TWT GmbH
Science & Innovation

Verification and Testing to Support Functional Safety Standards (VeTeSS)

Projekt-Abschlussbericht

TWT GmbH
Science and Innovation
Ernsthaldenstr. 17
70565 Stuttgart
Tel: +49.7 11.21 57 77.0

Sperrvermerk: nein
Gesperrt bis: nicht zutreffend

Dezember 2015

Verification and Testing to Support Functional Safety Standards (VeTeSS)

Projekt-Abschlussbericht

Das diesem Bericht zugrunde liegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS120001C gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

TWT GmbH Science & Innovation:

Geschäftsführung:	Dr. Dimitris Varziotis
Innovation Management:	Dr. Markus Pfeil
Projektmanagement:	Dr. Stefan Rieger
Dokumentation:	Dr. Stefan Rieger

Inhaltsverzeichnis

1 Kurzdarstellung 1
1.1 Aufgabenstellung 1
1.2 Voraussetzungen unter denen das Vorhaben durchgeführt wurde 2
1.3 Planung und Ablauf des Vorhabens 2
1.4 Wissenschaftlich-technischer Stand 4
1.5 Zusammenarbeit mit anderen Stellen 6
2 Eingehende Darstellung 7
2.1 TS 1 Informationsmanagement 7
2.2 AP 3 Anforderungsdefinition und -analyse 12
2.3 AP 4 Systemintegration, -verifikation und -test 13
2.4 AP 5 Verifikation und Testen von Hardwarekomponenten 20
2.5 AP 6 Verifikation und Testen von Softwarekomponenten 25
2.6 AP 7 Fallstudien und Demonstratoren 29

1 Kurzdarstellung

Neue Sicherheitsstandards wie ISO 26262 stellen für Unternehmen, die sicherheitskritische eingebettete Systeme produzieren, eine Herausforderung dar. Verifikation und Validierung werden heute oft manuell durchgeführt. Dabei wird für digitale und analoge Komponenten, sowie Hard- und Software unterschiedlich vorgegangen.

Das Ziel des VeTeSS-Projekts war die Entwicklung standardisierter Werkzeuge und Methoden zur Verifikation von sicherheitsrelevanten Systemen im Automobilbereich. Partner der gesamten Produktionskette sowie Forschungseinrichtungen haben gemeinsam automatisierte, quantitative Prozesse und Methoden für alle Phasen der Entwicklung konzipiert.

Durch die steigende Komplexität und Anzahl eingebetteter Systeme und zugehöriger Software im Automobil ergibt sich ein erhöhter Aufwand in der Absicherung und Verifikation, der sich signifikant auf Entwicklungskosten und –zeit niederschlägt. Hier ist insbesondere die Entwicklung von E/E-Komponenten zu nennen, die in verschiedenen Systemen und Anwendungsbereichen eingesetzt werden. Dies ist beispielsweise für Embedded-ECUs und den entsprechenden Prozessoren der Fall. Im Kontext der ISO 26262 werden solche Komponenten als „Safety Element out of Context“ (SEooC) besonders behandelt.

Die im VeTeSS-Projekt entwickelten Ansätze wirken dem Aufwandsanstieg entgegen und zielen insbesondere auch auf SEooC-Komponenten ab. Davon profitiert vor allem auch die europäische Zulieferer-Industrie, so dass Anbieter Standardkomponenten für verschiedene Anwendungsfelder und nicht nur Lösungen, die auf die speziellen Anforderungen eines bestimmten Kunden zugeschnitten sind, bereitstellen können. VeTeSS fokussiert insbesondere auf den strategisch wichtigen Automobilmarkt. Erfahrungs- und Wissensaustausch mit anderen Industriesektoren, die ähnliche Anforderungen haben, ist im Rahmen der Disseminationsaktivitäten erfolgt, beispielsweise mit dem ITEA2-Projekt openETCS, das im Bahnbereich angesiedelt ist.

Die VeTeSS-Projektergebnisse sind sowohl auf konventionelle als auch auf elektrische und Hybridfahrzeuge anwendbar und erreichen neben einer Steigerung der Sicherheit, Qualität und Zuverlässigkeit, eine Verbesserung der Wettbewerbsfähigkeit der europäischen Hard- und Softwareindustrie im Automobilbereich.

1.1 Aufgabenstellung

Das Ziel des Teilvorhabens bestand in der Konzeption und Evaluation von Methoden und Ansätzen zum Informations- und Anforderungsmanagement, zur Verifikation und Validierung von Systemen, Hardware und Software im Kontext von ISO 26262. Die von TWT im Projekt entwickelten methodischen Ansätze und Prozesse gehen dabei über den Stand der Technik hinaus, indem eine ganzheitliche Betrachtung der Entwicklung von E/E-Systemen zugrunde gelegt wird und bestehende technologische und wissenschaftliche Methoden in realen Anwendungsszenarien untersucht und die industrielle Einsetzbarkeit durch einen erhöhten Reifegrad oder eine Integration von bisher unabhängigen Methoden vorangetrieben wird.

Im Rahmen des Informations- und Anforderungsmanagements wurden beispielsweise Richtlinien und Umsetzungsempfehlungen definiert, die einen höheren Grad der Nachverfolgbarkeit

und Automatisierung des ISO 26262-Entwicklungsprozesses ermöglichen, sowie standardisierte Vorgaben für die Definition von Sicherheitsanforderungen und ihre Verknüpfung mit den Entwicklungsartefakten erarbeitet. Für die Systemverifikation wurde ein Ansatz zur formalen Verifikation von Echtzeitsystemen erarbeitet, der auf eine typische Toollandschaft im Automobilbereich (MatLab/Simulink) aufsetzt, die bisher keine formalen Analysemethoden unterstützt. Hier ergibt sich sowohl ein hohes Einsparpotenzial von Kosten als auch eine signifikante Verbesserung der Absicherung, da Designfehler identifiziert werden können, die durch reguläres Testen nur sehr schwer zu finden sind.

Ein von TWT entwickelter, simulativer Ansatz zur Verifikation der elektromagnetischen Verträglichkeit (EMV), der auch in Form eines prototypischen Software-Werkzeugs umgesetzt wurde dient der physikalischen Absicherung beim Systementwurf. Dieser Ansatz wurde mit einem abstrakten Steuergeräte-Hardware-Modell gekoppelt, um den Effekt von elektromagnetischer Inferenz in Zuleitungen von Steuergeräten zu simulieren. Die Simulationen wurden mittels realen Versuchsaufbauten validiert.

Auf Hardwareebene wurde zusätzlich ein Ansatz für stochastische Hardwaremodelle konzipiert. Dieser erlaubt es beispielsweise, Ausfallraten für Hardwarekomponenten basierend auf den jeweiligen Raten der Teilkomponenten zu berechnen. Dies kann etwa zur Bestimmung der benötigten Auslegung von redundanten Komponenten genutzt werden.

Für die Softwareentwicklung und Verifikation von Softwarekomponenten wurden Abstraktionsebenen definiert, die bestehende Standard-Konzepte aus der modellbasierten Softwareentwicklung („Model-Driven Architecture“ der OMG), dem AUTOSAR-Standard und der ISO-Norm vereinen. Diese sind insbesondere für SEooC-Software-Komponenten von Bedeutung. Für die Verifikation hardwarenaher und sicherheitsrelevanter Software wurde ferner ein Verfahren für das modellbasierte Testen entwickelt, das eine frühe Sicherheitsanalyse von Softwaredesigns ermöglicht.

1.2 Voraussetzungen unter denen das Vorhaben durchgeführt wurde

Das Vorhaben wurde auf Basis von relevanten Vorarbeiten im Bereich des Anforderungsmanagements, der formalen Verifikation von Software und auf Systemebene, des modellbasierten Testens, stochastischer Analysemethoden und Kenntnissen im Bereich der Simulation elektrischer Signale und elektronischer Komponenten durchgeführt. Die einzelnen Ansätze wurden signifikant erweitert und im Kontext der ISO 26262 miteinander integriert, um einen ganzheitlichen Ansatz für die verschiedenen Phasen eines ISO 26262-konformen Entwicklungsprozesses zu erhalten.

Für die Validierung des prototypischen Simulationswerkzeugs zur EMV-Analyse war ein realer Versuchsaufbau im Labor notwendig. Dies wurde in Kooperation mit dem Projektpartner Infineon Austria erreicht. Die restlichen Verfahren wurden anhand von industriellen Anwendungsfällen validiert.

1.3 Planung und Ablauf des Vorhabens

Abbildung 1 stellt eine Übersicht der inhaltlichen Arbeitspakete (AP) mit dem Haupt-Informationsfluss dar. Neben den AP 1 (Projektmanagement) und 2 (Verwertung und Öffentlichkeitsarbeit) war TWT

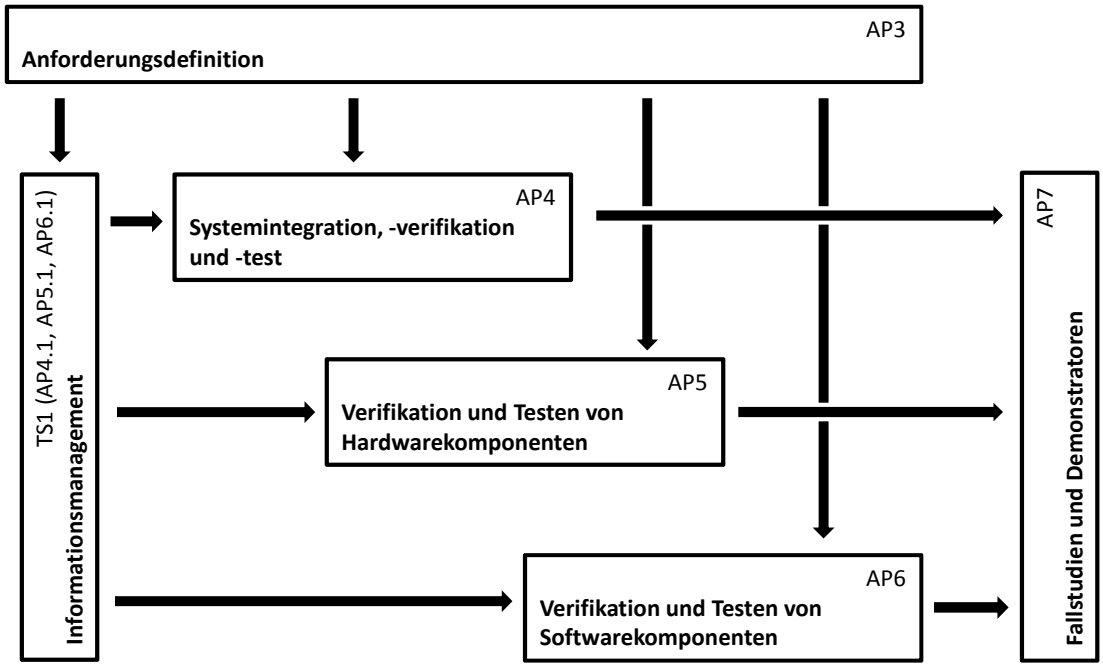

Abbildung 1: Struktur der inhaltlichen Arbeitspakete mit Informationsfluss

an den inhaltlichen AP 3-7 beteiligt. Begonnen wurde mit der Anforderungsdefinition in AP 3, das die Anforderungen für die AP 4, 5 und 6 sowie den *Technical Stream* 1 (Informationsmanagement) festlegt. Unter Technical Stream (TS) wird in VeTeSS ein AP-übergreifendes Thema verstanden. In den Hauptarbeitspaketen 4, 5 und 6 existiert für jeden Technical Stream ein Task. TS 1 hat eine gewisse Sonderstellung, da hier eine besonders enge Kollaboration stattfand und gemeinsame Liefergegenstände AP-übergreifend erarbeitet wurden. TWT hatte die Leitung von TS 1 inne.

Im Rahmen von TS 1 wurden AP- und prozessübergreifende Konzepte und Richtlinien zum Informationsmanagement erarbeitet, die für die inhaltlichen Arbeiten in den AP 4, 5 und 6 eine Rolle spielten. Dies betrifft beispielsweise die Nachverfolgbarkeit von Anforderungen, die Notation von Anforderungen und Testfällen, die Verwendung von semi-formalen Notationen in Anforderungen oder Prozesse zur organisationsübergreifenden Kommunikation im ISO 26262-Entwicklungsprozess.

Die in den AP 4, 5 und 6 erarbeiteten Methoden, Konzepte und Tools setzen die in AP 3 definierten Anforderungen um und wurden im Rahmen von AP 7 in industriellen Anwendungsfällen validiert. Aufgrund des Feedbacks der Anwendungspartner wurden die Ansätze entsprechend angepasst, um eine möglichst gute Abdeckung einer realen ISO 26262-Umsetzung zu erzielen.

1.4 Wissenschaftlich-technischer Stand

Schätzungen zufolge werden für Verifikation und Test einer elektronischen Komponente 50-70% des Entwicklungsaufwands benötigt. Hinzu kommt, dass Verifikationsprozesse schwierig und weniger effektiv sind als gemeinhin angenommen wird: Eine Studie von Electric Cloud in Kooperation mit Osterman Research zeigt, dass ein Großteil von Softwarefehlern auf unzureichende Testprozesse oder Limitierungen der Infrastruktur zurückzuführen sind und weniger auf Designfehler [11]. Vollständig automatisierte Testumgebungen sind selten, nur ungefähr 12% der Softwareentwicklungsunternehmen nutzen vollautomatisierte Testsysteme. Fast 10% berichteten, dass sämtliche Tests manuell durchgeführt werden.

Obwohl diese Studie Softwareentwicklung und Testen in einer Vielzahl von Industriezweigen, und nicht die Entwicklung von sicherheitsrelevanten Systemen im Speziellen beleuchtet, hat eine interne Studie von Organisationen im VeTeSS-Konsortium bestätigt, dass das Fehlen von standardisierten und automatisierten Prozessen für die Verifikation sicherheitsrelevanter Systeme ein schwerwiegendes Problem ist. Die Verifikation und der Test von Sicherheitsanforderungen erfolgt größtenteils manuell oder allenfalls teilautomatisiert.

1.4.1 Informationsmanagement im Kontext der ISO 26262

Dem Informationsmanagement kommt im Kontext der ISO 26262 eine große Bedeutung zu; es ist für die Erbringung des Sicherheitsnachweises essentiell und muss eine lückenlose Nachverfolgbarkeit der Systementwicklung beginnend mit den Sicherheitsanforderungen, über das System-, Hardware- und Softwaredesign bis hin zur Implementierung und zur Verifikation der initialen Sicherheitsanforderungen ermöglichen. In diesem Umfeld gibt es unzählige Ansätze und Werkzeuge, die jedoch nach derzeitigem Kenntnisstand keinen standardisierten und durchgängigen Ansatz bieten.

Meist bleibt es bei „Insellösungen“ , die bestimmte Teile des ISO 26262-Entwicklungszyklus abdecken, jedoch nicht in einen standardisierbaren, durchgängigen Prozess eingebettet werden.

1.4.2 Formale Verifikation von Echtzeiteigenschaften von Systemdesigns

In der Literatur gibt es zahllose Ansätze zur formalen Verifikation von Systemdesigns, die jedoch selten auf die im Automobilbereich in der Praxis sehr oft anzutreffenden MatLab/Simulink-Modelle anwendbar sind. Bestehende Ansätze haben oft Einschränkungen. So hat etwa der von MathWorks bereitgestellte Simulink Design Verifier [21] signifikante Schwächen, was die Art der zu verifizierenden Eigenschaften anbelangt, da er Nebenläufigkeit und die Verifikation von Echtzeiteigenschaften unzureichend unterstützt [20]. Auch der Ansatz von Barnat et al. [5] unterstützt nicht die Verifikation von Echtzeiteigenschaften.

Der Model Checker UPPAAL [6] setzt auf den Formalismus der *Timed Automata* [2] zur Verifikation von Echtzeiteigenschaften auf, unterstützt jedoch keine Simulink-Modelle und besitzt eine eigene Modellbeschreibungssprache.

1.4.3 Analyse elektromagnetischer Verträglichkeit

Die ISO 26262-Norm empfiehlt den Einsatz von Analysemethoden zur Bestimmung der elektromagnetischen Verträglichkeit auf Systemebene für alle ASIL-Stufen im Kontext der Robustheitsanalyse als auch der Analyse systematischer und sekundärer Fehler auf System und- Hardwareebene. Standardisierte Testmethoden zur Verifikation der Robustheit von Systemen und Komponenten hinsichtlich elektromagnetischer Interferenz (EMI) werden im Standard referenziert.

In Abhängigkeit des ASIL sind maximale Fehlerraten vorgegeben, die mit quantitativen Methoden wie Fehlerbaumanalyse ("Fault Tree Analysis" -- FTA) oder durch die Anwendung von Fehlermetriken bestimmt werden [23]. Um den quantitativen Einfluss von Ausfällen aufgrund gemeinsamer Ursachen wie EMI zu untersuchen, schlägt der Standard IEC 61508, auf dem die ISO 26262 basiert, die Anwendung einer β-Faktor-Analyse vor. Diese Analyse wird beispielsweise von den Werkzeugen FaultTree+ [17] und Medini Analyze [18] unterstützt.

Es gibt einige Software-Werkzeuge, die sich für die Modellierung und Simulation von EMI von Fahrzeugkomponenten eignen. Dazu gehören beispielsweise EMC Studio [12] und FEKO [1]. Beide Werkzeuge erlauben die Definition eines Kabelbaums, die Integration von Schaltungen und die Simulation der Interferenz in Form von parasitären Spannungen und Strömen sowie kritischen Frequenzbereichen. Andere hybride Werkzeuge wie COMSOL Multiphysics [9] ermöglichen die korrekte Simulation von EMI basierend auf der Physik des Elektromagnetismus.

Alle gennanten Werkzeuge sind für die Untersuchung von EMI mit gegebener Geometrie geeignet. Die Resultate einer solchen Simulation können zur Definition und Verfeinerung modellbasierter Kontrakte verwendet werden.

1.4.4 Stochastische Hardwaremodelle

Stochastische Modellierung und Analyse erlauben die explizite Berücksichtigung von Unsicherheit in einer quantitativen Analyse von Hardwaredesigns. Beispielsweise ist die Signallaufzeit auf einem Chip vom Herstellungsprozess abhängig und zeigt statistische Abweichungen, die als Zufallsvariable angesehen werden können. Für die Integration von stochastischen Fehlerraten in Hardwaremodelle gibt es derzeit keinen allgemein in der Praxis verbreiteten Ansatz, wohl aber für die quantitative stochastische Analyse. Hier ist eines der am weitesten verbreiteten Softwarewerkzeuge PRISM, das auf *Continuous-Time Markov Chains* (CTMCs) aufsetzt [19].

1.4.5 Abstraktionsebenen in der (eingebetteten) Softwareentwicklung

In der Industrie hat sich eine Vielzahl von Definitionen für Abstraktionsebenen für die Softwareentwicklung etabliert. Zu nennen ist hier insbesondere ein bestehendes Standard-Konzept aus der modellbasierten Softwareentwicklung: Die „Model-Driven Architecture" (MDA) der OMG [22]. Im Umfeld der eingebetteten Systeme für den Automobilbereich hat sich der AUTOSAR-Standard durchgesetzt, der eine Referenzarchitektur und standardisierte Abstraktionsebenen definiert. AUTOSAR und MDA wurden bisher getrennt voneinander behandelt und noch nicht in einem gemeinsamen Konzept vereinigt. Weiterhin ist noch die ISO 26262-Norm zu nennen, die ebenfalls Richtlinien vorgibt.

1.4.6 Modellbasierte Sicherheitsanalyse und Fehlerinjektion

Eine Herausforderung bei der Entwicklung sicherheitsrelevanter, eingebetteter Software im Automobilbereich ist das Testen von Sicherheitsmechanismen in Bezug auf Hardware- und Softwarefehler. Der Sicherheitsstandard ISO 26262 definiert Sicherheitsmechanismen als technische Lösungen, um Fehler zu erkennen und zu kontrollieren, um das System in einen sicheren Zustand zu bringen. Aktuelle Ansätze für das Testen von Sicherheitsmechanismen erfordern die Injektion von Fehlern in laufende Systeme, die aus einem Microcontroller und der dazugehörigen Software bestehen.

Hier ergibt sich das Problem, dass Fehler auch in die Zielhardware selbst injiziert werden müssen, was einen komplexen und kostenintensiven Prozess zur Folge hat. Ein Grund dafür sind die aufwändige Spezifikation von Tests und Testprotokollen sowie die notwendigen Iterationen in allen Entwicklungszyklen.

Es gibt viele Studien zur modellbasierten Sicherheitsanalyse basierend auf formalen Methoden. Einige konzentrieren sich auf die Untersuchung von fehlertoleranten Systemen, wie beispielsweise der Ansatz von Bruns und Sutherland [8] zur Verifikation von Kommunikationsprotokollen. Die Autoren von [16] zielen auf die Verifikation von Fehlertoleranz in Systemen ab, die als Zustandsautomaten formalisiert sind.

In [7] wird eine Methode für modellbasiertes Integrationstesten vorgestellt, die vor der eigentlichen Systementwicklung basierend auf formalen Modellen durchgeführt wird. Die Autoren wenden den Ansatz jedoch nicht im sicherheitsrelevanten Automobilbereich an und fokussieren mehr auf Systemintegration und nicht auf eingebettete Software. Hänsel et al. beschreiben einen Ansatz zur Testfallgenerierung von Netzwerken von *Timed Automata*, die auf ein Bremssystem angewandt werden [15]. Sie stoßen jedoch bei der Testfallgenerierung mittels des Werkzeugs UPPAAL CoVer auf auf Ressourcenprobleme und entwickeln einen evolutionären Algorithmus.

1.5 Zusammenarbeit mit anderen Stellen

TWT hat sich im Rahmen der Projektaktivität mit allen beteiligten Projektpartner intensiv ausgetauscht. Insbesondere die Kooperation mit Infineon Deutschland und Österreich, IKV und den Universitäten Oxford und Wien war sehr intensiv. Die in den einzelnen Arbeitspaketen entwickelten Verfahren und Ansätze wurden dabei in enger Zusammenarbeit mit Infineon Deutschland und Österreich anhand industrieller Einsatzszenarien evaluiert und bewertet.

Ein Austausch mit anderen Projekten fand ebenfalls statt. So hat TWT unter anderem gemeinsam mit Projektpartnern aus dem ITEA2-Projekt openETCS, das im Bahnbereich angesiedelt ist, einen Workshop auf der internationalen Konferenz INDIN 2013 organisiert, der zu einem intensiven, domänenübergreifenden Austausch geführt hat.

2 Eingehende Darstellung

In Folgenden werden die Aktivitäten, Arbeitsschritte und Ergebnisse der inhaltlichen[1] Arbeitspakete des Vorhabens beschrieben und in Relation zum wissenschaftlich-technischen Stand der Dinge gesetzt. Ferner wird der voraussichtliche Nutzen der Ergebnisse skizziert.

2.1 TS 1 Informationsmanagement

Der *Technical Stream* 1 Informationsmanagement liefert Beiträge zu den AP 4, 5 und 6. Die Anforderungen an das Informationsmanagement wurden in AP 3 definiert. Die in TS 1 erarbeiteten Inhalte wurden auch im Kontext von AP 7 evaluiert. TWT hatte die Leitung von TS 1 inne.

2.1.1 Aktivitäten, Arbeitsschritte und Ergebnisse

Im Laufe des Projekts wurden viele Gemeinsamkeiten zwischen den Teil-AP 4.1, 5.1 und 6.1 (= TS 1) identifiziert, so dass die entsprechenden Arbeiten schließlich AP-übergreifend durchgeführt und die Liefergegenstände der einzelnen Teil-AP als gemeinsame Dokumente bereitgestellt wurden. Dies führte zu einer sehr engen Zusammenarbeit im Verlaufe des Projekts.

Im Rahmen von TS 1 hat TWT in enger Zusammenarbeit mit den Projektpartnern die folgenden Ergebnisse erzielt, die den Weg für ein einheitliches und klar definiertes Informationsmanagement im Sinne der ISO 26262 ebnen.

Definition für die Begriffe Sicherheitsanforderung und Sicherheitsargumentation
Sicherheitsanforderungen sind Anforderungen, die Elementen eines *Items* im Sinne der ISO 26262 zugeordnet werden. Sicherheitsanforderungen sollten aus zwei Komponenten bestehen:

1. Einer funktionalen Komponente, die angibt, wie eine Gefahr (ISO 26262: *Hazard*) vermieden werden kann, und
2. einer Integritätskomponente, die angibt, wie oft die zugehörige Gefahr auftreten darf.

Anforderungen, die keine Gefahren reduzieren oder vermeiden und damit Sicherheit garantieren, sind keine Sicherheitsanforderungen. Beide Arten von Anforderungen werden als erfüllt angesehen, wenn der Verifikationsprozess einen Nachweis der Annahme (= Anforderung) liefert.

Die *Sicherheitsargumentation* ist der Nachweis, dass ein Entwicklungsprozess eines gegebenen Projekts ausreichend sicher ist. Dies basiert auf den folgenden drei Schritten:

1. Vorbereitung des Projekts und Definition eines sogenannten *Development Interface Agreement* (DIA), das nach der ISO 26262 die Verteilung der Verantwortung zwischen Fahrzeughersteller (*Original Equipment Manufacturer* - OEM) und den Zulieferern festlegt.

[1] AP 1 (Projektmanagement) und AP 2 (Öffentlichkeitsarbeit und Verwertung) werden in diesem Kapitel nicht betrachtet, da sie selbst keine technischen Ergebnisse liefern.

2. *Safety Management* während der Entwicklung, bei dem einzelne Aktivitäten identifiziert und ISO 26262-Anforderungen auf die ASIL-Zuordnung der/des einzelnen *Item(s)* zugeschnitten werden
3. Dokumentation der Entwicklungsaktivitäten, wo eine entsprechende Dokumentation gemäß den Erkenntnissen aus Punkt 2. erforderlich ist.

Einheitliches Metamodell für Sicherheitsanforderungen Um Sicherheitsziele und -Anforderungen für die Systementwicklung und -verifikation einheitlich zu handhaben, wurde ein Framework konzipiert, das in Form eines Metamodells einen Standard für Sicherheitsanforderungen vorgibt. Insbesondere wird die Struktur von Sicherheitsanforderungen, wie ihre Attribute und die Beziehung zu anderen Artefakten, wie beispielsweise Testfällen, festgelegt. Das Metamodell wurde in Form eines UML-Modells definiert. In Abbildung 2 ist es visualisiert.

Zusätzlich zum Metamodell für Sicherheitsanforderungen wurde ein Metamodell für Referenzen (*Traces*) und den Sicherheitsnachweis (*Safety Evidence*) erarbeitet.

Abstraktionsebenen für Sicherheitsanforderungen Abstraktionsebenen definieren eine hierarchische Ordnung für den Detailgrad eines/einer in Entwicklung befindlichen Systems, Hardware oder Software. Dabei wurden in TS 1 unterschiedliche Abstraktionsebenen für System, Hardware und Software definiert. Die Resultate stellen eine Generalisierung des in AP 6.2 von TWT entwickelten Konzepts der Abstraktionsebenen für Software dar (siehe dazu auch Abschnitt 2.5).

Vorgehensweise zum Dokumentenmanagement Für das Dokumentenmanagement im Sicherheits-Lebenszyklus (ISO 26262: *Safety-Lifecycle*) ist ein konsequenter Dokumentenmanagement-Prozess für jede Entwicklung zu definieren, um die Verfügbarkeit und Qualität der Dokumentation sicherzustellen. Dafür muss der Gesamtprozess geplant und organisiert werden. Dokumente müssen präzise und klar sein. Weiterhin sollten sie im Rahmen eines Konfigurationsmanagements verwaltet werden, beispielsweise, um exakte Referenzen im Rahmen von Gutachten und für bestimmte Versionen eines Systems zu ermöglichen. Weiterhin ist die korrekte und vollständige Festlegung von Meta-Daten notwendig.

Prozess für die organisationsübergreifende Kommunikation Die Kommunikation von Anforderungen und V&V-Aktivitäten ist eine essentielle Aufgabe im Rahmen der Entwicklung sicherheitsrelevanter Komponenten, sofern mehrere Organisationen an der Entwicklung beteiligt sind (z.B. OEM und Zulieferer). Im Rahmen von VeTeSS wurde dazu ein Prozess formal in der Notation BPMN definiert. Abbildung 3 zeigt mit der Kommunikation von sicherheitsrelevanten Informationen einen Ausschnitt aus der erarbeiteten Prozessdefinition.

Leitfaden zum Anforderungsmanagement Anforderungen für die Entwicklung sicherheitsrelevante Systeme zu schreiben ist eine wichtige aber nicht-triviale Aufgabe. In VeTeSS wurden Ansätze eines Leitfadens zum Schreiben textueller Anforderungen erarbeitet. Diese Anforderungen müssen eindeutig, verständlich, atomar, konsistent, plausibel und verifizierbar sein. Anhand von Beispielen werden typische Probleme beim Schreiben textueller Anforderungen gegeben.

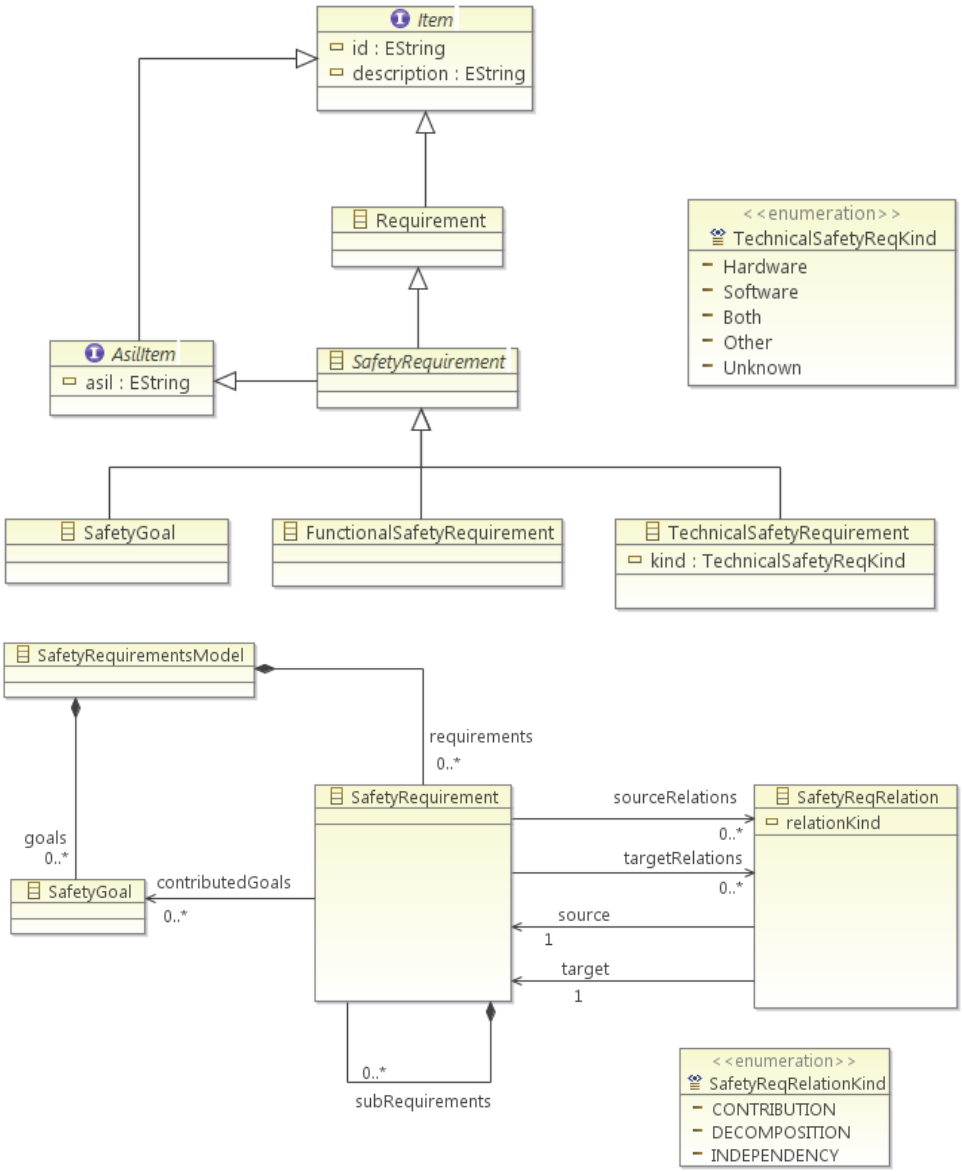

Abbildung 2: Metamodell für Sicherheitsanforderungen

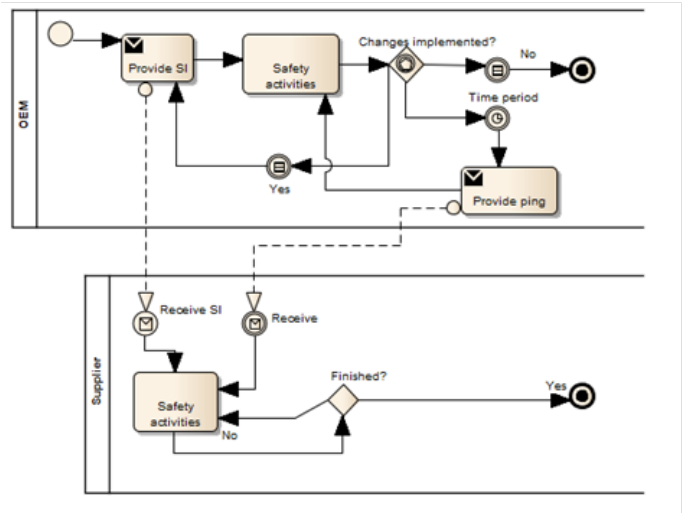

Abbildung 3: Kommunikation von sicherheitsrelevanten Informationen (in BPMN)

Weiterhin wurden Methoden zur Formalisierung von Anforderungen untersucht. Dies steht in Bezug zu VeTeSS TS 3 *Formale Methoden*. Hierzu wurde der in [13] beschriebene Ansatz sowie das im CESAR-Projekt [10] erarbeitete Konzept näher untersucht. Außerdem wurden auch semi-formale Notationen hinsichtlich ihrer Eignung für die Anforderungsspezifikation und der Unterstützung verschiedener Abstraktionsebenen untersucht. Folgende semi-formale Notationen spielten in der Untersuchung eine Rolle:

- UML
  - MARTE-Profil
  - Profil für Anforderungsmanagement
- URML
- SysML
- EAST-ADL
- ReqIF
- Programmiersprachen

Die Bewertung der Notationen fand anhand von Metriken wie Kosten, Grad der Standardisierung, Langzeit-Wartung/Unterstützung, Verbreitungsgrad und sinnvollen Abstraktionsebenen zur Anwendung der Notation eine Rolle. Eine abschließende Bewertung über den Einsatz in industriellen Projekten kann projektspezifisch anhand der Metriken erfolgen.

Schließlich wurden Möglichkeiten zur Handhabung von Anforderungen über Unternehmensgrenzen hinweg aufgezeigt, die die Nachverfolgbarkeit der Anforderungen sicherstellen.

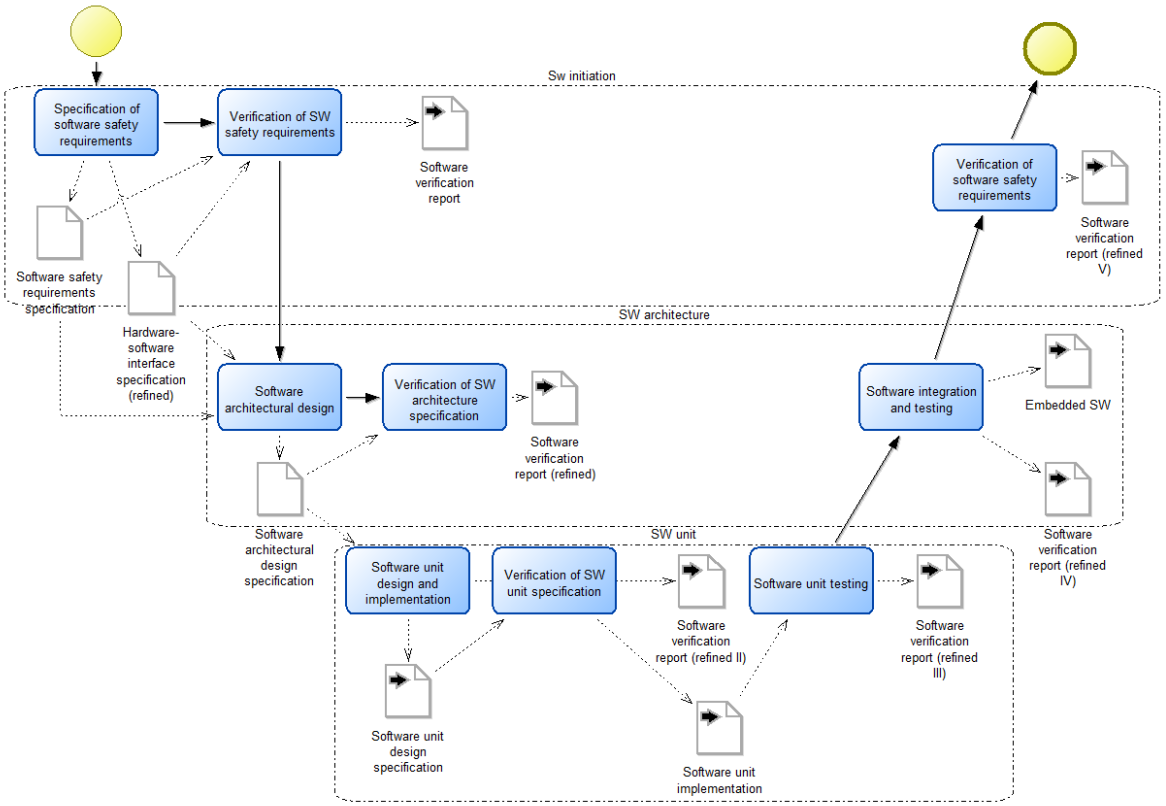

Abbildung 4: V&V-Prozess auf Softwareebene

Management von V&V-Aktivitäten Für jede Phase im Sicherheits-Lebenszyklus wurde ein Prozess erarbeitet, der beschreibt, wie Aktivitäten und Arbeitsprodukte miteinander in Beziehung stehen, um die Validierung und Verifikation von Anforderungen sicherzustellen. Der definierte V&V-Gesamtprozess setzt Entwicklungsartefakte mit Anforderungen und Verifikations- und Testaktivitäten in Beziehung. In Abbildung 4 ist dies exemplarisch für die Software-Ebene dargestellt. Ferner wurde eine (Meta-)Struktur für die Attribute von Testfällen vorgeschlagen.

2.1.2 Voraussichtlicher Nutzen der Ergebnisse

Die erarbeiteten Ergebnisse decken einen großen Teil des ISO 26262-Entwicklungsprozesses ab und ermöglichen daher ein standardisiertes Informationsmanagement, das die wesentlichen Anforderungen an sicherheitsrelevante Entwicklungen erfüllt. Bisher war kein derart ganzheitlicher Ansatz vorhanden.

TWT wird, basierend auf den obigen Ergebnissen, bestehenden Kunden Beratungsdienstleistungen zur Optimierung bestehender und zum Aufsetzen neuer Informationsmanagement-Lösungen im Kontext sicherheitsrelevanter Entwicklungen anbieten. Weiterhin ist längerfristig geplant, einzelne

Ansätze und Erkenntnisse, beispielsweise das Metamodell für Sicherheitshanforderungen, Methoden (semi-)formaler Anforderungsspezifikation und den Prozess zur organisationsübergreifenden Kommunikation von Sicherheitsinformationen, in neue Software-Produkte im Kontext des Systems Engineering zu integrieren.

2.1.3 Fortschritte auf dem Gebiet des Vorhabens

Fortschritte wurden vor allem hinsichtlich der ganzheitlichen Abdeckung des ISO 26262-Entwicklungsprozesses erreicht, so dass große Teile des Informationsmanagements basierend auf den in TS 1 entwickelten Konzepten und Ansätzen umsetzbar sind. Die Praxistauglichkeit wurde im Rahmen von AP 7 nachgewiesen; Anwendungspartner planen den operativen Einsatz von TS 1-Konzepten, wie etwa dem Anforderungs-Metamodell, in kommenden Projekten.

2.2 AP 3 Anforderungsdefinition und -analyse

Dieses Arbeitspaket diente der Anforderungsdefinition und -analyse für den weiteren Verlauf des VeTeSS-Projekts.

2.2.1 Aktivitäten, Arbeitsschritte und Ergebnisse

Im Rahmen dieses Arbeitspakets hat TWT signifikant zur Definition von Anforderungen hinsichtlich des Informationsmanagements in AP 3.5 beigetragen und weitere Beiträge der Projektpartner zur Anforderungsdefinition integriert. In AP 3.1 hat TWT zur gemeinsamen Projektterminologie beigetragen, sowie in den AP 3.2, 3.3 und 3.4 Anforderungen hinsichtlich formaler Methoden, Modellierung und Simulation sowie Informationsmanagement geliefert.

Die Anforderungen wurden integriert, konsolidiert und mit den Partnern abgestimmt und wurden schließlich in das gemeinsame Anforderungsspezifikationsdokument D3.6 übertragen.

Die Definition eines Metamodells für Sicherheitsanforderungen wurde ebenfalls in diesem AP erarbeitet. Mehr Details dazu finden Sie in Abschnitt 2.1.

2.2.2 Voraussichtlicher Nutzen der Ergebnisse

Die Anforderungen waren im weiteren Verlauf des Projekts für die weiteren Arbeiten in den AP 4, 5, 6 und 7 notwendig. Ein großer Teil der in AP 3 gesammelten und konsolidierten Anforderungen konnte schließlich bis Projektende abgedeckt werden.

2.2.3 Fortschritte auf dem Gebiet des Vorhabens

Es wurden Fortschritte hinsichtlich der Methodik zur Anforderungsspezifikation erzielt (z.B. das Metamodell). Die wesentlichen Ergebnisse waren jedoch die konsolidierten Anforderungen, die für den weiteren Projektverlauf relevant waren und selbst hauptsächlich als Referenz eine Rolle spielen.

2.3 AP 4 Systemintegration, -verifikation und -test

Dieses Arbeitspaket betrachtet die Verifikations- und Testmethoden auf Systemebene. Die Aktivitäten und Ergebnisse aus AP 4.1 wurden im Rahmen von TS 1 in Abschnitt 2.1 beschrieben. Im Folgenden werden die Arbeiten in den Teil-AP 4.2 und 4.3 näher beleuchtet.

2.3.1 Aktivitäten, Arbeitsschritte und Ergebnisse

Im Rahmen von AP 4.2 hat TWT einen simulativen Ansatz zur Verifikation der elektromagnetischen Verträglichkeit (EMV) und ein zugehöriges prototypisches Software-Werkzeug entwickelt. In AP 4.3 ist das Ergebnis ein Framework zur formalen Verifikation von Echtzeiteigenschaften von Simulink/Stateflow-Modellen.

Workflow und Werkzeug zur EMV-Analyse auf Systemebene Die von TWT entwickelten Methoden und Werkzeuge erlauben es, die EMV auf der Systemebene zu analysieren. Dadurch kann die Stärke der EMI (Elektromagnetische Interferenz) zwischen mehreren Systemelementen schnell und kosteneffizient abgeschätzt und beurteilt werden. Das Werkzeug unterstützt den Systementwickler mit einer qualitativen Vorhersage der EMI in der frühen Stufe der Systementwicklung. Zusätzlich tragen die TWT-Lösungen zum Prozess der automatischen EMV-Optimierungen bei.

Als Eingangsdaten für die qualitative Untersuchung von EMV dient ein vereinfachtes Komponentenmodell des DUT („Device Under Test“), beispielsweise ein Impedanzmodell. TWT definiert und implementiert den Arbeitsablauf zum Verbinden des DUT mit der entsprechenden Verdrahtung zum integrierten System, welches für die Simulation von EMV verwendet werden kann.

Die Ergebnisse der EMV-Analyse können als Basis für die Fehlerinjektion aus der Umgebung des DUT dienen. Darüber hinaus werden die Ergebnisse der EMV-Analyse auch für andere Aufgaben wie zum Beispiel physikalische Tests genutzt.

Das von TWT zu diesem Zweck prototypisch entwickelte Werkzeug *XTT (Cross Talk Tool)* ist MatLab-basiert. Es ermöglicht die Modellierung von Kabeln, die Berechnung der erforderlichen Kopplungsparameter und das Erstellen eines LTspice IV-Modells. Mit diesem Modell können dann Zeit- und Frequenzbereichanalysen durchgeführt werden. LTspice IV ist eine kostenlose Computersoftware, welche SPICE zur Simulation von vordefinierten Schaltungen verwendet. Das Tool unterstützt paralleles Multicore-Rechnen und -Simulieren, wodurch automatische Parameterstudien effizient ablaufen können. Die LTspice IV Modelle können sehr einfach auf andere Programme übertragen werden, welche SPICE nutzen, um die Schaltung zu simulieren.

Die Funktionalität umfasst die Modellierung und Simulation von zwei parallelen Leitungen mit oder ohne dielektrischer Isolationsschicht sowie geschirmter Kabel. Die Quelle für das Störsignal kann modelliert oder aus einer Datenbank importiert werden. Dafür müssen die richtigen Kabelanschlüsse und Lasten spezifiziert werden.

Die Modellierung verlustbehafteter Übertragungsstrecken berücksichtigt sowohl die internen als auch die externen Verluste. Erstere entstehen durch den Serienwiderstand des Leiters und verursachen eine Dämpfung und eine Verzögerung des Signals. Besteht der Leiter aus mehreren Teilleitern können die kapazitive und die induktive Kopplung zwischen den einzelnen Leitern

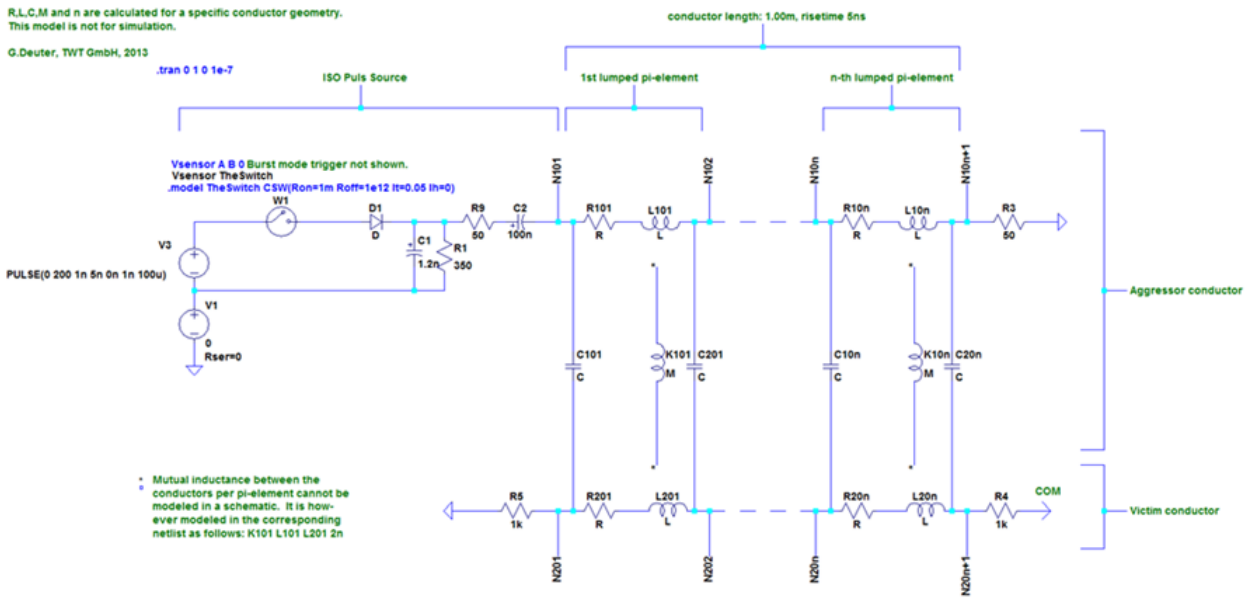

Abbildung 5: Verlustbehaftetes Übertragungsleitungsmodell

ebenfalls als virtuelle interne Verluste betrachtet werden. Externe Verluste werden hauptsächlich durch die induktive Kopplung mit umliegenden Leitern und die kapazitive Kopplung zur Masse oder zu einem Referenzleiter hervorgerufen. Die Kopplungsparameter zwischen den verschiedenen Komponenten werden berechnet, um eine sogenannte diskrete Pi-Element SPICE-Netzliste zu erzeugen. Die Zahl der notwendigen Elemente wird durch die Signalfrequenz und die Länge des Leiters bestimmt.

Um das nahe und ferne Übersprechen zwischen zwei Leitungen zu simulieren, müssen eine Leitung als *Aggressor* und eine andere als *Opfer* definiert werden. Eine schematische Darstellung eines verlustbehafteten Übertragungsleitungsmodells (*Lossy Transmission Line Model* - LTLM) ist in Abbildung 5 dargestellt.

Für das Modellieren der Aggressor-Quelle können sowohl spannungs- als auch stromgetriebene Schalter genutzt werden, um durch geeignete Trigger- und Timingsignale entsprechende Störpulse zu generieren. Die modellierten Quellen werden in einer *XTT*-Bibliothek gespeichert und können während der Konfiguration einer Simulation jederzeit abgerufen werden.

Das prototypische Werkzeug XTT besitzt eine modulare Struktur. Der Benutzer konfiguriert den Kabelbaum, die umgebende Grundgeometrie und die passenden Kabelabschlüsse und wählt die verbundenen Geräte aus. Für die Simulation des gegenseitigen Übersprechens können Randbedingungen wie kritische Spannungsspitzen an bestimmten Orten oder Parametervariationen definiert werden. Zusätzliche Aggressor- und Opferelemente können bei Bedarf ausgewählt bzw. definiert werden. *XTT* berechnet die Koppelparameter, berücksichtigt kritische Randbedingungen, bestimmt die notwendigen Messknoten für eine korrekte Analyse und berechnet die formellen SPICE Netzwerklistenanforderungen aus den Konfigurationen. Diese Netzwerklisten können mit *LTspice IV* oder mit jedem beliebigen anderen SPICE-Simulator geladen werden. *XTT* bietet

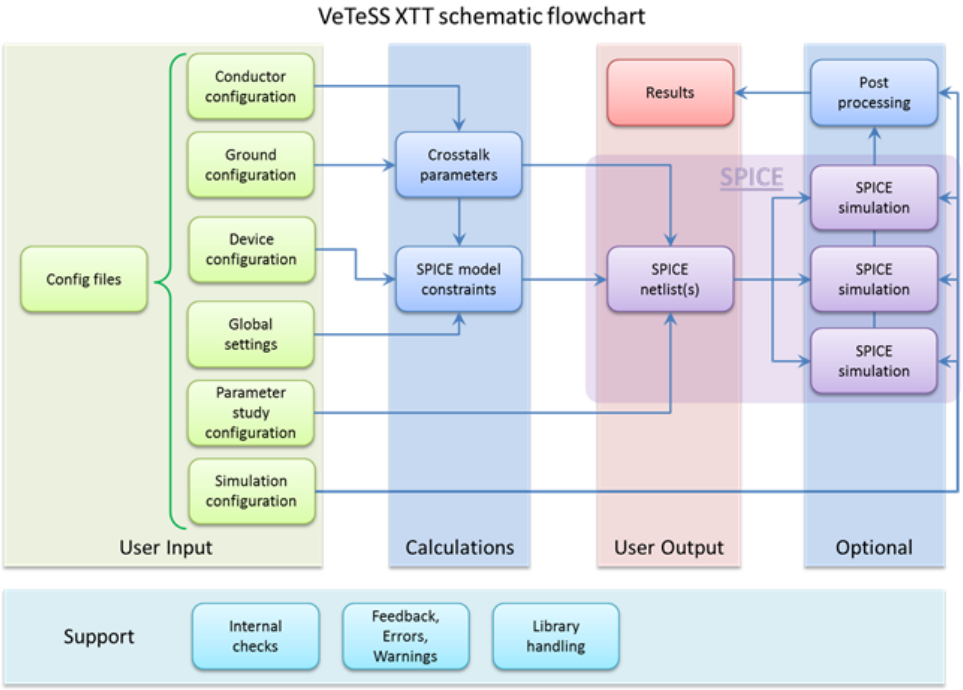

Abbildung 6: Schematische Darstellung der Arbeitsabläufe des Tools XTT.

außerdem eine optionale Schnittstelle, um mehrere Simulationen gleichzeitig parallel ablaufen zu lassen. Nach dem Postprocessing können die Daten gespeichert und geplottet werden. Der gesamte Workflow von XTT ist als grafische Übersicht in Abbildung 6 dargestellt.

Die Berechnung der Koppelparameter kann als multidimensionales, nichttriviales Problem betrachtet werden. Die freie Wahl beliebiger Geometrien, beliebiger Leitertypen sowie die Kombination beider erzeugt eine hohe Anzahl an Variablen. Die Kopplungsparameter, die Selbst- und Gegeninduktivitäten, die Widerstände und die Kapazitäten müssen für jede Kombination der Leiter untereinander und mit der Grundgeometrie bzw. dem Referenzleiter berechnet werden. Ein möglicher Spannungsversatz zwischen Erde und Referenzkabel oder der Schirmung muss ebenfalls berücksichtigt werden. Darüber hinaus können die für die einzelnen Komponenten verwendeten Materialien und somit die zu wählenden Materialkonstanten unterschiedlich sein. Viele dieser Parameter und Materialeigenschaften sind zusätzlich temperatur- und frequenzabhängig.

Alle Signaleingänge müssen sorgfältig geprüft werden. *XTT* selbst enthält Routinen, um kritische Fehler zu vermeiden. Eine Kollisionsprüfung für spezifizierte Geometrien ist ebenfalls implementiert. In der aktuellen Version können ausschließlich parallele Kabel oder Kabelsegmente simuliert werden, sowohl für Einfach- als auch für Koaxialleitungen. Dabei ist die Simulationstemperatur auf Raumtemperatur festgesetzt.

Die Validierung der Berechnungsergebnisse wurde sowohl durch den Abgleich mit realen Messungen als auch durch den Vergleich mit Simulationsergebnissen kommerzieller Softwaretools durchgeführt. Unter Berücksichtigung der Messfehler und Simulationsunsicherheiten zeigen die

Simulationsergebnisse und die Messergebnisse eine gute qualitative Übereinstimmung, wie in Abbildung 7 dargestellt.

Mehr Details finden sich in einer gemeinsamen Veröffentlichung von Infineon Österreich und TWT [3].

Formale Verifikation von Echtzeiteigenschaften Zur formalen Verifikation von Echtzeitsystemen wird ein Formalismus zur abstrakten Modellierung dieser Systeme benötigt, der Zeitaspekte unterstützt. „Klassische Automaten/Kripke-Strukturen" reichen dazu nicht aus. Stattdessen eigenen sich *Timed Automata* sehr gut für diese Aufgabe.

Wie auch beim modellbasierten Testen von Systemen ohne Echtzeitaspekte ist eine spezifische Notation erforderlich, welche die Eigenschaften beschreibt, die am System getestet werden sollen. Es gibt eine Vielzahl von Notationen, um Echtzeiteigenschaften zu modellieren. Dazu zählen beispielsweise Timed CTL (TCTL) oder Timed LTL (TLTL). Erstere ist eine Obermenge der in der UPPAAL-Suite [6] verwendeten Logik. Das UPPAAL-Toolkit wird ständig weiterentwickelt und verfügt sowohl über eine akademische als auch eine kommerzielle Lizenz. Zusätzlich gibt es eine vergleichsweise große Menge an Literatur über *UPPAAL*, inklusive zahlreicher einleitender und industrieller Beispiele. Das UPPAAL-Tool bietet verschiedene Features, die eine verhältnismäßig intuitive Modellierung ermöglichen, insbesondere durch spezielle Kommunikationsvariablen und die Möglichkeit der Festlegung, dass in einem bestimmten Zustand keine Zeit vergehen darf.

Im Automobilbereich finden zur Systemmodellierung vor allem MatLab/Simulink und Stateflow Anwendung, für die es jedoch keine direkte Möglichkeit der formalen Verifikation von Echtzeiteigenschaften gibt. Daher wurde ein prototypischer Ansatz erarbeitet, um dies durch eine geeignete Modelltransformation zum etablierten Werkzeug UPPAAL zu ermöglichen. Dieser basiert auf einem exemplarischen Stateflow-Modell auf Grundlage des Airbag-Anwendungsfalls aus AP 7.

In Abbildung 8 ist das Stateflow-Modell eines Airbag-Aktuators dargestellt, das zunächst manuell nach UPPAAL übersetzt wurde, um einen generischen Ansatz zur Transformation zu erarbeiten. Das Ergebnismodell ist in Abbildung 9 abgebildet; es handelt sich um mehrere, miteinander kommunizierende Timed Automata.

Der Entwurf des Transformationsverfahrens stellte sich als nicht trivial heraus; es galt einige Herausforderungen zu lösen. Dies waren u.a. die folgenden:

- *Hierarchien von Zuständen* werden von Stateflow unterstützt. Allerdings unterstützt UPPAAL dieses Konzept nicht, stattdessen müssen hierarchische Strukturen in flache umgewandelt werden, indem die Verschachtelung durch zusätzliche Zustände aufgelöst wird. Dabei kann es zu einem exponentiellen Anstieg der Anzahl relevanter Zustände kommen, welches zu einem Ressourcenproblem führen kann.
- *Das Zeitkonzept* wird nativ von UPPAAL unterstützt, allerdings nicht von Stateflow. Folglich besitzt das von uns untersuchte Modell keine entsprechenden Informationen. Stattdessen wurden Timing-Anforderungen in natürlicher Sprache verwendet, formalisiert und in das UPPAAL-Modell integriert. Die Verwendung von Timing-Informationen innerhalb von Stateflow-Modellen muss weiter untersucht werden.

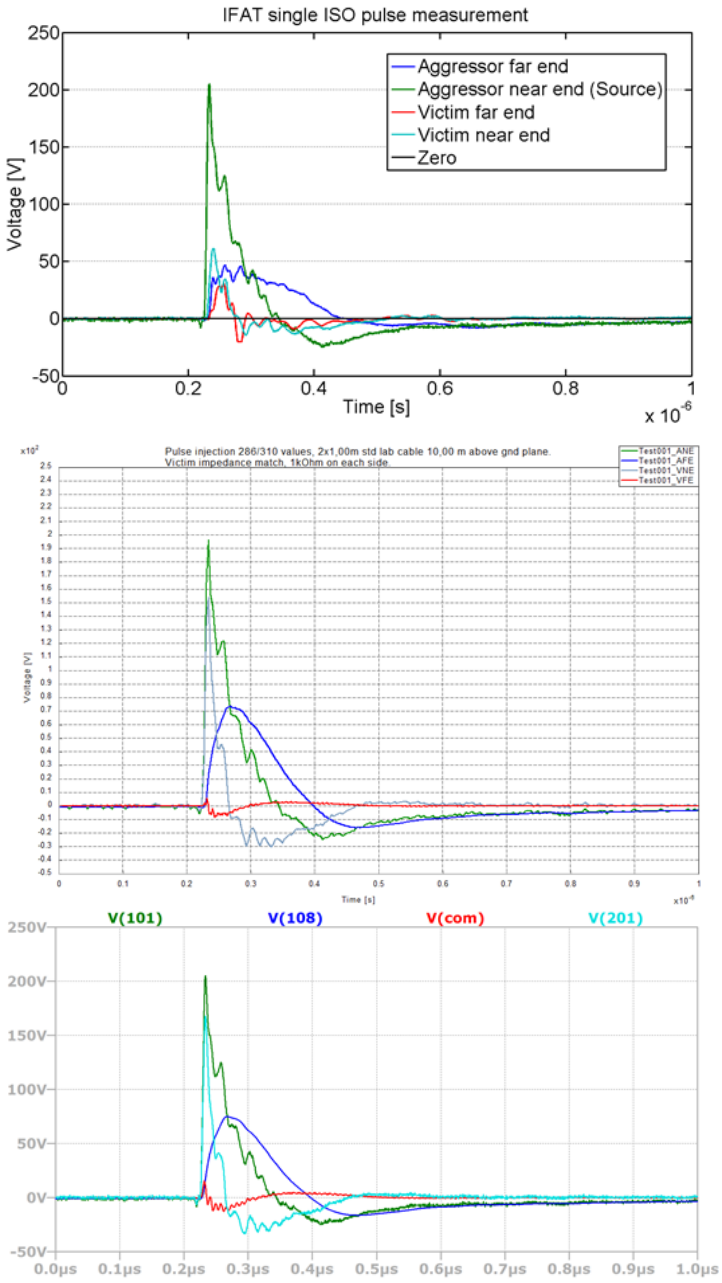

Abbildung 7: Messergebnis(oben), Simulationsergebnis mit der Software *EMC Studio* (Mitte) und Simulationsergebnis mit *XTT* (unten). Als Eingangssignal für die Simulationen dient der gemessene Aggressor-Impuls. Alle drei Ergebnisse zeigen gute qualitative Übereinstimmung.

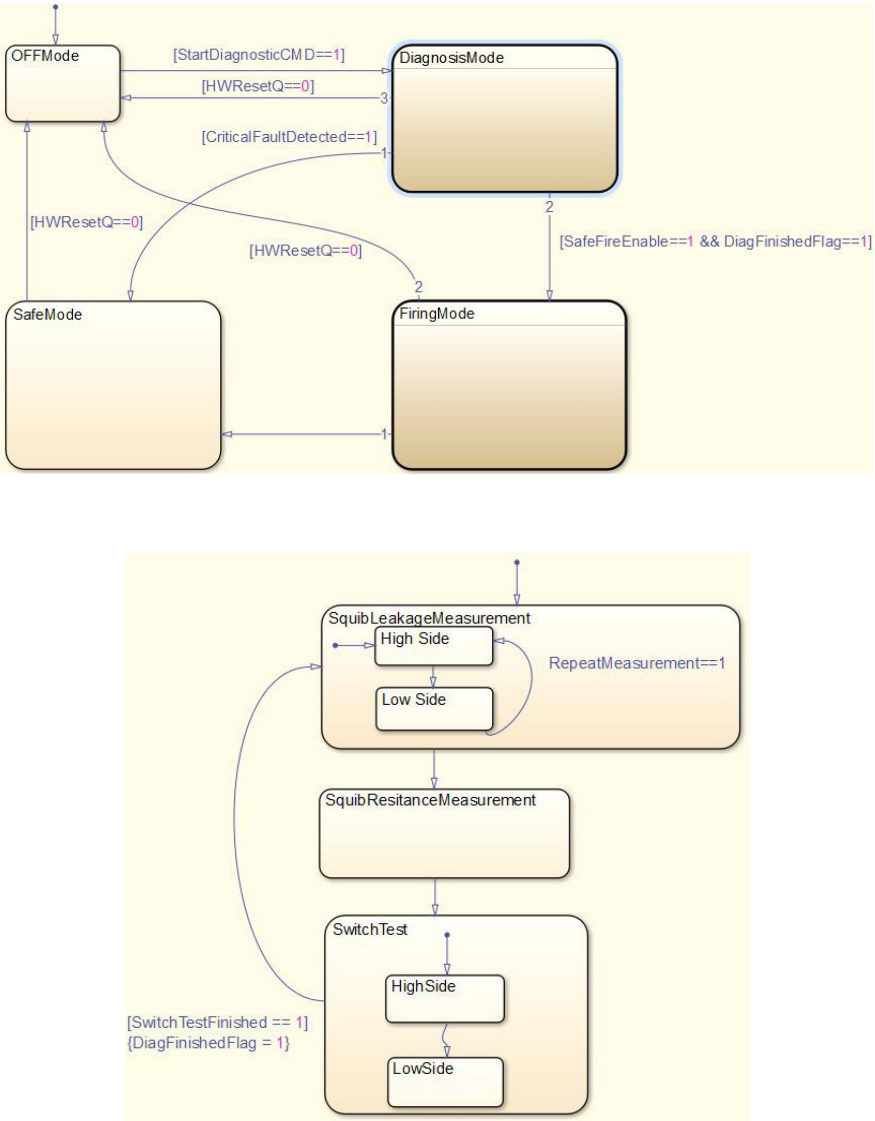

Abbildung 8: Hierarchisches Stateflow-Modell eines Airbag-Aktuators: Hauptzustände (oben) und Diagnosemodus (unten)

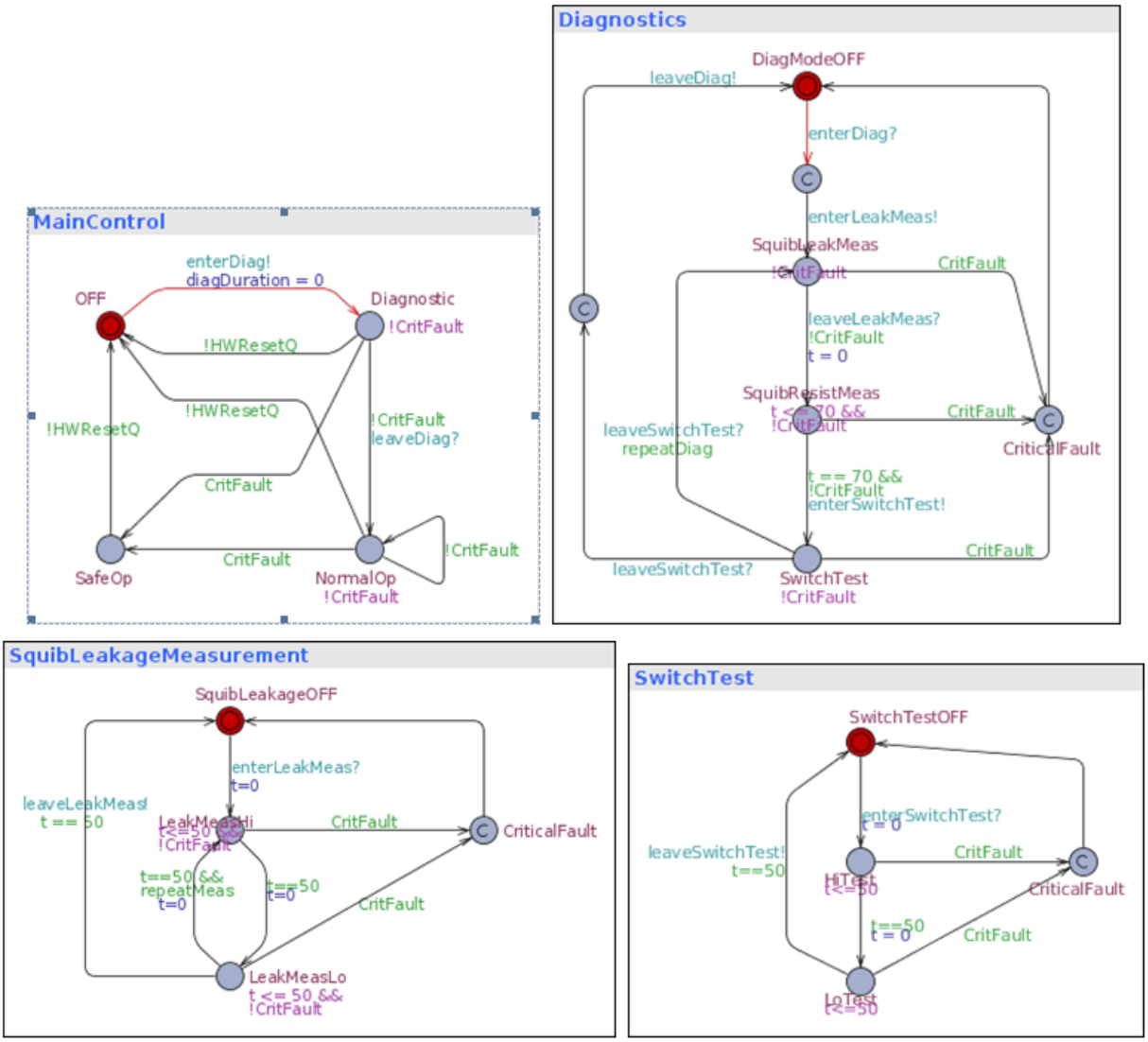

Abbildung 9: Ergebnis der Transformation des Stateflow-Modells aus Abbildung 8 in mehrere kommunizierende Timed Automata

- *Die Formalisierung der Anforderungen* kann nicht automatisiert durchgeführt werden, es sei denn, die Anforderungen sind bereits vor-formalisiert in irgendeiner Art und Weise. Die Zielsprache, ein TCTL-Dialekt, war ausreichend, um alle Aspekte der Anforderungen abzubilden. Insgesamt lässt sich sagen, dass die durch UPPAAL unterstützte Logik ausreicht, um die meisten (domänentyptischen) Anforderungen an eine Formalisierung zu erfüllen.
- *Die Sicherstellung der semantischen Äquivalenz* ist ein entscheidener Faktor bei einer Modelltransformation. Leider ist die Semantik von Stateflow nicht formal definiert, sondern implizit mit einer Reihe von Beispielen dokumentiert. Inkonsistenzen der Stateflow-Semantik sind bekannt; das Problem kann normalerweise mit Hilfe von Modellierungsrichtlinien umgangen werden, indem bestimmte Konstrukte vermieden werden, ohne die modellierte Funktionalität zu gefährden.

2.3.2 Voraussichtlicher Nutzen der Ergebnisse

Das Verfahren zur EMV-Analyse wurde bereits in einem prototypischen Werkzeug umgesetzt. Es ist geplant, dieses weiterzuentwickeln, um ein Software-Werkzeug von industrieller Qualität zu erhalten, das direkt als Produkt vermarktet werden kann. Gleichzeitig können die gewonnenen Erkenntnisse bereits vorher in industriellen Dienstleistungsprojekten im Rahmen der EMV-Absicherung eingesetzt werden.

Der Transformationsansatz von Simulink/Stateflow nach UPPAAL wurde anhand eines realen, industriellen Beispiels erarbeitet, und später verallgemeinert. Eine automatisierte Transformation ist bereits in der Entwicklung und wird in Form eines Produktes bestehenden und neuen Kunden im Automobilsektor und darüber hinaus angeboten werden. Es sind ebenfalls Dienstleistungen vorgesehen, um kundenspezifische Transformationen zu entwickeln.

2.3.3 Fortschritte auf dem Gebiet des Vorhabens

Der umgesetzte EMV-Analyseansatz ergänzt bestehende Arbeiten zur EMV-Analyse um funktionale Sicherheitsaspekte, insbesondere im Automobilbereich. Beginnend mit den Anforderungen in der Konzeptphase werden technische EMV-Anforderungen abgeleitet. EMV-Anforderungen werden explizit in Form von physikalischen Annahmen und Garantien für relevante Komponenten definiert.

Das Konzept zur formalen Verifikation von Simulink/Stateflow-Modellen mittels einer Transformation in Timed Automata fokussiert vor allem auf die reale Anwendung im Systemdesign sicherheitsrelevanter Systeme, die echtzeitkritisch sind. Für Simulink/Stateflow ist uns kein Werkzeug bekannt, das eine formale Verifikation von Echtzeiteigenschaften ermöglicht. Bestehende Ansätze zur Transformation von Statecharts setzen überwiegend auf UML oder ähnlichen Formalismen auf, die (noch) keine derartige Praxisrelevanz im Automobilbereich haben.

2.4 AP 5 Verifikation und Testen von Hardwarekomponenten

Dieses Arbeitspaket betrachtet die Verifikations- und Testmethoden auf Hardwareebene. Die Aktivitäten und Ergebnisse aus AP 5.1 werden im Rahmen von TS 1 in Abschnitt 2.1 beschrieben.

Im Folgenden werden die Arbeiten in den Teil-AP 5.3 und 5.4 näher beleuchtet.

2.4.1 Aktivitäten, Arbeitsschritte und Ergebnisse

Im Rahmen von AP 5.3 wurde ein generisches, elektrisches Steuergerätemodell zur Fehlerinjektion entwickelt. In AP 5.4 wurde ein Ansatz zu stochastischen Hardwaremodellen erarbeitet.

Generisches Steuergerätemodell TWT hat ein einfaches, generisches Steuergerätemodell entworfen, das es erlaubt, das korrekte elektrische Verhalten hinsichtlich der Stromversorgung und I/O-Ports zu simulieren, um den Zustand des Steuergeräts unter Berücksichtigung seiner elektrischen Limitierungen zu bestimmen.

Das generische Steuergerätemodell besteht aus Teilkomponenten wie Spannungsversorgung, DC/DC-Konvertern, EPROM und Sensoren. In einem ersten Modellierungsschritt wurden diese Teilkomponenten als atomare Elemente des Steuergeräts betrachtet. Bei der Modellierung in Simulink können mehrere Instanzen solcher Teilkomponenten zum Steuergerätemodel hinzugefügt und separat konfiguriert werden. Dies resultiert in unterschiedlichem Verhalten, obwohl ein gemeinsames Modell die Basis bildet.

Die entwickelten Teilkomponenten-Modelle wurden in einer Simulink-Bibliothek hinterlegt, um eine einfache Wiederverwendung in unterschiedlichen Steuergerätemodellen zu ermöglichen. Außerdem ist es möglich, jederzeit weitere Teilkomponenten-Modelle hinzuzufügen. Für die Simulation und die Fehlerinjektion müssen auch die internen Zustände der Komponenten abgebildet werden.

Ein einfaches, exemplarisches Steuergerätemodell in der Simulationsumgebung ist in Abbildung 10 dargestellt. Dieses entstammt dem Anwendungsfalls der Airbag-Steuerung von Infineon Österreich. Die Eingangsspannung wird durch zwei DC/DC-Konverter modifiziert und schließlich zu den Lastkomponenten „weitergereicht“. Diese sind das EPROM, beliebige interne Lasten, Sensoren, etc. Die Lastkomponenten haben eine Funktionalität zur Beobachtung der internen Spannungswerte und leiten davon einen funktionalen Zustand ab.

Die einzelnen Zustände und zusätzliche Trigger-Signale liefern die Eingangssignale eines Statecharts, das die Zustandsübergänge des Steuergeräts abbildet. Ein Beispiel ist in Abbildung 11 visualisiert. Weiterhin ist es möglich, periphere Komponenten, wie eine Airbag-Warnlampe oder Übertragungskabel mittels eines generischen Import-Blocks zu integrieren.

In das generische, elektrische Steuergerätemodell wurden Störimpulse basierend auf elektromagnetischer Interferenz (EMI) injiziert und das Verhalten des Modells analysiert. Der zugehörige Workflow wird in Abbildung 12 dargestellt.

Der Workflow kann genutzt werden, um ein gegebenes Steuergerätemodell gegen EMI-Fehler abzusichern. Der Aufwand für die Konstruktion und Analyse komplexer Modelle mit vielen Teilkomponenten ist überschaubar und der Ansatz flexibel erweiterbar.

Für eine nicht-stationäre Analyse ist jedoch detailliertes Wissen über die Teilkomponenten notwendig. Beispielsweise müssen zur Bestimmung des Effekts von Störspannungen auf den Versorgungsbahnen auf das Systemverhalten die exakten Spezifikationen des DC/DC-Konverters bekannt sein. Geringe Abweichungen (z.B. hinsichtlich der Stabilität der Parameter) können große

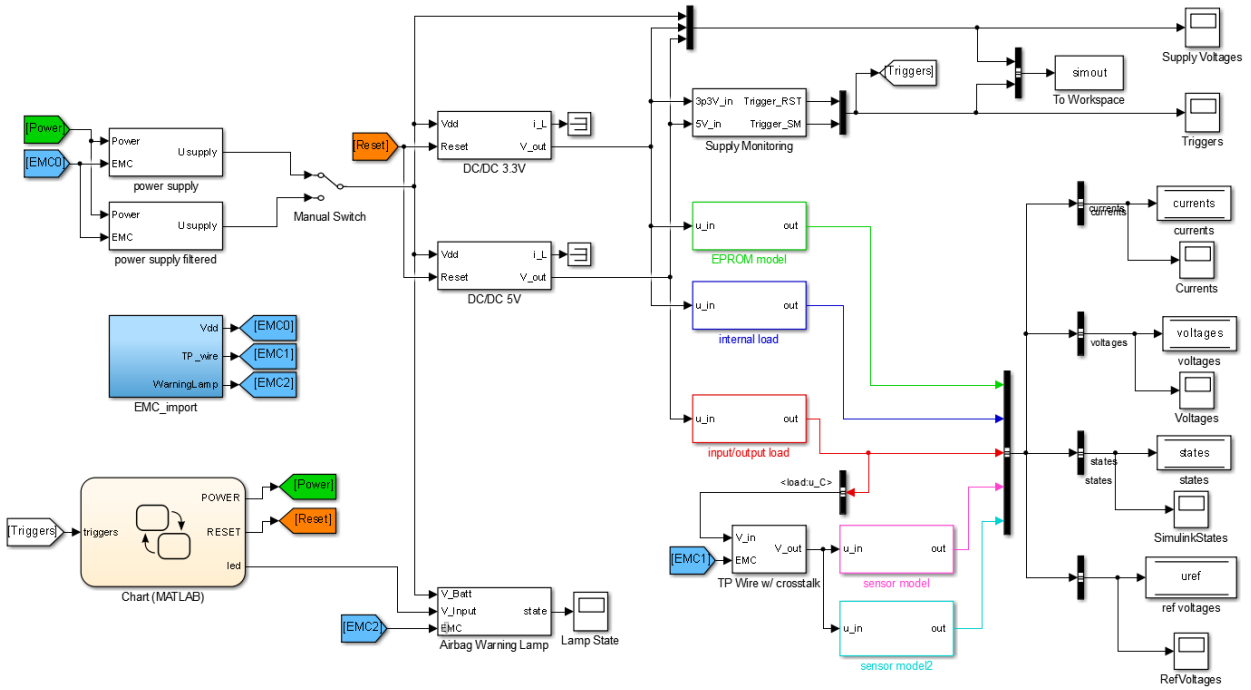

Abbildung 10: Beispiel für ein elektrisches Steuergerätemodell

Abweichungen vom realen Systemverhalten implizieren.

Somit wird der erarbeitete Ansatz dem ursprünglichen Anspruch, ein *einfaches*, *generisches* Steuergerätemodell zu entwickeln, nicht vollständig gerecht. Dieses Problem widersprecht jedoch nicht dem grundsätzlichen Vorgehen bzw. Workflow.

Eine weitere, wichtige Erkenntnis ist die Schwierigkeit der Integration detaillierter, elektrischer Modelle in Matlab/Simulink. Dies könnte durch die Verwendung von Beschreibungssprachen wie VHDL-AMS oder SystemC-AMS für das Steuergerätemodell, angegangen werden.

Stochastische Hardwaremodelle Stochastische Modellierung und Analyse erlauben die explizite Berücksichtigung von Unsicherheit in einer quantitativen Analyse von Hardwaredesigns. Beispielsweise ist die Signallaufzeit auf einem Chip vom Herstellungsprozess abhängig und zeigt statistische Abweichungen, die als Zufallsvariable angesehen werden können. Aus dem Blickwinkel des Automobilsektors sind beispielsweise die folgenden Aspekte in diesem Kontext von Interesse:

1. Lebensdauer von Hardwareelementen
2. Verzögerungen bei Berechnungen, z.B. auf RTL-Ebene
3. Probabilistische Fehler-Injektion
4. Unsicherheiten in Signalpfaden

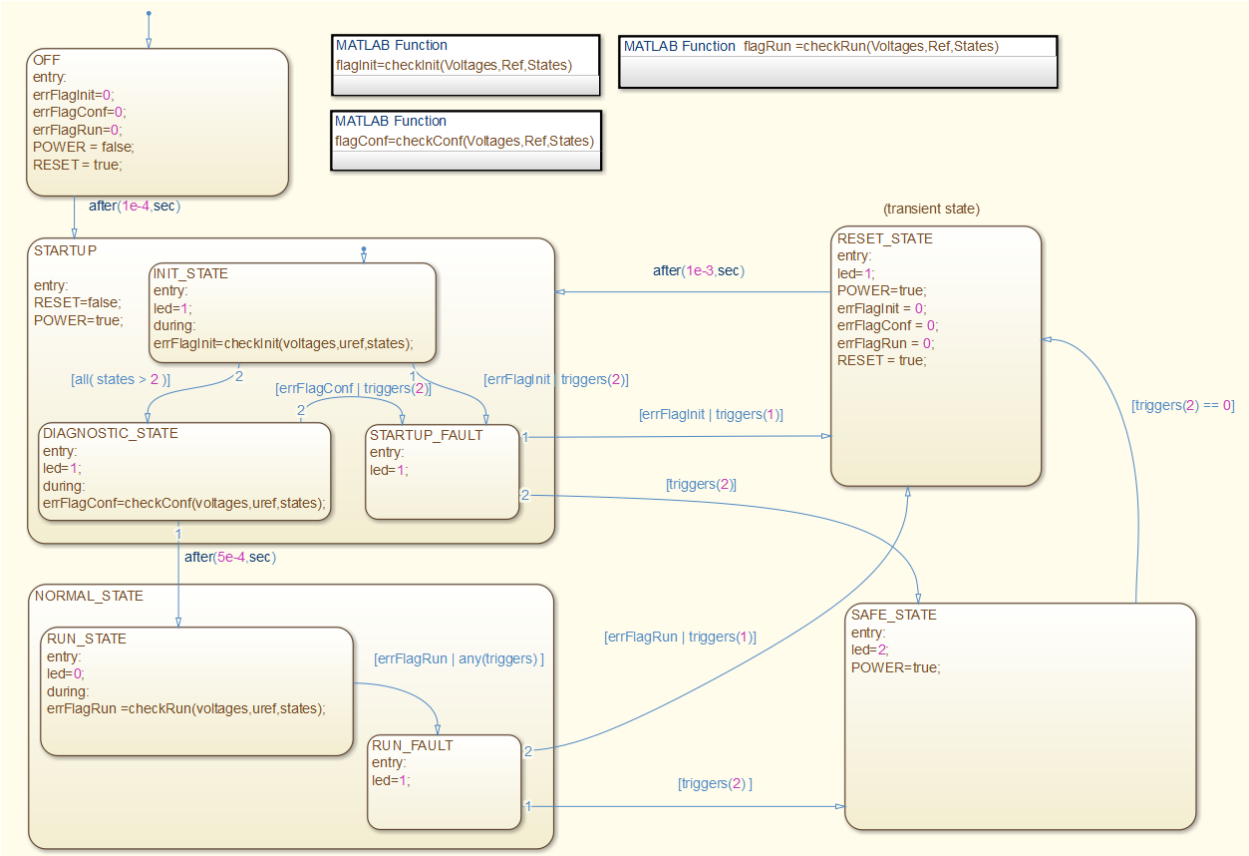

Abbildung 11: Zustandsmodell für das Beispiel-Steuergerätemodell

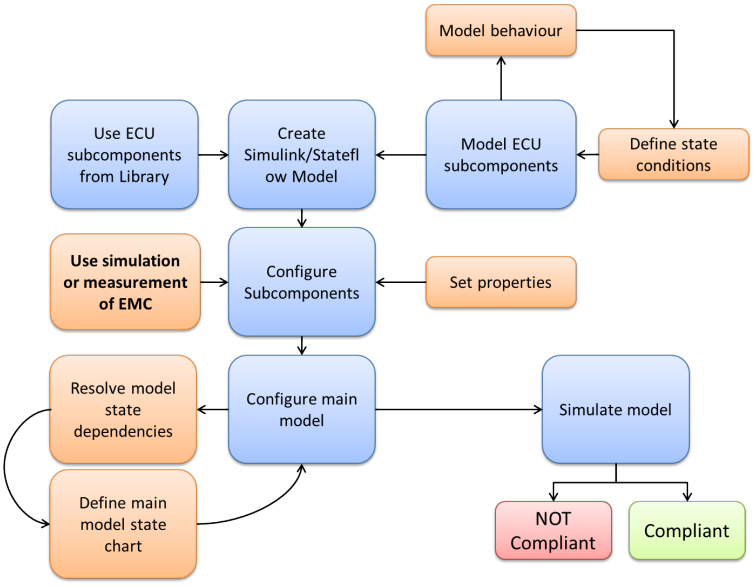

Abbildung 12: Workflow zur EMI-Injektion in elektrische Steuergerätemodelle

Die Aktivitäten in AP 5.4 haben sich auf methodische Arbeit konzentriert. Nach vielen Diskussionen mit den Partnern, insbesondere mit Infineon UK und der Universität Oxford, war die Abstraktionsebene, auf der das stochastische Modell basieren sollte, nicht klar. Die Anwendungspartner haben kontinuierliche Wahrscheinlichkeitsverteilungen und Fehlerraten nicht auf unterer Hardwareebene in Betracht gezogen. Folglich wurde die Betrachtung auf ein höheres Abstraktionsniveau verlagert.

Nach weitere Untersuchungen und einer detaillierten Befragung der Partner, wurden zwei zentrale Szenarien identifiziert:

1. *Stochastische Modellierung von Fehlerraten mit negativer Exponentialverteilung und Ableitung eines abstrakten Hardware-Modell aus einem sequenziellen RTC-Design*: In Kombination wurde ein PRISM-Modell erstellt, das probabilistische Anfragen erlaubt.
2. *Hardware-Fehlerinjektion und Wahrscheinlichkeit der Detektion innerhalb eines Zeitintervalls*: Dieses Szenario konnte im Projekt nicht vollendet werden; es wurden einige Herausforderungen identifiziert, die noch zu lösen sind. Diese wurden entsprechend dokumentiert.

2.4.2 Voraussichtlicher Nutzen der Ergebnisse

Der Workflow zur Erstellung generischer Steuergerätemodelle zur Fehlerinjektion wird von TWT voraussichtlich in Dienstleistungen bei bestehenden Kunden im Rahmen der Fahrzeugabsicherung eingesetzt werden. Das negative Ergebnis hinsichtlich der Steuergerätemodellierung stellt einen signifikanten Erkenntnisgewinn dar. In zukünftigen Modellen werden entsprechend andere Modellierungssprachen wie SystemC-AMS Anwendung finden. Der Ansatz ist im Rahmen der Absicherung sehr gut mit dem Verfahren der EMV-Analyse, das in Kapitel 2.3 beschrieben wurde, zu kombinieren.

Hinsichtlich stochastischer Hardwaremodelle wurden Erkenntnisse gewonnen, die mögliche Einsatzszenarien im Kontext der Hardwaremodellierung und der Hardwareauslegung aufzeigen. Diese wird TWT zur internen Weiterentwicklung des Konzepts nutzen, um einer praktischen Einsetzbarkeit näher zu kommen.

2.4.3 Fortschritte auf dem Gebiet des Vorhabens

Es wurde ein Workflow zur analogen Fehlerinjektion basierend auf generischen Steuergerätemodellen aufgezeigt. Zwar hat sich der Modellierungsansatz in Simulink als recht komplex herausgestellt, da er sehr stark von der Verfügbarkeit detaillierter Spezifikationen von Hardwarekomponenten abhängt, jedoch hat dies keinen Einfluss auf das generelle Vorgehen. Als Modellierungssprache sind andere Sprachen wie SystemC-AMS wahrscheinlich besser geeignet.

Stochastische Hardwaremodelle werden derzeit in der Hardwareentwicklung nicht oder kaum eingesetzt. Dementsprechend war es notwendig, die beteiligten Projektpartner, die Hardwareentwicklung betreiben, über die Möglichkeiten der quantitativen, stochastischen Analyse zu informieren und ihre Herausforderungen zu identifizieren. Dies ist sehr gut gelungen und es haben sich zwei Haupteinsatzszenarien herauskristallisiert, die in zukünftigen Projekten angegangen werden können.

2.5 AP 6 Verifikation und Testen von Softwarekomponenten

Dieses Arbeitspaket betrachtet die Verifikations- und Testmethoden auf Softwareebene. Die Aktivitäten und Ergebnisse aus AP 6.1 werden im Rahmen von TS 1 in Abschnitt 2.1 beschrieben. Im Folgenden werden die Arbeiten in den Teil-AP 6.2 und 6.3 näher beleuchtet. Im Verlauf der Projektdurchführung wurden viele Gemeinsamkeiten zwischen den Teil-AP 6.2, 6.5 und 6.6 identifiziert (AP 6.5 und 6.6 ohne TWT-Beteiligung). In sehr enger Zusammenarbeit wurden daher gemeinsame Liefergegenstände erarbeitet, die die Ergebnisse obiger Teil-AP zusammenfassen. TWT hatte in diesem Kontext die Leitung übernommen.

2.5.1 Aktivitäten, Arbeitsschritte und Ergebnisse

Für die Verifikation hardwarenaher und sicherheitsrelevanter Software wurde in AP 6.2 ein Verfahren für das modellbasierte Testen entwickelt, das eine frühe Sicherheitsanalyse von Softwaredesigns ermöglicht. Im Kontext von AP 6.3 wurden Abstraktionsebenen definiert, die bestehende Standard-Konzepte aus der modellbasierten Softwareentwicklung („Model-Driven Architecture“ der OMG), dem AUTOSAR-Standard und der ISO-Norm vereinen. Diese sind insbesondere für die Verifikation von SEooC-Software-Komponenten von Bedeutung.

Frühe Sicherheitsanalyse von Softwaredesigns Eine Herausforderung bei der Entwicklung sicherheitsrelevanter, eingebetteter Software im Automobilbereich ist das Testen von Sicherheitsmechanismen in Bezug auf Hardware- und Softwarefehler. Der Sicherheitsstandard ISO 26262 definiert Sicherheitsmechanismen als technische Lösungen, um Fehler zu erkennen und zu kontrollieren, um das System in einen sicheren Zustand zu bringen. Aktuelle Ansätze für das Testen von Sicherheitsmechanismen erfordern die Injektion von Fehlern in laufende Systeme, die aus einem Mikrocontroller und der dazugehörigen Software bestehen.

In enger Zusammenarbeit mit Infineon Deutschland hat TWT einen Ansatz zur frühen Analyse von Sicherheitsmechanismen in AUTOSAR-Treibersoftware entwickelt. Das Ziel ist das Finden von Designfehlern in Sicherheitsmechanismen bereits in der Konzeptphase im ISO 26262-Entwicklungsprozess. Der Ansatz basiert auf den Methoden *Model Checking* und modellbasiertem Testen (MBT). Mittels Model Checking können Fehler in einem abstrakten Modell des zu testenden Systems ausgeschlossen werden. Der MBT-Ansatz nutzt das abstrakte Modell schließlich, um automatisch Testfälle zur Verifikation der Treiber-Software zu generieren.

Abbildung 13 stellt den eingesetzten Prozess sowie die beteiligten Werkzeuge und Artefakte dar. Der Ansatz wurde im Rahmen eines realen, industriellen Anwendungsfalls erfolgreich evaluiert: Ein Analog/Digital-Konverter-Treiber als Teil des *Microcontroller Abstraction Layer* (MCAL) des AUTOSAR Software Stacks [4]. Es wurde im Speziellen eine Implementierung einer Motorsteuerung analysiert, wie sie beispielsweise in hybriden Elektrofahrzeugen zum Einsatz kommt. Abbildung 14 stellt dieses Szenario grafisch dar. Für den Microcontroller wurde ein virtuelles Modell der Hardware in SystemC verwendet, das ebenfalls eine Fehlerinjektion ermöglicht.

Eine detaillierte Beschreibung des Ansatzes, des Anwendungsfalls und der Ergebnisse findet sich in einer gemeinsamen Veröffentlichung [14], die im Rahmen des Projekts entstanden ist.

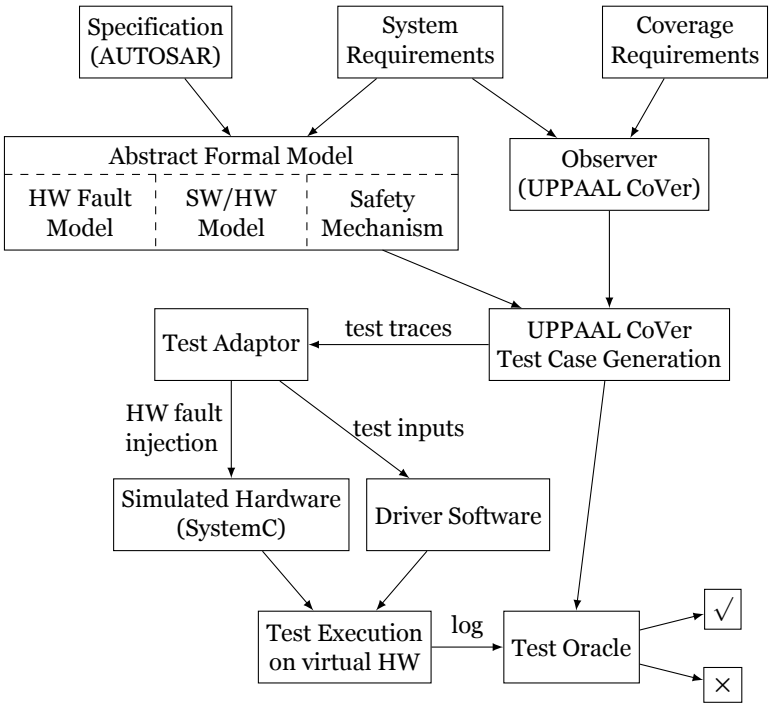

Abbildung 13: Eingesetzter Prozess zur Verifikation hardwarenaher Software im AUTOSAR-Kontext

Abstraktionsebenen für hardwarenahe Software Eine Softwarekomponente kann plattformspezifisch oder als *Safety Element out of Context* (SEooC) entwickelt werden. TWT hat in Abstimmung mit den Partnern einen Leitfaden für Abstraktionsebene für hardwarenahe Software konzpiert. Die Abstraktionsebenen dienen vor allem der Bewertung und Einordnung von Verifikations- und Testansätzen im Rahmen der Entwicklung von SEooC.

Da eine System-/Softwareentwicklung immer auf ein konkretes Ziel ausgerichtet ist, wurde ein abstrakter Ansatz gewählt, der auf die konkrete Anwendung anzupassen ist. Beispielsweise können feingranularere Abstraktionsebenen „eingezogen“ werden. Formale Verifikation kann auf jeder der Abstraktionsebenen durchgeführt werden, indem ein formales Modell basierend auf den Anforderungen und/oder (eines Teils des) Systems entworfen wird.

Eines der Ziele bei der Definition der Abstraktionsebenen war es, die Kompatibilität mit bestehenden Standard-Konzepten aus der modellbasierten Softwareentwicklung - wie beispielsweise der *Model-Driven Architecture* (MDA) der OMG [22] - mit den im AUTOSAR-Standard im Rahmen der Referenzarchitektur festgelegten Abstraktionsebenen sowie selbstverständlich mit der ISO 26262 kompatibel zu gestalten.

Im Folgenden ein Ausschnitt aus der erarbeiteten Definition:

A1 **Designmodelle**

(a) Abstraktionsebene:

i. Detailgrad der zur Analyse funktionalen und nicht-funktionalen Verhaltens von

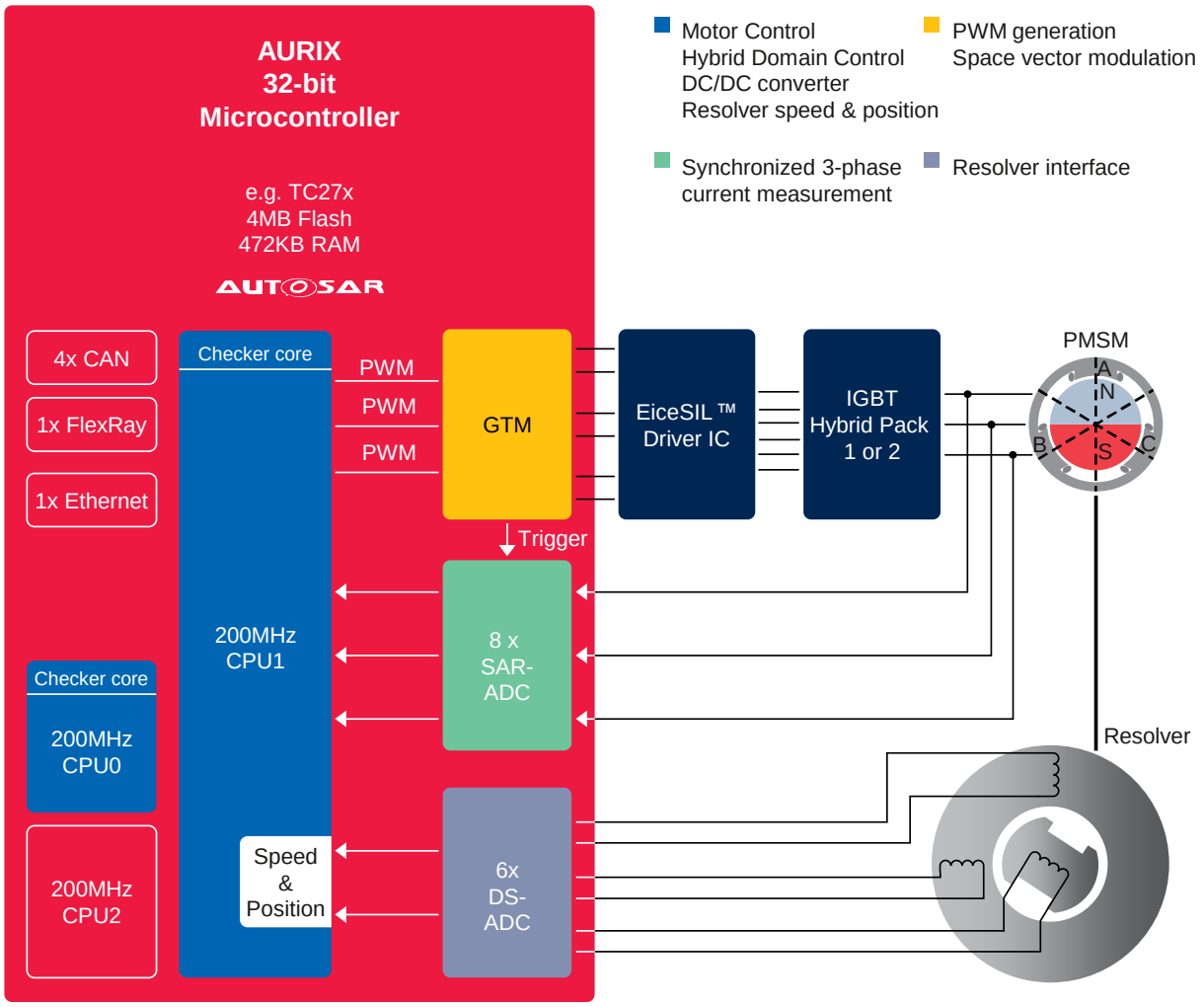

Abbildung 14: Elektrische Motorsteuerung, anhand der die in AP 6.2 entwickelte Analysemethode evaluiert wurde

Teilen oder des ganzen Systems in Bezug auf (Sicherheits-)Anforderungen notwendig ist.
   ii. Beinhaltet einfaches Umgebungsmodell, ggf. mit Eingabe-Stimuli.
(b) Technologien (Beispiele):
   i. Informell: Natürliche Sprache
   ii. Semi-Formal: DSL, UML/SysML
   iii. Formal: DSL, Timed Automata, Petri-Netze, Promela, Markov-Modell, CSP

A2 **Implementierungsmodelle**

(a) Abstraktionsebene:
   i. Systemspezifikation, die so detailliert ist, dass sie im Rahmen einer Simulation ausgeführt werden kann oder Analyseframework mit realistischen Eingabe-Stimuli.
   ii. Verfeinerter Detailgrad, der es erlaubt, die tatsächliche Implementierung vom Modell abzuleiten.
(b) Technologien (Beispiele):
   i. Simulink, SystemC

A3 **Implementierung**

(a) Abstraktionsebene:
   i. Software-Units liegen als Komponenten vor und ihre Interaktion ist definiert.
   ii. Software-Architektur ist definiert (Struktur, Ein-/Ausgabedaten, Typen, Schnittstellen, externe Abhängigkeiten)
(b) Technologien (Beispiele):
   i. Simulink, SystemC, Java, TargetLink, C, Assembler

A4 **Software auf Zielplattform**

(a) Abstraktionsebene:
   i. Software läuft auf der Zielhardware und ist als Objektcode geladen; direktes Testen im realen Systemkontext möglich.
(b) Technologien (Beispiele):
   i. Objektcode (HW-spezifisch)

2.5.2 Voraussichtlicher Nutzen der Ergebnisse

Beide Ergebnisse kann TWT im Rahmen von Dienstleistungen in der Systementwicklung bei bestehenden Kunden einsetzen, um deren Prozesse effizienter zu gestalten. Die Sicherheitsanalyse in der Konzept-/Designphase erlaubt eine frühe Einschätzung der Wirksamkeit von Sicherheitsmechanismen und verhindert u.U. schwerwiegende Fehler in einer frühen Phase der Entwicklung, die zu hohen Kosten führen können. Die Abstraktionsebenen unterstützen beim Design sicherheitsrelevanter Software sowie deren Einbettung in den ISO 26262-Verifikationsprozess und sind AUTOSAR-, ISO 26262- und MDA-konform.

2.5.3 Fortschritte auf dem Gebiet des Vorhabens

Wie bereits dargelegt, ermöglicht der Ansatz zur frühen Sicherheitsanalyse von hardwarenahen Softwaredesigns die frühzeitige Identifikation von Designfehlern, gerade was die Umsetzung von Sicherheitsmechanismen anbelangt. Die Praxistauglichkeit des Verfahrens wurde anhand eines industriellen Anwendungsfalls in Kooperation mit Infineon nachgewiesen und wird u.a. zur Reduktion äußerst kostenintensiver Re-Iterationen im Entwicklungsprozess und einem „sichereren" Software-Design führen. Somit lohnt sich das initiale Investment bei der Erstellung formaler Modelle.

Die Abstraktionsebenen vereinheitlichen verschiedene, im Automobilbereich (und darüber hinaus) verbreitete Ansätze unter dem besonderen Fokus der Funktionssicherheit nach ISO 26262. Gleichzeitig wird die Kompatibilität mit den bestehenden Konzepten nicht gefährdet und insbesondere die SEooC-Entwicklung berücksichtigt. Die Definition ermöglicht eine gute Basis zur Anpassung für konkrete, sicherheitsrelevante Entwicklungen.

2.6 AP 7 Fallstudien und Demonstratoren

2.6.1 Aktivitäten, Arbeitsschritte und Ergebnisse

Im Rahmen von AP 7 wurden wesentliche Projektergebnisse, die von TWT alleine oder in Kooperation mit den Partnern entwickelt wurden, evaluiert, um ihre Eignung für den industriellen Einsatz der Anwendungspartner zu bewerten. Die entsprechenden Resultate wurden bereits in den vorangegangenen Kapiteln zu AP 4, 5 und 6, sowie TS 1 beschrieben. Die Kooperation zwischen Infineon Deutschland und Österreich sowie TWT war in diesem Kontext besonders eng.

Außerdem hat TWT wesentlich zur Definition von Testfall-Attributen in Abstimmung mit TS 1 beigetragen und ein Konzept zur Sicherstellung der Nachverfolgbarkeit von VeTeSS-Anforderungen zu Anwendungsfällen erarbeitet. Letzteres ermöglicht es, festzustellen, welche der Anforderungen technisch in einem der Anwendungsfälle von WP7 erfüllt sind.

TWT hat zudem an der sogenannten „Mod/Sim"-Arbeitsgruppe aktiv partizipiert. In diesem Rahmen wurde gemeinsam mit weiteren Partnern eine exemplarische Instanziierung eines ISO 26262-konformen Anforderungs-, Analyse- und Verifikationsprozesses basierend auf dem Airbag-Anwendungsfall von Infineon Österreich erarbeitet.

2.6.2 Voraussichtlicher Nutzen der Ergebnisse

Die Evaluation der Ergebnisse der technischen AP 4, 5 und 6 sowie TS 1 hat einen signifikanten Beitrag zur Bewertung der einzelnen Methoden, Konzepte und Tools geliefert und eine Validierung praxisnah ermöglicht. Gleichzeitig wurden Möglichkeiten der Verbesserung des Reifegrades der Ergebnisse aufgezeigt, welche TWT im Rahmen der Weiterentwicklung berücksichtigen wird.

2.6.3 Fortschritte auf dem Gebiet des Vorhabens

Keine wissenschaftlicher Fortschritt sondern Evaluation, daher für AP 7 nicht zutreffend.

Literatur

[1] Altair HyperWorks. FEKO. http://www.feko.info.

[2] Rajeev Alur and David L Dill. A theory of timed automata. Theoretical computer science, 126(2):183--235, 1994.

[3] Gerhard Deuter Andreas Baumgart, Klaus Hörmaier. Model-based method to achieve emc for distributed safety-relevant automotive systems. In SIMUL 2014: The Sixth International Conference on Advances in System Simulation, 2014.

[4] AUTOSAR. Specification of ADC Driver. `http://www.autosar.org/fileadmin/files/releases/4-0/software-architecture/peripherals/standard/AUTOSAR_SWS_ADCDriver.pdf`, 2011.

[5] Jiri Barnat, Jan Beran, Lubos Brim, Tomas Kratochvila, and Petr Rockai. Tool chain to support automated formal verification of avionics simulink designs. In Marielle Stoelinga and Ralf Pinger, editors, Formal Methods for Industrial Critical Systems, volume 7437 of Lecture Notes in Computer Science, pages 78--92. Springer Berlin Heidelberg, 2012.

[6] Gerd Behrmann, Alexandre David, and Kim G Larsen. A tutorial on UPPAAL 4.0., 2006.

[7] NCWM Braspenning, J. M. van de Mortel-Fronczak, and J. E. Rooda. A model-based integration and testing method to reduce system development effort. Electronic Notes in Theoretical Computer Science, 164(4):13--28, 2006.

[8] Glenn Bruns and Ian Sutherland. Model checking and fault tolerance. In Algebraic Methodology and Software Technology, pages 45--59. Springer, 1997.

[9] COMSOL. COMSOL Multiphysics. `http://www.comsol.com`.

[10] CESAR Consortium. CESAR - Cost-efficient Methods and Processes for Safety-relevant Embedded Systems. Springer, 2013. ISBN-10: 3709113865; ISBN-13: 978-3709113868.

[11] Electric Cloud. Survey finds 58% of software bugs result from test infrastructure and process, not design defects. http://electric-cloud.com/company/news/press-releases/item/survey-finds-58-of-software-bugs-result-from-test-infrastructure-and-process-not-design-defects/, 06 2010.

[12] EMCoS. EMC Studio. `http://www.emcos.com`.

[13] A. Fleischmann, J. Hartmann, C. Pfaller, M. Rappl, S. Rittmann, and D. Wild. Concretization and formalization of requirements for automotive embedded software systems development. Technical report, Technische Universität München, Fakultät für Informatik, 2005.

[14] Stefan Gulan, Jens Harnisch, Sven Johr, Roberto Kretschmer, Stefan Rieger, and Rafael Zalman. Model-based analysis for safety critical software. In Floor Koornneef and Coen van Gulijk, editors, Computer Safety, Reliability, and Security, volume 9337 of Lecture Notes in Computer Science, pages 111--120. Springer International Publishing, 2015.

[15] J. Hänsel, D. Rose, P. Herber, and S. Glesner. An Evolutionary Algorithm for the Generation of Timed Test Traces for Embedded Real-Time Systems. In 2011 IEEE Fourth International Conference on Software Testing, Verification and Validation (ICST), pages 170--179, March 2011.

[16] Scott Hazelhurst and Jean Arlat. Specifying and verifying fault tolerant hardware. Proc. Designing Correct Circuits, 2002.

[17] isograph. Fault tree+. http://www.isograph.com.

[18] KPIT. Medini analyze Functional Safety Tool. http://www.kpit.com/.

[19] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In G. Gopalakrishnan and S. Qadeer, editors, Proc. 23rd International Conference on Computer Aided Verification (CAV'11), volume 6806 of LNCS, pages 585--591. Springer, 2011.

[20] Florian Leitner. Evaluation of the Matlab Simulink Design Verifier versus the model checker SPIN. Technical Report soft-08-05, Universität Konstanz, Department of Computer and Information Science, 2008.

[21] MathWorks. Simulink design verifier. http://de.mathworks.com/products/sldesignverifier/.

[22] Object Management Group. Omg model driven architecture. http://www.omg.org/mda/.

[23] W. Vesley, F. Goldberg, N. Roberts, and D. Haasl. Fault Tree Handbook. U.S. Nuclear Regulatory Commision, 1981.

Berichtsblatt

1. ISBN oder ISSN	2. Berichtsart (Schlussbericht oder Veröffentlichung) Abschlussbericht

3. Titel
Verification and Testing to Support Functional Safety Standards – Abschlussbericht der TWT GmbH Science & Innovation

4. Autor(en) [Name(n), Vorname(n)]	5. Abschlussdatum des Vorhabens 30.04.2015
Dr. Stefan Rieger - TWT GmbH Science & Innovation	6. Veröffentlichungsdatum k.A.
	7. Form der Publikation Bericht
8. Durchführende Institution(en) (Name, Adresse) TWT GmbH Science & Innovation Ernsthaldenstraße 17 70565 Stuttgart	9. Ber. Nr. Durchführende Institution k.A.
	10. Förderkennzeichen 01IS120001C
	11. Seitenzahl 34
12. Fördernde Institution (Name, Adresse) BMBF	13. Literaturangaben 23
	14. Tabellen -
	15. Abbildungen 14

16. Zusätzliche Angaben
k.A.

17. Vorgelegt bei (Titel, Ort, Datum)
Henry Sende, Deutsches Zentrum für Luft- und Raumfahrt e.V., Rosa-Luxemburg-Str. 2, 10178 Berlin

18. Kurzfassung

Der Abschlussbericht der TWT GmbH Science & Innovation beinhaltet eine Zusammenfassung der Forschungsaktivitäten und der erzielten Ergebnisse im Projekt VeTeSS (Verification and Testing to Support Functional Safety Standards). Weiterhin wird der Fortschritt über den aktuellen Stand der Technik und die geplante Verwertung der Projektergebnisse beleuchtet. Die Ergebnisse adressieren den gesamten Entwicklungsprozess sicherheitsrelevanter Systeme im Automobilbereich mit besonderem Fokus auf den Sicherheitsstandard ISO 26262.

Die Hauptergebnisse des Projekts sind:

- Ganzheitlicher Ansatz zum Informationsmanagement für die Entwicklung sicherheitsrelevanter Systeme im Automobilbereich
- Simulative Analyse der elektromagnetischen Verträglichkeit mit Entwicklung eines prototypischen Werkzeugs
- Fehlerinjektion basierend auf generischen Steuergerätemodellen
- Formale Verifikation von Echtzeiteigenschaften industrieller Systemmodelle
- Konzept zur Umsetzung stochastischer Hardwaremodelle
- Frühe Sicherheitsanalyse von Softwaredesigns basierend auf Model Checking und modellbasiertem Testen

19. Schlagwörter Sicherheit, ISO 26262, Verifikation, Modellierung, formale Analyse, elektromagnetische Verträglichkeit, Informationsmanagement	
20. Verlag k.A.	21. Preis k.A.

Document Control Sheet

1. ISBN or ISSN	2. type of document (e.g. report, publication) Final report

3. title
Verification and Testing to Support Functional Safety Standards – Abschlussbericht der TWT GmbH Science & Innovation

4. author(s) (family name, first name(s))	5. end of project
Dr. Stefan Rieger - TWT GmbH Science & Innovation	30.04.2015
	6. publication date n/a
	7. form of publication Report
8. performing organization(s) (name, address) TWT GmbH Science & Innovation Ernsthaldenstraße 17 70565 Stuttgart	9. originator's report no. n/a
	10. reference no. 01IS120001C
	11. no. of pages 34
12. sponsoring agency (name, address) BMBF	13. no. of references 23
	14. no. of tables -
	15. no. of figures 14

16. supplementary notes
n/a

17. presented at (title, place, date)
Henry Sende, Deutsches Zentrum für Luft- und Raumfahrt e.V., Rosa-Luxemburg-Str. 2, 10178 Berlin

18. abstract
The final project report of TWT GmbH Science & Innovation summarises the research activities and results in the VeTeSS project (Verification and Testing to Support Functional Safety Standards). Furthermore, the progress beyond the current state of the art and the planned exploitation of project results is addressed. The result comprise the entire development process of safety-relevant systems in the automotive domain, focusing especially on the current safety standard ISO 26262.
The main achievements in this project are:

- Wholistic approach to information management for the development of safety-relevant automotive Systems
- Simulation-based Analysis of electromagnetic compatibility including the development of a tool prototype
- Fault injection based on generic ECU models
- Formal verification of real-time properties of industrial system models
- Concept for realizing stochastic hardware models
- Early safety analysis of software designs based on model checking and model-based testing.

19. keywords
Safety, ISO 26262, Verification, Modelling, formal Analysis, electromagnetic compatibility, information management

20. publisher	21. price
n/a	n/a