- Erster Entwurf -

Sicherheitsanalyse - Kombination der Techniken

Hardware- und Softwaresysteme stellen zunehmend Funktionen bereit, deren Ausfall hohe Kosten oder sogar Gefährdungen von Menschenleben bewirken kann. Zu nennen sind beispielsweise computerkontrollierte Bremssysteme in der Automobiltechnik, Autopiloten in der Luftfahrtindustrie oder Kontrollstände in der Reaktortechnik. Bei der Entwicklung solcher sicherheitskritischer Systeme müssen in der Anforderungsanalyse neben den funktionalen Anforderungen (Was soll das System leisten?) insbesondere Anforderungen an die Sicherheit (Was darf nicht passieren?) analysiert werden. Die beiden Anforderungstypen erfordern zwei unterschiedliche Perspektiven auf das System. Während bzgl. der Funktionalität der Blick auf die Verlässlichkeit gerichtet wird, nimmt man bei der Sicherheitsanalyse mögliche Gefahren ins Visier. In manchen Systemen sind Funktionalität und Sicherheit identisch, allerdings spielen z.B. unübersichtliche Anzeige- und Bedieninstrumenten im Hinblick auf die Verlässlichkeit keine Rolle. Für die Sicherheit hingegen stellen sie ein großes Problem dar. In manchen Systemen, wie beispielsweise einem nuklearen Reaktor steht Zuverlässigkeit teilweise sogar im Widerspruch zur Sicherheit. Ein zuverlässig entworfener Reaktor wird auch bei einer Überhitzung versuchen, die Funktionalität so lange wie möglich aufrecht zu erhalten, was in letzter Konsequenz zu einer Katastrophe führen kann. In einem sicheren Design besteht die höchste Priorität darin, die Katastrophe zu verhindern, weshalb auch ein Shutdown in Kauf genommen werden muss. Um potentielle Gefahren aufzudecken, werden bei der Sicherheitsanalyse folgende Fragen gestellt: *Welche Gefahren gibt es? Wie können sie verursacht werden? Zu welchen Konsequenzen können sie führen? Wie kritisch und wahrscheinlich sind sie?* Zur Beantwortung stehen eine Reihe von Techniken zur Verfügung. Eine gründliche Sicherheitsanalyse kann erst unter Anwendung eines Mix von Techniken durchgeführt werden. Der vorliegende Text zeigt auf, warum dies so ist und welchen Beitrag die einzelnen Techniken liefern. Betrachtet werden die Hazard and Operability Study, Fehlerbaumanalyse, Fehler Modus und Effekt Analyse und die Ereignisbaumanalyse.

Schlüsselwörter: Sicherheitsanalyse, HAZOP; FTA, FMEA, ETA, Design.

I. Einleitung

Ziel der Sicherheitsanalyse muss es sein, Gefahren und deren Ursachen möglichst umfassend aufzudecken. Dabei muss jede Systemeinheit betrachtet und hinsichtlich ihres Gefahrenpotentials bewertet werden. Eine systematische Vorgehensweise kann mit Hilfe einer Reihe von Methoden erreicht werden. Aber was bieten die Methoden und welche Methode ist am besten geeignet? Die Auswahl wird in der Industrie in der Regel auf Basis von Traditionen getroffen. In der Automobilindustrie bevorzugt man die Fehler Modus und Effekt Analyse (FMEA), die Fehlerbaumanalyse (FTA) entspringt der Luft- und Raumfahrt und wird dementsprechend hier intensiv eingesetzt, ähnlich verhält es sich mit der Hazard and Operability Study (HAZOP) in der chemischen Industrie. Notwendig ist aber die Anwendung unterschiedlicher Methoden, da jede einzelne einen anderen Fokus auf das zu untersuchende System einnimmt (vgl. Abb. 1). Fenelon und McDermid [FM94] schlagen Kausalität als integrierendes Konzept der Techniken vor. Aus dieser Sicht erlaubt die Systematik von HAZOP insbesondere die Ableitung einer Menge unverknüpfter Fehlermöglichkeiten. Mit Hilfe der FTA und FMEA können diese in Relation gesetzt werden, wobei die FTA mit einem Systemfehler beginnt und in Form einer Rückwärtsanalyse multiple Ursachen ableitet. Die FMEA hingegen beginnt mit einer Ursache und leitet in entgegengesetzter Richtung im Rahmen einer Vorwärtsanalyse multiple Systemfehler als mögliche Konsequenzen ab. In diesem Text wird die Ereignisbaumanalyse (ETA), die bei Fenelon und McDermid nicht erwähnt wird, hinzugenommen. Bei der ETA liegt der Fokus auf Sequenzen von Systemfehlern, die nicht unbedingt in kausaler Beziehung stehen, aber dennoch eine wichtige Ergänzung der anderen Techniken darstellt. Es wird untersucht, wie effektiv Ausfälle des Systems durch vorhandene Backup-Methoden abgefangen werden können.

Die aus den Arbeiten von McDermid entstandene Methode HIP-HOP [PM99] integriert u.a. FTA und FMEA. Allerdings werden hier die Möglichkeiten der beiden Methoden nicht vollständig ausgeschöpft. In dem vorliegenden Aufsatz werden die vier genannten Methoden vorgestellt und anhand eines einfachen Beispiels wird illustriert, welche Möglichkeiten der gegenseitigen Ergänzung aufgrund der unterschiedlichen Foki bestehen. Insbesondere wird hierbei konsequent der in der Literatur oft nicht

explizit berücksichtigte Unterschied zwischen horizontalen (auf derselben Designebene) und vertikalen Fehlerbeziehungen (über unterschiedliche Designebenen) herausgestellt.

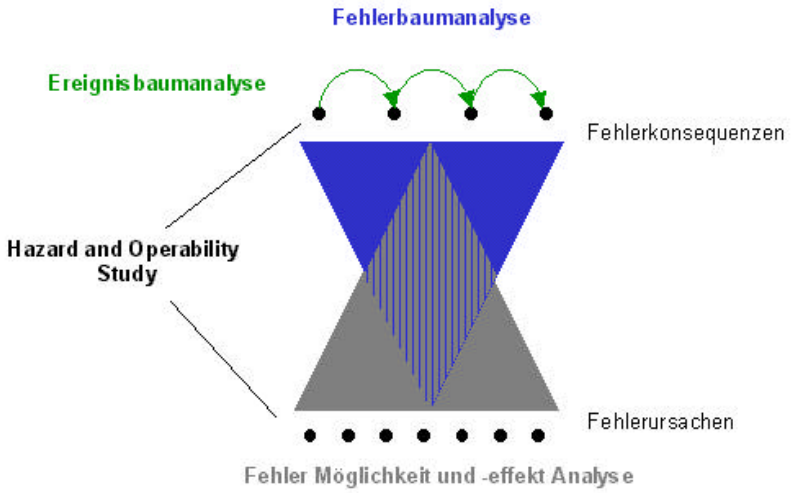

Abb. 1: Zusammenhang der Sicherheitsanalysetechniken

Zunächst wird im nächsten Abschnitt der Prozess beschrieben, in den die Techniken einzubetten sind. Hierbei wird ein einfaches System zur Berechnung von Bremssignalen, wie es beispielsweise in Automobilen zum Einsatz kommen kann, eingeführt. Dazu wird ein Referenzmodell angegeben, das alle Fehlermöglichkeiten des Systems enthält. In den folgenden vier Abschnitten werden die Techniken jeweils beschrieben und anschließend zur Analyse des Beispielssystems angewendet. Bezugnehmend auf das Referenzmodell wird deutlich gemacht, welchen Beitrag zur umfassenden Fehleraufdeckung die jeweiligen Methoden leisten können.

II. Der Sicherheitsanalyseprozess

Generell gilt bei der Entwicklung sicherheitskritischer Systeme: Je früher Gefahren erkannt und berücksichtigt werden, desto geringer ist der notwendige Aufwand. Aus diesem Grund wird mit der Sicherheitsanalyse bereits zu Beginn der Entwicklung begonnen und parallel dazu weitergeführt. Für die Entwicklung technischer Systeme existieren unterschiedliche Prozessmodelle [XX], die mindestens die Phasen Anforderungsanalyse, Entwurf, Implementation, Test und Auslieferung umfassen. Die Phasen werden sequentiell (z.B. Wasserfallmodell) oder iterativ (z.B. Spiralmodell) durchlaufen. Den folgenden Ausführungen liegt ein hierarchischer modellbasierter Prozess [XX] zugrunde. Dabei wird ausgehend von abstrakten Anforderungen ein Systemmodell entworfen, das zur Ableitung konkreterer Anforderungen dient. Iterativ werden System und Anforderungen verfeinert, bis eine Ebene verfügbarer oder leicht zu implementierender Komponenten erreicht ist. Der Zweck der Zerlegung in Systemelemente besteht darin, überschaubare Systemeinheiten zu erhalten, die gegebenenfalls unabhängig voneinander in unterschiedlichen Teams als integrierte Schaltungen oder Softwaremodule implementiert werden können.

In Abb. 2 ist ein hierarchischer Entwurf über drei Ebenen für ein Berechnungssystem von Werten zur Ansteuerung von Bremsaktuatoren zu sehen. Dieses Beispielsystem wird in den folgenden Abschnitten mit Hilfe der hier behandelten Sicherheitsanalysemethoden auf potentielle Gefahren untersucht. Es sind zwei Eingänge und drei Ausgänge vorgesehen. Der Wert des Eingangs 'Fahrerbefehl' (FB), der an einen Pedalsensor angeschlossen wird, gibt an, wie stark der Fahrer das Bremspedal durchtritt. Der Eingang 'Achslast' (AL) wird an einen Gewichtssensor zur kontinuierlichen Messung der Beladung des Fahrzeugs angeschlossen. Die Ausgänge 'Bremswert1' (BW1) und 'Bremswert2' (BW2) sind zur Ansteuerung von Bremsaktuatoren eines Fahrzeugs vorgesehen. Sobald das Bremspedal betätigt wird, soll unter Berücksichtigung der Fahrzeugbeladung ein Bremswert proportional zur Position des Bremspedals berechnet werden. Der zweite Bremswert ist redundant und dient für den Fall, dass am ersten Output kein Wert anliegt. Die Berechnung dieses Wertes ist simpler, denn sie berücksichtigt nicht die Fahrzeugbeladung. Im Regelfall werden beide Werte, im Fehlerfall aber mindestens der zweite geliefert. Die Auswahl zwischen den beiden muss vom umgebenen System vorgenommen werden. Zeitgleich zu den Bremswerten wird ein weiterer Output

erzeugt, der zur Ansteuerung der Bremsleuchten dient, so dass der Bremsvorgang nachfolgenden Fahrzeugen angezeigt wird. Die angeschlossenen Lichter sollen proportional zur Bremskraft aufleuchten. Im Verlauf der Designzerlegung entscheiden sich die Designer, die Berechnung der Brems- und Leuchtwerte und schließlich auch die Berechnung der beiden Bremswerte in unterschiedlichen Modulen durchzuführen. Die Bremsmodule schicken den berechneten Wert sowohl nach außen als auch an das Leuchtmodul, wo dann proportional dazu ein Leuchtwert berechnet wird.

Die Zerlegung wird fortgeführt bis auf der untersten Designebene Komponenten erreicht werden, die im Design als atomar angesehen werden. Dies können Hardwarekomponenten, zugelieferte Teilsysteme oder leicht zu implementierende Softwaremodule sein. Beim Bremssystem wird die Designzerlegung auf der dritten Ebene beendet, denn es sei angenommen, dass es sich bei den hier vorhandenen Systemelementen um Module handelt, die aus anderen Projekten bereits implementiert vorliegen. Bzgl. des Leuchtmoduls wird bereits auf der zweiten Ebene festgelegt, dass ein vorhandenes verwendet werden soll.

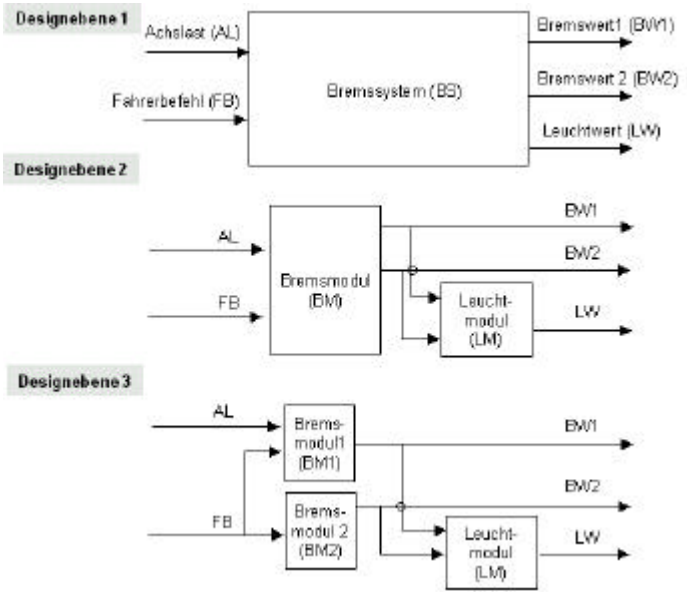

Abb. 2: Beispiel einer Designzerlegung für ein Bremssystem

Die Sicherheitsanalyse sollte parallel zum Design durchgeführt werden. Abb. 3 verdeutlicht wie diese beiden Prozesse ineinander greifen. Die Systemmodelle gehen in die Sicherheitsanalyse ein und werden dort auf potentielle Fehler untersucht. Auf Basis der aufgedeckten Fehler werden Sicherheitsanforderungen abgeleitet, die an den Designprozess gegeben und dort bei den folgenden Verfeinerungsschritten eingehalten werden müssen, um die Fehler zu eliminieren bzw. reduzieren. Bereits nachdem die erste Designebene vorliegt, kann mit der Sicherheitsanalyse begonnen werden. Zu diesem frühen Zeitpunkt der Entwicklung wird nach Systemfehlern gesucht. Dies sind Fehler, die an der Systemoberfläche bemerkbar sind, was sich darin äußert, dass eine gewünschte Funktionalität gar nicht mehr oder nur noch teilweise geliefert wird oder dass unerwünschte Nebeneffekte auftreten. Unter Abschätzung der Auswirkungen der Fehler auf die Systemumgebung wird die Kritikalität der Systemfehler bestimmt. Dabei muss abgeschätzt werden, wie sich die Beeinträchtigung der Funktionalität bei der Anwendung des Systems in unterschiedlichen Anwendungssituationen auswirken kann. Bei hoher Geschwindigkeit und dichtem Verkehr ist ein Versagen der Bremsung fataler als im Schritttempo auf einer freien Strasse. Durch Analyse der verfeinerten Systemmodelle werden Ursachen für die Systemfehler abgeleitet und die Sicherheitsanforderungen präzisiert. Die Analyse der untersten Designebene liefert Fehler der kleinsten Designeinheiten, die sogenannten Komponentenfehler. Bei den Komponenten handelt es sich in den meisten Fällen, um Designeinheiten, die von Zulieferern bezogen werden oder im eigenen Unternehmen vorhanden sind und zu denen bereits Fehlererfahrungen aus dem Einsatz in anderen Systemen existieren. Fehler pflanzen sich ausgehend von Komponenten durch das System fort und äußern sich schließlich als Systemfehler an der Systemoberfläche, so dass die Kausalität sich genau entgegengesetzt zum Designfortgang verhält. Dies bedeutet für die Sicherheitsanalyse, dass auf Basis der frühen Systemmodelle nur sehr hypothetische und allgemeine Fehlerprognosen möglich sind und konkrete

Aussagen erst abgeleitet werden können, wenn feststeht, aus welchen Komponenten das System bestehen soll. Auf der Komponentenebene ist es aufgrund der Erfahrungen auch möglich, Fehlerwahrscheinlichkeiten anzugeben und aufbauend darauf Wahrscheinlichkeiten für Systemfehler zu berechnen.

In der industriellen Praxis ist die Verzahnung von Design und Sicherheitsanalyse leider noch sehr schwach. McDermid et al. [FMPN94] sprechen bezeichnend von einem "over the wall process" und beschreiben damit, dass Design und Sicherheitsanalyse von unterschiedlichen Abteilungen mit mangelnder Kommunikation durchgeführt werden. Neben der organisatorischen Trennung ist für diesen Zustand die fehlende Integration der Techniken für Design und Analyse verantwortlich. Zusammen mit der in diesem Aufsatz behandelten Verschmelzung der Techniken innerhalb der Sicherheitsanalyse ist die Integration von Design und Analyse derzeit eine große Herausforderung (mehr zum letzteren in [FMPN94]).

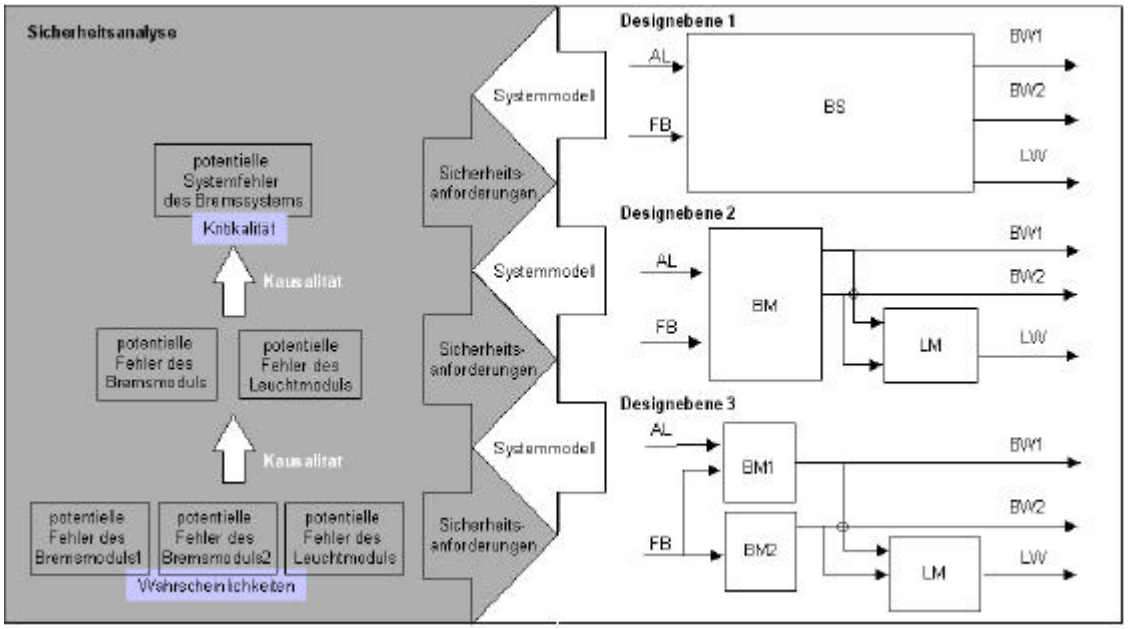

Abb. 3: Zusammenspiel zwischen Design und Sicherheitsanalyse

Die Sicherheitsanalyse wird in unterteilt in Gefahrenidentifikation, kausale Analyse und Risikoeinschätzung [Lev95]. Im folgenden werden diese drei Schritte beschrieben und anhand des Bremssystems unter Annahme gewisser Vereinfachungen durchgeführt. Dabei wird ein kausales Netz aufgebaut, das die Fehlermöglichkeiten und deren kausale Beziehungen aufzeigt. In diesem Abschnitt wird noch nicht beschrieben, wie der Inhalt des kausalen Netzes aus dem Systemmodell abgeleitet werden kann, vielmehr soll es als Referenzmodell für den nächsten Abschnitt dienen, in dem verdeutlicht wird, welchen Beitrag zur Aufdeckung der Fehler die einzelnen Sicherheitsanalysetechniken liefern.

Gefahrenidentifikation

Die Gefahrenidentifikation (hazard identification) wird auch als vorbereitende Gefahrenanalyse (preliminary hazard analysis - PHA) bezeichnet, da es sich im Wesentlichen um eine Vorbereitung für die anschließende kausale Analyse handelt. Die Aufgabe besteht darin, möglichst umfassend die Gefahren zu bestimmen, die beim Einsatz des Systems in konkreten Anwendungssituationen auftreten können. Im Zusammenhang mit dem Bremssystem, gibt es beispielsweise folgende Gefahren:

G1. Der Fahrer tritt das Bremspedal, aber das Fahrzeug wird nicht gebremst.
G2. Das Fahrzeug wird korrekt gebremst, aber die Bremsleuchten leuchten nicht auf.

Neben den aufgezählten gibt es noch weitere Gefahren, z.B. kann es passieren, dass das Fahrzeug gebremst wird, obwohl das Bremspedal nicht getreten wurde oder dass das Fahrzeug zu spät

gebremst wird. Um den Umfang der Analyse einzuschränken, werden im folgenden nur G1 und G2 untersucht.

Zusätzlich zur Identifikation muss eine Einschätzung der Kritikalität vorgenommen werden. G1 kann katastrophal sein, vor allem wenn die Bremsung bei einer hohen Geschwindigkeit und dichtem Verkehr nicht erfolgt. G2 ist weniger kritisch, dennoch sind Bremsleuchten ein wichtiges Sicherheitsfeature, das einen Beitrag zur Vermeidung von Auffahrunfällen leisten soll.

Auf Basis der Kritikalität der identifizierten Gefahren kann das Ausmaß des Risikos, dass mit der Entwicklung verbunden ist, eingeschätzt und notwendige Kontrollmechanismen, Verantwortlichkeiten und Ressourcen für die Entwicklung abgesteckt werden. Die Gefahrenidentifikation wird derzeit nur sehr schwach mit Methoden unterstützt, denn zu diesem Entwicklungszeitpunkt liegen kaum Strukturinformationen vor und man ist auf Erfahrungen aus früheren Projekten angewiesen. Diese Erfahrungen sind oftmals nur in den Köpfen der damals beteiligten Experten vorhanden, insofern beschränken sich die existierenden Methoden auf Techniken zur Moderation von Expertensitzungen [Lev95], bei denen durch gezielte "Was-wäre-wenn"-Fragen die Erfahrungen der Gruppenmitglieder gesammelt werden sollen.

Kausale Gefahrenanalyse

Bei der anschließenden kausalen Gefahrenanalyse (hazard causal analysis) wird untersucht, wodurch die zuvor identifizierten Gefahren verursacht werden können. Idealerweise wird sie parallel zum Design durchgeführt und nach jedem Designschritt verfeinert. Es ist wichtig, deutlich zu machen, auf welcher Abstraktionsebene die Kausalität untersucht wird. Sollen physikalische Eigenschaften von Systemmodulen berücksichtigt werden, z.B. dass Leiterbahnen unterbrochen sein können oder wird hiervon abstrahiert? Um Fragen zur Detaillierungsgrad der kausalen Analyse zu klären, wird ein Fehlermodell definiert. Das Fehlermodell muss selbstverständlich auf die im Design vorhandenen Informationen über das System abgestimmt sein. Für diesen Text wurde das Fehlermodell aus [VDA96] gewählt mit einer Unterscheidung eingehender, ausgehender und interner Fehler. Diese Betrachtungsweise beruht auf der Annahme, dass die Systemelemente Input erhalten, diesen verarbeiten und einen Output liefern, der dann an andere Systemelemente weitergeleitet wird. Ein ausgehender Fehler kann entweder durch einen eingehenden oder internen Fehler verursacht werden.

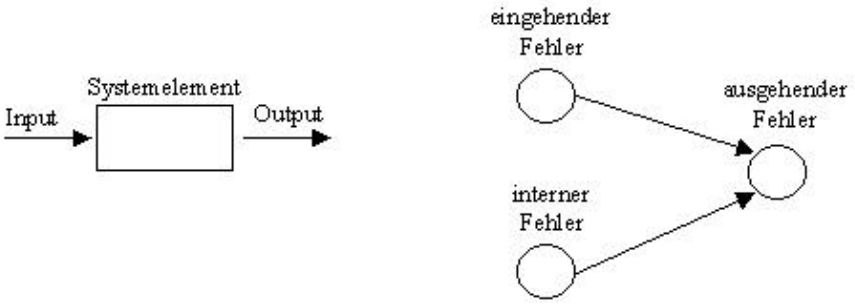

Abb.4: Fehlermodell

Bereits sehr früh im Entwicklungsprozess, wenn nur das Input-Output-Verhalten des Systems bekannt ist, kann eine System-Gefahrenanalyse (system hazard analysis) durchgeführt werden. Dabei wird die Interaktion des zu entwickelten Systems mit anderen Systemen betrachtet und insbesondere untersucht, welche Systemfehler es gibt und wie diese zu den identifizierten Gefahren beitragen können. Unter Bezug auf das Fehlermodell handelt es sich bei Systemfehlern um fehlerhafte Outputs des Systems. Zu G1 trägt das Bremssystem bei, wenn es keinen der beiden Bremswerte liefert. G2 kann durch das System verursacht werden, wenn der Leuchtwert ausbleibt. Betrachtet werden müssen also die ausgehenden Fehler: „Bremssystem liefert BW1 nicht“ (¬BS_BW1), „Bremssystem liefert BW2 nicht“ (¬BS_BW2)und „Bremssystem liefert LW nicht“ (¬BS_LW). Weiterhin soll der Fall betrachtet werden, dass die Werte zunächst an den Ausgängen anliegen, dann aber nach kurzer Zeit "verschwinden", obwohl der Fahrer noch das Bremspedal tritt: „Bremssystem liefert BW1 instabil“ (≈BS_BW1), „Bremssystem liefert BW2 nicht“ (≈BS_BW2)und „Bremssystem liefert LW nicht“ (≈BS_LW). Dieser Fehler ist im Zusammenhang mit den auf der dritten Designebene verwendeten Bremsmodule in der Vergangenheit aufgetreten und muss deshalb hier berücksichtigt werden.

Bei Systemen, die eine intensive Interaktion mit dem Benutzer erfordern (z.B. Autopiloten in Flugzeugen) wird zusätzlich eine operationale Gefahrenanalyse (operational hazard analysis) durchgeführt. Dabei werden zunächst die Aufgaben bestimmt, die der Benutzer durchzuführen hat, und es wird eingeschätzt, ob er diese in jeder Situation bewältigen kann und inwiefern mögliche Fehlbedienungen zu den identifizierten Gefahren beitragen können. Bremsen auf eisglatter Fahrbahn ist eine bekannte Gefahr. Der Fahrer muss in einer solchen Situation eine Stotterbremsung durchführen, allerdings treten die meisten Fahrer das Bremspedal instinktiv, durch und das Auto gerät ins Schleudern. Aus diesem Grund wurde ABS in bestimmte Autos eingebaut. Eine Reihe von Autoren bemängeln, dass der operationalen Analyse bisher im Designprozess zu wenig Beachtung geschenkt wurde (siehe z.B. [Bill97]). Für das Bremssystem ist sie nicht notwendig, da sich die Interaktion auf die Betätigung des Bremspedals beschränkt.

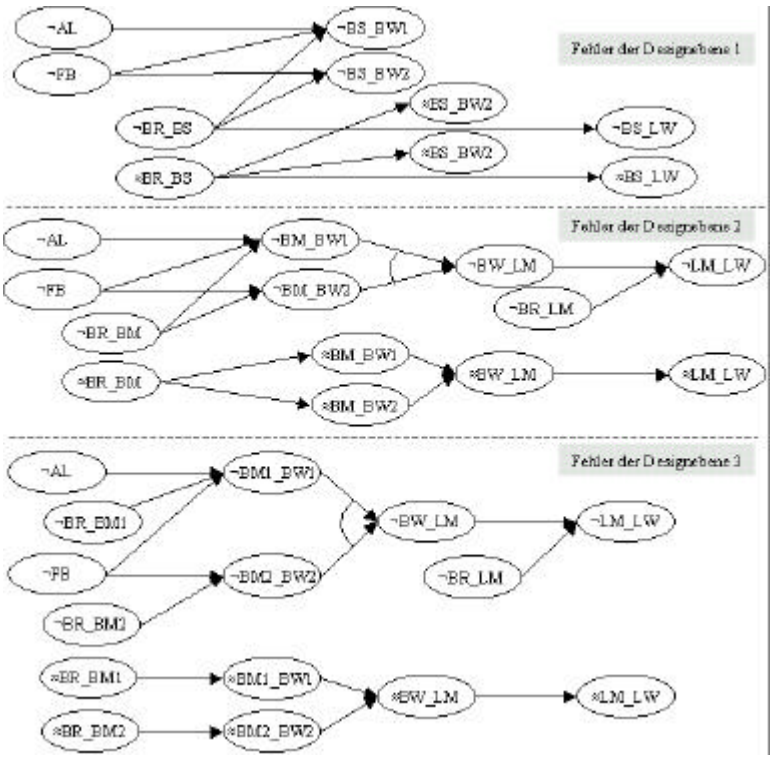

Abb. 5: Referenzmodell – normatives Ergebnis der kausalen Analyse für das Bremssystem

Im weiteren Verlauf der kausalen Analyse werden Ursachen für die ausgehenden Fehler des Systems aufgedeckt. In Abb. 5 ist eine vereinfachende Auswahl von Fehlermöglichkeiten zusammen mit den kausalen Verknüpfungen dargestellt. Dieses kausale Netz wird in den folgenden Abschnitten als normatives Referenzmodell für die vorgestellten Analysemethoden verwendet, indem gezeigt wird, welche Bestandteile mit den unterschiedlichen Methoden aufgedeckt werden können. In dem Referenzmodell wird zwischen horizontalen und vertikalen Beziehungen unterschieden. Horizontale Beziehungen herrschen zwischen Systemelementen auf derselben Designebene. Sie tauschen untereinander Daten aus. Wenn dabei ein Systemelement einen ausgehenden Fehler erzeugt, entweder weil ein interner Fehler oder ein eingehender Fehler vorlag, dann pflanzt sich dieser von Element zu Element fort. Bei diesem Prozess werden aus ausgehenden Fehlern eines Elementes eingehende Fehler des nächsten Elementes, wo der Fehler entweder abgefangen oder weitergeleitet wird. Auf der obersten Designebene existieren horizontal nur die Beziehungen zwischen den Inputs und Outputs des Gesamtsystems. Im Gegensatz zu ¬BS_BW2 hat ¬BS_BW1 einen eingehenden Fehler als mögliche Ursache mehr, da zu der Berechnung des zweiten die Achslast benötigt wird. Zusätzlich kann für beide das Ausbleiben der Berechnung (¬BR_BS) als interner Fehler verantwortlich sein. Die Instabilität der Bremswerte kann durch Instabilität bzgl. der Berechnung verursacht werden (≈BR_BS). Da für LW kein Input direkt verantwortlich ist, kommen sowohl für das Fehlen als auch für die Instabilität des Wertes lediglich die internen Fehler als Ursache in Frage. Auf der obersten Ebene soll hinsichtlich der internen Fehler nicht weiter differenziert werden, da zu diesem Zeitpunkt noch nicht feststeht, ob die Berechnungen von einem oder mehreren Modulen ausgeführt werden sollen. Im ersten Fall hätte man ein Berechnungsmodul, dass so flexibel angelegt ist, dass es

je nach Parametereinstellung unterschiedliche Berechnungen durchführen kann. Im zweiten Fall würde man Spezialroutinen anstreben. Die Entscheidung hierzwischen wird erst auf den nächsten Ebenen getroffen.

Vertikale Beziehungen bestehen zwischen den Ebenen und können analysiert werden, wenn bei fortschreitendem Design im Rahmen einer Subsystem-Gefahrenanalyse (subsystem hazard analysis) die Systemzerlegung sichtbar wird. Interne Fehler der Ebene i werden dabei durch interne Fehler der Ebene i+1 konkretisiert. Für das Bremssystem werden die Berechnungsfehler (¬BS_BR und ≈BS_BR) der obersten Designebene auf der zweiten Ebene auf ein Bremsmodul (¬BM_BR und ≈BM_BR) und ein Leuchtmodul (¬LM_BR) verteilt. Ausgehende und eingehende Fehler können konkretisiert werden, indem angegeben wird, welche Systemelemente den fehlerhaften Output liefern bzw. den fehlerhaften Input erhalten. Der fehlende und der instabile Leuchtwert wird beispielsweise auf der zweiten Ebene dem Leuchtmodul zugeschrieben (¬LM_LW und ≈LM_LW). Der fehlende Fahrerbefehl und die fehlende Achslast wirken sich auf das Bremsmodul aus (¬FB und ¬AL). Bzgl. der kausalen Relationen zwischen den Fehlern werden von Ebene zu Ebene mehr Zwischenstationen sichtbar. Tritt beispielsweise ein interner Fehler im Bremsmodul auf, dann kann es passieren, dass keiner der beiden Bremswert geliefert wird, woraufhin diese Werte als Input des Leuchtmoduls fehlen und folglich kein Leuchtwert geliefert wird. Fehlt lediglich einer der Bremswerte, kann trotzdem ein korrekter Leuchtwert geliefert werden. Für instabile Bremswerte gilt dies nicht, denn die Auswahl des Bremswertes als Basis der Berechnung erfolgt in dem eingesetzten Leuchtmodul nicht-deterministisch und es ist nicht in jedem Fall in der Lage, die Instabilität zu erkennen. Deshalb kann bereits ein instabiler Bremswert einen instabilen Leuchtwert verursachen. Im Gesamtzusammenhang muss allerdings berücksichtigt werden, dass fehlerhafte Leuchtwerte verursacht durch fehlerhafte Bremswerte von zweitrangiger Bedeutung sind. Z.B. in dem Fall, dass kein Bremswert geliefert wird, erfolgt auch keine Bremsung und somit müssen auch die Bremslichter nicht leuchten. Dieses Problem muss durch Verbesserung der Brems- und nicht der Leuchtmodule behoben werden.

Man könnte annehmen, dass z.B. der Fehler ¬BM_ LW und ¬LW_LM gleichgesetzt werden können. Allerdings gilt zwar ¬BM_ LW ⇒ ¬LW_ LM aber umgekehrt gilt nicht ¬BM_ LW ⇐ ¬LW_ LM, wenn man Transferfehler berücksichtigt. In diesem Fall kann ein eingehender Fehler auch durch einen defekten Transfer z.B. durch ein brüchiges Kabel verursacht werden, es muss nicht unbedingt vorher ein ausgehender Fehler vorgelegen haben. Da keine weiteren Informationen über die Verbindungen vorhanden sind, wird an dieser Stelle der Transferfehler zwar nicht explizit als eigener Fehler aber dennoch implizit berücksichtigt, indem die Gleichsetzung von ausgehenden und damit kausal verbundenen eingehenden Fehlern vermieden wird.

Auf Basis der untersten Designebene werden die Berechnungsfehler des Bremsmoduls auf BM1 und BM2 verteilt (¬BR_BM1, ≈BR_BM1, ¬BR_BM2 und ≈BR_BM2). Entsprechend besteht für beide Module die Möglichkeit, dass der Bremswert nicht (¬BM1_BW1 und ¬BM2_BW2) bzw. instabil (≈BM1_BW1 und ≈BM2_BW2) geliefert wird. Die Fehler, die von außen in das System eingebracht werden (¬FB und ¬AL) werden nicht auf die Bremsmodule verteilt, denn eine Unterteilung von z.B. ¬FB in ¬FB_BM1 und ¬FB_BM2 würde verschleiern, dass beide dieselbe Ursache haben: Der Fahrerbefehl wird nicht zum Bremssystem geliefert, obwohl der Fahrer das Bremspedal tritt. Um diesen sogenannten Common Cause explizit zu machen, wird durchgängig auf eine Zuweisung der eingehenden Systemsfehler auf die einzelnen Module verzichtet.

Damit ist beschrieben, welche potentiellen Fehler und Fehlerpfade des Bremssystems in der kausalen Analyse entdeckt werden müssen. Im folgenden Abschnitt wird die Frage nach dem „Wie“ beantwortet und dabei untersucht, welchen Beitrag die Techniken HAZOP, FTA, FMEA und ETA liefern. Bei der bereits oben erwähnten HIP-HOP Methode von McDermid und Papadopoulus [PM99] wird die Unterscheidung von vertikalen und horizontalen Fehlerbeziehungen vorgenommen, allerdings wird die FTA lediglich verwendet um die vertikalen und die FMEA lediglich um die horizontalen Beziehungen zu analysieren. Die folgenden Abschnitte zeigen, dass es sinnvoll ist, beide Beziehungsarten sowohl mit der FMEA als auch mit der FTA zu untersuchen. Die zusätzliche Verwendung von HAZOP und ETA bietet einen weiteren Schritt in Richtung einer integrierten Sicherheitsanalyse.

Der Vollständigkeit halber wird vorher noch die Phase der Risikoeinschätzung (risk assessment), mit der die Sicherheitsanalyse abschließt, beschrieben.

Risikoeinschätzung

Die Risikoeinschätzung setzt an, nachdem das Design abgeschlossen und das System implementiert ist. Ziel ist es, am fertigen Produkt zu belegen, dass die Designanstrengungen die erforderte Sicherheit gewährleisten konnten. Zu diesem Zweck wird jede Gefahr hinsichtlich der in der kausalen Analyse aufgedeckten Ursachen und der im Design vorgenommenen Vermeidungsmaßnahmen dokumentiert. Für Systeme, die zertifiziert werden müssen, ist es zusätzlich notwendig, dass mit dem Einsatz des Systems verbundene Risiko, quantitativ zu bestimmen. Das Risiko setzt sich aus der Wahrscheinlichkeit und Kritikalität von Gefahren zusammen. Zusätzlich zu den Wahrscheinlichkeiten der ausgehenden Systemfehler müssen Wahrscheinlichkeiten bestimmter Umgebungssituationen eingeschätzt werden, denn ein Systemfehler wirkt sich immer abhängig von der vorliegenden Umgebungssituation aus. Zu diesem Zweck werden häufig Simulatorstudien und Berechnungen auf Basis von Umgebungsmodellen durchgeführt. Da diese Modelle in den meisten Fällen die Wirklichkeit nur ungenau simulieren, sind auch die Abschätzungen ungenau.

III. Hazard And Operability Study

Die Hazard And Operability Study (HAZOP) wurde in den späten 60ern für die chemische Industrie entwickelt [CISHEC87] und basiert auf dem Prinzip, dass eine Gefahr entsteht, wenn das System oder Teile des Systems beim Betrieb von den Designintentionen abweichen. Intentionen sind dabei immer für den physikalischen Fluss von Stoffen über physikalische Leitungen zwischen Systemkomponenten formuliert. Somit liegt der Fokus von HAZOP nicht auf den Komponenten sondern auf den Verbindungen zwischen ihnen. Im folgenden wird gezeigt, dass diese Methode angewendet werden kann, um im ersten Schritt der kausalen Analyse potentielle eingehende und ausgehende Fehler aufzudecken. Die Analyse von Ursache-Wirkungs-Beziehungen wird mit HAZOP nicht systematisch unterstützt. Hierfür eignen sich die Methoden FTA, FMEA und ETA in den nächsten Abschnitten.

Ablauf einer HAZOP

In der chemischen Industrie fließen zwischen den Systemelementen Chemikalien. Eine wichtige Rolle spielen dabei die Temperatur, das Volumen, der Druck und andere Parameter der fließenden Stoffe. Voraussetzung für HAZOP ist ein Systemdesign, in dem diese Parameter für jeden Fluß hinsichtlich zulässiger Wertebereiche spezifiziert sind. In der chemischen Industrie werden solche Designs als "Pipe and Process"'-Diagramm bezeichnet. Die Absicht der Analyse ist es, mögliche Abweichungen von den zulässigen Bereichen zu identifizieren. Zusätzlich sollen Vorschläge dazu unterbreitet werden, wie die Betriebsfähigkeit (operability) trotz der Abweichungen aufrechterhalten werden kann. HAZOP lässt sich auf Computersysteme übertragen, wenn statt der chemischen Flüsse Datenflüsse betrachtet werden [BP93, CCR95, FH94, MP94]. Für Datenflüsse sind zulässige Wertebereiche einerseits implizit durch den Datentypen und andererseits explizit durch Spezifizierung zulässiger Daten gegeben, z.B. sind BW1 und BW2 vom Typ Float und müssen im Intervall [0, BW_max] liegen.

HAZOP wird in Teamsitzungen durchgeführt. Zu dem Team gehören üblicherweise ein Leiter, ein Protokollant, Designer, Benutzer und verschiedene Experten, z.B. für Human Factors. Weitere Experten werden zum gegebenen Zeitpunkt hinzugezogen. Um in der Teamsitzung Abweichungen abzuleiten, werden sogenannte Leitwörter (guidewords) angewendet. Typische Leitwörter aus der chemischen Industrie sind: „kein“, „mehr“, „weniger“, „genau so viel wie“, „ein Teil von“, „rückwärts“, „anders als“. Jeder Fluss wird unter Anwendung der Leitwörter analysiert. Die Leitwörter sollen das Nachdenken über das System in bestimmte Bahnen lenken und Ideen entlocken, sie sollen die Diskussion anregen und die Chancen erhöhen, dass die Gefahren möglichst umfassend identifiziert werden. Insofern sind sie ein Instrument, das kreative Denkprozesse anstößt, dementsprechend bezeichnen McDermid und Pumfrey [MP94] HAZOP als ein "system of imaginative anticipation of hazards".

Für die Analyse von Computersystemen müssen die Leitwörter angepasst werden. Hierzu gibt es zwei grundsätzliche Vorgehensweisen. Chudleigh et al. [CCR95] schlagen vor, die klassischen Leitwörter auch für die Analyse von Computersystemen beizubehalten und lediglich neu zu interpretieren. "Kein" bedeutet dann beispielsweise, dass Daten erwartet, aber über den Datenkanal nicht gesendet werden. Hinzu kommen lediglich Leitwörter für Zeitaspekte: „zu früh“, „zu spät“, „vorher“, „nachher“. Diese Wörter spielen insbesondere für Realzeitsysteme [Kop97] eine wichtige Rolle, da hier Zeitrestriktionen unbedingt eingehalten werden müssen, und z.B. ein zu spät gesendetes Datum eventuell nicht mehr verarbeitet wird. McDermid und Pumfrey [MP94] gehen einen anderen Weg und schlagen ganz neue Leitwörter zur Analyse von Computersystemen vor. Sie orientieren sich an den

Fehlerklassen von Ezhilchelvan und Shrivastava [ES89] und unterscheiden drei Klassen mit jeweils zugehörigen Leitwörtern: Service Provision mit den Leitwörtern "omission“ (Daten werden erwartet, aber nicht geliefert) und "comission“ (Daten werden geliefert, aber nicht erwartet), Service Timing mit den Leitwörtern "early“ (Daten werden zu früh geliefert) und "late" (Daten werden zu spät geliefert) und schließlich Service Value mit den Leitwörtern "coarse“ (Daten sind inkorrekt und der Fehler ist entdeckbar) und "subtle“ (Daten sind inkorrekt und der Fehler ist nicht entdeckbar).

Um den Umfang der beispielhaften Analyse des Bremssystems nicht zu sprengen, werden im folgenden unter bezug auf das Referenzmodell nur die Leitwörter „Kein“ und anschließend "ein Teil von" angewendet. Auf der obersten Designebene des Bremssystems gibt es die Outputs BW1, BW2 und LW. Durch Anwendung von "Kein“ kann der ausgehende Fehler "BS liefert BW1 nicht“ abgeleitet werden. Analog lassen sich die ausgehenden Fehler "BS liefert BW2 nicht“ und "BS liefert LW nicht“ ableiten. Durch Anwendung der Leitwörter auf die Inputs, werden eingehende Fehler abgeleitet. Mittels „Kein“ ergeben sich folgende eingehende Fehler: „AL liegt nicht an“ und „FB liegt nicht an“. Wir nehmen an, dass in der Teamdiskussion bei Anwendung von "ein Teil von" auf die Outputs des Systems die Instabilität der Outputs zur Sprache kommt. Die Experten halten diese Fehler jedoch auf Basis der obersten Designebene für unrealistisch und verwerfen sie wieder. Als Ergebnis von HAZOP liegt nach Analyse der obersten Ebene eine Liste von fehlerhaften System-Inputs und –Outputs (Systemfehlern) vor. Die Ableitung interner Fehler wird nicht unterstützt, weil der Fokus von HAZOP auf den Verbindungen liegt. Nach demselben Muster wird die Analyse für die weiteren Ebenen durchgeführt. Bereits aufgedeckte Fehler werden dabei differenzierter auf die dort vorhandenen Module verteilt. Beispielsweise wird auf der zweiten Ebene aus "BS liefert BW1 nicht" ein ausgehende Fehler des Bremsmoduls: "BM liefert BW1 nicht". Hinzu kommt aufgrund der Verbindung zwischen dem Brems- und dem Leuchtmodul der eingehende Fehler "BW liegt nicht am LM an". Spätestens auf der dritten Designebene ist es nicht mehr sinnvoll, HAZOP anzuwenden. An dieser Stelle liegen bereits Erfahrungen bzgl. der Module vor, so dass das spekulative Vorgehen durch empirische Daten abgelöst werden kann.

Um zu entscheiden, ob die aufgedeckten Abweichungen auftreten können oder nicht, ist es notwendig zu betrachten, wie sich Fehler von Modul zu Modul fortpflanzen. Da solche kausalen Pfade mit HAZOP nicht systematisch analysiert werden können, müssen zu diesem Zweck andere Techniken hinzugezogen werden.

Vor- und Nachteile

HAZOP bietet eine systematische Analyse von Fehlermöglichkeiten, die weitestgehend sicherstellen soll, dass kein Fehler übersehen wird. Die Methode ist so angelegt, dass auch Systeme untersucht werden können, für die bisher noch keine Fehlererfahrungen vorliegen. Die generierten kritischen Fehlermöglichkeiten können von anderen Methoden aufgegriffen und weitergehend analysiert werden. Insofern ist HAZOP eine wichtige Methode, die Input für andere Methoden liefert. Wird HAZOP als einzige Analysemethode verwendet, dann ist sicherlich zu bemängeln, das die Systematik auf die Ableitung der Fehler beschränkt bleibt und die Analyse der Fehlerkonsequenzen und –ursachen nicht unterstützt. Diese müssen aufgrund der Erfahrung der Teammitglieder in der Diskussion ermittelt werden. Neben der Systematik ist also die lebhafte und kreative Interaktion zwischen den Teammitgliedern ein Schlüsselfaktor für den Erfolg des HAZOP Prozesses.

IV. Fehlerbaumanalyse

Die Fehlerbaumanalyse (fault tree analysis - FTA) stammt ursprünglich aus der Luft- und Raumfahrt und Reaktortechnik [Ves81]. Sie ist die meistangewandte Methode zur Sicherheitsanalyse. Mit ihr können auf Basis eines Systemmodells Ursachen für Fehler abgeleitet werden. Dabei wird entlang der Systemzerlegung rückwärts der Weg des Fehlers verfolgt, bis schließlich die Systemelemente erreicht werden, bei denen der Feher entsteht. Es handelt sich also um eine deduktive Methode. Die Analyse kann begonnen werden, sobald die erste Ebene der Designzerlegung vorliegt und relevante Fehler, identifiziert worden sind. Nach jedem weiteren Designschritt wird die Analyse fortgeführt.

Ablauf einer Fehlerbaumanalyse

Bei der FTA wird Schritt für Schritt ein Fehlerbaum aufgebaut. Er besteht aus Knoten und Kanten (Abb. 6[1]). Es gibt zwei Knotenarten: Ereignisknoten zur Repräsentation von Fehlerereignissen und Gatterknoten zur logischen Verknüpfung der Ereignisknoten. Basisereignisse stehen für auslösende Ereignisse bilden die Blätter der Bäume. Bedingungen werden für bestimmte Gattertypen gebraucht, nicht-entwickelte Ereignisse werden nicht weiter analysiert, entweder, weil keine weiteren Informationen vorliegen oder weil sie von geringer Bedeutung sind. Externe Ereignisse, finden außerhalb des Systems statt. Alle anderen Ereignisse werden als innere Ereignisse repräsentiert. Die Knoten werden über Gatter miteinander in Relation gesetzt. Die Nachfolger eines Gatters werden als Input-Knoten, die Vorgänger als Output-Knoten bezeichnet, wobei die Input-Knoten kausal verantwortlich für die Output-Knoten sind. Es sind Gatter für die logischen Operatoren AND, OR und XOR vorgesehen, darüber hinaus gibt es Inhibit- und Priority-And Gatter. Das Inhibit-Gatter ist eine spezielle Art des And-Gatters. Es besitzt genau ein Input-Ereignis und eine Bedingung. Das Output-Ereignis tritt dann ein, wenn vorher das Input-Ereignis eingetreten und zusätzlich die Bedingung erfüllt ist. Das Priority-And-Gatter hat mehrere Inputs, die in einer festen Reihenfolge, von links nach rechts eintreten müssen.

Abb. 6: Syntax der Fehlerbäume

Die FTA kann bereits auf der ersten Designebene begonnen werden, wenn die Inputs und Outputs des Systems festgelegt sind und mit Hilfe einer HAZOP mögliche Systemfehler abgeleitet wurden. Die Systemfehler bilden jeweils die Wurzel eines Fehlerbaums, das bedeutet, dass jeder Systemfehler separat untersucht wird. Die Suche nach Ursachen kann einerseits horizontal auf jeder Ebene und andererseits vertikal über die unterschiedlichen Ebenen durchgeführt werden. Beispielhaft werden im folgenden die Fehlerbäume für die Systemfehler „BS liefert BW1 nicht“ (BW2 wird analog analysiert) und „BS liefert LW nicht“ aufgebaut.

Auf Basis der ersten Designebene können bereits horizontale Fehlerbeziehungen analysiert werden. ¬BS_BW1 kann durch fehlende Inputs (¬FB oder ¬AL) verursacht werden. Zusätzlich kann eine nicht erfolgte Berechnung innerhalb des Bremssystems (¬BR_BS) Ursache sein. Der Fehler ¬BR_BS wurde bei der HAZOP Analyse noch nicht aufgedeckt. Dies liegt daran, dass die Systematik von HAZOP interne Fehler nicht berücksichtigt. Bei der FTA werden sie nach der Heuristik, dass prinzipiell jede Komponente fehlerhaft sein kann, generiert. Um welche Fehler es sich tatsächlich handelt, kann erst auf der untersten Ebene festgestellt werden, wenn Komponenten mit bekannten Fehlermöglichkeiten vorhanden sind.

Für den fehlenden Leuchtwert (¬BS_LW) kommt auf der ersten Ebene nur ein interner Fehler als Ursache in Frage, weil der Zusammenhang zwischen Bremswert und Leuchtwert noch nicht im Systemmodell ersichtlich ist. Die beiden zur ersten Designebene gehörenden Fehlerbäume sind im oberen Teil (a) von Abb. 7 zu sehen.

Im nächsten Designschritt wird für die Sicherheitsanalyse sichtbar, aus welchen Modulen das System besteht. Zwischen dem internen Fehler ¬BR_BS des Systems und denen der Module auf der zweiten Ebene besteht eine vertikale kausale Beziehung. ¬BR_BS wird entweder durch einen Berechnungsfehler des Bremsmoduls (¬BR_BM) oder des Leuchtmoduls (¬BR_LM) verursacht. Die vertikalen Beziehungen können ebenfalls in einem Fehlerbaum dargestellt werden (Abb. 8). Dabei liegt der Fokus nicht auf der Fehlerpropagierung sondern auf den internen Fehlern und deren Konkretisierungsbeziehungen. Ein interner Fehler auf der i-ten Designebene wird konkretisiert, indem auf Basis der Designinformationen eingeschätzt wird, durch welche Kombinationen von internen Fehlern auf der i+1-ten Designebene er entstehen kann.

[1] Im folgenden wird die Syntax der US Nuclear Regulatory Commission [Ves81] beschrieben. Je nach Anwendungskontext werden Variationen dieser Syntax verwendet [DIN81, RAC90].

Die horizontale Fehleranalyse kann auf der zweiten Ebene verfeinert werden. Dabei werden einerseits die Fehler bzgl. der Module konkretisiert und andererseits werden die horizontalen Beziehungen der ersten Ebene aufgebrochen und entsprechend der zerlegten Systemstruktur weitere Propagierungsstationen dazwischengehängt. So können beispielsweise für den nicht gelieferten Leuchtwert (¬LM_LW) die Ursachen ¬BR_LM und ¬BW_LM aufgedeckt werden. Der Bremswert fehlt als eingehender Wert erst dann am Leuchtmodul, wenn weder BW1 noch BW2 geliefert werden, denn das Modul ist in der Lage, zu erkennen, wenn einer der beiden Werte nicht anliegt und greift dann automatisch auf den anderen zu. Fehlende Bremswerte können jeweils entweder durch eine fehlende Berechnung (¬BR_BM) oder nicht anliegenden Inputs (¬FB und ¬AL) verursacht werden. Die zur zweiten Designebene gehörenden Fehlerbäume sind in der Mitte (b) von Abb. 7 zu sehen. Die für ¬BM_BW1 und ¬BM_BW1 entwickelten Fehlerbäume links in der Abb. sind identisch mit den im rechten Baum entwickelten Teilbäumen für diese Werte.

Da das Bremsmodul auf der letzten Ebene in zwei Module zerlegt wird, kann der interne Fehler des Bremsmoduls im Fehlerbaum für die vertikalen Beziehungen weiter konkretisiert werden (vgl. Abb. 8). Diese Konkretisierung spiegelt sich auch in den Fehlerbäumen für die horizontalen Beziehungen wider, indem ¬BR_BM als Ursache für nicht gelieferte Bremswerte durch ¬BR_BM1 bzw. ¬BR_BM2 ausgetauscht wurde. Ansonsten sind die horizontalen Fehlerbäume der letzten Ebene mit denen der zweiten identisch.

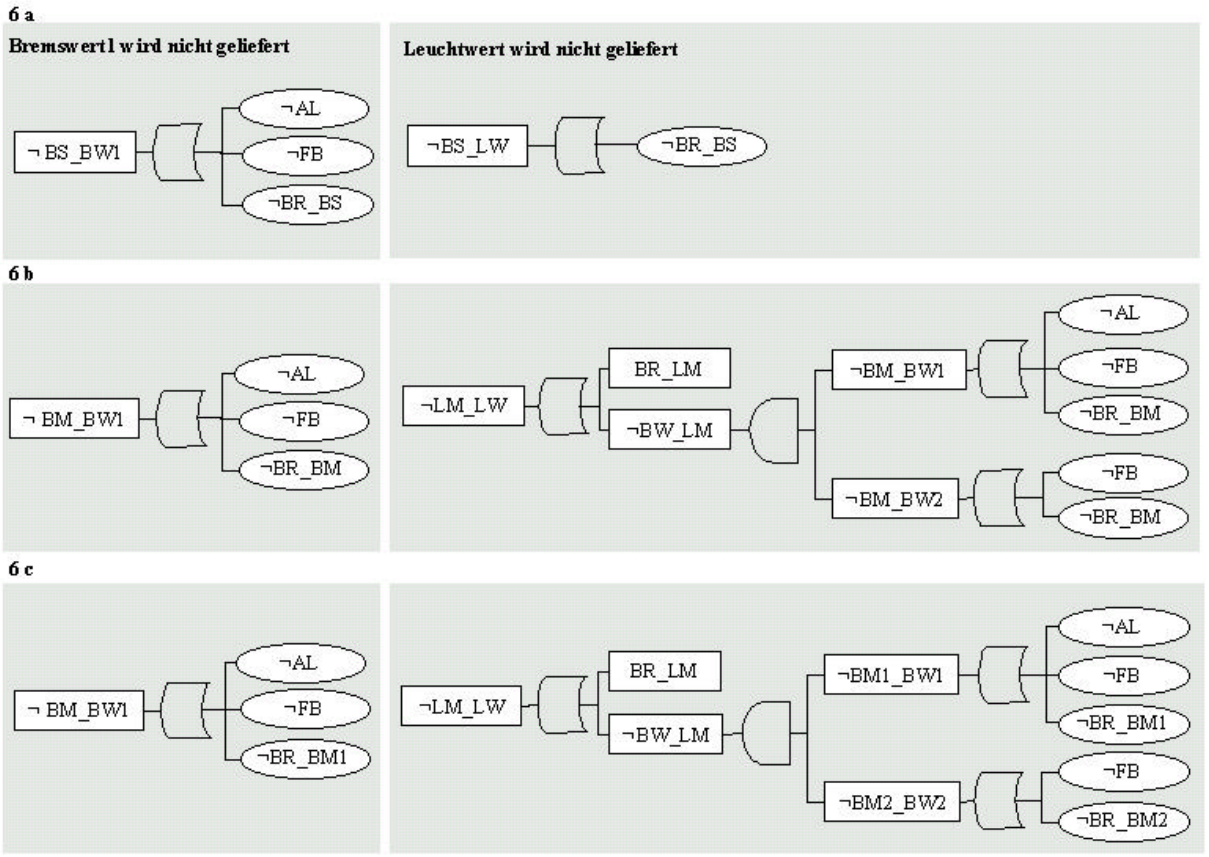

Abb. 7: Fehlerbäume für die horizontalen Fehlerbeziehungen

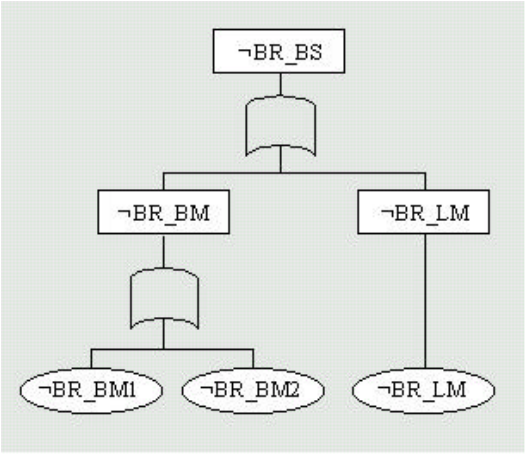

Abb. 8: Fehlerbäume für die vertikalen Fehlerbeziehungen

Leveson und Harvey haben mit der Methode SoftwareFTA (SFTA) [LH83] die klassische FTA speziell für die Analyse von Software angepasst. Die SFTA wird direkt auf dem Programmcode durchgeführt. Die Designstruktur, an der sich die klassische FTA orientiert, wird ersetzt durch elementare Strukturen einer Programmiersprache. Für jede dieser Strukturen wird eine allgemeine Baumstruktur definiert, die bei der Analyse dem Programmcode entsprechend instantiiert und miteinander kombiniert werden. Als Fehlerereignisse werden Abweichungen der Variablen vom vorgegebenen Wertebereich betrachtet. In [LH83] werden elementare Baumstrukturen für If-Then-Else-Anweisungen, Zuweisungen, Funktionsaufrufe und While-Schleifen aufgeführt. Fenelon und McDermid [FM93] kritisieren an der SFTA, dass es sich nicht um einen top-down Ansatz handelt. Die SFTA wird nur auf der Ebene des Programmcodes durchgeführt, während die FTA wie oben gezeigt bereits auf den obersten Designebenen begonnen werden kann und dann entsprechend der Verfeinerung des Designs fortgeführt wird. Die SFTA vernachlässig also die vertikalen Fehlerbeziehungen völlig. Aus diesen Gründen ist der Ansatz wenig intuitiv für Ingenieure, die den traditionellen Ansatz gewohnt sind. Außerdem wird der gesamte Code untersucht, statt eine Fokussierung auf kritische Softwareregionen vorzunehmen: "Kann es irgendwo passieren, dass X einen Wert größer 100 einnimmt?" statt "Wenn in diesem Modul der Wert von X 100 überschreitet, wird ein kritischer Systemfehler erzeugt." Fenelon und McDermid [FM93] haben gezeigt, dass eine FTA für Software genauso durchgeführt werden kann, wie für physikalische Systeme, wenn, wie oben für das Bremssystem durchgeführt, statt der physikalischen Designstruktur die Modulzerlegung zugrunde gelegt wird.

Nachdem die Fehlerbäume aufgebaut wurden, kann sowohl eine qualitative als auch eine quantitative Auswertung vorgenommen werden. Bei der qualitativen Analyse werden durch ein Reduktionsverfahren die Mengen von Basisereignissen bestimmt, die hinreichend sind, um das Ereignis in der Wurzel, den Systemfehler, zu verursachen. Diese Mengen werden Cutsets genannt. Wenn alle Ereignisse in einem Cutset eintreten, dann tritt auch der Systemfehler ein, jedes einzelne Ereignis ist dabei notwendig. Cutsets liefern wertvolle Informationen für das Design, denn es ist nicht zwingend notwendig, die Fehlerwahrscheinlichkeit aller Basisereignisse durch Designmaßnahmen zu reduzieren. Es genügt, wenn aus jedem Minimal Cutset die Fehlerwahrscheinlichkeit genau eines Ereignisses reduziert wird. Für die Bestimmung der Minimalen Cutsets wird der Fehlerbaum entsprechend der Gattersymbole in eine äquivalente logische Formel übersetzt. Unter Anwendung von Reduktionsregeln kann die resultierende Formel vereinfacht werden. Wenn keine weitere Reduktion mehr möglich ist, liegt eine minimale Form vor. Man erhält eine Formel in disjunktiver Normalform. Für das Bremssystem kann diese Reduktion einmal für die horizontalen und zusätzlich für die vertikalen Beziehungen durchgeführt werden. Die horizontalen Beziehungen in den beiden Fehlerbäumen, die sich aus der untersten Designebene ableiten, sind am konkretesten und zeigen deshalb exakte Angriffspunkte zur Reduzierung der Fehlerwahrscheinlichkeit. Die entsprechenden logischen Formeln haben folgende Gestalt:

$$\neg BR\_BM1 \vee \neg FB \vee \neg AL \Rightarrow \neg BM1\_BW1$$
$$((\neg BR\_BM1 \vee \neg FB \vee \neg AL) \wedge (\neg BR\_BM2 \vee \neg FB)) \vee \neg BR\_LM \Rightarrow \neg LM\_LW$$

Der erste Formel ist bereits minimal, somit gibt es für den fehlenden Bremswert1 die drei minimalen Cutsets {¬BR_BM1}, {¬FB} und {¬AL}. Die zweite Formel kann reduziert werden zu:

$$(\neg BR\_BM1 \wedge \neg BR\_BM2) \vee \neg FB \vee (\neg AL \wedge \neg BR\_BM2) \vee \neg BR\_LM \Rightarrow \neg LM\_LW$$

Darin sind vier minimale Cutsets für das fehlende Leuchtsignal enthalten: {¬BR_BM1, ¬BR_BM2}, {¬FB}, {¬AL, ¬BR_BM2} und {¬BR_LM} gibt.

Für die vertikalen Beziehungen gibt es nur einen Fehlerbaum, dieser hat die Minimal-Cutset-Form ¬BR_BM1 ∨ ¬BR_BM2 ∨ ¬BR_LM, so dass drei ein-elementige Cutsets existieren: {¬BR_BM1}, {¬BR_BM2} und {¬BR_LM}.

Bei der quantitativen Analyse werden ausgehend von der Fehlerrate der Basisereignisse die Unzuverlässigkeit und Nichtverfügbarkeit des Systems berechnet. Voraussetzung ist bei diesem Vorgehen, dass die Basisereignisse statistisch voneinander unabhängig sind. Die quantitativen Annahmen sind wichtige Information für das Design, da man durch geeignete Maßnahmen versuchen wird, die Unzuverlässigkeit und Nichtverfügbarkeit unterhalb der Akzeptanzgrenzen zu halten. Solche Akzeptanzgrenzen werden z.B. in der Luftfahrt von der Luftfahrtbehörde vorgegeben [XX]. In der Automobilbranche gibt es derzeit noch keine behördlich vorgeschriebenen Grenzen.

Bei Softwarefehlern wird häufig auf quantitative Angaben verzichtet, da diese als deterministisch betrachtet werden - entweder ist in dem Programm ein Bug oder nicht. Durchaus üblich ist es aber dennoch, Fehlerwahrscheinlichkeiten auf Basis von Testergebnissen zu berechnen: "In diesem Programmteil wurde in *n* Testdurchläufen, mit 80% der möglichen Variablenbelegungen kein Fehler gefunden, deshalb ist es unwahrscheinlich, dass hier ein Fehler vorliegt."

Vorteile und Nachteile

Mit Hilfe der FTA können auf Basis des Systemmodells systematisch Ursachen für Fehler abgeleitet werden. Die Menge der möglichen Fehler muss allerdings bereits vorher vorliegen. Zu diesem Zweck kann, wie oben gezeigt wurde, HAZOP angewendet werden. Mi Hilfe der FTA werden in Form einer Rückwärtsanalyse multiple Ursachen abgeleitet, es ist allerdings nicht möglich, für einen Fehler multiple Konsequenzen aufzudecken. Hierfür ist eine Vorwärtsanalyse, wie sie mit der FMEA im nächsten Abschnitt durchgeführt wird, notwendig.

Ein weiterer Nachteil der FTA ist, dass keine zeitlichen Aspekte berücksichtigt werden können, z.B. dass es zu einem Fehler kommt, wenn ein Ereignis B fünf Sekunden nach einem Ereignis A eintritt. Dies ist ein Problem, wenn z.B. sowohl A als auch B für die Funktionalität des Systems notwendig sind. In einem solchen Fall ist es nicht angebracht, A oder B zu eliminieren. Es muss vielmehr dafür gesorgt werden, dass B frühestens fünf Sekunden nach A eintritt. Solche Aussagen können nicht aus einem Fehlerbaum heraus generiert werden.

V. Fehler-Modus und -Effekt Analyse

Die Fehler-Modus und –Effekt Analyse (failure mode and effect analysis – FMEA) wurde Mitte der sechziger Jahre in den USA von der NASA für das Apollo Projekt entwickelt und insbesondere von der Automobilindustrie übernommen [Sch93]. Im Gegensatz zur FTA handelt es sich um eine induktive Methode, bei der ausgehend von Systemelementen, deren Fehlermodi bekannt sind, für jeden Fehlermodus die Auswirkungen auf das Gesamtsystem abgeleitet werden. Bekannte Fehlermodi liegen z.B. für zugelieferte Systemelemente vor, wobei sie vom Zulieferer angegeben werden, oder von Elementen, die bereits in anderen Systemen eingesetzt werden, zu denen also „in-service“ Erfahrungen vorliegen.

Ablauf einer Fehler-Modus und –Effekt Analyse

Die klassische FMEA [Sch93] wird auf der Komponentenebene durchgeführt. Zu den Komponenten liegen meist gesicherte Fehlerdaten vor, insbesondere wenn es sich um Standardkomponenten handelt. Die Hersteller dieser Komponenten führen intensive Tests durch und werten Fehlerdaten aus dem realen Einsatz aus, soweit vorhanden. Begonnen werden kann, sobald ein vollständiger Systementwurf vorliegt. Für jede Komponente wird ein FMEA-Formular (Abb. 9) ausgefüllt, das pro Fehlermodus eine Zeile enthält. Fehlermodi für einen elektrischen Motor sind beispielsweise: „Motor startet nicht“, „Motor dreht zu schnell“, „Motor dreht zu langsam“. Die innere Struktur dieser Komponenten wird in der FMEA nicht weiter untersucht. Von Interesse sind vielmehr die

Auswirkungen der Fehlermodi auf die Funktionsweise des Gesamtsystems. Diese werden im FMEA-Formular in die Spalte „Konsequenzen“ eingetragen. Zu diesem Zweck müssen die Fehlermodi als ausgehende Fehler („Motor liefert keine Drehung“, „Motor liefert eine zu schnelle Drehung“, „Motor liefert eine zu langsame Drehung“) interpretiert und ihre horizontalen Beziehungen zu anderen Komponenten bis zu dem Punkt, an dem sich der Fehlermode als ausgehender Fehler an der Systemoberfläche zeigt, untersucht werden. Die klassische FMEA bietet keine Unterstützung bei der Analyse oder Darstellung der Konsequenzen.

Failure Modes and Effects Analysis (FMEA)					
System: Subsystem: Author:		FMEA Description:		Date: Sheet of Rev:	
PART NUMBER	PART TYPE	FAILURE MODE	FAILURE EFFECT	DETECTION METHOD	COMMENTS

Abb. 9: FMEA-Formular

Die FMEA diente ursprünglich der Sicherheitsanalyse von Hardware-Komponenten. Einer Übertragung auf die Softwareanalyse steht folgendes Zitat kritisch gegenüber: „It is not practicable to perform an FMECA on Software since software ‘components‘ do not fail“ [Con91]. Software unterschiedet sich von Hardware, denn Software führt genau die Berechnungen aus, die dem Programmcode entsprechen. In Hardware-Komponenten können durchaus spontane Fehler auftreten, z. B. aufgrund eines unvorhergesehenen Materialfehlers. Also, obwohl die Hardware richtig realisiert wurde, können interne Fehler auftreten. Dies ist bei Software nicht möglich. Dennoch gibt es Fehlermöglichkeiten, die sowohl für Hardware als auch für Software gelten. Interne Fehler können auftreten, wenn die Realisierung einer Komponente z.B. aufgrund von Programmierfehlern von den Anforderungen abweicht. Diese Fehler entsprechen fehlerhaften Hardwarekonstruktionen. Da auf den obersten Designebene noch nicht bekannt ist, wie die Module programmiert werden, müssen potentielle Programmierfehler berücksichtigt werden. Sie liefern dem Programmierer wertvolle Hinweise, worauf er bei der Implementierung besonders achten muss.

Da die klassische FMEA erst auf der untersten Designebene beginnt, ist sie anders als die FTA nicht geeignet, den Designprozess zu begleiten. Dieses Manko wird durch die System-FMEA, die im Auftrag des Vereins Deutscher Automobilhersteller (VDA) [VDA96] entwickelt wurde, beseitigt. Die System-FMEA bietet eine methodische Verankerung der FMEA in der hierarchischen Systemstruktur. Sie wird in fünf Schritten durchgeführt. Im ersten Schritt wird das System in seine Elemente zerlegt und entsprechend der Systemstruktur hierarchisch angeordnet. Wenn die Sicherheitsanalyse den Designprozess begleitet, kann diese Information aus dem Design entnommen werden. Jedes Systemelement hat unterschiedliche Funktionen bzw. Aufgaben, diese werden im zweiten Schritt bestimmt und den einzelnen Systemelementen zugeordnet. Auch diese Information gehört originär zum Designprozess und sollte von dort übernommen werden können. Es wird unterschieden zwischen eingehenden, internen und ausgehenden Funktionen. Das in diesem Text verwendete Fehlermodell (Abb. 4) ist hieran angelehnt. Um das Zusammenwirken von Funktionen zu verdeutlichen, wird ein Funktionsbaum erstellt. Interne Funktionen, die in der Summe eine interne Funktion des übergeordneten Systemelementes ausmachen, werden über ein logisches UND mit der übergeordneten Funktion verknüpft. Falls dieser Funktionsbaum im Designprozess nicht erstellt wird, muss er auf Basis der vorhandenen Designinformationen für die Sicherheitsanalyse rekonstruiert werden. Er ist eine wichtige Voraussetzung für die folgende Analyse der Fehlerpropagierung. Der Funktionsbaum für das Bremssystem ist in Abb. 10 dargestellt. Im ursprünglichen Funktionsbaum der VDA werden nur vertikale Funktionsbeziehungen dargestellt, zum besseren Vergleich der Methoden ist diese Darstellung in Abb. 10 um horizontale Funktionsbeziehungen erweitert.

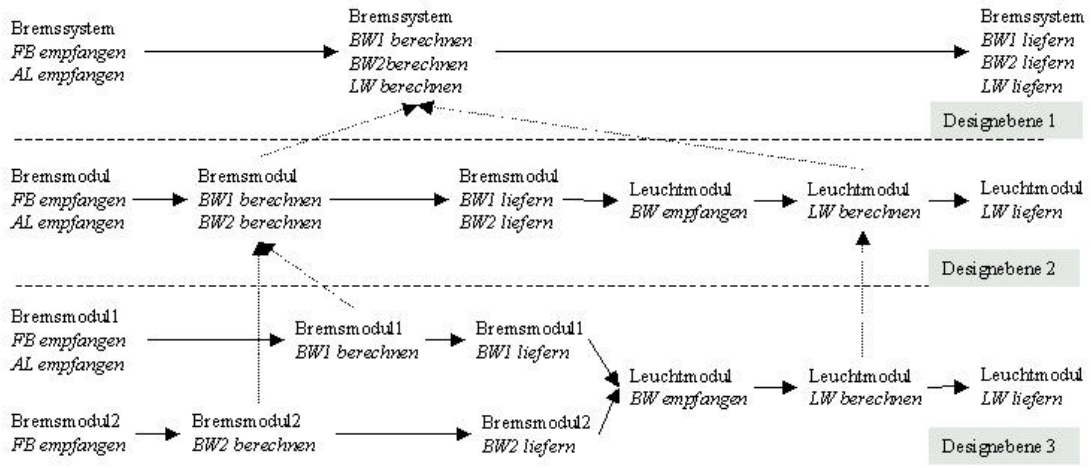

Abb. 10: Funktionsstruktur des Bremssystems

Im dritten Schritt werden mögliche Fehlfunktionen untersucht. Diese werden für jedes Systemelement aus den Funktionen abgeleitet. Im einfachsten Fall sind dies Negationen oder Einschränkungen der Funktionen. Auf Basis der Funktionsstruktur und der Fehlfunktionen kann eine Fehlfunktionsstruktur erstellt werden, die zeigt, wie sich Fehler im System fortpflanzen können. Die Fehlfunktionsstruktur zeigt somit die horizontalen und vertikalen Fehlerbeziehungen. Sie kann nach jedem Designschritt durchgeführt werden. Bei der FMEA handelt es sich um eine Vorwärtsanalyse, deshalb beginnt man anders als bei der FTA nicht mit den Fehlerkonsequenzen, sondern mit den Fehlerursachen. Fehlerursachen können eingehende und interne Fehler sein. Im folgenden werden für alle eingehenden und internen Funktionen des Bremssystems mögliche Fehler durch Negation abgeleitet und hinsichtlich ihrer Konsequenzen analysiert.

Auf der ersten Designebene lassen sich aus den eingehenden Funktionen die Fehlfunktionen „AL wird nicht empfangen“ bzw. ¬AL und "FB wird nicht empfangen“ bzw. ¬FB ableiten. Beide Fehler werden jetzt jeder für sich bzgl. ihrer Konsequenzen analysiert. Der erste Fehler führt dazu, dass BW1 nicht berechnet (¬BR_BS_BW1) und in der Folge nicht geliefert wird (¬BS_BW1). Der zweite Fehler führt dazu, dass sowohl BW1 als auch BW2 nicht berechnet und in der Folge nicht geliefert werden (¬BS_BW1 und ¬BS_BW2). Unter Bezug auf das in diesem Text verwendete Fehlermodell (Abb. 4) wird bei der Fehlerpropagierung nur betrachtet, welche Fehler in ein Modul ein- und als Konsequenz ausgehen, ohne explizit zu benennen, welche internen Funktionen von den eingehenden Fehlern betroffen sind. Unter dieser Prämisse wurde die Fehlfunktionsstruktur in Abb. 11 aufgebaut. Neben den eingehenden Fehlern werden in der Fehlfunktionsstruktur auch interne Fehler als mögliche Ursachen berücksichtigt. Da bei der System-FMEA die Fehler aus den Funktionen abgeleitet werden, wird bzgl. der internen Fehler auf der ersten Designebene genauer differenziert als bei der FTA, bei der die internen Fehler auf Basis der vorhandenen Systemelemente abgeleitet werden. Bei der FMEA werden drei, bei der FTA nur ein interner Fehler abgeleitet: BW1 bzw. BW2 werden nicht berechnet (¬BR_BS_BW1, ¬BR_BS_BW2) und LW wird nicht berechnet (¬BR_BS_LW). Bei der FTA wurde allgemeiner nur die fehlende Berechnung betrachtet (¬BR_BS).

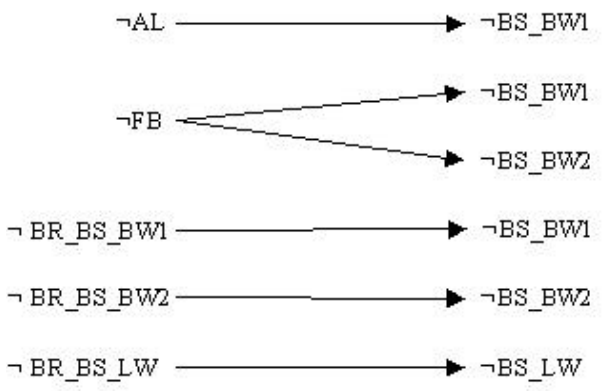

Abb. 11: Fehlfunktionsstruktur für die erste Designebene des Bremssystems

Auf der zweiten Designebene werden die Funktionen auf die neuen Module verteilt. Darüber hinaus können neue Funktionen hinzukommen oder bestehende differenziert werden. Beim Bremssystem kommt auf der zweiten Ebene die Funktion "BW empfangen" für das Leuchtmodul hinzu, da auf dieser Ebene entschieden wird, dass die Bremswerte nicht nur nach außen, sondern auch an das Leuchtmodul geschickt werden, wo dann proportional dazu der Leuchtwert berechnet wird. Es reicht aus, wenn einer der beiden Bremswerte am Leuchtmodul anliegt. Die übrigen Funktionen bleiben gleich und werden lediglich auf die neuen Module verteilt (Abb. 10). In der abgeleiteten Fehlfunktionsstruktur (Abb. 12) kommt der eingehende Fehler des Leuchtmoduls "BW wird nicht empfangen" bzw. ¬BW_LM mit der Konsequenz, dass kein Leuchtwert geliefert wird, hinzu. Ansonsten werden die bereits bekannten Fehlfunktionen entsprechend der Funktionszuordnung den neuen Modulen zugeteilt.

Der Zusammenhang zwischen den ausgehenden Fehlern des Bremsmoduls und dem eingehenden Fehler des Leuchtmoduls wird bei der FMEA nicht aufgedeckt, weil jeder Fehler isoliert von den anderen analysiert wird. Isoliert haben jedoch die fehlenden Berechnungen keine Auswirkungen auf das Leuchtmodul, denn wenn der eine Bremswert nicht geliefert wird, kann dieser Fehler immer noch durch den anderen ausgeglichen werden. Erst die UND-Verknüpfung der beiden Fehler stellt tatsächlich ein Problem dar. UND-Verknüpfungen von Fehlermodi werden allerdings bei der Vorwärtsanalyse der FMEA nicht berücksichtigt. Da vorher eine FTA, die auch UND-Verknüpfungen berücksichtigt, durchgeführt wurde, ist dies nicht tragisch. Hier wird wieder deutlich, dass es notwendig ist, sowohl vorwärts als auch rückwärtsgerichtete Techniken anzuwenden.

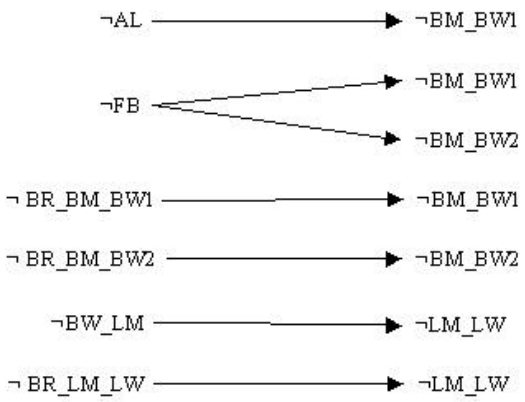

Abb. 12: Fehlfunktionsstruktur für die zweite Designebene des Bremssystems

Auf Basis der zweiten Designebene kann bereits mit der Vorwärtsanalyse der vertikalen Fehlerbeziehungen begonnen werden (Abb. 14). Dabei wird mit einem Fehlermodus einer unteren Designebene gestartet und untersucht, welche Fehlermodi auf den höheren Ebenen dadurch verursacht werden, bis auf der obersten Designebene Fehler des Systems abgeleitet werden. Die internen Fehler der beiden Bremsmodule bewirken jeweils, dass auf der ersten Ebene die entsprechenden Bremswerte des Gesamtsystems nicht berechnet werden. Die fehlende Berechnung des Leuchtwertes innerhalb des Leuchtmoduls bewirkt, dass auf der ersten Ebene die Berechnung des Leuchtwertes innerhalb des Gesamtsystems ausbleibt.

Basierend auf der dritten Designebene kann die Fehlfunktionsstruktur in Abb. 13 generiert werden. Die Fehler, die sich auf BW1 beziehen, werden dem Bremsmodul1 und die, die sich auf BW2 beziehen dem Bremsmodul2 zugeordnet. Zusätzlich kommt die Instabilität der Bremswerte als neuer Fehlermodus hinzu. Er wird erst an dieser Stelle aufgedeckt, weil auf der dritten Designebene entschieden wird, vorhandene Berechnungsmodule in diesem Design wiederzuverwenden. Diese Module wurden bereits in anderen Systemen eingesetzt und die Instabilität dabei beobachtet. Ein instabiler Bremswert pflanzt sich zum Leuchtmodul fort und verursacht einen instabilen Leuchtwert. Alle drei Ausgänge des Systems können somit instabile Werte liefern. Bzgl. der Funktionalität des Leuchtmoduls ist zu diesem Zeitpunkt im Design noch nicht berücksichtigt worden, dass bei Instabilität des einen der andere Bremswert als Basis für den Leuchtwert herangezogen werden soll, insofern kann sowohl die Instabilität von BW1 als auch die von BW2 zu einem instabilen Leuchtwert führen. Die neu aufgedeckten ausgehenden Fehler müssen anschließend mit einer FTA daraufhin untersucht

werden, ob es weitere Ursachen insbesondere UND-verknüpfte Ursachen gibt. Dies ist beim Bremssystem jedoch nicht der Fall.

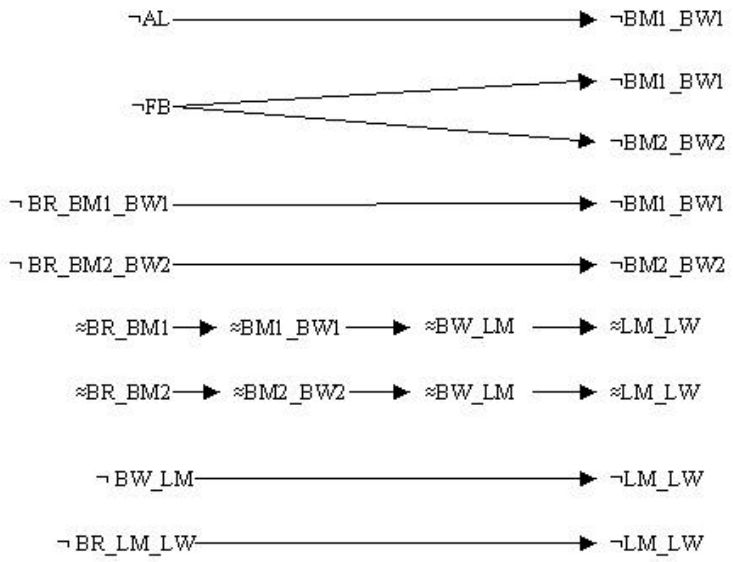

Abb. 13: Fehlfunktionsstruktur für die dritte Designebene des Bremssystems

Bei der Analyse der vertikalen Beziehungen auf Basis der dritten Designebene werden aufgrund des neuen Fehlermodus zusätzliche Systemfehler abgeleitet. Eine instabile Berechnung der Bremswerte innerhalb der einzelnen Bremsmodule bedingt eine instabile Berechnung innerhalb des Bremsmoduls auf der zweiten Ebene, was schließlich zur Instabilität innerhalb des Gesamtsystems führt. Der neu aufgedeckte Fehler des Systems muss auch hier anschließend mit einer FTA untersucht werden, um festzustellen, ob er durch weitere Komponentenfehler verursacht werden kann.

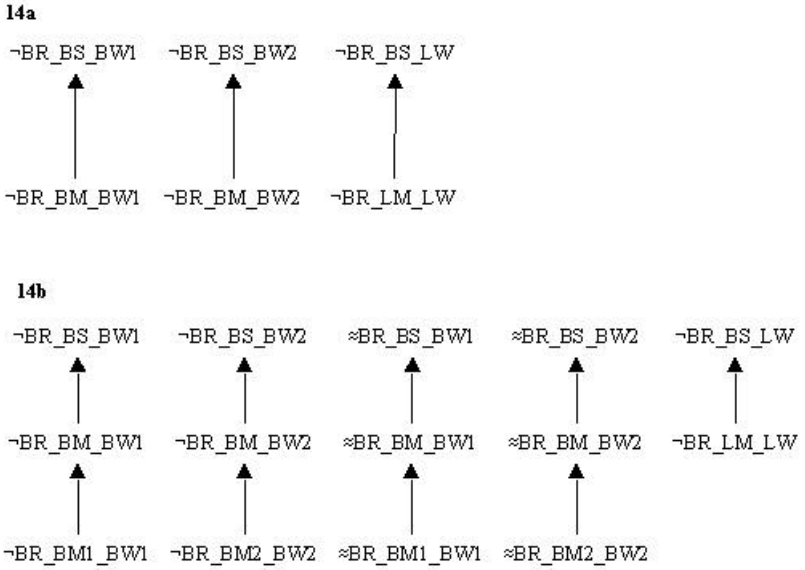

Abb. 14: Fehlfunktionsstruktur für die vertikalen Fehlerbeziehungen des Bremssystems

Nach dem Aufbau der Fehlerstrukturen sind bereits alle qualitativen Informationen für die klassischen FMEA-Formblätter (Abb. 9) vorhanden. Die erstellten Strukturen werden in die Spalten "mögliche Fehlerfolge", "möglicher Fehler" und "mögliche Fehlerursache" übertragen. Damit ist der dritte Schritt der System-FMEA abgeschlossen.

Im vierten Schritt wird eine quantitative Risikobewertung vorgenommen. Bei der klassischen FMEA wird ursprünglich keine quantitative Risikoanalyse durchgeführt. Allerdings wurde die Methode zur Fehlermode und -effekt und Kritikalität Analyse (failure modes and criticality analysis - FMECA) erweitert [SAE67], bei der eine Risikoprioritätszahl berechnet wird. Durch diese Zahl können die

Komponenten bzgl. ihrer Gefährlichkeit eingeschätzt werden. Berücksichtigt werden die Kritikalität der Konsequenzen, die Häufigkeit des Fehlers und die Wahrscheinlichkeit, mit welcher der Fehler entdeckt wird. Diese drei Aspekte werden für jeden Fehlermodus auf einer Skala von 1 bis 10. Die Risikoprioritätszahl ergibt sich durch Multiplikation der drei Werte. Dieses Vorgehen erfolgt genauso auch bei der System-FMEA.

Im fünften und letzten Schritt der FMEA muss für die Optimierung gesorgt werden. Zu diesem Zweck werden Maßnahmen geplant und Verantwortlichkeiten für die Durchführung bzw. Kontrolle bestimmt. Die Maßnahmen können darauf abzielen, die Fehlerfolgen zu eliminieren bzw. zu lindern, die Verlässlichkeit zu erhöhen, d.h. das Auftreten von Fehlfunktionen zu verhindern, oder effektivere Erkennungsmethoden für die Fehlerursachen vorzusehen.

Vorteile und Nachteile

Voraussetzung für die klassische FMEA ist ein vollständiger Systementwurf, was bedeutet, dass diese Analyse den Designprozess nicht begleitet sondern erst zum Abschluss durchgeführt werden kann. Durch die Erweiterung zur System-FMEA wird dieses Manko behoben. FMEA ist auf der Hardwareebene wohl verstanden. In vielen Industrien bieten Lieferanten von Equipment standardisierte Listen mit potentiellen Fehlermodi für ihre Produkte an. Die Fehlermodi können darüber hinaus aus den Funktionen der Systemelemente abgeleitet werden. Dabei ist allerdings Intuition gefragt. Sinnvoll wäre es, wenn Leitwörter für Funktionen, ähnlich den Leitwörtern für Flüsse bei der HAZOP angewendet werden könnten. Eine umfassende Definition solcher Leitwörter liegt allerdings nicht vor.

Ein klarer Nachteil der FMEA ist, dass im Gegensatz zur FTA UND-Kombinationen von Fehlern nicht betrachtet werden. Weiterhin wird anders als bei der FTA eine blinde Suche durchgeführt. Jeder potentielle Fehlermode muss analysiert werden, aber nicht jeder führt tatsächlich zu einem kritischen Fehler auf der Systemebene. Aufgrund dieser Vorgehensweise ist die FMEA eine sehr sorgfältige aber auch sehr arbeitsintensive Analyse. Ein Vorteil gegenüber der FTA ist, dass multiple Konsequenzen abgeleitet werden können.

VI. Ereignisbaumanalyse

Die Ereignisbaumanalyse (event tree analysis – ETA) entstand aus dem missglückten Versuch, die FTA in einem sehr umfangreichen Projekt anzuwenden [Ras1990]. Aufgrund der extremen Größe des entstehenden Fehlerbaumes wurde die aus der Ökonomie bekannte Entscheidungstheorie genutzt, um das Problem in Teilprobleme zu unterteilen, auf die dann wiederum die FTA angewandt werden konnte. Während sich Fehlerbäume und Fehlfunktionsstrukturen gut dazu eignen, kausale Abhängigkeiten zwischen Ereignissen zu beschreiben, repräsentieren Ereignisbäume die sequentielle Abfolge von Ereignissen.

Ablauf einer Ereignisbaumanalyse

In der ETA werden Beziehungen zwischen Systemfehlern untersucht. Bzgl. des in diesem Text verwendeten Fehlermodells sind dies die ausgehenden Fehler des Systems. Sie bilden die Wurzeln der Fehlerbäume für horizontale Beziehungen, insofern kann die ETA dazu dienen, Zusammenhänge zwischen den Wurzelereignissen von Fehlerbäumen aufzudecken. Man nutzt die ETA in den meisten Fällen, um die Fähigkeiten eines Systems zu untersuchen, Systemfehler mit den vorhandenen Backup-Methoden abzufangen, wobei das Versagen einer Backup-Methode wiederum ein Systemfehler ist. Beim Bremssystem wird ein fehlerhafter BW1 durch BW2 ausgeglichen.

Zum Aufbau eines Ereignisbaums werden die Outputs des Systems zunächst geordnet. Dabei müssen die Outputs, die für einen Fehlerfall vorgesehen sind, hinter den Outputs aufgeführt werden, die für den Normalfall gedacht sind. Diese Reihenfolge bestimmt die Anordnung des Ereignisbaums (vgl. Abb. 15). Es geht im folgenden darum, alle möglichen Kombinationen von korrekten und inkorrekten Outputs zu untersuchen. Jede Kombination steht für ein Szenario, dem eine Wahrscheinlichkeit und eine Kritikalität zugeordnet werden kann. In dem Baum werden die Szenarien durch die Äste repräsentiert. Ein initiales Fehlerereignis bildet die Wurzel, beispielsweise ¬BS_BW1 (vgl. Abb15a). Der Baum wird jetzt von links nach rechts entwickelt. Für jeden Systemoutput gibt es einen Punkt, an dem entsprechend der Fehlermöglichkeiten verzweigt wird. Für den Output BW2

existieren drei Verzweigungsmöglichkeiten: BW2 wird korrekt, instabil bzw. nicht geliefert. Analog für LW. Um das erste Szenario zu erstellen, wird an jedem Verzweigungspunkt nach oben verzweigt, jeder Output nach dem initialen Fehlerereignis ist also korrekt. Dies ist der günstigste Fall, ¬BS_BW1 wird durch BW2 ausgeglichen und auch LW wird korrekt geliefert. Sukzessiv werden alle Verzweigungskombinationen aufgeführt und die entsprechenden Szenarien bzgl. der Kritikalität bewertet. In den Fällen, dass beide Bremswerte nicht geliefert werden, bzw. der erste nicht und der zweite nur instabil, wurde in Abb. 15a nicht weiter unterschieden, ob nun LW fehlerhaft geliefert wird oder nicht, denn zwei fehlerhafte Bremswerte sind mit oder ohne Leuchtsignal sehr kritisch bzw. kritisch und dieser Fall muss weitestgehend verhindert werden.

In Abb. 15 ist unten (b) der Ereignisbaum für das initiale Fehlerereignis ≈BS_BW1 dargestellt. Grundsätzlich muss für jede Fehlermöglichkeit des Outputs, der durch Backup-Methoden abgefangen werden soll, ein separater Ereignisbaum erstellt werden.

Abb. 15: Ereignisbäume für das Bremssystem

Neben der Kritikalität wird auch die Wahrscheinlichkeit jedes Szenarios bestimmt. Voraussetzung ist, dass die Fehlerwahrscheinlichkeiten der einzelnen Fehlerereignisse bekannt sind. Da bei der FTA die Wahrscheinlichkeiten für die Wurzelereignisse berechnet werden, können die Werte von dort übernommen werden. Die Szenariowahrscheinlichkeiten ergibt sich durch Multiplikation der Wahrscheinlichkeiten aller Ereignisse des entsprechenden Astes. Voraussetzung ist wie bei der FTA die statistische Unabhängigkeit der Ereignisse.

Vor- und Nachteile

Mit der Ereignisbaumanalyse kann die Sicherheit des Backupsystems untersucht werden, wobei es möglich ist, multiple Systemfehler zu betrachten. Sie ergänzt die kausalen Analysen und setzt unterschiedliche Fehlerbäume zueinander in Beziehung.

Allerdings werden die Ereignisbäume aufgrund der Kombinationsmöglichkeiten sehr groß und unübersichtlich, vor allem wenn feinere Fehlerabstufungen analysiert werden sollen, z.B. „Wert ist zu klein/groß/wird zu spät geliefert/wird zu früh geliefert“.

VIII. Zusammenfassung

Im vorangehenden Text wurde beschrieben, wie sich HAZOP, FTA, FMEA und ETA ergänzen können. Abb. 16 zeigt eine Gegenüberstellung der Techniken. Mit Hilfe von HAZOP lassen sich für das Gesamtsystem und für jedes Modul ausgehende und eingehende Fehler ableiten. Mit der FTA wird dann eine Rückwärtsanalyse zur Ableitung potentieller Ursachen der ausgehenden Fehler durchgeführt. Während die FMEA dazu dient, in Form einer Vorwärtsanalyse die Konsequenzen eingehender Fehler aufzudecken. Diese dreistufige Analyse kann auf jeder Designebene durchgeführt werden, um horizontale Fehlerbeziehungen abzuleiten. Bei der Analyse vertikaler Fehlerbeziehungen wird mit Hilfe der FTA abgeleitet, welche Systemmodule maßgeblich zu internen Systemfehlern beitragen. Dabei wird nach dem Prinzip vorgegangen, dass prinzipiell jedes Modul fehlerhaft sein kann. Mit Hilfe der FMEA werden interne Fehler von Modulen auf unteren Designebenen differenzierter betrachtet. Für jede Komponente wird eine Fehlerliste entweder aus den Funktionen abgeleitet oder, falls es sich um eine Standardkomponente handelt, vom Hersteller bezogen. Ausgehend hiervon werden die Auswirkungen auf das Gesamtsystem bestimmt. FTA und FMEA untersuchen Propagierungen von Fehlern durch das System. Teilweise sind dies bei beiden Techniken dieselben Pfade, aber aufgrund der unterschiedlichen Analyserichtung können auch unterschiedliche Pfade entdeckt werden. Mit der FTA werden insbesondere solche Pfade aufgedeckt, auf denen mehrere Ursachen UND-verknüpft zusammenwirken. Die FMEA erlaubt hingegen die Analyse von Pfaden, auf denen Fehler mehrere Konsequenzen haben. Für die FTA müssen Fehlerkonsequenzen als Startpunkt für die Analyse bekannt sein, ausgehend hiervon können nicht bekannte Fehlerursachen entdeckt werden. Bei der FMEA hingegen müssen Fehlerursachen als Startpunkt bekannt sein, von denen aus nicht bekannte Fehlerkonsequenzen aufgedeckt werden können. Die ETA rundet die Sicherheitsanalyse ab, indem sie für die oberste Systemebene die Analyse multipler Systemfehler und damit der Robustheit des Backup-Systems ermöglicht.

	HAZOP	**FTA**	**FMEA**	**ETA**
Analyserichtung	keine	kausal rückwärts	kausal vorwärts	zeitlich vorwärts
Was wird aufgedeckt?	eingehende und ausgehende Fehler	Fehlerpropagierungen – multiple Urtsachen	Fehlerpropagierungen – multiple Konsequenzen	Fehlerszenarien – Kombinationen von Systemfehlern
Berücksichtigung interner Fehler	nicht	Undifferenziert nach dem Prinzip: Jede Komponenet kann ausfallen	Differenziert abgeleitet aus den Funktionen, bzw. auf Basis von fehlerlisten vom Hersteller	nicht

Abb. 16: Vergleich HAZOP, FTA, FMEA und ETA

In diesem Text wurden die Fehlermöglichkeiten aus dem Systemmodell abgeleitet, dabei gilt zu beachten, dass dieses Verfahren von den Informationstypen, die im Design repräsentiert sind abhängig ist. In dem Bremssystembeispiel wurden physikalische Aspekte außer Acht gelassen, z.B. ob die Daten über einen Datenbus oder direkte Verbindungen übertragen werden. Jede Modellierungsmethode legt den Schwerpunkt auf bestimmte Aspekte des Designs. Um die Fehlermöglichkeiten möglichst umfassend aufzudecken, muss die Analyse auf unterschiedlichen Repräsentationen durchgeführt werden, so dass alle Systemaspekte, die zu Fehlern beitragen können, berücksichtigt werden.

LITERATUR

[Bill97] C.E Billings. *Aviation Automation: The Search for a Human-Centered Approach*. Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers, 1997.

[Bis90] P.G. Bishop. Dependability of Critical Computer Systems 3 – Techniques Directory. London, New York: Elsevier Applied Science, 1990.

[BP93] D.J Burns, R.M. Pitblado. A Modified Hazop Methodology For Saferty Critical System, in F. Redmill und T. Anderson (eds.), Directions in safety critical systems: SCS Symposium, Bristol, 1993, Springer Verlag.

[CCR95] M. Chudleigh, J.R. Catmur, F. Redmill (1995). A Guideline for HAZOP Studies on Systems which include a Programmable Electronic System. In Proceedings of *SAFECOMP'95*, Springer-Verlag.

[CISHEC87] *A guide to Hazard and Operability Studies*. The Chemical Industry Safety and Health Council of the Chemical Industries Association Ltd., 1977.

[Con91] P.D.T O'Connor. Practical Reliability Engineering (3rd Edition). J. Wiley, 1991.

[DIN81] Deutsches Institut für Normierung e.V. DIN 25 424 Fehlerbaumanalyse - Methode und Bildzeichen, 1981.

[ES89] P.D. Ezhilchelvan, S.K. Shrivastava. A Classification of faults in systems, University of Newcastle upon Tyne, 1989.

[FMPN94] P. Fenelon, J.A. McDermid, D.J. Pumfrey, M. Nicholson. Towards Integrated Safety Analysis and Design. In *ACM Applied Computing Review*, 1994.

[FH94] P. Fenelon, B. Hebbron. Applying HAZOP to Software Engineering Models. In Proceedings of *SARSS '94*.

[FM93] P. Fenelon, J.A McDermid. An integrated toolset for software safety analysis. In *Journal of Systems and Software*, July, 1993.

[Kop97] H. Kopetz. Real-Time Systems - Design Principles for Distributed Embedded Applications. Kluwer Academic Publishers, 1997.

[Lev95] N.G. Leveson. *Safeware: System Safety and Computers*. Addison-Wesely Publishing Company, 1995.

[LH83] N.G. Leveson, P.R. Harvey. Analysing Software Safety. In IEEE *Transactions on Software Engineering*, SE-9(5), 1983.

[MP94] J.A. McDermid, D.J. Pumfrey. Development of Hazard Analysis to Aid Software Design. In COMPASS, 1994.

[PM99] Y. Papadopoulos, J.A. McDermid. Hierarchically Performed Hazard Origin ans Propagation Studies. In Lecture Notes in Computer Science, 1698, proceedings of SAFECOMP'99, 18th International Conference on Computer Safety, Reliability and Security, Springer Verlag, 1999.

[RAC90] Fault Tree Analysis Application Guide. Reliability Analysis Center, IIT Research Institute, 1990

[Sch93] M. Schubert. *Fehlermöglichkeits- und Einflußanalyse: FMEA - Leitfaden*. Deutsche Gesellschaft für Qualität, Beuth Verlag, 1993.

[SAE67] Design analysis procedure for failure modes, effects and criticality analysis (FMECA). Society of Automotive Engineers, Detroit, USA, 1967 (ARP926).

[VDA96] Verband der Automobilindustrie e. V.. Sicherung der Qualität vor Serieneinsatz - System-FMEA. In der Reihe *Qualitätsmanagement in der Automobilindustrie*, Band 4, Teil II.

[Ves81] W.E. Vesley. *Fault Tree Handbook*. Division of the System Safety Office of Nuclear Reactor Regulation, US Nuclear Regulatory Commission, Washington DC, 1981.