mehr zum thema:
vierfores.de
nutzfahrzeuge.fraunhofer.de

SICHERHEIT IN DER FAHRZEUGTECHNIK: SOFTWARE IN EINGEBETTETEN SYSTEMEN

Die Fahrzeugtechnik ist einer der wichtigsten Treiber für Innovationen bei der Entwicklung komplexer Systeme. Stetig erhöht sich hier der Anteil softwaregesteuerter Komponenten und auch der Einsatz autonomer Fahrzeuge ist heute vorstellbar. Trotz der wachsenden Komplexität muss Qualität sichergestellt werden. Das erfordert leistungsfähige Modellierungs- und Analysetechniken sowie geeignete Entwicklungsprozesse und Verfahren, die das Verstehen der komplizierten Sachverhalte erleichtern. Der Artikel beleuchtet aus Sicht der Qualitätssicherung Probleme und Einflussfaktoren, die in der Entwicklung von eingebetteten Systemen zumeist nicht hinreichend berücksichtigt werden. Unzulänglichkeiten existierender Verfahren und Techniken sowie mögliche Ansätze zur Verbesserung werden diskutiert und potenzielle Anwendungen im Kontext der Entwicklung von autonomen Systemen beschrieben.

Verfahren und Modelle in der Qualitätssicherung

Pkw-Produzenten haben früher bei ihren Fahrzeugen vornehmlich Leistungsdaten in den Vordergrund gestellt. Aktuell prägen Schlagwörter wie Sicherheit und Komfort die Entwicklungen der Automobilindustrie maßgeblich mit. Fahrerassistenz-Systeme, wie z. B. Spurhalte-, Notbrems- oder Einpark-Systeme, vermitteln mehr Sicherheit durch eine Entlastung und Unterstützung des Fahrers in unterschiedlichsten Situationen. Derartige Systeme erfordern entsprechend leistungsfähige Softwarelösungen. Die technischen Herausforderungen bei der Realisierung solcher Systeme sind immens. Vor allem Aspekte wie Zuverlässigkeit, Verfügbarkeit und Sicherheit sind von besonderer Bedeutung (vgl. [Sch12]). Sicherheit adressiert hier vor allem Safety- und Security-Eigenschaften der Systeme, d. h. die Einflüsse des Systems auf die Umwelt (*Safety*) und umgekehrt (*Security*). Die genannten Aspekte werden zunehmend durch Software in ihrer Interaktion mit elektronischen Komponenten, mechanischen Bauteilen und der Umgebung des Systems beeinflusst. Daher muss auf die Sicherstellung der damit verbundenen Eigenschaften innerhalb der Software ein besonderes Augenmerk gelegt werden.

Während Zuverlässigkeit das Ausfallverhalten eines Systems in Abhängigkeit von Betriebsbedingungen und Laufzeit beschreibt, thematisiert der Begriff Sicherheit im Sinne von *Safety* die kritischen bzw. unerwünschten Systemzustände, deren Ursachen und Auftrittsverhalten. Eine im Zusammenhang mit Safety weit verbreitete Technik zur Analyse von Ursache-/ Wirkzusammenhängen von Ausfällen ist die Fehlerbaum-Analyse (vgl. [DIN90] und [DIN07]). Dabei handelt es sich um eine Methode, die Ausfälle z. B. auf Komponentenebene qualitativ und quantitativ in Relation zu gefährlichen Wirkungen im System setzt. Das heißt, durch diese Methode lassen sich zum einen die Art und Weise eines Systemausfalls und zum anderen die Wahrscheinlichkeit für ein Systemversagen erfassen. Der Terminus „gefährlich“ bezeichnet in diesem Zusammenhang eine Situation des Systems, in der der Eintritt eines Ereignisses, das zur Verletzung von Personen oder zur Beschädigung bzw. Verlust von Eigentum führen kann, möglich ist. **Abbildung 1** zeigt den prinzipiellen Aufbau des bei der

Die in diesem Artikel beschriebenen Arbeiten finden im Pilotprojekt „Virtuelle und Erweiterte Realität für höchste Sicherheit und Zuverlässigkeit von Eingebetteten Systemen“ (ViERforES) im Rahmen des BMBF-Programms „Spitzenforschung und Innovation in den Neuen Ländern“ statt. ViERforES ist Bestandteil der Innovationsallianz Virtuelle Techniken – ein Zusammenschluss mehrerer vom Bundesministerium für Bildung und Forschung (BMBF) unter dem Dach der Hightech-Strategie IKT 2020 geförderter Projekte. Beteiligt an den Forschungsarbeiten sind die Otto-von-Guericke-Universität Magdeburg, die TU Kaiserslautern, das Fraunhofer-Institut für Experimentelles Software Engineering in Kaiserslautern und das Fraunhofer-Institut für Fabrikbetrieb und -automatisierung in Magdeburg.

Kasten 1: *Das Projekt ViERforES.*

die autoren

Dr. Patric Keller
(pkeller@cs.uni-kl.de)
ist wissenschaftlicher Mitarbeiter am Lehrstuhl Software Dependability im Fachbereich Informatik der TU Kaiserslautern. Er forscht zu den Themen Sicherheit, Zuverlässigkeit und Verfügbarkeit von Systemen.

Dr. Veit Köppen
(vkoeppen@ovgu.de)
ist geschäftsführender Leiter des Center for Digital Engineering an der Otto-von-Guericke-Universität Magdeburg und wissenschaftlicher Assistent im Institut für Technische und Betriebliche Informationssysteme.

Prof. Dr. Peter Liggesmeyer
(liggesmeyer@cs.uni-kl.de)
ist Inhaber des Lehrstuhls Software Dependability im FB Informatik der TU Kaiserslautern, wissenschaftlicher Direktor des Fraunhofer-Instituts für Experimentelles Software Engineering, und Vizepräsident der Gesellschaft für Informatik.

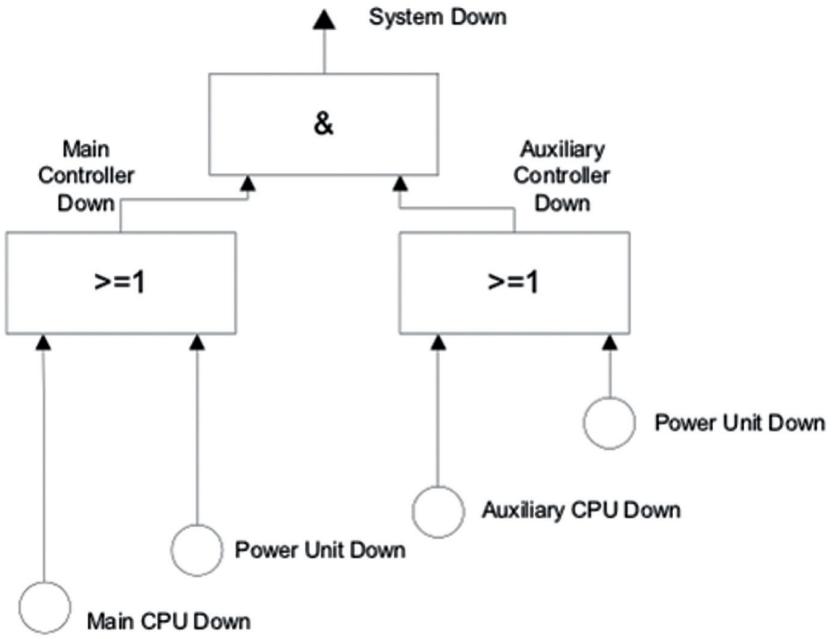

Abb. 1: Fehlerbaum, der die Bedingung für einen Systemausfall mittels grafischer Elemente beschreibt.

Analyse verwendeten Fehlerbaummodells anhand eines einfachen Beispiels.

Die Fehlerbaum-Analysetechnik eignet sich, um Fragen nach den Bedingungen, unter denen ein bestimmter unerwünschter Systemzustand erreicht werden kann, zu beantworten. Das kann auch die Abschätzung der Wahrscheinlichkeit für dessen Eintritt einschließen. In den letzten Jahren wurden das Verfahren und das zu Grunde liegende Modell um zusätzliche Methoden und Elemente erweitert. Daneben gibt es eine Vielzahl weiterer formaler und informaler Techniken, die gegenwärtig zum Einsatz kommen (eine Übersicht geben [Lig00] und [Lig09]).

Weiterführende Ansätze

Im grundlegenden Aufbau ähneln sich die genannten Systeme größtenteils: Auf Basis von Sensordaten werden Entscheidungen entsprechend vordefinierten Regeln getroffen, die das Fahrverhalten direkt (durch aktiven Eingriff, z. B. Abbremsen) oder passiv (z. B. durch Warnung des Fahrers) beeinflussen können. Systemausfälle sind dabei auf Fehler in der Hard- oder Software zurückzuführen. Die Eigenschaft solcher Systeme, Echtzeitbedingungen einhalten zu müssen, deren Adaptivität sowie deren Komplexität machen es schwierig, bestehende Analysetechniken und -verfahren anzuwenden.

Geeignete Analyseverfahren müssen zum einen das Versagen von Hardwarekomponenten berücksichtigen. Zum anderen müssen sie auch systematische Fehler, d. h. Fehler, die sich während der Konzeption, des Designs oder der Implementierung in den für die Steuerung relevanten Softwaremodulen manifestieren, in die Analysen mit einbeziehen. Außerdem spielen umweltbedingte Störeinflüsse, wie erschwerte Witterungsbedingungen und die Verfälschung von Sensordaten, die sich auf die Qualität der Sensordaten auswirken und letztendlich die Entscheidung über das Eingreifen in das Fahrverhalten mitbestimmen, eine nicht zu vernachlässigende Rolle.

Hinzu kommen Betrachtungen zu Reaktionen auf unvorhergesehene Ereignisse, wie Teilsystemausfälle oder die Verletzung temporaler Bedingungen (kritisch in Echtzeitsystemen). All diese Aspekte und viele weitere müssen durch die Analyse abgedeckt sein.

Die aktuellen Forschungsarbeiten konzentrieren sich darauf, existierende Modellierungskonzepte, wie Komponentenfehler-Bäume (vgl. [Kai03]) und *State/Event* Fehlerbäume (vgl. [Kai06]) zu erweitern und anzupassen, um die gewünschten Analysen zu ermöglichen. Dies beinhaltet in einem ersten Schritt die Definition geeigneter Modelle, die nicht nur in der Lage sein müssen, statische, sondern auch komplexe dynamische Abhängigkeiten (z. B. Adaption, zeitliche Abhängigkeiten) abzubilden. Als Referenzsystem dient hierbei unter anderem der Entwurf eines *Acc*-Teilsystems (*Adaptive Cruise Control*). Dessen Hauptaufgabe ist die Koordination des Fahrverhaltens eines Fahrzeugs in einem Verbund. Im Rahmen der Untersuchung an diesem System gilt es, die relevanten Ursache- und Wirkzusammenhänge bezüglich Sensorversagen und Verfälschung von Sensordaten formal zu erfassen und zu bewerten. Abbildung 2 zeigt einen Ausschnitt eines Komponentenfehler-Baums.

Schadcode und mögliche Einflüsse auf die Sicherheit

Während die Abschätzungsmethodik im *Safety*-Bereich bereits etabliert ist, müssen auch Abhängigkeitsstrukturen hinsichtlich einer umfassenden Sicherheit beachtet werden. Bei komplexen, offenen Systemen muss damit gerechnet werden, dass eine mangelhafte Datensicherheit (*Security*) Gefährdungen im Sinne von *Safety* hervor-

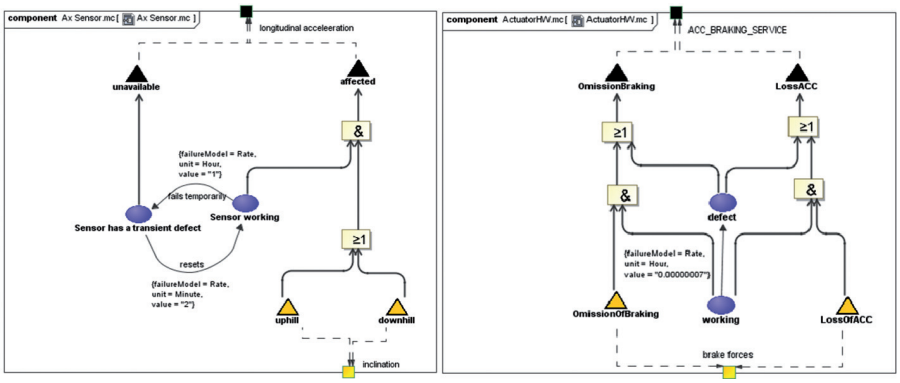

Abb. 2: Auszug der erweiterten Komponenten-Fehlerbäume, erstellt im Rahmen der ACC-Analysearbeiten.

rufen kann. Bei der Betrachtung von komplexem Systemverbünden, wie sie z. B. im Fahrzeug eingesetzt werden, muss man bei der Angriffssicherheit zwischen eingeschleustem Schadcode und Angriffen von außen unterscheiden.

Schadcode-Erkennung und die durch den Nutzer einzuleitenden Maßnahmen unterscheiden sich in der Domäne eingebetteter Systeme grundlegend von der im Desktop-Bereich. Die Gründe hierfür sind die zur Verfügung stehenden Ressourcen und die Funktionalität des Systemverbunds. Bereits bei der Entwicklung müssen potenzielle Eigenschaften von Schadcode betrachtet werden – das schließt unter anderem die folgenden Eigenschaften ein (vgl. hierzu auch [Kil06]):

- Verbreitungsmethode.
- Aktivierung.
- Unterbringung auf dem System, insbesondere bei verteilten Komponenten.
- Wirkungsweise auf das System, inklusive Fragen der Steuerung
- Nutzung und Art der Kommunikationsinfrastruktur.
- Funktionalität des Schadcodes, vom Kommunikationsmanagement bis hin zur Spionage.
- Selbstschutzmechanismen der Schadsoftware, wie Polymorphie oder Tarnung.

Um die Analyse während der Entwicklung effizient zu unterstützen, sind eine formale Klassifikation potenzieller Schadcode-Software und deren Zusammenspiel in den Softwarekomponenten zielführend. Hierbei können die obigen Eigenschaften entsprechend ihren möglichen Ausprägungen einfach aufgenommen und während der Entwicklung entsprechend den Szenarien, Ressourcen und Anforderungen getestet werden.

Eine weitere Frage bei der Berücksichtigung der Domäne eingebetteter Systeme ist die Behandlung von identifiziertem Schadcode. Dabei ist zu berücksichtigen, dass einerseits das Gesamtsystem nach Möglichkeit weiter funktionieren muss und eine Intervention durch den Nutzer möglichst gering gehalten werden sollte. Andererseits müssen Informationen über die Entdeckung und mögliche Vorgehensmaßnahmen aufgrund kleiner Displays einfach und verständlich gestaltet sein. Hier muss eine Vielzahl von Fragen der Visualisierung bezüglich Eindeutigkeit und Verständnis berücksichtigt werden. Nur durch eine frühzeitige Einbeziehung potenzieller Nutzer und ein entsprechendes Anforderungsprofil in der Bedienung kann sich an dieser Stelle ein System erfolgreich in der Praxis bewähren. Das schließt darüber hinaus auch Fragen der Interaktion zwischen Nutzer und System ein. In Abhängigkeit vom Nutzerprofil ist dabei der Informations- und Interaktionsgrad adaptiv anzupassen.

Interoperabilitätsebene	Beispiel in der Fahrzeugtechnik
Technisch	CAN-Bus.
Pragmatisch	Stelle ich die Bremse am richtigen Rad?
Sozial	Nutzerakzeptanz.
Semantisch	Spezifikation der Maßeinheiten, Mnemonische Namen der Datenfelder.
Politisch / Menschlich	Gutes ABS vs. schlechte Atomenergie.
Organisatorisch	Unfallfreie Fahrt bzw. Schadensminimierung.
Juristisch	Darf die Bremse autonom gestellt werden?
Physisch	Steckerform, Pin-Belegung.
International	Eignung für den weltweiten Markt.
Empirisch	ABS / ESP im Fahrsicherheitstraining.
Dynamisch	Ausfallerkennung der Drehgeber.
Syntaktisch	Datenfelder und -typen der CAN-Nachrichten.
Konzeptionell	Kompatibler Drehgeber mit anderem Wirkprinzip des Sensors.

Tabelle 1: Interoperabilitätsebenen in der Fahrzeugtechnik.

Faktoren Interoperabilität und Sicherheit

Das Zusammenspiel unterschiedlicher Systeme bedeutet zunehmend auch eine Komplexität bei der Kopplung und Integration innerhalb des Gesamtsystems. Eine Infrastruktur im eingebetteten Systembetrieb muss sicherstellen, dass der Ausfall und der Austausch defekter oder zu ersetzender Komponenten einfach gestaltet sind. Das bedeutet, dass bereits im Ent-

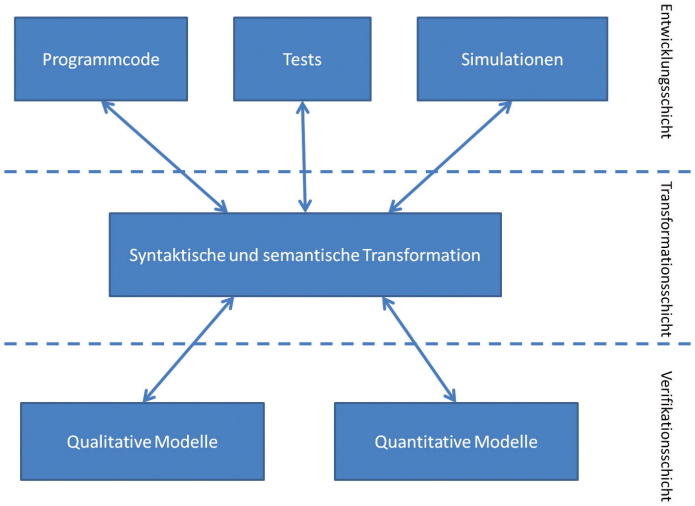

Abb. 3: Transformationen zwischen Entwicklung und Verifikation (vgl. [Güd12]).

Abb. 4: RAVON, ausgestattet mit multiplen Sensoren zur Umwelterfassung (Quelle: uni-kl.de/ravon).

wicklungsprozess geeignete Maßnahmen definiert werden müssen.

Interoperabilität erfordert ein korrektes Zusammenspiel zwischen unabhängigen und zumeist heterogenen Komponenten innerhalb eines Systems, um gemeinsam eine definierte Funktionalität zu erbringen. Theoretisch kann man durch standardisierte Austauschformate und Kommunikationsstrukturen eine einfach beherrschbare Struktur erreichen. Jedoch stellt die Einhaltung der Standards eine in der Praxis häufig sehr hohe Hürde bei der Entwicklung von Komponenten dar. Das liegt nicht nur an der Weiterentwicklung von Techniken, sondern häufig auch an den strikten oder einschränkenden Vorgaben der Standards. M. Manso, M. Wachowicz und M. Bernabé diskutieren in [Man09] eine Vielzahl von Ebenen der Interoperabilität, wie in Tabelle 1 beschrieben. Der Transfer auf Beispiele der Fahrzeugtechnik anhand von ABS & ESP (Antiblockier-System und Electronic Stability Control) findet sich ebenfalls dort.

Die Angriffssicherheit ist im großen Maße abhängig von der Interoperabilitätssteuerung, d. h. von möglichen Kommunikationsinfrastrukturen, der Architektur im Systemverbund, Kopplungskonzepten und Schnittstellen. Die Preise für eingebettete Systeme fallen stetig. Das ermöglicht den Einsatz redundanter Systemkomponenten, die sowohl die Ausfall- als auch die Angriffssicherheit erhöhen können. Redundanz ermöglicht die Validation der Reaktionen der einzelnen Komponenten. Darüber hinaus können effiziente und sichere Implementierungen parallel genutzt werden. So kann in definierten Abständen eine Validierung der Reaktionen erfolgen, ohne die Effizienz des Systems stark einzuschränken.

Bei der Entwicklung von Systemen existieren häufig Zielkonflikte. Diese Konflikte müssen bereits in der Entwicklung adressiert werden, weshalb sie bereits zu diesem Zeitpunkt erkannt und Lösungen gefunden werden sollten. Leistungsfähige Modellierungstechniken und Simulationen spielen hier eine tragende Rolle. Aktuell werden in den Domänen „Virtuelles Engineering“, „Digitales Engineering“ und „Rapid-Prototyping“ neue Methoden, Techniken und Werkzeuge eingesetzt, um bereits in den frühen Phasen der Systementwicklung Eigenschaften der Systeme, aber auch der zu Grunde liegenden Entwicklungsprozesse zu bewerten und entsprechend zu beeinflussen.

Sicherheits- und Zuverlässigkeitsanforderungen

Bei der Entwicklung neuer Systeme steht oft die effiziente Konstruktion im Vordergrund. Das kann sich auf unterschiedliche Aspekte, wie *Time-to-Market* oder Kosten, beziehen, die im Unterschied zur Qualität leicht erfasst werden können. Qualität erfordert die Beachtung vieler Aspekte. Neben Fragen der Sicherheit sind unter anderem auch Fragen zur Zuverlässigkeit, Korrektheit, Verfügbarkeit und Fehlertoleranz zu beantworten. Allein eine nachträgliche Überprüfung des Systems ist dabei nicht zielführend. Bereits in frühen Phasen der Konzeption und des Entwurfs – aber auch während der Entwicklung – müssen zweckdienliche Maßnahmen ergriffen werden. Modellbasierte Ansätze, formale

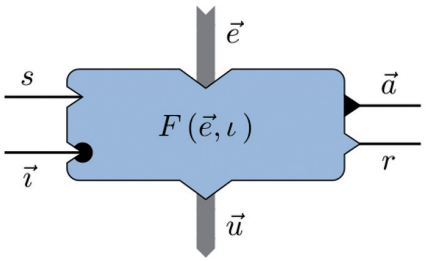

Abb. 5: Grafische Darstellung eines Verhaltensmoduls mit Eingangs- und Ausgangssteuerwerten (Quelle: [Pro10]).

Methoden und stochastische Verfahren können hier geeignete Lösungen bieten.

Um noch nicht realisierte Komponenten eines Systems bei dieser Analyse mit betrachten zu können, ist die Integration entsprechender Modelle notwendig. Aus diesen Modellen müssen die zu betrachtenden Eigenschaften abgeleitet werden. Dabei ist es nicht hinreichend, das System – sowohl software- als auch hardwareseitig – zu modellieren, sondern es müssen darüber hinaus Aspekte der Umgebung in die Analyse einbezogen werden. Im Anschluss kann – unter Verschmelzung von Eigenschaftsbeschreibung und Modell – eine Analyse erfolgen. Hierbei ist festzuhalten, dass die stochastische Analyse durch eine Vielzahl von Parametern und eine hohe Anzahl an Teilkomponenten erschwert wird. Auch die Definition der Ziele kann konfliktbehaftet sein. Ein möglicher Ausweg aus diesem Dilemma ist die Identifikation von potenziellen Kompromissen durch eine Pareto-Optimierung

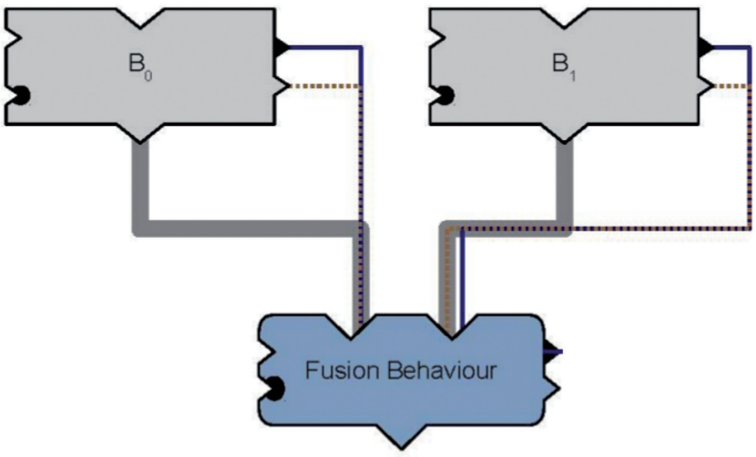

Abb. 6: Beispiel für die Kopplung zweier Verhaltensmodule mittels eines Fusionsmoduls (Quelle: [Pro10]).

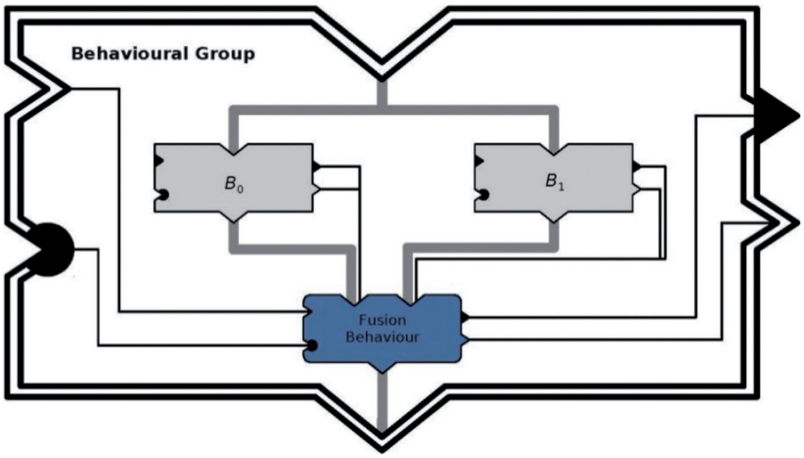

Abb. 7: Beispiel für die hierarchische Organisation auf unterschiedlichen Ebenen (Quelle: [Pro10]).

(vgl. [Ste86]). So kann der Multidimensionalität im Entwicklungsprozess Rechnung getragen werden und gleichzeitige Fehlentwicklungen können vermieden werden.

Aufgrund der unterschiedlichen Methoden und einzusetzenden Tools ist eine Integration in den Entwicklungsprozess unabdingbar, um eine Steuerung der nicht-funktionalen Eigenschaften der Systeme handhabbar zu gestalten (siehe Abbildung 3). So ermöglicht es beispielsweise eine formale Beschreibung in der Sprache SAML (vgl. [Güd12]), Sicherheitsanforderungen zu spezifizieren und bereits im Entwicklungsprozess den Einfluss dieser Faktoren zu steuern.

Literatur & Links

[DIN90] DIN 25424-2:1990-04, Fehlerbaumanalyse – Handrechenverfahren zur Auswertung eines Fehlerbaums, Beuth Verlag 1990

[DIN07] DIN EN 61025:2007-08, Fehlzustandsbaumanalyse, Beuth Verlag 2007

[Güd12] M. Güdemann, M. Lipaczewski, S. Struck, F. Ortmeier, Unifying Probabilistic and Traditional Formal Model Based Analysis, in: Proc. of 8. Dagstuhl-Workshop MBEES 2012 - Model-Based Development of Embedded Systems, 2012

[Kai03] B. Kaiser, P. Liggesmeyer, O. Mäckel, A new component concept for fault trees, in: Proc. of 8th Australian Workshop on Industrial Experience with Safety Critical Systems and Software – SCS 2003

[Kai06] B. Kaiser, State/event fault trees. A safety and reliability analysis technique for software-controlled systems, Verlag Dr. Hut 2006

[Kil06] S. Kiltz, A. Lang, J. Dittmann, Klassifizierung der Eigenschaften von Trojanischen Pferden, D-A-CH Security 2006, Syssec 2006

[Lig00] P. Liggesmeyer, Qualitätssicherung softwareintensiver technischer Systeme, Spektrum-Verlag 2000

[Lig09] P. Liggesmeyer, Software-Qualität, Spektrum-Verlag 2009

[Man09] M. Manso, M. Wachowicz, M. Bernabé, Towards an Integrated Model of Interoperability for Spatial Data Infrastructures, Transactions in GIS 13 (1), 2009

[Pro10] M. Proetzsch, Development Process for Complex Behavior-Based Robot Control Systems, ser. RRLab Dissertations, Verlag Dr. Hut 2010

[Sch12] F. Schmidt, Funktionale Absicherung kamerabasierter Aktiver Fahrerassistenzsysteme durch Hardware-in-the-Loop-Tests, Dissertation, TU Kaiserslautern 2012

[Ste86] R.E. Steuer, Multiple Criteria Optimization: Theory, Computation and Application, John Wiley & Sons 1986

Autonome Fahrzeuge

Im Vergleich mit den Sicherheitsanforderungen von Fahrassistenz-Systemen übersteigen die Anforderungen von autonomen Fahrsystemen die vorherigen deutlich. Eine wesentliche Funktion autonomer Fahrzeuge ist das Erreichen eines festgelegten Zielpunkts, ausgehend vom definierten Startpunkt. Hierbei werden die von der Sensorik generierten Umgebungsinformationen genutzt, um dem System die Planung und das sichere Abfahren von Routen zu ermöglichen.

Eine besondere Eigenschaft derartiger Systeme ist es, dass sie eigenständig, also ohne Eingriff von außen, spezifiziertes Verhalten korrekt, zuverlässig und sicher umsetzen. Dabei sollen Gefährdungen (z. B. Hindernisse oder Personen auf der Strecke) selbstständig erkannt und – wenn notwendig – entsprechende Gegenreaktionen initiiert werden. Ähnlich wie bei den Assistenzsystemen wird eine Vielzahl von heterogenen Sensoren zu einem Verbund gekoppelt. Die produzierten Daten fusionieren zu einem gemeinsamen Umgebungsmodell, das für die Navigation und Kollisionsvermeidung verwendet wird.

Die Forschungsarbeiten im Rahmen der Entwicklung neuer Techniken für Sicherheitsanalysen nutzen unter anderem das von der Arbeitsgruppe Robotik an der TU Kaiserslautern entwickelte *RAVON*-System (*Robust Autonomous Vehicle for Off-Road Navigation*) (siehe Abbildung 4) als Evaluierungsplattform. Dieses zeichnet sich durch die Fähigkeit aus, auf der Grundlage eines komplexen Verhaltensnetzwerks (dem „Integrated Behavior-based Control“ (iB2C) Network), Umgebungsinformationen in Navigations- und Steuerungsbefehle umzuwandeln. Im Kern setzt sich dieses Netzwerk aus einzelnen Modulen, den so genannten Verhaltensmodulen, zusammen (vgl. [Pro10], siehe Abbildung 5). Jedes dieser Module übernimmt bestimmte Aufgaben, wie z. B. die Berechnung von Abstands- oder Geschwindigkeitswerten aus gegebenen Eingabewerten ($\vec{e}$). Die errechneten Ausgabevektoren ($\vec{u}$) können wiederum als Eingabe für andere Module dienen. Durch Stimulation (s), Inhibition ($\vec{i}$) oder Aktivitätssignale ($\vec{a}$) können sich diese Module gegenseitig beeinflussen. Um komplexes Verhalten zu realisieren, werden Gruppen von Verhaltensmodulen mittels so

genannter Fusionsmodule zusammengeschaltet (siehe Abbildung 6).

Das entstehende Netzwerk organisiert sich auf unterschiedlichen Hierarchieebenen (siehe Abbildung 7). Dabei gibt es übergeordnete Ebenen, die z. B. für die Routenplanung verantwortlich sind, und untergeordnete Verhaltensebenen, deren Aufgabe ist es z. B., Anweisungen in Steuerdaten für Lenkung, Motor usw. umzuwandeln.

Ab einem bestimmten Punkt erreichen diese Netze eine Komplexität, bei der es nicht mehr möglich ist, einfache Zusammenhänge betreffend Sicherheit und Verlässlichkeit zu erfassen (beispielsweise setzt sich der RAVON-Demonstrator aus über 500 Modulen zusammen). Fragestellungen, die speziell in der Verwendung von diesen Verhaltensnetzen auftauchen, sind unter anderem:

- Wie wirken sich Ausfälle einzelner Module auf die Sicherheit und Zuverlässigkeit des Gesamtsystems aus?
- Gibt es Module, die sich gegenseitig blockieren und nicht-spezifiziertes Verhalten hervorrufen können?
- Wie wirkt sich Sensorrauschen auf das Gesamtverhalten aus?
- Unter welchen Bedingungen ist der Systembetrieb nicht mehr sicher?
- Wie wahrscheinlich ist ein Systemversagen unter bestimmten Bedingungen?

Aufgrund der Komplexität und der starken Abhängigkeit der Elemente der Verhaltensnetze sind Verfahren wie einfache klassische Fehlerbaum-Analysen zur Bewertung der Systemsicherheit nicht geeignet. Der Einsatz von *State-Event*-Fehlerbäumen ermöglicht es, dynamische Aspekten (z. B. Verhaltensadaption) in die Untersuchungen einzubeziehen. Die Module werden als Fehlerkomponenten modelliert und deren Abhängigkeiten in Form von Ereignissen beschrieben. Die Modellierung konzentriert sich dabei ausschließlich auf Bereiche des Systems, die zu den Sicherheitsmechanismen in Beziehung stehen. Zur Analyse werden die Modelle in eine Form von Petri-Netzen überführt (vgl. [Kai06]). Auf diese Weise können die internen Zustände von Modulen sowie deren Abhängigkeiten zu unerwünschten Systemzuständen in Relation gesetzt werden.

Ein Ansatz zur Bewertung der Zuverlässigkeit verfolgt die Transformation von relevanten Teilen der Verhaltensnetze in endliche Automaten. Das ermöglicht weitere Analysen, z. B. die Untersuchung der Erreichbarkeit und die Berechnung der Wahrscheinlichkeit des Eintritts eines bestimmten Zustands. Eine besondere Herausforderung im Rahmen von quantitativen Analysen stellt dabei das Abschätzen der Ausfallwahrscheinlichkeiten für die Verhaltensmodule dar, da diese als Software realisiert sind und entsprechende Kennwerte somit nur schwierig zu erfassen sind. ■