

# *Strukturierte Erstellung von Sicherheitspezifikationen in UML mit Hilfe der FMEA-Methode<sup>1</sup>*

Friedemann Bitsch, Ercüment Canver, Adam Moik  
FORMS-Workshop 12/99, Braunschweig

## **Kurzfassung:**

Die Entwicklung von einem System mit Sicherheitsverantwortung mit Hilfe formaler Methoden erhöht das Vertrauen in seine korrekte Funktion. Bestandteil der formalen Entwicklung ist der formale Nachweis eines funktionalen Systemmodells gegenüber einem Sicherheitsmodell. In einer sicherheitsrelevanten Anwendung gibt es eine ganze Reihe von Anforderungen, die zur Gewährleistung der Sicherheit immer eingehalten werden müssen. Solch ein Sicherheitsmodell wird in den meisten Fällen aus dem Lasten- bzw. Pflichtenheft extrahiert oder sogar ad-hoc aufgestellt.

In diesem Beitrag wird ein Lösungsweg beschrieben, wie für die Aufstellung von sicherheitskritischen Anforderungen systematisch vorgegangen werden kann. Zunächst werden globale Sicherheitsanforderungen zusammengestellt, die aus den rechtlichen Anforderungen und aus den Prozesseigenschaften folgen. Dieses Vorgehen wird auf System-/Subsystemebene fortgeführt, noch ohne Realisierungsaspekte zu berücksichtigen. Danach werden mit den Mitteln der Fehler-Möglichkeiten- und Einfluss-Analyse (FMEA) zusätzliche technische Sicherheitsanforderungen (auch realisierungsabhängige) erstellt und die Sicherheit des Systemkonzepts bewertet.

Um die so hergeleiteten Sicherheitsanforderungen für einen formalen Sicherheitsnachweis verwenden zu können, müssen die Sicherheitsanforderungen in einer formalen Notation dargestellt werden. Die Unified Modeling Language (UML) genießt heutzutage in ingenieurwissenschaftlichen Anwendungen eine hohe Popularität als objektorientierte Beschreibungstechnik und Methode zur Softwareentwicklung. Deshalb erscheint es sinnvoll, die Ergebnisse aus den vorangegangenen Schritten in UML zu integrieren. Dabei können Invarianten in Klassendiagrammen integriert und mit der Object Constraint Language (OCL) dargestellt werden. Anforderungssequenzen können durch die Einbettung von Message Sequence Charts (MSC) in eine Temporallogik spezifiziert werden.

Diese Vorgehensweisen wurden an dem Beispiel der Referenzfallstudie „Eingleisiger Bahnübergang im Funk-Fahr-Betrieb“ des DFG-Schwerpunktprogramms „Integration von Techniken der Softwarespezifikation für ingenieurwissenschaftliche Anwendungen“ erprobt.

## **1. Einleitung**

### **1.1 Stand der Technik für die Vorgehensweise bei der Erstellung sicherheitskritischer Systeme**

Software enthält Fehler. Fehler in technischen Systemen können fatale Folgen haben. Diese Auswirkungen reichen von wirtschaftlichen Nachteilen durch Rückrufaktionen, bis hin zu Unfällen mit hohen Sach- und/oder Personenschäden [Lev95]. Die Entwicklung von Systemen mit Sicherheitsverantwortung im Eisenbahnwesen (Eisenbahnsicherungssysteme) wird heute von einer Sicherheitsanalyse bzw. einem Sicherheitsnachweis begleitet, die bzw. der von einer Aufsichts- oder Zulassungsbehörde abgenommen werden muss.

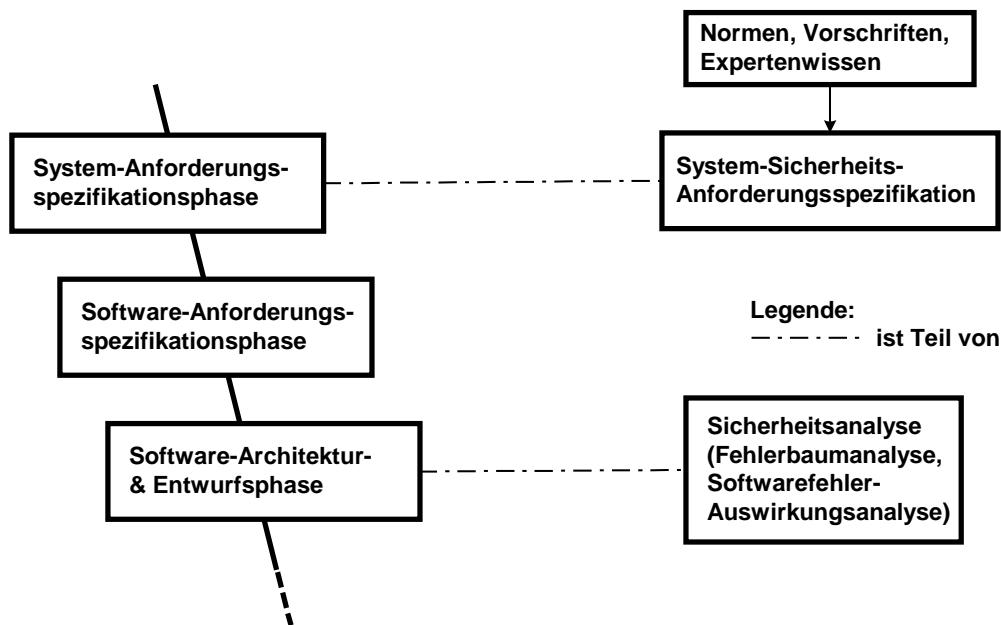
Ein Verkehrsautomatisierungssystem, wie es auch ein Eisenbahnsystem darstellt, besteht aus drei Teilen: Dem technischen Prozess, der Datenverarbeitung inklusive Kommunikation und dem Menschen [LbGö99]. Die Rahmenbedingungen für die Automatisierung eines Systems werden durch den technischen Prozess vorgegeben. Davon ausgehend werden die Anforderungen für die Entwicklung der steuernden bzw. regelnden Hard- und Software abgeleitet. Ein Eisenbahnsicherungssystem ist ein System, das Funktionen mit

---

<sup>1</sup> Die Arbeiten wurden zum Teil von der Deutschen Forschungsgemeinschaft im Rahmen des Schwerpunktprogramms (1064) „Integration von Techniken der Softwarespezifikation für ingenieurwissenschaftliche Anwendungen“ gefördert.

Sicherheitsverantwortung<sup>2</sup> wahrnimmt. Deshalb müssen die sicherheitskritischen Anforderungen für die gesamte Entwicklung dominierend sein. Das bedeutet, dass die Entwicklung der Systemfunktionen von Anfang an unter Berücksichtigung der geforderten Sicherheitsverantwortung des Systems entwickelt werden muss. Auf diese Weise wird die Sicherheit in das System schon von Anfang an hineinentwickelt.

Um dieser Situation Rechnung zu tragen und der Entwicklung von sicherheitskritischen Systemen den notwendigen Rahmen zu geben, wurden in verschiedenen Anwendungsbereichen Vorschläge und Normen erstellt (z.B. EN 50126, prEN 50128 und ENV 50129, DIN V VDE0801 für die Eisenbahntechnik). Die nachfolgend vorgeschlagene Vorgehensweise orientiert sich hauptsächlich an der Norm prEN 50128 „Bahnanwendungen - Software für Eisenbahnsteuerungs- und Überwachungssysteme“.



**Abbildung 1: Ausschnitt aus dem Entwicklungs-Lebenszyklus nach prEN 50128**

Als Ausgangsbasis für die Entwicklung eines Eisenbahnsicherungssystems dient in der Regel ein physikalischer Vorgang oder eine Systemfunktion in einer bestimmten Umgebung. In der Systemanforderungsspezifikationsphase erstellen die Ingenieure, die Experten des Anwendungsgebiets sind, auf der Grundlage von Vorschriften, von Normen und ihres Expertenwissens die Systemsicherheitsanforderungen (Systemsicherheitsanforderungsspezifikation) (Abbildung 1).

In der Software-Architektur- und Entwurfsphase empfiehlt dann die Norm prEN 50128 für die Softwareentwicklung die Fehlerbaumanalyse und die Softwarefehler-Auswirkungsanalyse. Bei den höheren Sicherheitsanforderungsstufen sind diese Analyseverfahren dringend empfohlen.

Zuerst wird die Fehlerbaumanalyse (FTA) durchgeführt. Dabei wird nicht nur die Software betrachtet, sondern die Analyse geschieht auf der Systemebene, um die potentiellen Gefahren, ihre Auswirkungen und auch ihre Ursachen zu erkennen. Bei der Fehlerbaumanalyse wird von einem unerwünschten Ereignis ausgegangen und nach allen Ursachen gesucht, die zu diesem Ereignis führen. Dabei werden auch logische Kombinationen von Ursachen betrachtet.

Die Softwarefehler-Auswirkungsanalyse wird auf der Grundlage der Ergebnisse der FTA durchgeführt. Während mit der FTA die Fehlerursachen bestimmt werden, werden mit der FMEA hauptsächlich die Wirkungen der Fehler analysiert. Dabei werden alle Einzelausfälle der Komponenten oder Betrachtungseinheiten mit ihren Auswirkungen auf das System oder auf die Teilsysteme untersucht. Ziel der Softwarefehler-Auswirkungsanalyse ist die Auffindung von Schwachstellen und ein qualitativer Nachweis der Systemsicherheit durch eine sicherheitstechnische Bewertung des Gesamtsystems. Sie wird erst durchgeführt, wenn Details über die Realisierung des Systems bekannt sind. Sie wird während der gesamten Entwicklungszeit fortgeführt und bei Änderungen wiederholt, um die Sicherheit des entwickelten Systems zu bewerten und ggf. zu verbessern. Die Softwarefehler-Auswirkungsanalyse kann zum Beispiel mit Hilfe der Fehler-Möglichkeiten- und Einfluss-

<sup>2</sup> Funktionen deren Versagen gefährliche Auswirkungen haben kann.

Analyse (FMEA) unter Berücksichtigung der von [VDA96] empfohlenen Vorgehensweise erfolgen. Die FMEA hat in Deutschland in der Entwicklung von Kfz-Steuergeräten heutzutage eine weite Verbreitung gefunden.

Ein erster Schritt bei der Durchführung der FMEA ist die Identifizierung sicherheitskritischer Systemelemente inklusive aller möglichen Fehlerursachen und Auswirkungen. Während einer FMEA werden Funktionen und Fehlfunktionen im System ausschließlich umgangssprachlich beschrieben. Dies hat den Nachteil, dass die Bedeutung seitens des Lesers Interpretationsspielräume zulässt und somit eine Quelle für Fehler sein kann. Vorteilhaft dagegen ist die für komplexe Systeme geeignete analytische und strukturierte Vorgehensweise bei der Fehleranalyse [Lev95]. Die FMEA setzt auf der Systemarchitektur auf und hilft hier Sicherheitslücken bzw. -schwächen zu finden. Anhand dieser Kenntnisse werden Maßnahmen abgeleitet, die die Hardware und/oder Software-Architektur dahingehend beeinflussen sollen, dass das Risiko minimiert wird.

## **1.2 Problemstellung**

Zur Erhöhung des Vertrauens in Eisenbahnsicherungssysteme erscheint es sinnvoll neben der Sicherheitsanalyse auch formale Techniken für den Sicherheitsnachweis einzusetzen. Unter formalen Techniken sollen nachfolgend Techniken und Vorgehensweisen zur Systementwicklung verstanden werden, die auf mathematischen Modellen basieren (z.B. einer Logik) und eine eindeutig definierte Syntax und Semantik besitzen. Sie erlauben den Nachweis von Eigenschaften durch einen mathematischen Beweis.

Beim formalen Sicherheitsnachweis wird der mathematische Beweis geführt, dass eine funktionale Verhaltensbeschreibung die formal spezifizierten Sicherheitsanforderungen in der Sicherheitspezifikation nicht verletzt bzw. immer erfüllt. Um die formale Entwicklung zu unterstützen, wurde z.B. in [Gön95, CGM97] eine methodische Vorgehensweise vorgeschlagen.

Die für einen formalen Sicherheitsnachweis benötigten Sicherheitsanforderungen müssen in der Art und Weise formuliert werden, so dass sie auch für den formalen Sicherheitsnachweis verwertbar sind. In Kapitel 2 dieses Artikels wird aufgezeigt, wie sich dies durch systematische Herleitung der Sicherheitsanforderungen aufgrund der Sicherheitsanalyse erreichen lässt. Die Sicherheitsanalyse wird dabei auf der Basis der Systemarchitektur bzw. des Entwurfs durchgeführt. Deshalb können die Sicherheitsanforderungen in der Weise formuliert werden, dass sie für den formalen Sicherheitsnachweis angewandt werden können. Das hat den Grund darin, dass die Begriffe und Größen, die zur Formulierung der Sicherheitsanforderung verwendet werden, die selben sind wie in der Architektur und im Entwurf. Der Einfachheit halber wird im folgenden in Bezug auf die Sicherheitsanalyse nur die FMEA betrachtet. In Kapitel 3 wird erläutert, wie diese Sicherheitsanforderungen mit Hilfe der Unified Modeling Language (UML) formal so spezifiziert werden können, dass die Darstellungsweise für Signalingenieure im Eisenbahnwesen, für Gutachter und die Zulassungsbehörde in der Praxis akzeptabel und einfach anwendbar ist. Dies wird dadurch erreicht, indem mit UML eine ingenieurgerechte Darstellungsweise verwendet wird, die den Einsatz von Spezifikations- und Entwurfswerkzeugen (UML-Werkzeugen) ermöglicht, die in der Industrie schon angewandt werden. In Kapitel 4 schließen wir unsere Betrachtungen mit einer Zusammenfassung ab.

## **2. Strukturierte Erstellung von Sicherheitsanforderungen mit der FMEA**

### **2.1 Vorgehensweise bei der FMEA**

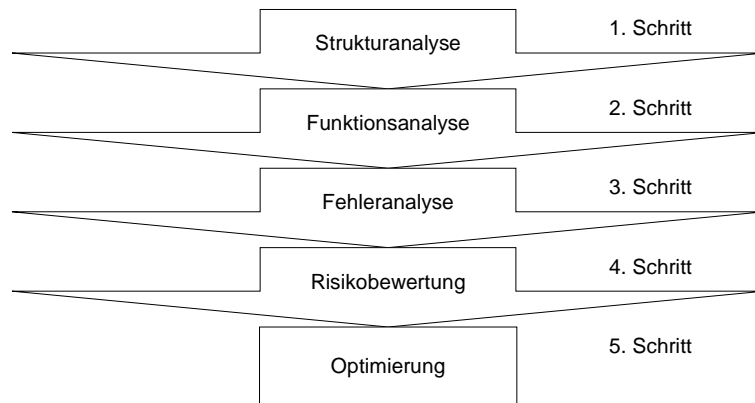
Die FMEA-Methode wurde in den 60er Jahren von der NASA entwickelt und eingeführt. Aus den FMEA-Formblättern lässt sich nicht ohne großen Aufwand auf die Systemstruktur, auf das funktionale und das Fehlverhalten des untersuchten Systems schließen. Dieser Nachteil wird ausgeglichen, indem die FMEA durch Vorgaben zur Vorgehensweise bei der Systemstrukturierung und bei der Dokumentation unterstützt wird (System-FMEA, vgl. [VDA96]). Dadurch werden die FMEA-Formblätter bei Prüfungen nachvollziehbarer.

Die Erstellung einer FMEA wird in fünf Schritte eingeteilt (siehe Abbildung 2):

1. Strukturierung des Systems in Systemelemente (Strukturanalyse)
2. Identifizierung der Funktionen und ihren Zusammenhängen (Funktionsanalyse)
3. Identifizierung der Fehlfunktionen und ihrer Zusammenhänge (Fehleranalyse)
4. Risikobewertung
5. Konzeptoptimierung

Während der Strukturanalyse wird ein System in seine Bestandteile zerlegt. Diese Zerlegung erfolgt hierarchisch in Form einer Baumstruktur. Die Zerlegung kann auf einer beliebigen Hierarchiestufe begonnen und so weit notwendig weitergeführt werden. Das FMEA-Modell wird während der Entwicklung fortgeschrieben und verfeinert.

Das zu entwickelnde System muss bestimmte Anforderungen bzw. Funktionen erfüllen. Um dies zu ermöglichen, müssen die untergeordneten Systemelemente ihrerseits bestimmte Teilfunktionen erfüllen. Der Detaillierungsgrad nimmt mit jeder Hierarchiestufe zu. Die gegenseitigen Abhängigkeiten der Funktionen und Teilfunktionen eines Systems werden mit Hilfe eines Funktionsnetzes dargestellt.



**Abbildung 2: FMEA**

Die Beschreibung der Systemstruktur und der Funktionen ist Voraussetzung für die sich anschließende Fehleranalyse. Während der Fehleranalyse werden zunächst zu allen Funktionen und Teilfunktionen die möglichen Fehlfunktionen bestimmt. Im zweiten Schritt erfolgt die Erstellung eines Fehlernetzes, das die Abhängigkeiten der einzelnen Fehler voneinander darstellt. Damit stellt ein Fehlernetz für ein System mögliche Fehlerfolgen, mögliche Fehler und mögliche Fehlerursachen dar. Diese werden dann in die FMEA-Formblätter eingetragen.

Durch die Angabe einer Risikoprioritätszahl (RPZ) wird das Risiko jeder Fehlerfolge quantitativ bewertet. Die Risikoprioritätszahl wird anhand des mathematischen Produktes dreier Einzelfaktoren ermittelt:

- Bedeutung der Fehlerfolge (B)
- Auftretenswahrscheinlichkeit der Fehlerursache (A)
- Entdeckungswahrscheinlichkeit der aufgetretenen Fehlerursache (E)

Die Einzelfaktoren haben einen Wertebereich von 1 bis 10. Dies hält alle RPZ-Angaben innerhalb der FMEA vergleichbar und erlaubt eine Rangfolge für die Optimierung zu bilden. Während der Optimierung werden alle Systemelemente mit hohen RPZ-Angaben oder Bedeutungen der Fehlerfolgen (B) betrachtet. Ziel ist es hier eine Senkung des Risikos durch die Festlegung zusätzlicher Maßnahmen zu erreichen. Kommen verschiedene alternative Maßnahmen in Frage, so werden die geeignetsten ausgewählt. Gut geeignet sind meistens Maßnahmen, die kostengünstig und kurzfristig durchführbar sind. Das Risiko muss dabei merklich reduziert werden können. Die Risikobewertung und die Optimierung werden in Formblättern dokumentiert.

## **2.2 Gewinnung von für den formalen Nachweis verwendbare Sicherheitsanforderungen mit der FMEA**

Die Sicherheitsanforderungen, die in der Sicherheitsanforderungsspezifikationsphase auf der Basis vorhandener Normen und Vorschriften durch Experten im Eisenbahnwesen erstellt werden, werden globale Sicherheitsanforderungen bzw. globale Sicherheitsspezifikation genannt. Die globalen Sicherheitsanforderungen lassen sich in zwei Klassen einteilen: Funktionsorientierte und nicht funktionsorientierte Sicherheitsanforderungen. Die nicht funktionsorientierten Sicherheitsanforderungen betreffen z.B. die Qualität, die Organisation, die Inbetriebnahme, die Durchführung von Instandsetzungsarbeiten oder den Einsatz des Systems. Hier gibt es keine Möglichkeit einen formalen Beweis zu führen, dass diese Sicherheitsanforderungen in jeder denkbaren Situation eingehalten werden. Lediglich kann zum Teil durch stochastische Betrachtungen und durch Wahrscheinlichkeitsrechnung die Einhaltung solcher Sicherheitsanforderungen belegt werden. Die

funktionsorientierten Sicherheitsanforderungen betreffen direkt die Funktion des Systems oder seiner Bestandteile und können in den meisten Fällen formal spezifiziert und verifiziert werden. Deshalb sind sie grundsätzlich für formale Sicherheitsnachweise verwendbar. Häufig sind sie jedoch auf konkrete Software-Architektur bzw. -Entwurf nicht direkt prüfbar, da sie zu abstrakt sind. Das bedeutet, dass sie oft mit Hilfe physikalischer Größen wie Beschleunigung oder Kraft ausgedrückt werden. So soll zum Beispiel ein Bremswunsch eines Zugführers, ausgedrückt durch die Betätigung des Bremshebels, zu einer Verzögerung des Fahrzeuges führen. Nachfolgend werden ausschließlich funktionsorientierte Sicherheitsanforderungen betrachtet, auch wenn allgemein von Sicherheitsanforderungen gesprochen wird.

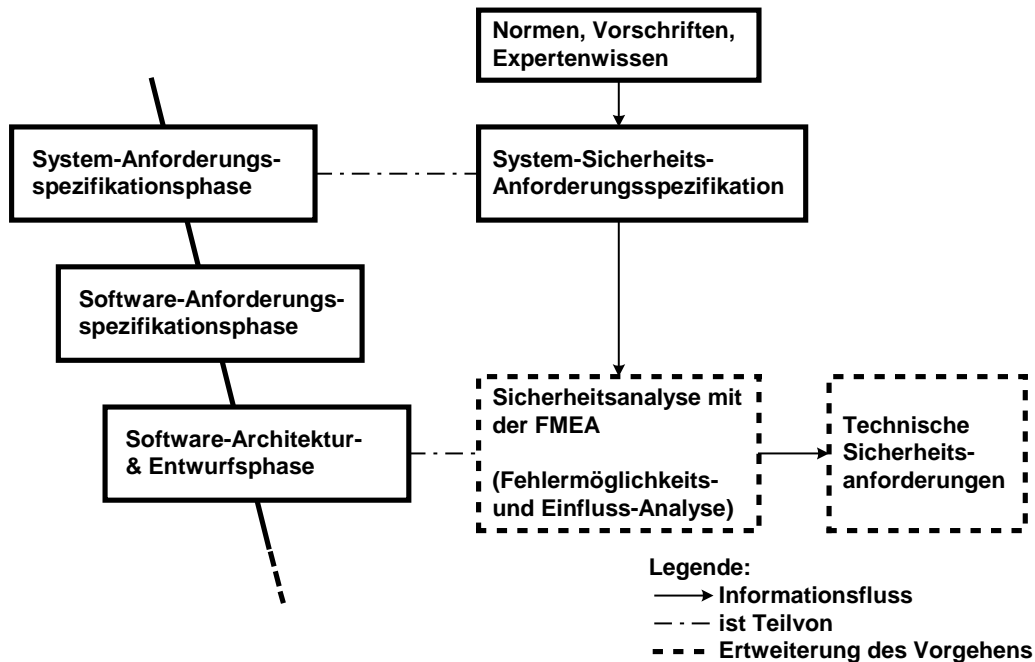


Abbildung 3: Ableitung von Sicherheitspezifikationen aus Sicherheitsanalysen

Die Einhaltung der globalen Sicherheitsanforderungen lässt sich in Bezug auf eine konkrete Realisierung häufig nicht direkt prüfen, da die Steuerungssoftware intern oftmals mit anderen Größen arbeitet (z.B. Druck oder Fluss). Damit müssen die sicherheitskritischen Anforderungen im Kontext der konkreten Realisierung ausgedrückt werden. Um diese Sicherheitsanforderungen von den globalen Sicherheitsanforderungen zu unterscheiden, wird nachfolgend von technischen Sicherheitsanforderungen gesprochen (technische Sicherheitspezifikation). Eine technische Sicherheitsanforderung kann z.B. fordern, dass ein durch einen Sensor gemessener Wert an die Steuerungssoftware weitergeleitet wird oder, dass das Anlegen einer bestimmten Spannung an eine elektrische Radbremse zu einer Verzögerung des gesamten Fahrzeuges führt.

Es erscheint sinnvoll, das Verfahren der FMEA zu erweitern, indem die Ergebnisse der Sicherheitsanalyse, für die Aufstellung einer technischen Sicherheitspezifikation verwendet werden (Abbildung 3).

Für die Gewinnung formaler technischer Sicherheitsanforderungen wurden die ersten fünf Schritte der FMEA übernommen. Damit ist es möglich, die Systemstruktur zu erfassen, sicherheitsrelevante Funktionen und Fehlfunktionen auf allen Systemebenen eines komplexen Systems zu identifizieren, das System auf Risiken zu untersuchen und ggf. zu optimieren. Auf der Grundlage der untersuchten Funktionen, der dazugehörigen Fehlfunktionen und der damit verbundenen Fehlerauswirkungen können dann entsprechende technische Sicherheitsanforderungen erstellt werden.

### 2.3 Beispiel: Ausschnitt aus der Entwicklung eines eingleisigen Bahnübergangs im Funkfahrbetrieb (FFB)

Das vorgestellte Vorgehen wird in diesem Kapitel anhand eines Beispiels erklärt. Als Beispiel für ein Eisenbahnsicherungssystem dient die Referenzfallstudie „Eingleisiger Bahnübergang im Funkfahrbetrieb“ [Jan99] des DFG-Schwerpunktprogramms „Integration von Techniken der Softwarespezifikation für ingenieurwissenschaftliche Anwendungen“. Die funktionale Modellierung des Beispiels mit der Unified

Modeling Language (UML) wird in [ArGa99] beschrieben. In den vorliegenden Betrachtungen erfolgt die Ergänzung dazu durch einen Einblick in die Modellierung der entsprechenden Sicherheitsanforderungen.

Auf der Grundlage von bestehenden Normen und Vorschriften für das Eisenbahnwesen bestimmen Experten globale Sicherheitsanforderungen. Eine globale Sicherheitsanforderung, die mit Hilfe von Expertenwissen erstellt wurde, lautet z.B.: „Ein Bahnübergang darf nur befahren werden, wenn der Zug die Meldung über die ordnungsgemäße Sicherung vom Bahnübergang erhalten hat.“ Im folgenden wird nun beschrieben, wie technische Sicherheitsanforderungen mit Hilfe der FMEA erstellt werden. Die FMEA-Struktur des Gesamtsystems ist in Abbildung 4 dargestellt. Nachfolgend soll nur der Bahnübergang betrachtet werden.

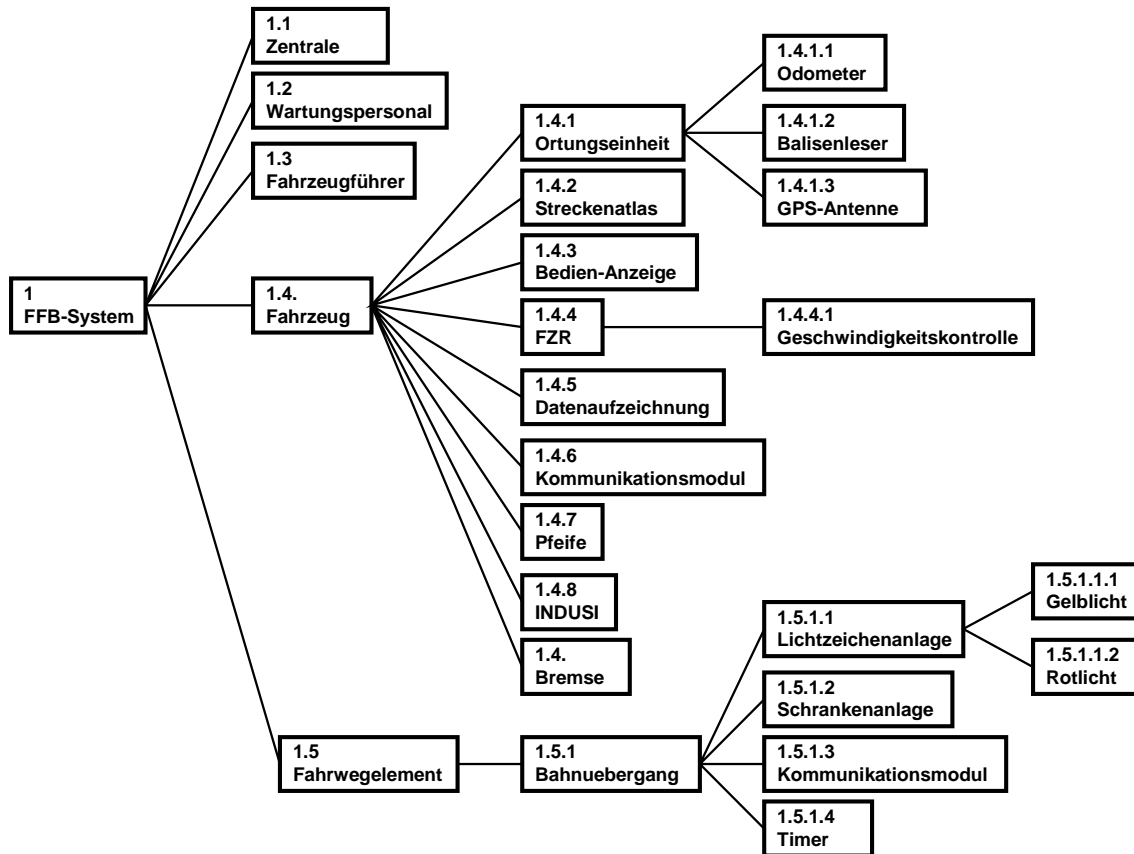
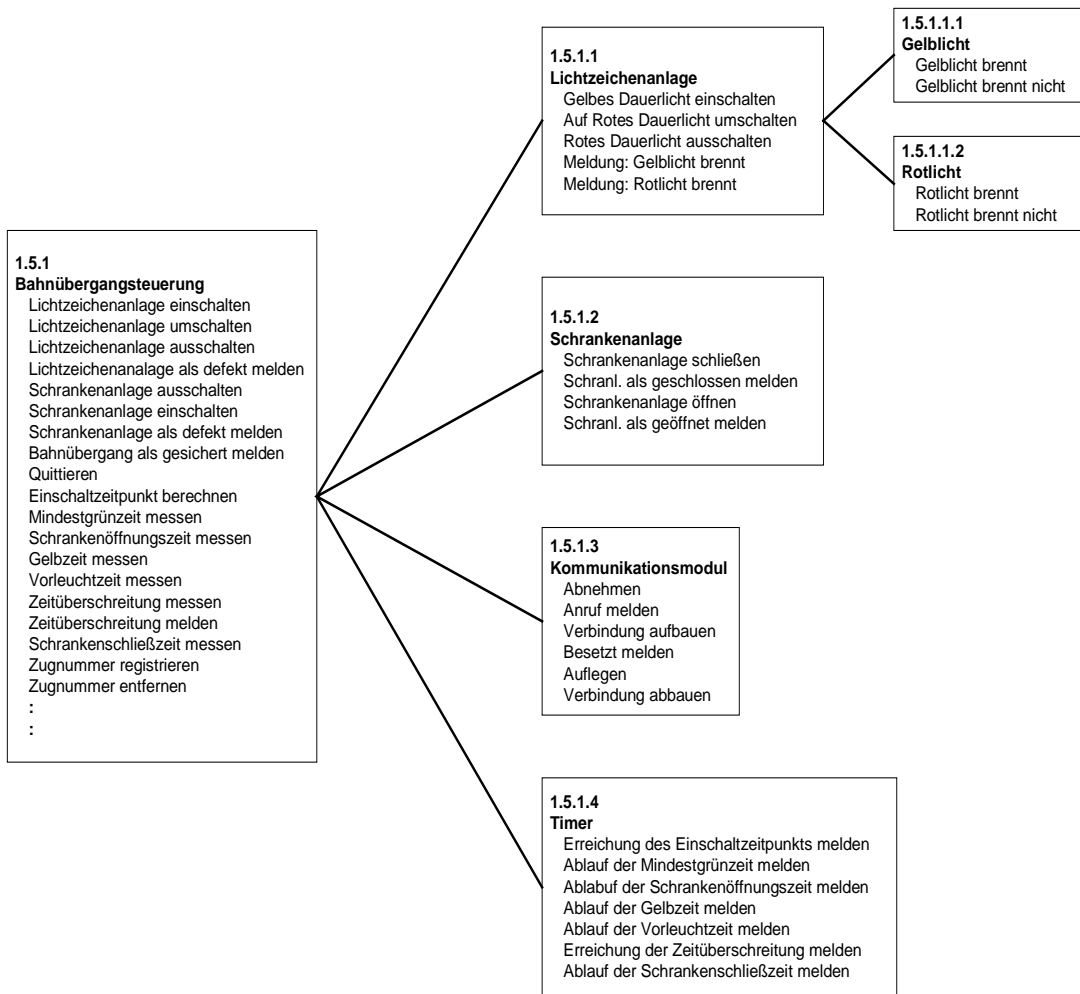


Abbildung 4: Grobe Struktur des FFB-Systems

Der Bahnübergang seinerseits besteht aus einer Lichtzeichenanlage, einer Schrankenanlage einem Kommunikationsmodul und einem Timer. Nun werden zu den einzelnen Systemelementen die jeweils dazugehörigen Funktionen aufgestellt (vgl. Abbildung 5). Im folgenden soll beispielhaft die Funktion „Auf rotes Dauerlicht umschalten“ genauer betrachtet werden. Nachdem die Gelbzeit abgelaufen ist und der Zeitpunkt zum Einschalten der Sicherungsanlage erreicht wurde, gibt die Bahnübergangssteuering der Lichtzeichenanlage die Anweisung „Auf rotes Dauerlicht umschalten“. Die Lichtzeichenanlage schaltet daraufhin das Gelblicht aus und anschließend das Rotlicht ein. Das Rotlicht brennt nun und kennzeichnet somit den Straßenverkehrsteilnehmern die geschlossenen Schrankenbäume, um den Bahnübergang zu sichern.



**Abbildung 5: FMEA mit Funktionsdefinitionen**

Zu allen bestimmten Funktionen werden nun die entsprechenden Fehlfunktionen aufgestellt (vgl. Abbildung 6).

Die technischen Sicherheitsanforderungen werden anschließend im Kontext der konkreten Realisierung auf Grund der gegebenen Funktionalitäten, Fehlfunktionalitäten und deren Fehlerauswirkungen (Fehlfunktionszusammenhänge) abgeleitet. Aus dem in Abbildung 6 gezeigten Beispiel folgen beispielsweise folgende technische Sicherheitsanforderungen:

1. Leuchtet das rote Dauerlicht nicht, so darf die Schrankenanlage nicht eingeschaltet werden.
2. Das gelbe und das rote Dauerlicht dürfen zu keinem Zeitpunkt gemeinsam brennen.

Damit die Sicherheitsanforderungen für den formalen Nachweis verwendet werden können, müssen sie nun noch formalisiert werden. Um eine ingenieurgerechte formale Darstellungsweise der Sicherheitsanforderungen geht es im folgenden Kapitel.

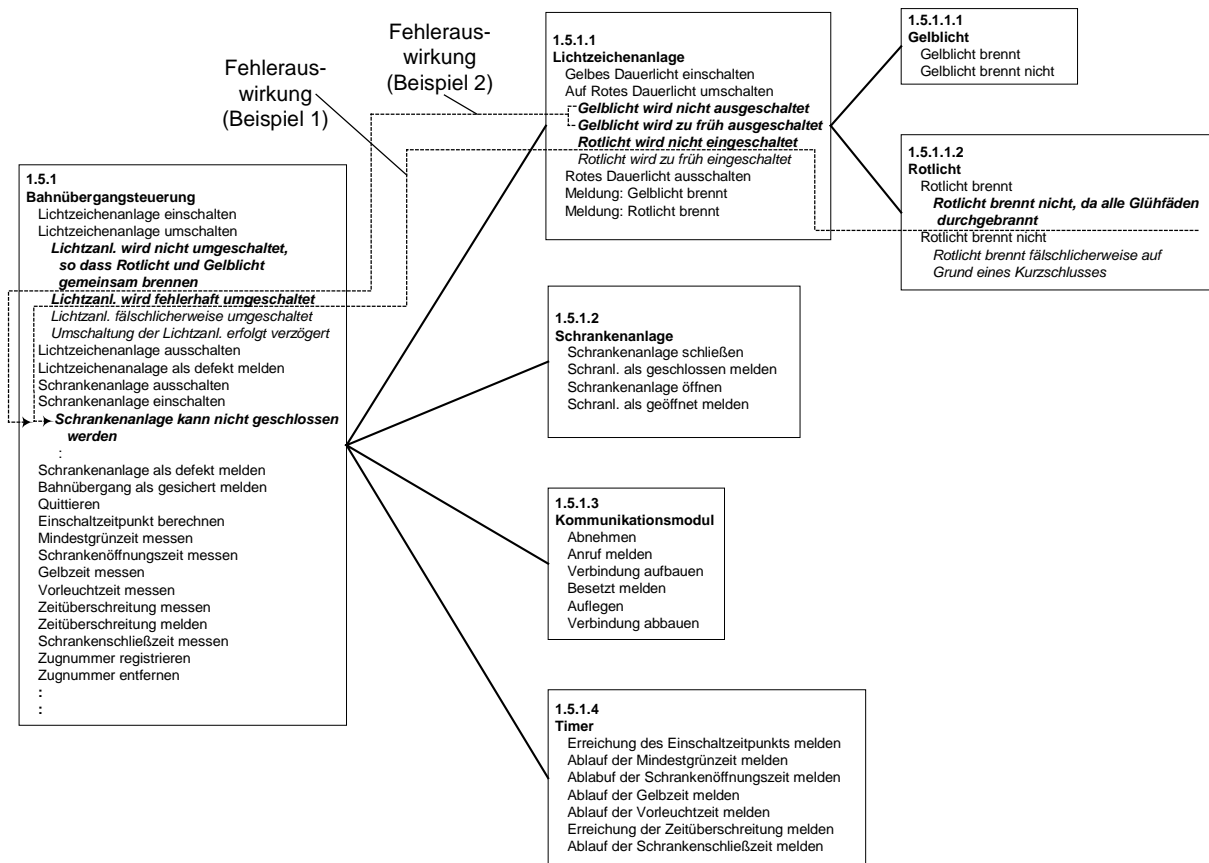


Abbildung 6: Beispiele für die Definition der Fehlfunktionen

### 3. Formalisierungsansatz für die Darstellung von Sicherheitsanforderungen in UML

#### 3.1 Formalisierungsansatz in UML

In ingenieurwissenschaftlichen Anwendungen werden zunehmend moderne objektorientierte Beschreibungstechniken und Methoden zur Softwareentwicklung eingesetzt. Unter den verschiedenen Ansätzen genießt insbesondere die UML-Notation eine hohe Popularität. In [ArGa99] wird die Eignung der UML für die Entwicklung von Eisenbahnsicherungssystemen mit einem sehr positiven Ergebnis untersucht. Sicherheitsanforderungen, die nur mit Hilfe von logischen Formeln dargestellt werden, sind für Ingenieure nicht intuitiv verständlich. Von den grafischen Beschreibungsmitteln, wie zum Beispiel UML, die Ingenieure für funktionale Spezifikationen verwenden, sind sie eine andere Art und Weise der Darstellung von Spezifikationen gewohnt. Dies stellt ein Hindernis für den praktischen Einsatz formaler Techniken für den Sicherheitsnachweis dar und ist ein Grund, warum formale Techniken im Ingenieurwesen nur sehr wenig Akzeptanz finden. Ein weiterer wichtiger Grund dafür ist, dass formale Verifikationswerkzeuge vorhandene bzw. Ingenieuren vertraute Modellierungswerkzeuge nicht unterstützen. In diesem Kapitel beschäftigen wir uns damit, Sicherheitsanforderungen mit verständlichen bzw. leicht erlernbaren Darstellungsformen mit vorhandenen Werkzeugen zu spezifizieren. Dazu werden sie in die UML integriert, so dass sie mit UML-Werkzeugen spezifiziert werden können, wodurch für die Spezifikation der Architektur, des Entwurfs sowie der Sicherheitsanforderungen ein und dasselbe Werkzeug verwendet werden kann.

#### 3.2 Arten von Sicherheitsanforderungen

Man unterscheidet verschiedene Arten von funktionsorientierten Sicherheitseigenschaften. Für unsere Betrachtungen sind insbesondere Invarianten und Anforderungssequenzen (Ausführungssequenzen, engl.:

execution sequences) zu unterscheiden (vgl. [Hei98]), da dies nach unserer Erfahrung die am häufigsten auftretenden Arten von Sicherheitsanforderungen sind. Sonstige Arten von Sicherheitsanforderungen wie z.B. Anforderungen, die Ausfallwahrscheinlichkeiten enthalten, werden in diesen Betrachtungen zurückgestellt.

Invarianten beschreiben Eigenschaften, die in allen erreichbaren Zuständen oder Transitionen des Systems erfüllt sein müssen und niemals verletzt werden dürfen [BhHe97] wie zum Beispiel, dass in allen erreichbaren Systemzuständen die BOOLE'schen Variablen  $x$  und  $y$  niemals beide gleichzeitig auf „wahr“ gesetzt sein dürfen. Anforderungssequenzen stellen Sicherheitsanforderungen dar, die eine bestimmte Folge von Zuständen bzw. Ereignissen fordern oder verbieten, die auftreten bzw. verhindert werden muss, sobald das System einen bestimmten Zustand erreicht oder ein bestimmtes Ereignis auftritt bzw. ausbleibt.

Insbesondere die Darstellung von Anforderungssequenzen in einem formalen logischen Gerüst (z.B. Temporallogik oder  $\mu$ -Kalkül – [Kru94]) erweist sich als äußerst komplex. Die Erfahrung zeigt zwar (vgl. [Hei98]), dass in der Praxis Invarianten die häufiger vorkommende Anforderungsart in Sicherheitspezifikationen für Software in Automatisierungssystemen ist, dennoch gibt es Sicherheitsanforderungen, die nur als Anforderungssequenzen dargestellt werden können.

Im folgenden wird beschrieben, wie man sowohl Sicherheitsanforderungen, die als Invarianten dargestellt werden, als auch Sicherheitsanforderungen, die als Anforderungssequenzen beschrieben werden müssen, spezifizieren kann. Dazu werden Beschreibungsmittel, die in UML zur Verfügung stehen, geeignet adaptiert. Dadurch wird eine Spezifikation von Sicherheitseigenschaften ermöglicht, der eine präzise definierte formale Semantik zu Grunde liegt und die in einer für Ingenieure vertrauten Spezifikationsumgebung erstellt werden kann.

### **3.3 Integration von Invarianten in Klassendiagramme**

Die Beschreibung eines Systems umfasst im wesentlichen zwei Aspekte: Seinen strukturellen Aufbau und sein Verhalten. Der strukturelle Aufbau ist gegeben durch die Komponenten bzw. funktionellen Einheiten aus denen das System besteht und den Beziehungen zwischen diesen Bestandteilen. Das Verhalten ergibt sich aus den Interaktionen zwischen den Bestandteilen oder aus Zustandswechseln, die in den Komponenten auftreten; es wird in UML mittels Zustandsübergangsdigrammen (Statecharts) beschrieben.

In UML-Klassendiagrammen wird der strukturelle Aufbau eines Systems beschrieben. Dabei werden auch die Gemeinsamkeiten und Ähnlichkeiten zwischen den Klassen entsprechend der objekt-orientierten Vorgehensweise als Vererbungshierarchien dargestellt. Die verschiedenen Komponenten eines Systems stehen in wechselseitigen Beziehungen zueinander. Häufig sind Sicherheitsanforderungen an diese Beziehungen geknüpft.

Die Eigenschaften eines Systems (zugrundegelegte Annahmen sowie geforderte Anforderungen) werden oft durch Aussagen über Relationen zwischen den Klassen und/oder durch Aussagen über den in den Klassen auftretenden Attributen und ihrer Werte formuliert. Daher bietet es sich an, Annahmen und Anforderungen als Invarianten in den Klassendiagrammen darzustellen. Sicherheitsanforderungen betreffen häufig die Abhängigkeiten zwischen den Zuständen der verschiedenen Komponenten eines Systems. Diese Zustände können zwar laufend Veränderungen unterworfen sein, allerdings stehen die Zustände dann in einer bestimmten Relation zueinander. Als Eigenschaften sollen hier Invarianten betrachtet werden.

UML bietet mit der Object Constraint Language (OCL) Möglichkeiten solche Anforderungen in einer deklarativen Weise anzugeben. Beispielsweise kann Sicherheitsanforderung (2) aus Abschnitt 2.3, wie in Abbildung 7 dargestellt, mittels OCL spezifiziert werden. Die Anforderung wird in den Kontext der betreffenden Klasse (hier Lichtzeichenanlage) gestellt. Die Anforderung soll für alle Instanzen dieser Klasse gelten

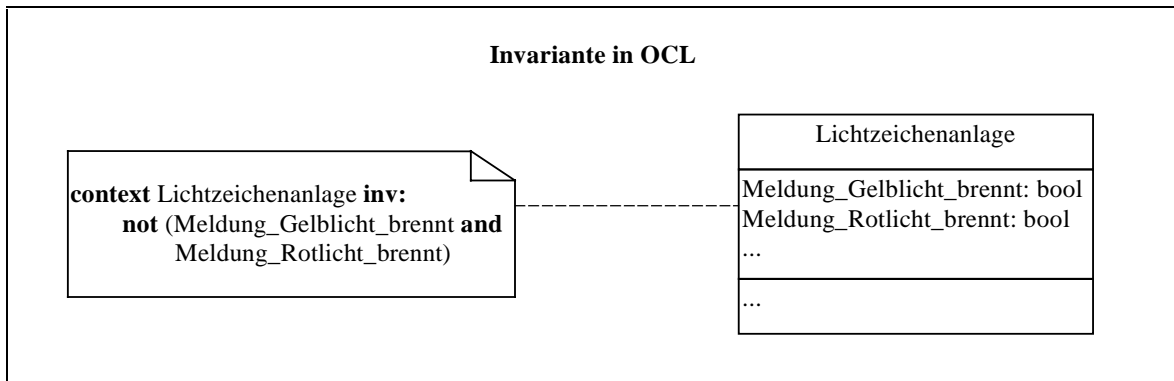


Abbildung 7: Spezifikation einer einfachen Invariante in OCL

In UML müssen Invarianten, die in OCL dargestellt werden, immer in den Kontext einer bestimmten Klasse gestellt werden. Sollen Anforderungen über Relationen zwischen Klassen angegeben werden, muss über den Relationen bzw. Klassen eine Metaklasse definiert werden, in deren Kontext man die Sicherheitsanforderung stellt, die somit mehrere Klassen in einer symmetrischen Weise übergreift.

Ziel unserer Arbeiten ist es, Sicherheitsanforderungen nachzuweisen. Dazu wollen wir zwischen den zu beweisenden Eigenschaften und den zugrundegelegten Annahmen explizit unterscheiden können. OCL unterstützt diese Unterscheidung nicht. Daher sollen geeignete Verfahren zur Spezifikation von Annahmen und Anforderungen in Anlehnung an OCL untersucht werden. Die zugrundegelegten Annahmen werden als Axiome und Definitionen und die Anforderungen als Invarianten bezeichnet.

Zur Motivation soll eine typische Anforderung im Fall des Bahnübergangs dienen, die lautet: „Ein Fahrzeug darf sich nur innerhalb der Grenzen seines ihm zugeteilten Fahrwegs befinden“. Diese Anforderung soll „stets“ gelten, stellt somit also eine erwünschte Invariante dar. Abbildung 8 illustriert den von uns untersuchten Ansatz, UML-Annotationen zur Spezifikation von Sicherheitsanforderungen einzusetzen.

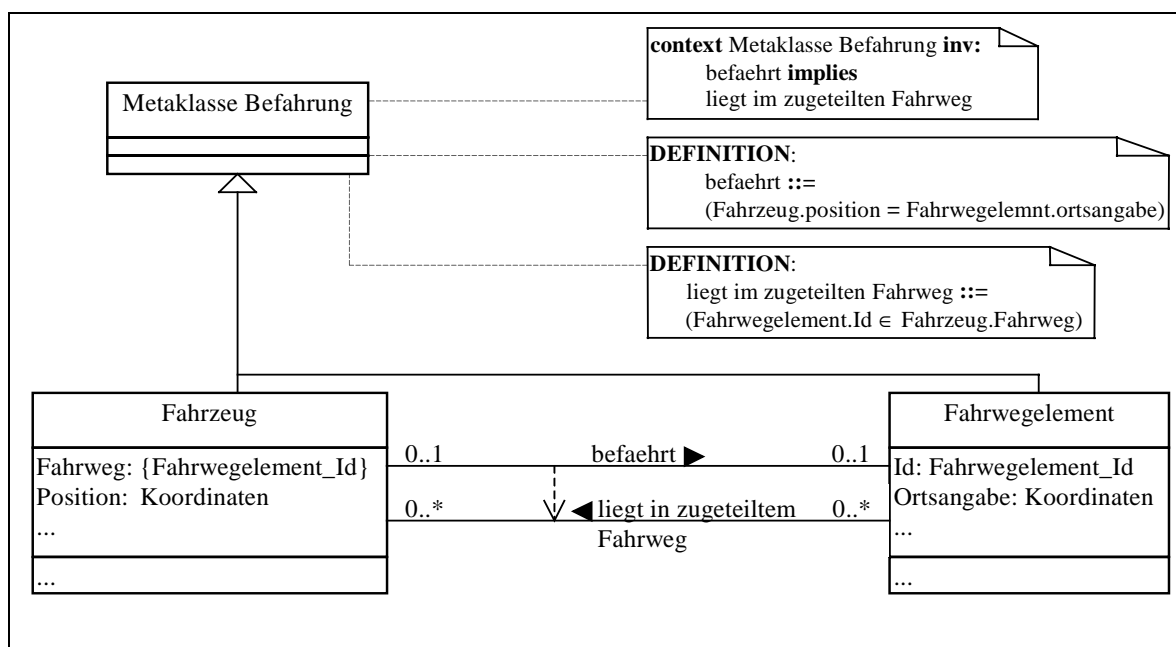


Abbildung 8: Klassendiagramm mit Spezifikation von Invarianten in Annotationen

Interessant an dieser Sicherheitsanforderung ist, dass sie zwei Relationen zwischen zwei Klassen in Beziehung zueinander setzt. Die formulierte Invariante soll für alle Instanzen der Klasse Fahrzeug und für alle Instanzen der Klasse Fahrwegelement gelten. Damit die Invariante des Beispiels für alle Instanzen der Klasse Fahrzeug und für alle Instanzen der Klasse Fahrwegelement gilt, muss eine Metaklasse definiert werden, weil in UML eine mit

OCL dargestellte Invariante immer im Kontext einer Klasse stehen muss. Die Invariante ist somit als Aussage über den Relationen zwischen den Instanzen zweier Klassen formuliert. Diese Relationen selber werden definiert, indem Aussagen über Attributwerte („Fahrzeug.Position“, „Fahrwegelement.Ortsangabe“) formuliert werden. Wie in Abbildung 8 zu sehen ist, ist im Beispiel für eine Beweisunterstützung die Definition von „befahren“ und „liegt in zugeteiltem Fahrweg“ notwendig. Diese Assoziationen könnten auch in einem funktionalen Modell realisiert werden, was wiederum die Verständlichkeit der Anforderung erschwert, da Teile der Anforderung operationell<sup>3</sup> dargestellt werden müssten. In unserem Ansatz sind diese Relationen als Abstraktionen über den Werten von Attributen in definitorischer oder deklarativer Weise (Axiome) beschrieben.

Ob die Anforderungen tatsächlich Invarianten über dem System darstellen, hängt von seinem Verhalten ab, d.h. sie werden über einem funktionalen Modell des Systems bewiesen. In UML ist dieses Modell in der Regel durch Zustandsübergangsdiagramme gegeben. Die Beweisverpflichtungen<sup>4</sup> ergeben sich aus dem funktionalen Modell, den Annahmen und den Anforderungen: Aus dem funktionalen Modell und den zugrundegelegten Annahmen sollen die gewünschten Anforderungen ableitbar sein. Im Fall endlicher und ausreichend kleiner Systeme kann die Überprüfung dann unter Einsatz eines Model-Checkers<sup>5</sup> erfolgen. Falls das zu überprüfende System zu groß sein sollte, um Model-Checker einzusetzen, kann man versuchen, den Zustandsraum des Systems durch Abstraktionsverfahren [BhHe97] zu verkleinern. In der Regel wird man dann die Korrektheit der Abstraktion mit Verfahren aus dem Bereich des Theorembeweisens nachweisen müssen.

### 3.4 Darstellung von Anforderungssequenzen als MSCs

Oft sind Sicherheitsanforderungen an den betrieblichen Ablauf eines Systems geknüpft. Sie sollen sicherstellen, dass bestimmte Folgen von Ereignissen eintreten bzw. nicht eintreten sollen. UML bietet bereits mit den von MSCs (Message Sequence Charts) abgeleiteten Sequenzdiagrammen ein Mittel zur Beschreibung von Folgen von Ereignissen. Allerdings eignen sie sich nur begrenzt zur Spezifikation von Anforderungen an ein System, da nicht festgelegt ist, unter welchen Rahmenbedingungen ein Sequenzdiagramm gelten soll [DaHa99]: Werden alle oder nur manche Abläufe des Systems betrachtet, soll die Folge nur einmal oder immer wieder auftreten?

Ein weiteres Manko der Sequenzdiagramme tritt insbesondere dann zutage, wenn Sicherheitsanforderungen beschrieben werden sollen. Auf einer abstrakten Beschreibungsebene will man nicht sämtliche erwünschte Folgen von Ereignissen auflisten, sondern hat in der Regel eine gut umrissene Vorstellung von den unerwünschten Ereignisfolgen. Dies spiegelt sich auch in vielen der im Rahmen der Fallstudie aus der FMEA gewonnenen Sicherheitsanforderungen wieder. Allerdings lassen sich Sequenzdiagramme in UML nicht als unerwünscht kennzeichnen. Daneben kann es sinnvoll sein, logische Kombinationen von (bzw. Abhängigkeiten zwischen) Ereignisfolgen darzustellen. Auch dies ist in UML nicht möglich. Eine Abhilfe besteht darin, diese Diagramme in eine erweiterte Notation einzubetten, die diese Ausdrucksmöglichkeit bietet [Can99].

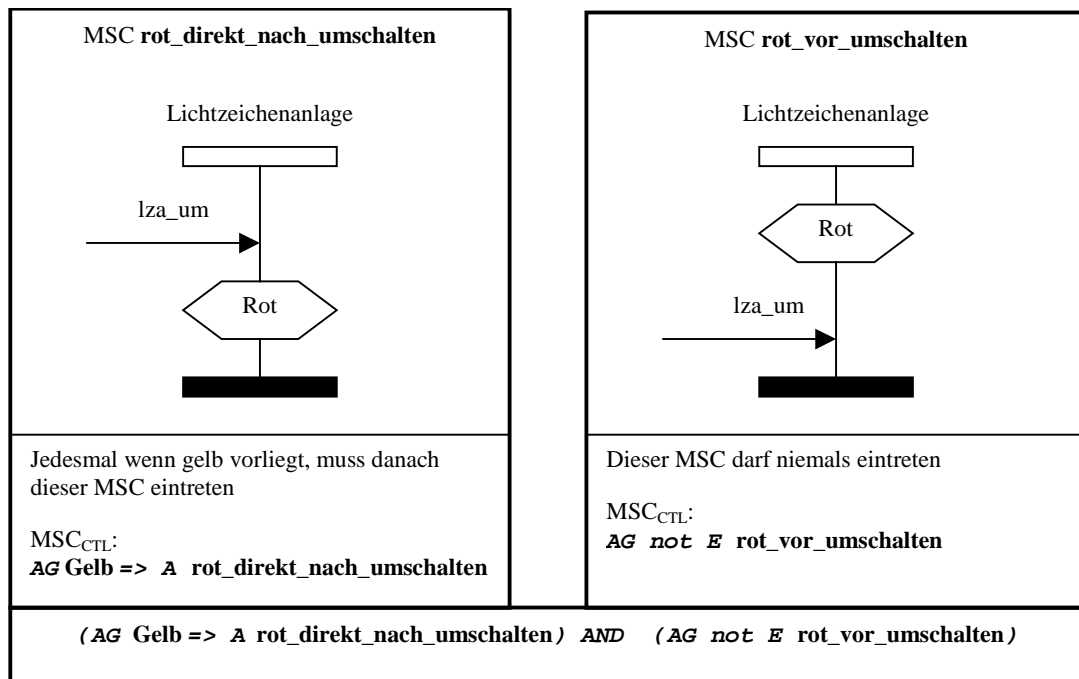
Als Beispiel zur Illustration des Ansatzes dient hier eine technische Sicherheitsanforderung: „Das Rotlicht darf erst und muss direkt dann eingeschaltet werden, nachdem die Lichtzeichenanlage einen Befehl zum Umschalten von gelb nach rot erhalten hat“ (vgl. Abbildung 10).

Die formale Semantik der  $MSC_{CTL}$ -Formeln für die Anforderungen ergibt sich aus ihrer Umsetzung in die Temporallogik CTL. Das Anforderungsbeispiel besteht aus zwei Anforderungsteilen, die einfach durch ein logisches AND miteinander kombiniert werden können. Die Darstellung des ersten Teils in  $MSC_{CTL}$  bedeutet, dass für alle erreichbaren Berechnungspfade im Zustandsraum vom Anfangszustand an immer gilt („AG“), dass sobald das gelbe Dauerlicht brennt, in jedem weiteren Berechnungspfad im Zustandsraum („A“) als nächstes der MSC „rot\_direkt\_nach\_umschalten“ ausgeführt werden muss. Der zweite Anforderungsteil ist ein Beispiel für ein unerwünschtes Ereignis. Er hat in  $MSC_{CTL}$  ausgedrückt die Bedeutung, dass für alle erreichbaren Berechnungspfade im Zustandsraum vom Anfangszustand an immer gilt („AG“), dass es nicht sein darf („not“), dass in irgendeinem Berechnungspfad des Zustandsraums („E“ - dies ist der Existenzquantor) der MSC „rot\_vor\_umschalten“ ausgeführt wird.

<sup>3</sup> Mit „operationell“ bezeichnet man Beschreibungen, die ausführbar sind.

<sup>4</sup> Mathematische Sätze, deren Nachweis sicherstellen soll, dass die formulierten Anforderungen von dem modellierten System erfüllt werden.

<sup>5</sup> Model Checking ist eine Methode zur formalen Verifikation. Dabei durchlaufen Algorithmen den gesamten Zustandsraum des funktionalen Modells vollständig und überprüfen dabei die Einhaltung der Anforderungen.



**Abbildung 10: Spezifikation einer Sicherheitsanforderung in MSC<sub>CTL</sub>**

Ob das System die Anforderungen erfüllt, wird wiederum über seinem spezifizierten funktionalen Verhalten (Zustandsübergangsdiagramme) geprüft. Ist der Zustandsraum des Systems hinreichend klein, kann die Überprüfung automatisch mit einem Model-Checker durchgeführt werden. Wie auch im letzten Abschnitt erwähnt, kann man für den Fall, dass das zu prüfende System zu groß ist, versuchen, Abstraktionsverfahren zur Reduzierung der Größe des Zustandsraums einzusetzen.

## 4. Zusammenfassung

Im ersten Teil des Berichts wurde dargestellt, wie mit Hilfe der FMEA nicht nur eine Sicherheitsanalyse von Softwarespezifikationen für Eisenbahnsicherungssysteme durchgeführt werden kann, sondern wie mit Hilfe der FMEA auch Sicherheitsanforderungen aus der Systemarchitektur und dem Systementwurf systematisch abgeleitet werden können. Dadurch werden die globalen Sicherheitsanforderungen, die in der Systemanforderungsspezifikationsphase erstellt werden, verfeinert und so formuliert, dass sie für den formalen Sicherheitsnachweis verwendbar sind. Das hat darin seinen Grund, dass die somit gewonnen technischen Sicherheitsanforderungen auf der Beschreibungsebene der Architektur und des Entwurfs ausgedrückt werden.

Im zweiten Teil wurde erläutert, wie man für den formalen Sicherheitsnachweis sowohl Sicherheitsanforderungen, die als Invarianten beschrieben werden, als auch Sicherheitsanforderungen, die als Anforderungssequenzen beschrieben werden müssen, formal spezifizieren kann. Hierbei wurde versucht, verständliche bzw. leicht erlernbare Darstellungsformen einzusetzen. Dazu werden Beschreibungsmittel, die in UML zur Verfügung stehen, geeignet adaptiert. Dadurch wird eine Spezifikation von Sicherheitsanforderungen ermöglicht, der eine präzise definierte formale Semantik zu Grunde liegt und die mit einem bestehenden und Ingenieuren vertrauten UML-Spezifikationswerkzeug erstellt werden kann. Invarianten werden dabei in Klassendiagrammen durch OCL dargestellt. Dafür wird OCL erweitert, um zwischen Invarianten, Definitionen und Annahmen unterscheiden zu können. Anforderungssequenzen werden mit Hilfe von MSCs dargestellt, die in eine Temporallogik eingebettet werden (MSC<sub>CTL</sub>). Es ist geplant, die in UML dargestellten Sicherheitsanforderungen in eine Sprache automatisiert zu extrahieren, von der aus eine einfache Umsetzung in die Eingangsspezifikationssprachen verschiedener Model-Checker möglich ist.

Die strukturierte Erstellung von Sicherheitsspezifikationen setzt damit auf industriell eingeführten Methoden und Werkzeugen auf (FMEA-Werkzeug und UML-Modellierungswerkzeug) und lässt sich somit leicht in die vorhandenen Entwicklungsprozesse integrieren. Die Erprobung des erläuterten Verfahrens soll noch an anderen

Beispielen durchgeführt werden. Es ist geplant, das vorgeschlagene Vorgehen in ein Vorgehensmodell einzubetten, welches die Anforderungen von Normen (insbesondere prEN 50128 und IEC 61508) und die Anforderungen aus der alltäglichen Praxis bei der Entwicklung von Systemen mit Sicherheitsverantwortung berücksichtigt.

## Literatur

- [ArGa99] Arabestani, S., Gayen, J.-T.  
*Ein Weg zur Einsetzbarkeit Formaler Methoden für Ingenieure im Eisenbahnwesen.*  
FORMS-Workshop, Braunschweig, 1999
- [BhHe97] Bharadwaj, R., Heitmeyer, C.  
*Model Checking Complete Requirements Specifications Using Abstraction.*  
Memorandum Report NRL/MR/5540-97-7999, Naval Research Laboratory, Washington, DC 20375. Nov. 10, 1997
- [Can99] Canver, E.  
*Model-Checking zur Analyse von Message Sequence Charts über Statecharts.*  
Ulmer Informatik Berichte 99-04, Universität Ulm, 1999
- [CGM97] Canver, E., Gayen, J.-T. und Moik, A.  
*Formale Spezifikation von Steuerungssoftware am Beispiel einer Weiche.*  
atp - Automatisierungstechnische Praxis, 39(5):57-64, Mai 1997
- [DaHa99] Damm, W., Harel, D.  
*LSCs: Breathing Life into Message Sequence Charts.*  
in Proc. of FMOODS'99, Kluwer Academic Publ., 1999.
- [DIN VDE 0801] Grundsätze für Rechner in Systemen mit Sicherheitsaufgaben.  
Ausgabe 01.90 (Vornorm)
- [EN 50126] *Bahnanwendungen – Spezifikation und Nachweis der Zuverlässigkeit, Instandhaltbarkeit, Sicherheit (RAMS)*
- [prEN 50128] *Bahnanwendungen – Software für Eisenbahnsteuerungs- und –Überwachungssysteme*
- [ENV 50129] *Bahnanwendungen – Sicherheitsrelevante elektronische Systeme für Signaltechnik*
- [Gön95] Göhner, P.  
*Spezifikation und Verifikation von sicheren Softwaresystemen*  
atp, 4/1995, pp. 24-31.
- [Hei98] Heitmeyer, C., et. al.,  
*Using Abstraction and Model Checking to Detect Safety Violations in Requirements Specification.*  
IEEE Transactions on Software Engineering, Vol. 24, No. 11, November 1998
- [Jan99] Jansen, L.  
*Referenzfallstudie Bahnübergang.* Referenzfallstudie im Bereich Verkehrsleittechnik des DFG-SPP Softwarespezifikation (Verion 1.1), Braunschweig, Juni 1999
- [Kru94] Krushan, R.P.  
*Computer-Aided Verification of Coordinating Processes: The Automata-Theoretic Approach.*  
Princeton Univ Press 1994.
- [LbGö99] Lauber, R., Göhner, P.  
*Prozessautomatisierung; Bd. 1.*  
Springer-Verlag Berlin Heidelberg, 3. Aufl. - 1999
- [Lev95] Leveson, N.  
*Safeware: system safety and computers.*  
Adison-Wesley, 1995
- [VDA96] *Sicherung der Qualität vor Serieneinsatz, FMEA,*  
VDA-Band 4, Teil 2  
Verband der Automobilindustrie e.V., Frankfurt, 1996.