriff der modellbasierten Entwicklung versehen.

Im Zentrum der modellbasierten Methodik steht ein ausführbares experimentierbares Modell zur Analyse des komplexen Systems. Dieses wird zu Beginn eines Projektes mit einem sehr hohen Abstraktionsgrad erzeugt (Modellierung, Modellbildung, hier: Blackbox) und iterativ immer weiter verfeinert (Modellierung, Modellbildung, hier: Grey- und Whitebox). In jeder Iteration ist dabei der jeweils angestrebte Analysezweck maßgeblich für den notwendigen Detailgrad.

Neben dem eigentlichen System erfolgt auch eine Modellbildung eventueller Einsatzszenarien (Missionen) inklusive der jeweiligen Umweltaspekte und Störeinflüsse. Auch hier gilt der besagte Zusammenhang zwischen Analysezweck und Detailgrad.

In Analogie zur anforderungsbasierten Methodik kann auch in der modellbasierten Entwicklung eine Vergabe einzelner Blackbox-Teilmodelle zur weiteren Ausarbeitung an jeweils eigenständige Expertenteams stattfinden.

Nach Fertigstellung einer Modellversion für den angestrebten Analysezweck kann nun eine simulative Untersuchung stattfinden. Diese kann einerseits der Validierung und Verifikation des dynamischen Verhaltens dienen. Andererseits können Leistungsbewertungen dieses Verhaltens durchgeführt werden, z.B. um etwaige Ressourcenengpässe aufzudecken oder Designvarianten gegenüberzustellen. Die Ergebnisse der Analyse fließen im Anschluss wieder in das Modell zurück.

Ist nach mehreren Iterationen das System hinreichend detailliert modelliert, so kann die tatsächliche Realisierung stattfinden. Anschließend besteht die Möglichkeit, umgesetzte Teile einzeln sowie kombiniert zu validieren und zu verifizieren. Da das dynamische Verhalten allerdings in der Regel bereits im Modell geprüft wurde, fällt der Aufwand hier wesentlich geringer aus.

Bild 2 fasst den Prozess des modellbasierten Entwurfes grafisch zusammen.

Der Vorteil modellbasierter Entwicklung ist die frühzeitige Überprüfbarkeit des dynamischen Systemverhaltens. In diesem Zusammenhang können Reaktionen auf Störeinflüsse und Umweltszenarien geprüft sowie Designalternativen gegenübergestellt werden.

Allerdings erfordert dies den Einsatz mathematisch validierter Werkzeuge zur Modellierung und Simulation, in denen die notwendigen Ausführungsmodelle (z. B. Discrete Event, DE, Finite State Machine, FSM, Synchronous Data Flow, SDF) umgesetzt sind.

Nicht nur die notwendige Einarbeitung in neue Werkzeuge zur geeigneten Unterstützung begründet die teils geringe Akzeptanz unter erfahrenen Systementwicklern. Auch der Paradigmenwechsel hin zu einem umfangreicheren Definitions- und Designprozess zu Projektbeginn trägt entscheidend dazu bei.

4 MODELLGETRIEBENE ANFORDERUNGSBASIERTE ENTWICKLUNGSMETHODIK

Wie die vorangegangenen Ausführungen aufgezeigt haben, existieren in der anforderungsbasierten Methodik fundamentale Schwachpunkte bezüglich der konsistenten, widerspruchsfreien und vollständigen Beschreibung. Ferner existiert mit dieser Methodik keine Möglichkeit zur frühzeitigen Validierung des dynamischen Systemverhaltens.

Auf der anderen Seite leidet die Reinform der modellbasierten Entwicklung unter Akzeptanzproblemen.

Die im Folgenden skizzierte Methodik der Systementwicklung hat zum Ziel, die Vorzüge beider Methoden zu kombinieren. Zu diesem Zweck wird im Zentrum des Verfahrens weder auf Anforderungen noch auf Modelle verzichtet, sondern vielmehr eine geeignete Zusammenführung beschrieben.

4.1 Anforderungserfassung mit UML- und SysML-Unterstützung

Das Erfassen der Anforderungen beginnt auf Basis der erwünschten grundlegenden Fähigkeiten des Systems (TL-Anforderungen). Analog zur anforderungsbasierten Entwicklungsmethodik erfolgt anschließend eine schrittweise Verfeinerung und somit die Dekomposition in kleinere, besser beherrschbare Teilsysteme.

In Ergänzung zur etablierten Verfahrensweise wird vom vorgeschlagenen Prozess allerdings ein wesentlich umfangreicherer Einsatz der UML und SysML zur Qualitätssteigerung der interdisziplinären Kommunikation gefordert. Ferner liegt das Ziel der Nutzung teilformaler grafischer Beschreibungsmittel wie UML/SysML in der Bereitstellung eines gesamtheitlichen und konsistenten Anforderungsdatenmodells des Systems.

Auf diese Weise kann schon zu frühen Projektphasen insbesondere die Widerspruchsfreiheit zwischen den einzelnen Anforderungsdokumenten erreicht werden.

4.2 System- und Anforderungsdekomposition mit SysML-Unterstützung

In Analogie zum anforderungsbasierten Prozess wird anschließend eine systematische Zerlegung durchgeführt. Hierbei erfolgt eine schrittweise Dekomposition über Systeme und Subsysteme bis hin zu elementaren Funktionen (Top-Down).

Im Gegensatz zur anforderungsbasierten Vorgehensweise soll nun jedoch die Arbeit auf textbasierten Anforderungen von Beginn an deutlich stärker durch einen hierarchisch strukturierten Anforderungsbaum unterstützt werden. Dies stellt insbesondere eine eindeutig nachvollziehbare Dekomposition sicher.

Zu diesem Zweck kommt die Notation des SysML-Anforderungsdiagrammes ([7]) zum Einsatz. Diese Beschreibungsform bietet den zentralen Vorteil, dass Abhängigkeiten wesentlich besser abbildbar sind. Durch das Erzwingen einer konsistenten baumartigen Struktur (vgl. BILD 3) wird die Traceability der Anforderungen gesteigert. Analog zu Abschnitt 4.1 ist eine präzisierende Annotation der einzelnen Anforderungsknoten mit weiteren UML- und SysML-Diagrammen sinnvoll.

4.3 UML-Modellierung des elementaren Systemverhaltens

Nach erfolgreicher Zerlegung der TL-Anforderungen bis auf die elementare Ebene können die ermittelten Funktionen nun durch UML-Verhaltensdiagramme wie z.B. Aktivitätsdiagramme oder Zustandsautomaten [6] umgesetzt werden.

Nichtfunktionale Anforderungen wie z.B. Leistungskriterien bleiben hingegen zunächst in textueller Form erhalten. Diese müssen später in der Simulation ausgewertet werden, da sie wichtige Parameter zur Validierung darstellen.

4.4 Validierung und Verifikation des elementaren Systemverhaltens

Als Ergebnis der Systemdekomposition liegen in der untersten Ebene elementare Anforderungen vor. Den funktionalen Anteil bilden dabei UML-Modelle ab, nichtfunktionale Parameter sind annotiert.

Auf Basis der UML-Verhaltensdiagramme erfolgt eine isolierte Validierung und Verifikation der elementaren Funktionen. Ein geeignetes Modellierungswerkzeug vorausgesetzt, kann dieser Arbeitsschritt direkt in der UML-

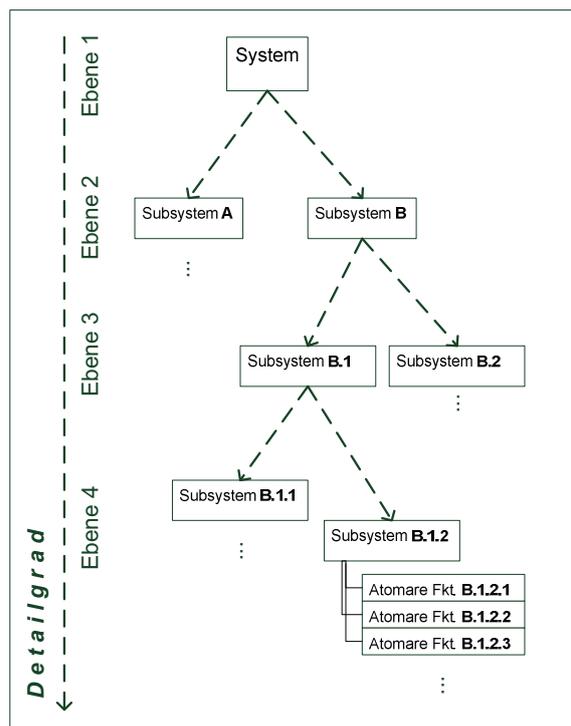


BILD 3: Baumstruktur der Systemdekomposition

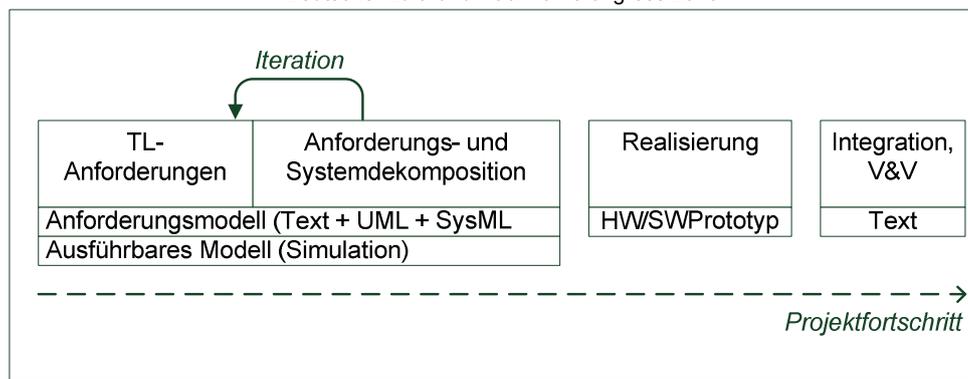


BILD 4: Schema modellgetriebene anforderungsbasierte Entwicklung

Modellierungsumgebung erfolgen. Alternativ könnte das in Abschnitt 4.5 unerläßliche Simulationswerkzeug Anwendung finden.

Mit dem Nachweis der korrekten Funktionsweise, der korrekten Umsetzung des spezifizierten Verhaltens sowie der Einhaltung eventuell geforderter nichtfunktionaler Randbedingungen ist die Untersuchung des elementaren Verhaltens (tiefste Ebene der Anforderungshierarchie) abgeschlossen. Eventuell aufgedeckte Fehler gilt es, in die UML-Modelle sowie bei Bedarf auch in die Anforderungen zurückfließen zu lassen.

4.5 Integration und V&V im Modell („virtueller Prototyp“)

Nach erfolgreicher Validierung und Verifikation der elementaren Funktionen können diese in aufsteigender Richtung des Anforderungsbaumes integriert werden (Bottom-Up).

Auf diesem Weg werden sukzessive immer größere Teilbäume der Anforderungshierarchie mit modelliertem Verhalten gefüllt. Nicht hinreichend detaillierte Teilbäume werden gegebenenfalls zunächst als Blackbox-Modelle belassen.

Nach Fertigstellung eines Teilbaumes kann eine simulative Validierung des dynamischen Verhaltens der jeweiligen Teilsysteme stattfinden. Auch Leistungsbewertungen sind möglich. Die Ergebnisse dieser Analyse fließen im Anschluss wieder in das Modell sowie in die Anforderungen zurück.

Mittels einer geeigneten Simulationsumgebung ist es nun auch möglich, das dynamische Verhalten jedes Teilsystems auf jeder Dekompositionsebene zu überprüfen. Wie bereits angedeutet wurde, können hierbei Teilbäume mit der bereits ausmodellierten tatsächlichen Funktion versehen werden. Es ist jedoch ebenso möglich, Teilbäume als Blackboxes und ausschließlich mit minimaler Funktionalität vorzuhalten, sofern dies der Zweck der simulativen Analyse zulässt.

Aus Gesamtsystemsicht bildet dieses ausführbare Modell einen „virtuellen Prototypen“, der alle bereits modellierten Funktionen enthält. Alle bis dato nicht enthaltenen Funktionen werden durch Blackboxes mit minimalem Verhalten ersetzt. Eine Simulation dieses Gesamtmodells kann zu jedem Zeitpunkt Aufschlüsse über das dynamische Verhalten des Systems liefern, wobei die Qualität der

Analyseergebnisse vom Grad der bereits als Whitebox vorliegenden Teile abhängt.

4.6 Realisierung des modellierten Systemverhaltens

Ist das System in allen Teilbäumen modelliert, verifiziert und validiert, findet die tatsächliche Realisierung statt.

Da das Modell bereits während der Simulation ausgeführt wurde, ist hier ein hoher Grad an Automatisierung möglich. So können die UML-Modelle beispielsweise zur Erzeugung von Quellcode oder Hardwarelogik Wiederverwendung finden.

Eine anschließende Validierung und Verifikation der Realisierung kann in der Regel wesentlich weniger umfangreich gestaltet werden als in der rein anforderungsbasierten Entwicklung, da das dynamische Verhalten bereits im Modell untersucht wurde.

BILD 4 fasst den entwickelten Prozess in zeitlicher Reihenfolge nochmals zusammen.

5 ZUSAMMENFASSUNG

Die vorgestellte Methodik der modellgetriebenen anforderungsbasierten Entwicklung ist insbesondere durch den Einsatz von UML- und SysML-Modellstrukturen sowie deren simulative Analyse gekennzeichnet. Sie bietet den entscheidenden Vorteil, die Validierung und Verifikation bereits vor Beginn der eigentlichen Realisierung durchführen zu können. Im Bedarfsfall ist so eine weniger arbeits- und kostenintensive Fehlerbeseitigung in den frühen Phasen der Systementwicklung möglich.

Trotz dieser grundlegenden Veränderung in der etablierten Verfahrensweise kann die vorgestellte Kombination anforderungs- und modellbasierter Methoden in bereits bestehende anforderungsbasierte Entwicklungsprozesse integriert werden.

6 AUSBLICK

Die skizzierte modellgetriebene anforderungsbasierte Methodik kann und muss effektiv durch geeignete Entwicklungsumgebungen unterstützt werden.

So muss es beispielsweise möglich sein, einfach und konsistent die geforderte Baumstruktur der

Anforderungsdekomposition zu erzeugen. Des Weiteren ist es notwendig, dass in eben diesen Anforderungsbaum auch UML-Diagramme als Blätter eingepflegt und konsistent mit anderen Diagrammen verknüpft werden können. Darüber hinaus sollten eine effiziente Speicherung und ein Export in XML [9] oder einem ähnlichen XML-basierten Datenformat vorgesehen sein, um einen hohen Grad an Interoperabilität und Automatisierungen zu ermöglichen.

Dieses Nutzungsprofil bedienen laut Datenblatt die meisten der etablierten CASE-Tools (Computer-Aided Software Engineering). Somit ist auch in Bezug auf das einzusetzende Werkzeug eine relativ nahtlose Integration in vorhandene Entwicklungsprozesse zu erwarten. Neben der toolunterstützten Erzeugung, Verwaltung und Speicherung des Anforderungsmodells erfordert der neue Prozess eine geeignete Simulationsumgebung. Diese muss die Möglichkeit bieten, die Spezifikationen auszuführen und so zu verifizieren und zu validieren.

Bezüglich einer derartigen Umgebung sprechen die Fähigkeit zum Umgang mit einer Vielzahl verschiedener Ausführungsmodellen sowie die Datenhaltung in einem XML-Format für das

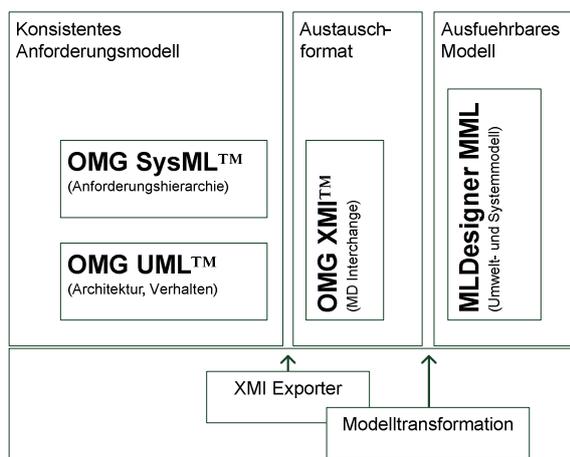


BILD 5: Workflow von Anforderungsmodell zum ausführbaren Modell

Simulationswerkzeug MLDesigner ([10]).

In BILD 5 ist der entwickelte Workflow von Anforderungsmodellierung bis zur Ableitung des ausführbaren Modells nochmals grafisch veranschaulicht.

Da die prototypische Umsetzung des skizzierten Arbeitsablaufes inklusive der Werkzeugunterstützung sowie der Schnittstellen derzeit noch nicht beendet ist, kann eine abschließende Bewertung des in Aussicht gestellten Workflows zu diesem Zeitpunkt noch nicht erfolgen.

7 LITERATURVERZEICHNIS

- [1]. **EADS Corporate Communications.** Airbus hält am Zieltermin für die A380-Zulassung und Erstauslieferung zum Jahresende fest, kündigt aber Verschiebung im Produktionsprogramm an. [Online] 2006. http://www.eads.net/1024/de/pressdb/archiv/2006/2006/Airbus/20060613_airbus_a380.html.
- [2]. —. Airbus bestätigt weitere Verzögerung bei der A380 und leitet unternehmensweite Umstrukturierungsmaßnahmen ein. [Online] 2006. http://www.eads.net/1024/de/pressdb/archiv/2006/2006/Airbus/20061003_airbus_a380_delay.html.
- [3]. **Flottau, Jens.** A400M Problems Range Far Beyond Engines - Aviation-Week. [Online] The McGraw-Hill Companies, Inc, 2010. http://www.aviationweek.com/aw/generic/story_channel.jsp?channel=defense&id=news/A400M012309.xml.
- [4]. **Salzwedel, Horst.** Mission Level Design of Avionics. *AIAA-IEEE DASC 04 - The 23rd Digital Avionics Systems Conference 2004.* 2004.
- [5]. **Rupp, Chris.** *Requirements-Engineering und -Management.* 2004. 3446228772.
- [6]. **Object-Management-Group.** OMG Unified Modeling Language. [Online] <http://www.uml.org>.
- [7]. —. OMG Systems Modeling Language. [Online] <http://www.omgsysml.org>.
- [8]. **Wikarski, Prof. Dr.** *Systemanalyse I, Aufgaben- und Organisationsanalyse.* 2002.
- [9]. **Object-Management-Group.** Catalog of OMG Modeling and Metadata Specifications. [Online] 2010. http://www.omg.org/technology/documents/modeling_spec_catalog.htm#XML.
- [10]. **MLDesign Technologies, Inc.** MLDesign Technologies, Inc. [Online] <http://www.mldesigner.com>.