IAS-Forschungsberichte

Forschungsbericht
Institut für Automatisierungs- und Softwaretechnik

Hrsg.: Prof. Dr.-Ing. Dr. h. c. P. Göhner

Uwe Biegert

Ganzheitliche modellbasierte Sicherheitsanalyse von Prozessautomatisierungssystemen

Band 2/2003

Universität Stuttgart

Ganzheitliche modellbasierte Sicherheitsanalyse von Prozessautomatisierungssystemen

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von
Uwe Biegert
aus Heilbronn

Hauptberichter:	Prof. Dr.-Ing. Dr. h. c. Peter Göhner
Mitberichter:	Prof. Dr. Hans-Joachim Wunderlich
Tag der Einreichung:	17.12.02
Tag der mündlichen Prüfung:	21.07.03

Institut für Automatisierungs- und Softwaretechnik
der Universität Stuttgart

2002

IAS-Forschungsberichte

Band 2/2003

Uwe Biegert

Ganzheitliche modellbasierte Sicherheitsanalyse von Prozessautomatisierungssystemen

D 93 (Diss. Universität Stuttgart)

Shaker Verlag
Aachen 2003

Bibliografische Information Der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

Zugl.: Stuttgart, Univ., Diss., 2003

ISBN 3-8322-1904-8
ISSN 1610-4781

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen
Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9
Internet: www.shaker.de • eMail: info@shaker.de

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Assistent am Institut für Automatisierungs- und Softwaretechnik (IAS) der Universität Stuttgart.

Herrn Prof. Dr.-Ing. Dr. h. c. Peter Göhner, dem Leiter des Instituts, danke ich sehr herzlich für die Förderung dieser Arbeit, für viele wertvolle Anregungen und kritische Diskussionen, sowie die Übernahme des Hauptberichts.

Mein herzlicher Dank gilt auch Herrn Prof. Dr. Hans-Joachim Wunderlich für das Interesse und die Übernahme des Mitberichtes.

Ich bedanke mich ebenfalls sehr herzlich bei allen Kolleginnen und Kollegen für die gute Zusammenarbeit, die kollegiale Unterstützung und die konstruktiven Diskussionsbeiträge. In diesem Zusammenhang möchte ich auch die Studenten nicht vergessen, deren Diplom- und Studienarbeit bei der praktischen Umsetzung der Konzepte eine große Hilfe waren.

Schließlich möchte ich mich bei meiner Familie bedanken. Meinen Eltern, die durch ihre Unterstützung und Förderung viel von dem Erreichten beigetragen haben, sowie Sabine für ihr Verständnis, Einfühlsamkeit und Ansporn während der Entstehung dieser Arbeit.

Stuttgart, im November 2002

Uwe Biegert

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ASCII	**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
ASW	**A**utomatisierungs**s**oft**w**are
CTL	**C**omputation **T**ree **L**ogic
COM	**C**omponent **O**bject **M**odel
DIN	**D**eutsches **I**nstitut für **N**ormung
ETA	**E**vent **T**ree **A**nalysis
FMEA	**F**ehler **M**öglichkeits- und **E**influss- **A**nalyse, **F**ailure-**M**ode and **E**ffect **A**nalysis
FTA	**F**ailure **T**ree **A**nalysis
GUI	**G**raphical **U**ser **I**nterface
HAZOP	**Haz**ard and **Op**erability Studies
MODAS	**Mod**ellierungs**as**sistent
MB	**M**enschliche **B**edieneingriffe
QPT	**Q**ualitative **P**rocess **T**heory
PAAG	**P**rognose, **A**uffinden der Ursachen, **A**bschätzen der Auswirkungen und **G**egenmaßnahmen
QSIM	**Q**ualitative **S**imulation
ROOM	**R**ealTime **O**bject **O**riented **M**odelling
SA-RT	**S**tructured **A**nalysis **– R**eal**T**ime
SQMA	**S**ituationsbasierte **Q**ualitative **M**odellbildung und **A**nalyse
THERP	**T**echnique for **H**uman **E**rror **R**ate **P**rediction
TS	**T**echnisches **S**ystem
UML	**U**nified **M**odelling **L**anguage
UML-RT	**U**nified **M**odelling **L**anguage **R**eal**T**ime
VBA	**V**isual **B**asic for **A**pplications
VSE	**V**erification **S**upport **E**nvironment

Begriffsverzeichnis

Aggregation	Verknüpfung mehrerer Elemente zu einer Einheit.
Automatisierungs-software	Umfasst die gesamte Software, welche zur Ausführung von Automatisierungsfunktionen benötigt wird.
Bedienfehler	Menschlicher Bedieneingriff mit unerwünschtem Ergebnis.
Beschreibungsmittel	Einer Modellierungsmethode zur Verfügung stehendes Mittel, um Strukturen und Sachverhalte eines Modellelementes oder Modells zu beschreiben.
Bestimmungsgemäßer Betrieb	Erwarteter Betrieb eines Prozessautomatisierungssystems (Normalbetrieb).
Bestimmungsgemäße Situation	Eine bestimmungsgemäße Situation beschreibt ein Szenarium für einen bestimmungsgemäßen Betrieb.
Fehler	Allgemein: Abweichung zwischen Soll- und Ist-Verhalten bzw. Soll- und Ist-Ergebnis.
Gefahr	Sachlage, bei der das Risiko größer als das Grenzrisiko ist.
Gefahrenanalyse	Identifizierung und Untersuchung des Gefahrenpotenzials eines Systems.
Gesamtmodell	Modell eines Prozessautomatisierungssystems, welches das Verhalten des technischen Systems, der Automatisierungssoftware und der menschlichen Bedieneingriffe beschreibt.
Grenzrisiko	Ist das größte noch vertretbare Risiko eines bestimmten technischen Vorganges oder Zustands.
Menschlicher Bedieneingriff	Menschliche Handlung zur gewollten Manipulation des Prozessgeschehens. Ein Eingriff kann direkt auf das technische System oder indirekt über die Automatisierungssoftware erfolgen.
Modell	Vereinfachtes Abbild einer Realität unter einer bestimmten Sicht.
Operator	Eine Person, die ein Prozessautomatisierungssystem bedient.
PlugIn	Erweiterung eines bestehenden Programms. Bei Microsoft Office Programmen enthält ein PlugIn zusätzliche Prozeduren (Makros) und erweitert die Standardfunktionalität des Programms.
Prozessautomatisierungssystem	Ein System, mit dem Ziel der Automatisierung von technischen Prozessen. Prozessautomatisierung ist ein Fachgebiet der Automatisierungstechnik.
Risiko	Möglichkeit, Schaden zu erleiden. Setzt sich aus der Wahrscheinlichkeit des Auftretens des zum Schaden führenden Ereignisses (Eintrittswahr-

	scheinlichkeit) und das beim Ereigniseintritt zu erwartende Schadenausmaß zusammen.
Schaden	Beeinträchtigung für Leib und Wohl von Menschen sowie für die Umwelt.
Semantik	Bedeutung eines Beschreibungsmittels.
Sicherheit	Sachlage, bei der das Risiko kleiner als das Grenzrisiko ist.
Sicherheitsanalyse	Analyse, welche untersucht, ob ein Prozessautomatisierungssystem während seines Betriebs jederzeit und unter allen Umständen, auch im fehlerhaften Betrieb, sicher ist bzw. keine sicherheitskritische Situationen auftreten.
Sicherheitsanforderung	Anforderung an die Sicherheit eines Prozessautomatisierungssystems.
Sicherheitskritische Situation	Situation, welche die Sicherheit für Mensch und Umwelt in Frage stellt.
Sicherheitsmaßnahme	Maßnahme zur Verringerung des Risikos, welche entweder die Eintrittswahrscheinlichkeit oder das Ausmaß eines Schadens (oder beides) einschränken.
Situation	Quasi-stationärer Zustand eines Systemelements oder Systems, vergleichbar mit einer Momentaufnahme. Beim Verfahren SQMA setzt sich eine Situation aus den Kombinationen von Intervallwerten zusammen und beschreibt auf diese Weise ein mögliches Szenarium des Systemelements bzw. Systems.
Softwarebaustein	Systemelement der Automatisierungssoftware.
Syntax	Aufbau und Anordnung von Beschreibungsmitteln.
System	Eine Menge von kooperierenden Elementen, die zusammen eine bestimmte Aufgabe erfüllen. In dieser Arbeit wird unter System immer das gesamte Prozessautomatisierungssystem verstanden.
Systemelement	Einheiten, aus denen ein System aufgebaut wird. Systemelemente besitzen Schnittstellen und ein Verhalten.
Technisches Bauelement	Systemelement des technischen Systems.
Technisches System	Ermöglicht den Betrieb und die Manipulation des technischen Prozesses. Im technischen System läuft der technische Prozess ab.
Systembestandteil	Bestandteil eines Systems. Systembestandteil ist der Oberbegriff für technisches System, Rechnersystem bzw. System der Automatisierungssoftware und Bedienpersonal.
Transition	Bestimmt mögliche Übergänge zwischen Situationen. Transitionen beschreiben das dynamische Verhalten eines Systems oder eines Systemelements.

Kurzfassung

Der Betrieb von Prozessautomatisierungssystemen ist immer mit einem gewissen Risiko verbunden. Ein Prozessautomatisierungssystem gilt dann als sicher, wenn das vorhandene Risiko zu keiner Zeit ein so genanntes Grenzrisiko überschreitet. Wird das Grenzrisiko überschritten, so droht Menschen und Umwelt unmittelbar ein Schaden. Mit Hilfe von Sicherheitsanalysen kann das vorhandene Risiko untersucht und abgeschätzt werden. Klassische Sicherheitsanalysen betrachten in der Regel nur einzelne Bestandteile eines Prozessautomatisierungssystems, welches aber im Allgemeinen aus drei verschiedenen Bestandteilen besteht: dem technischen System, dem Rechnersystem und dem Bedienpersonal. Was passiert aber, falls im technischen System ein Bauelement ausfällt, die Automatisierungssoftware Fehler enthält und zur gleichen Zeit das Bedienpersonal falsche Bedieneingriffe ausführt? Solche Fragen können mit klassischen Sicherheitsanalysen nur unzureichend beantwortet werden. Hinzu kommt, dass bei den meisten klassischen Sicherheitsanalysen die eigentliche Analyse des Systems in Form von Brainstorming-Prozessen durchgeführt wird. Dabei kann der Mensch niemals alle möglichen Kombinationen des Zusammenspiels zwischen den Bestandteilen überblicken und bewerten.

In der vorliegenden Arbeit wird ein modellbasierter Ansatz zur Durchführung einer ganzheitlichen Sicherheitsanalyse vorgestellt, welche alle Bestandteile eines Prozessautomatisierungssystems berücksichtigt. Die Ausführung erfolgt rechnergestützt. Auf Grund der Komplexität von Prozessautomatisierungssystemen wird eine qualitative komponentenorientierte Modellierungsmethode gewählt. Die Systemgrößen werden durch qualitative Intervallvariablen beschrieben, wobei die definierten Intervallbereiche zusätzlich durch qualitative Ausdrücke kommentiert werden. Durch Kombination von Intervallbereichen entstehen kommentierte Situationen, die das Verhalten wiedergeben. Dabei wird sowohl der bestimmungsgemäße als auch der fehlerhafte Betrieb berücksichtigt. Anhand der Systemstruktur werden die Modelle der Bestandteile miteinander kombiniert, um alle möglichen Situationen des gesamten Prozessautomatisierungssystems zu erhalten. Anschließend werden die ermittelten sicherheitskritischen Situationen des Prozessautomatisierungssystems bewertet und es wird entschieden, ob Sicherheitsmaßnahmen notwendig sind.

Durch das rechnergestützte Vorgehen lassen sich im Unterschied zu klassischen Methoden beliebig viele Fehlerkombinationen analysieren und damit Sicherheitslücken im Prozessautomatisierungssystem ermitteln. Das komplexe Zusammenspiel der Bestandteile wird mit Hilfe des qualitativen Modells transparent und analysierbar. Das Modell ist auf Grund seines qualitativen Charakters einfach anzuwenden und die Ergebnisse können leicht interpretiert werden.

Abstract

The operation of an automation system carries certain risks. A system is considered safe if a maximum acceptable risk is not exceeded at any time. If the maximum acceptable risk is exceeded, damage to human life and environment may occur. With the help of a safety analysis, the risk associated with a system can be examined and estimated. In general, conventional safety analysis methods consider only one particular part of an automation system. However, automation systems consist of three different parts: the technical system, the computer system and the operating crew. But what happens, if an element of the technical system is damaged, the software has bugs and the operator takes inappropriate actions at the same time? Using conventional safety analysis methods such questions can be answered inadequately, only. An additional lack of conventional methods of safety analysis is, that the analysis process is done by brainstorming. Therefore the user can hardly evaluate all possible interactions between the three counterparts.

In this work a model-based concept for an integral safety analysis is presented. It takes all parts of an automation system into account and the analysis is carried out by a computer. Because of the complexity of automation systems a qualitative component-oriented modelling method is chosen. The quantities of the system are described by qualitative interval variables. A qualitative value is given to each interval. Commented situations which represent the behaviour of a system part arise by combining the intervals. In the model, both the normal and the faulty behaviour are taken into account. Base on the system structure the models of the system parts are combined to get all possible situations of the complete automation system. Any resulting safety critical situations of the automation system have to been judged with respect to their risk and may have to be avoid by adding extra safety functions.

In difference to classic methods, the computer-aided procedure is able to analyse any numbers of combinations of failures to find safety-gaps in the system. With the help of the qualitative model, the complex interaction of the system parts becomes transparent and analysable. Because of its qualitative character, the model is simple to apply and the results can be interpreted easily.

Computers do not produce new sorts of errors. They merely provide new and easier opportunities for making the old errors.
- Trevor Kletz (Wise After the Event) -

1 Einführung in die Problematik von Sicherheitsanalysen

1.1 Bedeutung von Sicherheitsanalysen in der Prozessautomatisierung

In den vergangenen Jahren wurden im Bereich der Prozessautomatisierung zunehmend Aufgaben durch Rechnersysteme übernommen, die früher noch der Mensch durchgeführt hat. Damit wurde die Steuerung und Kontrolle von immer komplexeren Vorgängen möglich [Leve95]. Durch die Erweiterung ihres Aufgabenbereichs übernahmen Rechnersysteme zunehmend auch Funktionen, die die Sicherheit des Systems betreffen. Heute werden viele sicherheitskritische Vorgänge von Rechnersystemen gesteuert oder überwacht, ohne dass der Mensch eingreift. Seit gut vier Jahren verkehrt z. B. in der französischen Hauptstadt Paris die Metro auf der Linie „Météor“ ferngesteuert ohne Triebfahrzeugführer [Bahn00]. Dabei wird auch das Ein- und Aussteigen der Passagiere automatisch überwacht. Bei solchen hoch automatisierten Systemen muss bereits während der Entwicklung sichergestellt werden, dass im späteren Betrieb keine gefährlichen Situationen für den Menschen und für die Umwelt entstehen können. Das System muss daher über eine definierte Sicherheit verfügen. In vielen Fällen genügt schon der Ausfall eines Systemelements, um gefährliche Situationen herbeizuführen. Die Absturzursachen einiger Flugzeugkatastrophen waren z. B. verstopfte, vereiste oder für Wartungszwecke zugeklebte Staurohre von Geschwindigkeitsmessern [BFU02]. Bei der Softwareentwicklung des Autopiloten wurde diese Art der Störung nicht berücksichtigt. Der Autopilot reagierte mit Zurücknahme der Schubkraft bis zum Abriss des Luftstroms.

Für die Entwickler von Prozessautomatisierungssystemen stellt sich in diesem Zusammenhang die Frage: „Wie sicher ist sicher genug?“ – Eine Verbesserung der Systemsicherheit steht in vielen Fällen im Widerspruch zur Systemzuverlässigkeit und bedeutet zusätzliche Investitionskosten bei der Entwicklung [Mont00]. So erhöht zum Beispiel die Notbremse in Zügen die Sicherheit des Systems, bedeutet aber gleichzeitig höhere Produktionskosten. Die Zuverlässigkeit des Verkehrsmittels wird wiederum aufgrund des Missbrauchs der Notbremse gesenkt.

Um generell Informationen über die Sicherheit eines Systems zu erhalten, werden Sicherheitsanalysen durchgeführt. Ziel dabei ist die Ermittlung von sicherheitskritischen Situationen. Neben dem geplanten, bestimmungsgemäßen Verhalten des Systems wird ebenfalls das Verhalten

im Fehlerfall untersucht. Auch im Fehlbetrieb muss die Sicherheit eines Systems gewährleistet sein.

Zur Durchführung einer Sicherheitsanalyse wird eine Vielzahl verschiedener Methoden mit unterschiedlichen Zielsetzungen eingesetzt. Es existieren quantitative Methoden, bei denen die Sicherheit eines Systems in Kenngrößen (z. B. Risikowert, Ausfallswahrscheinlichkeit) angegeben wird und qualitative Methoden, bei denen verschiedene Systemzustände analysiert werden. Die Sicherheitsanalysen werden parallel zum Entwicklungsprozess eines Prozessautomatisierungssystems eingesetzt. Die zeitlich günstigste Einflussmöglichkeit auf die Produktionskosten besteht in frühen Entwicklungsphasen. In den frühen Entwicklungsphasen können Fehler mit einem wesentlich geringeren Zeitaufwand behoben werden [WeOs98]. Der Mangel an Informationen in den frühen Entwicklungsphasen stellt allerdings ein großes Problem dar. Um quantitative Methoden einsetzen zu können, müssen detaillierte Informationen zu den Systemelementen vorliegen. Qualitative Ansätze unterstützen in der Regel ein systematisches Vorgehen bei der Durchführung von Sicherheitsanalysen. Die Qualität einer Analyse ist abhängig von dem Kenntnisstand und der Erfahrung der beteiligten Personen. Die Analyse findet in Form von Diskussionen und im Wesentlichen als Brainstorming-Prozess statt. Schwierig gestaltet sich dabei die Reproduktion der Ergebnisse bei einer nachträglichen Prüfung durch andere Personen.

Die Schwierigkeit bei der Durchführung von Sicherheitsanalysen liegt in der Beherrschung der Komplexität eines Prozessautomatisierungssystems. Zur Beherrschung dieser Komplexität werden große Systeme in Teilsysteme zerlegt oder die Sicherheitsanalyse wird nur auf eine gewisse Teilfunktionalität beschränkt, z. B. die Betrachtung eines speziellen Systemzustands. Die Sicherheitsanalyse des gesamten Systems besteht in diesem Fall aus einer Sammlung von Ergebnissen der Einzeluntersuchungen. Das kausale Zusammenwirken zwischen den Teilsystemen wird dabei oft nur unzureichend oder gar nicht berücksichtigt. Der vermehrte Einsatz von Rechnern bei Prozessautomatisierungssystemen (und damit auch von Software) führt zu einer noch höheren Komplexität und erschwert dabei zusätzlich den Brainstorming-Prozess der klassischen Sicherheitsanalysen.

1.2 Zielsetzung der ganzheitlichen Sicherheitsanalyse

Ziel dieser Arbeit ist die Entwicklung eines Konzepts für eine ganzheitliche Sicherheitsanalyse. Es sollen sicherheitskritische Situationen eines Prozessautomatisierungssystems unter Beachtung nachfolgender Aspekte ermittelt werden:

- Die Sicherheitsanalyse soll sich nicht - wie bei klassischen Verfahren - auf spezielle Bestandteile eines Prozessautomatisierungssystems beschränken, sondern das System ganzheitlich betrachten. Insbesondere soll die Untersuchung des komplexen Zusammenspiels der Bestandteile mit in die Analyse einfließen.

- Mögliche Fehler in den Bestandteilen aber auch bestandteil-übergreifende Fehlerkombinationen sollen bei der ganzheitlichen Sicherheitsanalyse betrachtet werden. Fehlerkombinationen sind nach [Leve95], [Mont00] die häufigsten Ursachen für Unfälle.

- Die ganzheitliche Sicherheitsanalyse muss weiterhin in den frühen Entwicklungsphasen eines Systems anwendbar sein. Bei einer rechtzeitigen Entdeckung sicherheitskritischer Entwurfsentscheidungen oder Sicherheitslücken im geplanten System sind Korrekturen oder Sicherheitsmaßnahmen einfach und kostengünstig zu realisieren. Es ist zu beachten, dass Systeminformationen in den frühen Entwicklungsphasen oft nur unvollständig vorliegen. Die Auswirkungen von definierten Sicherheitsmaßnahmen müssen weiterhin durch das Verfahren selbst erprobt werden können.

- Um eine vollständige, systematische und profunde Sicherheitsanalyse zu realisieren, soll der Anwender weitgehend durch den Einsatz eines Rechners unterstützt werden. Eine Rechnerunterstützung ist nur mit einem zugrunde liegenden Modell zu realisieren. Die Verständlichkeit und Interpretierbarkeit des Modells ist eine wichtige Qualitätseigenschaft. Die Anwender einer Sicherheitsanalyse sind in der Regel Ingenieure aus den Fachbereichen Physik, Chemie, Elektrotechnik, Verfahrenstechnik und Maschinenbau. Da fundierte mathematische Fachkenntnisse nicht in jedem Fall vorausgesetzt werden können, soll das zugrunde liegende Modell leicht anwendbar sein und Erfahrungen sowie Fachkenntnisse des Anwenders berücksichtigen.

1.3 Gliederung der Arbeit

Diese Arbeit gliedert sich in insgesamt zehn Kapitel, die im Folgenden kurz beschrieben werden.

Das *zweite Kapitel* erläutert die für die Arbeit wichtigen Begriffe, um die Sicherheit bei Prozessautomatisierungssystemen zu definieren. Es werden die allgemeinen Bestandteile eines Prozessautomatisierungssystems sowie deren Bedeutung für die Systemsicherheit dargestellt. Das zweite Kapitel schließt mit der Darstellung der Aufgaben einer Sicherheitsanalyse und mit einem Überblick über relevante Normen und Gesetze.

Verfahren für die Sicherheitsanalyse werden im *dritten Kapitel* anhand eines kleinen Beispiels vorgestellt und bewertet. Zur Ermittlung eines geeigneten Modellkonzepts werden die in den Verfahren verwendeten Modelle untersucht. Die Ergebnisse bilden die Basis, um konkrete Anforderungen an das Konzept der ganzheitlichen modellbasierten Sicherheitsanalyse auszuarbeiten.

Im *vierten Kapitel* wird das in der Arbeit verwendete Modellierungskonzept vorgestellt. Dabei werden die wesentlichen Merkmale des qualitativen Modellierungskonzepts beschrieben. Zur Veranschaulichung wird ein Beispiel aufgeführt.

Im *fünften Kapitel* wird das Prinzip des Ansatzes vorgestellt, das maßgeblich das Konzept für die ganzheitliche modellbasierte Sicherheitsanalyse bestimmt. Das entwickelte Konzept setzt sich aus zwei Schritten zusammen: Die Modellierung eines Prozessautomatisierungssystems und die Durchführung der Sicherheitsanalyse.

Das *sechste Kapitel* befasst sich mit der Modellierung von Prozessautomatisierungssystemen und geht auf wichtige Aspekte bei der Systemmodellierung ein. Es wird gezeigt, wie das technische System, die Automatisierungssoftware und die Bedieneingriffe separat modelliert und anschließend zu einem Gesamtmodell verknüpft werden können.

Die Modellauswertung der modellbasierten Sicherheitsanalyse ist Gegenstand des *siebten Kapitels*. Es wird erläutert, wie sicherheitskritische Situationen des Prozessautomatisierungssystems mit Hilfe des Gesamtmodells interpretiert und bewertet werden. Die Überprüfung bestehender Sicherheitsanforderungen anhand des Gesamtmodells und die Diskussion von Sicherheitsmaßnahmen sind weitere Inhaltspunkte des siebten Kapitels. Der Anwender der modellbasierten Sicherheitsanalyse wird sowohl während der Modellerstellung als auch bei der Durchführung der Analyse durch Softwarewerkzeuge unterstützt. Die Funktionen und Eigenschaften der im Rahmen dieser Arbeit entwickelten Softwarewerkzeuge werden kurz vorgestellt.

Die praktische Anwendung der modellbasierten Sicherheitsanalyse steht im *achten Kapitel* im Vordergrund. Anhand einer kommerziellen Waschschleudermaschine wird der praktische Einsatz des Verfahrens erläutert. Die möglichen Betriebsszenarien der Waschschleudermaschine werden mit Hilfe der erstellten Teilmodelle ermittelt und im Gesamtmodell in Form von Situationen dargestellt. Die Auswertung sicherheitskritischer Situationen und die Ergreifung von Sicherheitsmaßnahmen stellen weitere Inhaltspunkte des Kapitels dar.

Das *neunte Kapitel* beinhaltet eine zusammenfassende Beschreibung und Bewertung der Ergebnisse der Arbeit sowie einen Ausblick auf weiterführende Aspekte.

2 Prozessautomatisierungssysteme und Sicherheit

In diesem Kapitel wird der allgemeine Aufbau eines Prozessautomatisierungssystems vorgestellt. Anschließend werden die wichtigsten Begriffe aus dem Themenfeld Sicherheit erläutert und definiert. Das Kapitel schließt mit einer Beschreibung der Aufgaben von Sicherheitsanalysen und mit einem Überblick über relevante Normen und Gesetze.

2.1 Aufbau eines Prozessautomatisierungssystems

Prozessautomatisierung ist ein Fachgebiet der Automatisierungstechnik und behandelt die Automatisierung von beliebigen technischen Prozessen. Ein Prozessautomatisierungssystem besteht aus einem technischen System, einem Rechnersystem und dem Bedienpersonal. Diese verschiedenen Bestandteile stehen zueinander in Wechselwirkung. Eine allgemeine Darstellung ist in Abbildung 2.1 gegeben.

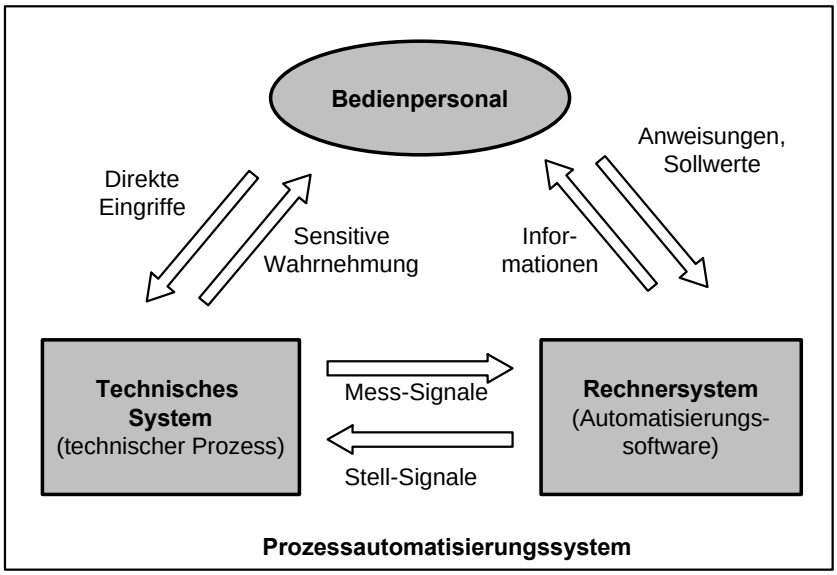


Abbildung 2.1: Bestandteile eines Prozessautomatisierungssystems

Im technischen System läuft ein technischer Prozess ab. Die Zustände des technischen Prozesses lassen sich als physikalische Größen mit technischen Mitteln erfassen und stehen der Automatisierung als Mess-Signale zur Verfügung. Zustandsänderungen des technischen Prozesses werden anhand von Stell-Signalen über Aktoren realisiert. Auf diese Weise lässt sich ein Prozess automatisieren. Das Verhalten des Rechnersystems ist geprägt durch die Automatisierungssoftware, die die automatisch ablaufenden Vorgänge im technischen Prozess bestimmt. Je nach Automatisierungsgrad besitzt das Bedienpersonal unterschiedliche Tätigkeiten. Das Bedienpersonal kann direkt in die Vorgänge des technischen Prozesses eingreifen oder indirekt über die Auto-

matisierungssoftware. Bei vollautomatisierten Prozessen beschränkt sich die Tätigkeit des Bedienpersonals auf Überwachungsaufgaben.

2.1.1 Technisches System und technischer Prozess

Den Kern eines Prozessautomatisierungssystems bildet der technische Prozess, welcher in einem technischen System eingebettet ist. Das technische System kann ein Gerät, eine Maschine oder eine technische Anlage darstellen. Mit Hilfe des technischen Systems wird der technische Prozess manipuliert. Diese Manipulation wirkt sich in einer Zustandsänderung eines Materials, einer Energie oder Information aus. Der Begriff „technischer Prozess“ umfasst eine Vielfalt von Vorgängen. Einige Klassifizierungsansätze sind in [LaGö99a] aufgeführt. Unterschieden werden:

- *Kontinuierliche Vorgänge* (flussorientierte Vorgänge, Fließprozesse)
  Kontinuierliche Vorgänge sind durch kontinuierliche Prozessgrößen geprägt. Es handelt sich in der Regel um physikalische Zustandsgrößen, die einem zeitlichen Verlauf zugeordnet werden können.
- *Sequenzielle Vorgänge* (Folgevorgänge)
  Diese Vorgänge werden durch diskrete Größen bestimmt. Es handelt sich hierbei um binäre Prozessgrößen, die Zustandsübergängen zugeordnet werden können.
- *Objektbezogene Vorgänge* (Stückgutvorgänge)
  Objektbezogene Vorgänge sind durch objektbezogene Größen geprägt, die den identifizierbaren Objekten des technischen Prozesses zugeordnet werden können.

Zur Beschreibung von technischen Systemen werden Phasenmodelle und Fließbilder eingesetzt. Je nach Abstrahierungsgrad finden unterschiedliche Darstellungsarten Verwendung. Beim Phasenmodell [Polk94] wird der Ablauf des technischen Prozesses in einzelne Vorgänge (Teilprozesse) unterteilt, die jeweils die Transformation (Zustandsänderung) eines Ausgangsproduktes in ein Zwischen- bzw. Endprodukt beschreiben. Eine detaillierte Darstellung des gesamten technischen Systems erfolgt mit Rohrleitungs- und Instrumentierungsfließbildern (RI-Fließbilder), [DIN 28004]. Beim Entwurf eines technischen Systems werden im ersten Schritt Fließbilder entwickelt, die alle notwendigen Maschinen und Apparate beinhalten. Diese enthalten ebenfalls sämtliche Durchflüsse, Mengen und Energieträger, aber auch Betriebsbedingungen [Polk94] [EnMü93]. In den nächsten Entwurfsschritten werden die Fließbilder immer weiter konkretisiert bis schließlich das RI-Fließbild fertig gestellt ist. Dieses umfasst die gesamte technische Ausrüstung und alle Zusatzinformationen.

Der technische Prozess spielt im Rahmen einer Sicherheitsanalyse die zentrale Rolle, da nur von diesem Gefahren für Mensch und Umwelt drohen können. Ein technischer Prozess ohne Gefahrenpotenzial gilt per Definition als sicher. Verfahrenstechnische Prozesse gehören zu den häu-

figsten Prozessen, die im Rahmen von Sicherheitsanalysen untersucht werden. Sie besitzen meistens ein hohes Gefahrenpotenzial [Pilz85].

Bauelemente des technischen Systems sind meist durch hohe physikalische Beanspruchung und Umwelteinflüsse besonders fehleranfällig. Mit zunehmender Betriebszeit tauchen Verschleißerscheinungen auf, die ein Versagen des Bauelements hervorrufen können. Allerdings stellen Fehler in technischen Bauelementen allein noch keine Gefahr dar, sie können allerdings die Ursache für das Auslösen einer Gefahr bzw. das Eintreten eines Schadens sein.

2.1.2 Rechnersystem und Automatisierungssoftware

Je nach Umfang und Art eines Prozessautomatisierungssystems werden unterschiedliche *Rechnersysteme* eingesetzt. Bei kleinen Automatisierungsproblemen werden häufig Mikrocontroller verwendet, bei mittleren Systemen kommen Industrie-PCs (IPC) und Speicherprogrammierbare Steuerungen (SPS) zum Einsatz. Bei großen Systemen, insbesondere bei der Anlagenautomatisierung, werden oft herstellerspezifische Prozessleitsysteme eingesetzt.

Die Aufgaben der Automatisierungssoftware hängen in erster Linie vom Automatisierungsgrad bzw. von der gewählten Struktur eines Prozessautomatisierungssystems ab und sind demnach von System zu System sehr unterschiedlich, siehe Kapitel 2.1.4. Bei hochautomatisierten Systemen übernimmt die Software die gesamte Prozessführung sowie das „Hochfahren / Herunterfahren“ einer Anlage bzw. das „Initialisieren / Abschalten“ eines Gerätes. Die Automatisierung eines Prozesses erfolgt anhand von erfassten Prozessgrößen (Mess-Signalen), aus denen nach einem festgelegten Konzept Stellgrößen (Stell-Signale) berechnet werden, die den technischen Prozess in gewünschter Weise manipulieren. Das zugrunde liegende Konzept garantiert eine immer gleiche, exakte Ausführung von Funktionen. Darüber hinaus werden Funktionen der Prozessüberwachung und Visualisierung des Prozessgeschehens mit Hilfe der Automatisierungssoftware realisiert.

Eine gängige Klassifizierung der Automatisierungssoftware richtet sich nach den Entwurfstechniken, die den Aufbau und die Eigenschaften der Software prägen. Unterschieden werden nach [Schn99] folgende Kategorien von Techniken:

- *Datenflussorientierte Techniken*
  Es werden Datenstrukturen und Datentransformationen des zu entwerfenden Systems analysiert und festgelegt. Aus diesen Informationen können anschließend Regeln zur Erzeugung von Ausgangsdaten (Stell-Signale) aus den Eingangsgrößen (Mess-Signale) abgeleitet werden.
- *Kontrollflussorientierte Techniken*
  Hierbei werden Aktionen in Abhängigkeit von einem Ereignis oder von der Erfüllung einer Bedingung beschrieben. Diese Aktionen besitzen eine bestimmte Ausführungsdauer und ei-

nen definierten Beginn und ein Ende. Der Kontrollfluss bestimmt die Reihenfolge, in der die Aktionen ausgeführt werden.

- *Objektorientierte Techniken*
  Ein Objekt ist eine Abstraktion einer Problemstellung und basiert auf einer Identität, einem Zustand und einem Verhalten. Alle Objekte gleichen Typs werden in einer Klasse zusammengefasst und durch deren Struktur beschrieben. Ein objektorientiertes System besteht ausschließlich aus miteinander kommunizierenden Objekten. Objekte können verschiedene Beziehungen untereinander aufweisen. Der interne Aufbau des Objekts ist allerdings von außen nicht erkennbar (Kapselung). Definierte Schnittstellen bilden den Kontakt nach außen.
- *Komponentenbasierte Techniken*
  Der Aspekt der Mehrfachverwendung spielt eine maßgebende Rolle. Komponenten sind komplexer als Objekte und können aus diesen aufgebaut sein. Funktionen von Komponenten werden autark ausgeführt. Komponentenbasierte Techniken verfolgen eine viel strengere Schnittstellendefinition als objektorientierte Techniken. Während bei Objekten gewisse (freigegebene) Funktionen von außen aufgerufen werden können, ist dies bei Komponenten nicht möglich. Komponenten können nur über ihre Schnittstellen Nachrichten austauschen.

Bei der eigentlichen Entwicklung der Automatisierungssoftware besitzen *Vorgehensmodelle* einen wichtigen Stellenwert [LaGö99b]. Mit der Festlegung der Tätigkeiten in verschiedenen Entwicklungsschritten (Phasen) und den anfallenden Produkten wird nicht nur ein strukturiertes Vorgehen gefördert, sondern ebenfalls die Kommunikation und insbesondere die Abstimmung der Vorstellungen zwischen Entwickler und Kunde. Aus dem geplanten Automatisierungskonzept lassen sich direkt Anforderungen an die zu entwickelnde Automatisierungssoftware ableiten. Dabei werden *funktionale* und *nicht-funktionale Anforderungen* unterschieden. *Funktionale Anforderungen* beschreiben das Aufgabenspektrum, d.h. das, was die Automatisierungssoftware leisten soll bzw. muss. *Nicht-funktionale Anforderungen* beschreiben die Rahmenbedingungen der Entwicklung der Automatisierungssoftware. Sie legen z. B. die Entwicklungsmethode oder die Programmiersprache fest. Sicherheitsanforderungen sind ebenfalls funktionale Anforderungen und werden in dieser Arbeit wie folgt verstanden.

Definition Sicherheitsanforderungen: Funktionale Anforderungen, die Aussagen über die Sicherheitseigenschaften eines Prozessautomatisierungssystems machen, werden als Sicherheitsanforderungen (safety requirements) bezeichnet.

Die Automatisierungssoftware selbst besitzt kein Gefahrenpotenzial. Allerdings wird der Software unterstellt, dass diese nie völlig frei von Fehlern ist [Bell98]. Diese Fehler können sicherheitskritische Situationen im technischen System hervorrufen. In [MeRe99] werden konzeptionelle Fehler bei der Automatisierungssoftware als häufige Unfallursache aufgeführt.

Die Automatisierungssoftware besitzt nur inhärente Fehler, da sie im Gegensatz zu technischen Bauelementen nicht durch Verschleiß beeinträchtigt werden kann bzw. keine neuen Fehler im

Betrieb entstehen können [HaKo99]. Unterschieden werden Anforderungs-, Entwurfs- und Programmierfehler. Die Automatisierungssoftware kann in Bezug auf ihre Anforderungen fehlerfrei, aber im Betrieb – im Zusammenspiel zwischen technischem System und menschlichen Bedieneingriffen – für einen eingetretenen Schaden verantwortlich sein [Leve95]. In einem solchen Fall sind die Anforderungen an die Automatisierungssoftware fehlerhaft oder unvollständig. Entwurfs- und Programmierfehler entstehen während der Entwicklung der Automatisierungssoftware und führen in der Regel dazu, dass die Automatisierungssoftware die gestellten Anforderungen nicht erfüllt. Im schlimmsten Fall bleiben diese Fehler unentdeckt und treten erst im Betrieb des Prozessautomatisierungssystems durch ungewöhnliche Betriebsbedingungen auf.

2.1.3 Bedienpersonal und menschliche Bedieneingriffe

Das Aufgabenspektrum des Bedienpersonals ist ebenfalls von System zu System unterschiedlich. Bei Prozessautomatisierungssystemen lassen sich mentale Aufgaben, Überwachungs-, Kontroll- und Steuerungsaufgaben identifizieren. [Bubb92].

- Von *mentalen Aufgaben* wird gesprochen, falls auf der Basis von erlernten Sachverhalten oder Informationen, die im Gedächtnis gespeichert wurden, Entscheidungen abgeleitet oder neu aufgetretene Probleme gelöst werden. Diese erfordern ein Höchstmaß an Eigenkontrolle und sind abhängig von Ausbildung und Erfahrung des Bedienpersonals.
- *Überwachungsaufgaben* liegen dann vor, wenn der Ablauf eines technischen Prozesses fortlaufend zu prüfen und gegebenenfalls zu korrigieren ist. Die belastende Wirkung dieser Tätigkeit wird in erster Linie durch den Verantwortungsdruck und den Zwang zur Daueraufmerksamkeit geprägt, bei gleichzeitig geringer Eigenaktivität.
- *Kontrollaufgaben* sind Aufgaben, bei denen das Bedienpersonal die Qualität und Quantität eines erzeugten Produktes mit vorgegebenen Normen vergleicht und gegebenenfalls eine Produkteinstufung in verschiedene Güteklassen durchzuführen hat. Die hauptsächlichen Belastungsarten werden durch Entscheidungszwang, Zwang zur Daueraufmerksamkeit und Zeitdruck sowie Bekämpfung von Monotonieeinflüssen und ständigem Fixierungswechsel der Augen bestimmt.
- *Steuerungsaufgaben* stellen die häufigsten Tätigkeiten bei Prozessautomatisierungssystemen dar. Der Mensch greift hierbei direkt oder indirekt über die Automatisierungssoftware in den Ablauf des technischen Prozesses ein. Nach festgelegten Mustern wird meist eine optimale Nutzung des Prozessautomatisierungssystems als Ziel verfolgt. Als Belastungsarten sind bei Steuerungsaufgaben ebenfalls der Zwang zur Daueraufmerksamkeit sowie erhöhter Verantwortungsdruck zu nennen.

Die Ausführung der geschilderten Aufgaben ist von der Art und der Vertrautheit dieser Aufgaben abhängig. Nach [Rasm83] lassen sich für das Bedienpersonal in technischen Anlagen hierzu drei verschiedene Ausführungsebenen unterteilen:

- Die *fähigkeitsbasierte Ebene* betrifft in erster Linie Routine-Aufgaben. Das Bedienpersonal speichert Tätigkeiten bzw. einzelne Anweisungen in Form von Abläufen im Gedächtnis ab. Die Leistungen werden in erster Linie durch häufige Wiederholungen dieser Tätigkeiten verbessert (z. B. Autofahren). Fehler auf dieser Ebene beruhen oft auf Veränderungen der gewohnten Aufgabe bzw. Umwelt.
- Die *regelbasierte Ebene* behandelt vertraute Probleme. Der Mensch löst diese Aufgaben, indem er auf gespeicherte Regeln des Typs *„wenn (Zustand) dann (Diagnose)“* oder *„wenn (Zustand) dann (hilfreicher Bedieneingriff)“* zurückgreift. In dieser Ebene gehen Fehler typischerweise mit einer Fehleinschätzung von Situationen einher, was zur Anwendung einer falschen Regel führt.
- Die *wissensbasierte Ebene* beschreibt Handlungen in neuartigen Situationen. Die Handlungen werden auf Basis von bewussten analytischen Prozessen und gespeichertem Wissen geplant. Fehler in dieser Ebene ergeben sich aus der Beschränkung der vorhandenen Analysemittel und unvollständigem oder fehlerhaftem Wissen.

Mit wachsender Erfahrung des Bedienpersonals bewegt sich die primäre Entscheidungsfindung von der wissensbasierten zur fähigkeitsbasierten Ebene. Jedoch können auch jederzeit alle drei Ebenen nebeneinander bestehen.

Bei Prozessautomatisierungssystemen erfolgt die Beschreibungen der menschlichen Bedienhandlungen in der Regel in Form von Benutzungshandbüchern bzw. Benutzungsanweisungen. Bei nicht bestimmungsgemäßen Betriebssituationen bzw. Notfallsituationen existieren spezielle Anweisungskataloge oder Handlungsrichtlinien. Zum Beispiel befinden sich in jedem Flugzeug Handlungsrichtlinien für das Verhalten im Notfall in unmittelbarer Reichweite der Passagiere. Die Piloten verfügen über Anweisungskataloge, die vorschreiben, wie sie in Notfallsituationen zu reagieren haben.

Bei sicherheitskritischen Systemen werden meist Sicherheitsübungen absolviert, um die Reaktionszeit des Bedienpersonals zu verkürzen und die Ausführungssicherheit zu erhöhen. Dabei ist das Ziel, die Entscheidungsfindung auf eine niedrigere Ebene zu verlegen. Falsche menschliche Bedieneingriffe stellen ausnahmslos nicht-inhärente Fehler dar, da diese erst nach der Inbetriebnahme auftreten können und damit oft unvorhersehbar sind.

2.1.4 Zusammenspiel der Bestandteile

Das Zusammenspiel der drei Bestandteile technisches System, Rechnersystem und menschliche Bedieneingriffe hängt in erster Linie von der Struktur des betrachteten Prozessautomatisierungssystems ab. Die Struktur hat Auswirkungen auf den Informationsfluss zwischen den drei Bestandteilen und auf den *Automatisierungsgrad*. Je höher der Automatisierungsgrad ist, desto weniger Funktionen werden vom Bedienpersonal übernommen. Bei einem voll automatisierten

Prozess beschränken sich die Aufgaben des Bedienpersonals lediglich auf die Überwachung des Prozessgeschehens und die Wartung des gesamten Systems.

Je nach Einsatz und Aufgabe des Rechnersystems werden, wie in Abbildung 2.2 dargestellt, folgende Grundstrukturen unterschieden:

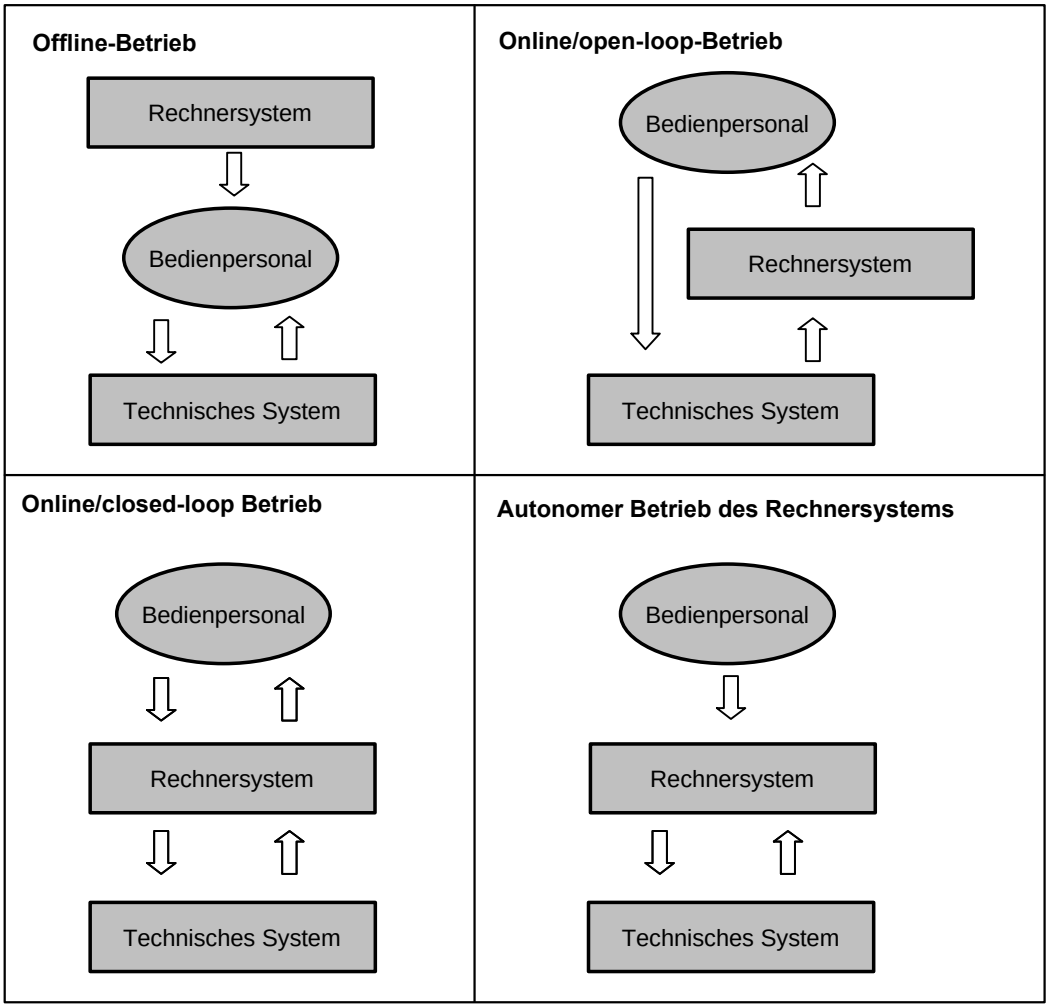


Abbildung 2.2: Grundstrukturen von Prozessautomatisierungssystemen

- *Offline-Betrieb des Rechnersystems*.
  Das Rechnersystem dient zur Beratung des Bedienpersonals, z. B. über die Berechnung zusätzlicher Kenngrößen. Die Prozessführung wird vollständig vom Bedienpersonal vollzogen.
- *Online/open-loop-Betrieb des Rechnersystems*.
  Die Aufgabe des Rechnersystems liegt in der Aufbereitung von Prozessgrößen und der Bereitstellung einer informationsorientierten Sichtweise, d.h. der Verknüpfung von vielen unterschiedlichen Mess-Signalen zu verständlichen Informationen. Die Prozessführung liegt weiterhin beim Bedienpersonal.
- *Online/closed-loop-Betrieb des Rechnersystems.*
  Diese Struktur ist bei halbautomatisierten Systemen üblich. Die Prozessführung wird zwischen Rechnersystem und Bedienpersonal aufgeteilt. Bei den meisten Anlagen und bei sicherheitskritischen Systemen wird dieser Ansatz gewählt, da das Bedienpersonal alle Aus-

führungen des Rechners überwacht und folgenschwere Entscheidungen bzw. Aufgaben selbst durchführt.

- *Autonomer Betrieb des Rechnersystems.*
  Die Prozessführung erfolgt vollständig durch das Rechnersystem. Solche Strukturen finden sich auch häufig bei der Produktautomatisierung wieder und ermöglichen einen hohen Automatisierungsgrad. Hierbei übernimmt der Rechner ebenfalls Überwachungsaufgaben. Die Aufgabe des Bedienpersonals beschränkt sich auf die Parametrisierung des Prozesses und auf high-level Aufgaben, wie zum Beispiel das An- und Ausschalten des Systems.

2.2 Sicherheit, Risiko und Gefahr

Die Begriffe *Zuverlässigkeit* und *Sicherheit* werden in der Literatur oft in ein und demselben Kontext genannt. Insbesondere im Hinblick auf Sicherheitsanalysen müssen diese Begriffe allerdings klar getrennt werden. Bei der Betrachtung der *Zuverlässigkeit* steht die Funktionalität eines Systems im Vordergrund. Nach DIN 40041 ist der Begriff Zuverlässigkeit gleichzusetzen mit der Fähigkeit eines Systems, für eine gegebene Zeit korrekt zu arbeiten. Dabei wird vorausgesetzt, dass das System zur Inbetriebnahme korrekt arbeitet und lediglich Ausfälle zu Unkorrektheit führen können. Bei der Betrachtung der *Sicherheit* geht es hingegen um die Verhinderung von Gefahren, die dem Menschen oder der Umwelt durch ein Prozessautomatisierungssystem drohen. Sicherheit ist in diesem Zusammenhang im Sinne des englischen Begriffs „safety“ aufzufassen. Unter „Security“ wird der Schutz eines Systems vor äußeren Gefahren verstanden.

Der Begriff *Sicherheit* ist untrennbar mit dem Begriff *Risiko* verbunden. Dies liegt in der Natur der Sache, das kein technisches System absolut sicher sein kann. Mit dem Betrieb ist immer ein gewisses Risiko verbunden, das toleriert werden muss. Um den Begriff Sicherheit zu präzisieren, wurde in DIN 31000 ein Konstrukt aus folgenden Begriffen definiert: *Schaden, Risiko, Grenzrisiko, Sicherheit, Gefahr* und *Schutz*. Diese Festlegung stellt die Grundlage für alle sicherheitstechnischen Betrachtungen dar und ist für das Verständnis der Arbeit von Bedeutung.

Definition Schaden: Unter *Schaden* wird im Zusammenhang mit der Definition der Sicherheit vorrangig die Beeinträchtigung von Leben und Wohl der Menschen sowie der Umwelt verstanden. Wirtschaftliche Schäden treten bei Sicherheitsaspekten in den Hintergrund.

Definition Risiko: Das *Risiko* setzt sich zusammen aus der Wahrscheinlichkeit des Auftretens des zum Schaden führenden Ereignisses und dem beim Ereigniseintritt zu erwartenden Schadenausmaß.

In VDI/VDE 3542 wurde vorgeschlagen, das Risiko als Produkt aus der Wahrscheinlichkeit oder Häufigkeit (H) eines zum Schaden führenden Ereignisses und dem Erwartungswert für das Schadenausmaß (S) zu quantifizieren.

$$R = H \cdot S \qquad (1)$$

Sowohl eigene Arbeiten [Bieg00a] wie auch andere Berichte [Litz98] [Leve95] haben jedoch gezeigt, dass diese Quantifizierung in vielen Fällen nicht möglich ist. Risiko ist in diesem Sinne eher qualitativ als quantitativ zu verstehen. Als Maßstab, um zu entscheiden, ob ein System sicher oder gefährlich ist, wird der Begriff des Grenzrisikos eingeführt.

Definition Grenzrisiko: Das Grenzrisiko ist das größte noch vertretbare Risiko eines bestimmten technischen Vorganges oder Zustands. Das Grenzrisiko ist eine Funktion des technisch-wissenschaftlichen Könnens und des wirtschaftlich Machbaren [LaGö99a].

Definition Gefahr: Wird dieses Grenzrisiko überschritten, so erfolgt der Übergang in den Bereich „Gefahr“, siehe Abbildung 2.3. Das Gefahrenpotenzial ist als drohender Schaden für Mensch und Umwelt zu verstehen.

Definition Sicherheit: Ein System ist sicher, falls das Risiko aller Einzelvorgänge kleiner als das Grenzrisiko ist.

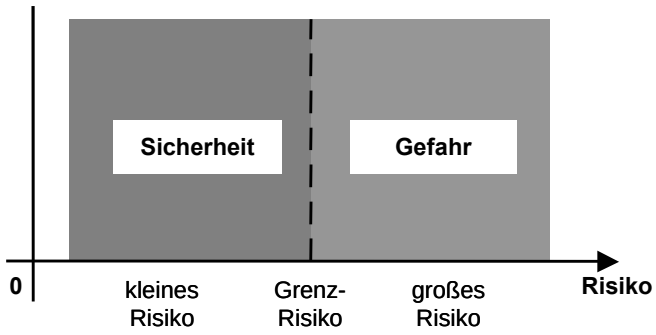


Abbildung 2.3: Zusammenhang von Sicherheit und Gefahr

Definition Schutz- und Sicherheitsmaßnahme: Die Verringerung des Risikos durch Maßnahmen, die entweder die Eintrittshäufigkeit oder das Ausmaß eines Schadens oder beides einschränken, wird als *Schutz*- bzw. *Sicherheitsmaßnahme* bezeichnet.

Für Sicherheitsanalysen aber auch für Zuverlässigkeitsanalysen ist die Betrachtung von möglichen *Fehlern* eines Systems wichtig.

Definition Fehler: Ein Fehler ist die Abweichung des tatsächlichen Verhaltens von einem vorgegebenen spezifizierten Sollverhalten. In der Regel ist dies die Abweichung eines berechneten, beobachteten oder gemessenen Werts von dem wahren, spezifizierten oder theoretisch richtigen Wert. Fehler, die schon vor der ersten Inbetriebnahme vorhanden sind, werden als inhärente Fehler bezeichnet. Nicht inhärente Fehler sind dagegen Fehler, die erst nach Inbetriebnahme auftreten bzw. begangen werden.

2.3 Aufgaben einer Sicherheitsanalyse

Durch den engen Zusammenhang der Begriffe Sicherheit, Risiko und Gefahr, vergleiche Abbildung 2.3, wird in der Literatur oft von Sicherheits-, Risiko- oder auch Gefahrenanalyse gesprochen. Für diese Arbeit wird folgende Definition einer Sicherheitsanalyse herangezogen:

Definition Sicherheitsanalyse: Im Rahmen einer *Sicherheitsanalyse* wird untersucht, ob ein Prozessautomatisierungssystem während des Betriebs jederzeit und unter allen Umständen, auch im nicht-bestimmungsgemäßen Betrieb, sicher ist bzw. keine sicherheitskritische Situation auftritt.

Definition Sicherheitskritische Situation: Eine Situation eines Prozessautomatisierungssystems wird als sicherheitskritisch bezeichnet, falls sie die Sicherheit für Mensch und Umwelt in Frage stellt.

Um Aussagen über sicherheitskritische Situationen zu treffen, müssen im Rahmen einer Sicherheitsanalyse die nachfolgenden Fragestellungen erörtert werden:

- Welche Gefahren sind in einem Prozessautomatisierungssystem enthalten?
- Wie hoch ist das Schadenausmaß, falls diese Gefahren eintreten?
- Wie wahrscheinlich ist das tatsächliche Eintreten dieser Gefahren?

Die beiden letzten Fragen beziehen sich auf das verbleibende Risiko beim Betrieb eines Prozessautomatisierungssystems. Die Identifizierung und Untersuchung des Gefahrenpotenzials eines Systems wird oft auch als Gefahrenanalyse bezeichnet. Im Rahmen einer Sicherheitsanalyse findet eine Beschreibung des Systems bzw. dessen Bestandteile statt. Es ist zu erörtern, ob und unter welchen Umständen die identifizierten Gefahren auftreten können. Zusätzlich werden der nicht-bestimmungsgemäße Betrieb eines Systems betrachtet und die möglichen Folgen von Fehlern abgeschätzt. Führen diese zu einem sicherheitskritischen Betrieb oder gar zu einem Unfall, so müssen Maßnahmen getroffen werden, um diese zu verhindern.

Eine Sicherheitsanalyse wird sowohl bei der Entwicklung von Systemen mit Sicherheitsverantwortung, wie auch für die Bewertung bestehender Systeme angewendet. Generell gilt: Je früher sicherheitskritische Situationen erkannt werden, desto geringer ist der notwendige Aufwand für die Korrektur. Die Vorgehensweise einer Sicherheitsanalyse und die zugrunde liegenden Verfahren sind nicht standardisiert, es muss lediglich gezeigt werden, dass eine systematische Vorgehensmethode angewendet wurde [Lauf96]. Der nächste Abschnitt geht daher kurz auf die gegenwärtige Situation bei gesetzlichen Bestimmungen und Normen ein.

2.4 Gesetzliche Bestimmungen und Normen

Artikel 2, Absatz 2 des Grundgesetzes „Recht auf körperliche Unversehrtheit“ schützt Leib und Leben einzelner Individuen bzw. der Allgemeinheit und stellt die Basis einiger Gesetze, Verordnungen und Richtlinien für technische Systeme dar. Für Prozessautomatisierungssysteme gilt allgemein, dass diese Gesetze, Verordnungen und Richtlinien in den seltensten Fällen bereits die endgültige Problemlösung beschreiben. Sie geben vielmehr den Rahmen vor, innerhalb dessen sich die Lösung bewegen soll, d.h. es besteht meistens ein großer Gestaltungsspielraum. Erschwerend wirkt sich das vielfältige, interdisziplinäre Anwendungsgebiet der Prozessautomatisierungstechnik auf die Erstellung von klaren allgemeingültigen Vorschriften aus. Bei der Entwicklung von verfahrenstechnischen Anlagen gilt z. B. das Bundes-Immissionsschutzgesetz, insbesondere die Störfallverordnung. Hinweise für Aufzugsanlagen sind in der Aufzugsverordnung des Gerätesicherheitsgesetzes zu finden und beim Schienenverkehr spielt das allgemeine Eisenbahngesetz eine Rolle. Eine gute Übersicht über Gesetze und Richtlinien in der Automatisierungstechnik bietet [Beck97]. Hersteller von automatisierten Produkten bzw. Anlagen besitzen in der Regel eine Vielzahl von internen Richtlinien. Es ist zu beobachten, dass der Grad der Präzisierung vom Grundgesetz über Gesetze und Richtlinien spezieller Einrichtungen bis hin zu den firmen-internen Richtlinien zunimmt. Institutionen wie NAMUR, VDI/VDE und DIN erstellen ihrerseits wiederum aus den vorhandenen Richtlinien und Gesetzen insbesondere in Zusammenarbeit mit erfahrenen Experten (Fachkreisen) Empfehlungen und Richtlinien – die technischen Normen. Für die Sicherheit von Prozessautomatisierungssystemen sind die technischen Normen VDI/VDE-Richtlinie 3542 (die im Abschnitt Begriffsbestimmung häufig zitiert wurde), DIN 31000 und die internationale Sicherheitsnorm IEC 61508 zu nennen. Die technischen Normen sind prinzipiell rechtlich nicht bindend, außer sie werden in Gesetzen und Verordnungen zitiert, stellen aber anerkannte Regeln der Technik dar. Es ist festzustellen, dass keine allgemeine Richtlinie zur Sicherheitsanalyse von Prozessautomatisierungssystemen existiert.

In diesem Kapitel wurden die wichtigsten Begriffe, Aufgaben und gesetzlichen Bestimmungen für Sicherheitsanalysen vorgestellt. Es ist hervorzuheben, dass eine absolute Sicherheit beim Betrieb eines Prozessautomatisierungssystems nicht existiert. Wird allerdings ein vertretbares Grenzrisiko nicht überschritten, so gilt das Prozessautomatisierungssystem als sicher. Zusammenfassend ist die Bedeutung der Bestandteile eines Prozessautomatisierungssystems im Hinblick auf die Sicherheit zu erwähnen. Der technische Prozess läuft in einem technischen System ab und enthält das Gefahrenpotenzial, welches durch Fehler im technischen System oder/und durch Fehler in der Automatisierungssoftware oder/und durch falsche Bedieneingriffe oder durch ein nicht vorhergesehenes Zusammenspiel der Bestandteile zu einem Schaden für Mensch und Umwelt führen kann.

3 Sicherheitsanalyse bei Prozessautomatisierungssystemen

In diesem Kapitel wird ein Überblick über unterschiedliche Verfahren zur Sicherheitsanalyse gegeben und anschließend Modellkonzepte behandelt, auf denen die vorgestellten Verfahren beruhen und Modellkonzepte, für die Beschreibung von Automatisierungssystemen. Basierend auf den gewonnenen Erkenntnissen werden Folgerungen für eine modellbasierte Sicherheitsanalyse von Prozessautomatisierungssystemen abgeleitet, aus denen dann konkrete Anforderungen an das zu entwickelnde Konzept formuliert werden.

3.1 Verfahren zur Sicherheitsanalyse

In [LaGö99a], [Leve95], [Bish90], [Pilz85] und [Dech87] werden verschiedene Verfahren für die Sicherheitsanalyse vorgestellt und diskutiert. Die Auswahl der im Rahmen dieses Kapitels vorgestellten Verfahren richtet sich nach zwei Aspekten: nach der Relevanz ihres Einsatzes (verbreitete Verfahren) und insbesondere nach der Art des zugrunde liegenden Modells. Verfahren, die die Untersuchung von Sicherheitsanforderungen zum Ziel haben, werden ebenfalls berücksichtigt.

Die Verfahren werden anhand eines kleinen Beispiels „Hochdruckanlage“ vorgestellt. In Abbildung 3.1 ist das System in Form eines vereinfachten RI-Fließbilds dargestellt. Es besteht aus den Systemelementen Verdichter, Druckbehälter, Überdruck- und Auslassventil. Mit Hilfe eines einfachen Steuerungsprogramms wird der Druck des Behälters auf einen gewünschten, modifizierbaren Solldruck eingestellt. Falls der Druck im Druckbehälter sinkt, so wird der Verdichter eingeschaltet und dadurch Druck aufgebaut. Die Entnahme der komprimierten Luft erfolgt mit dem Auslassventil. Das Überdruckventil öffnet automatisch, falls der Druck im Druckbehälter einen kritischen Wert erreicht.

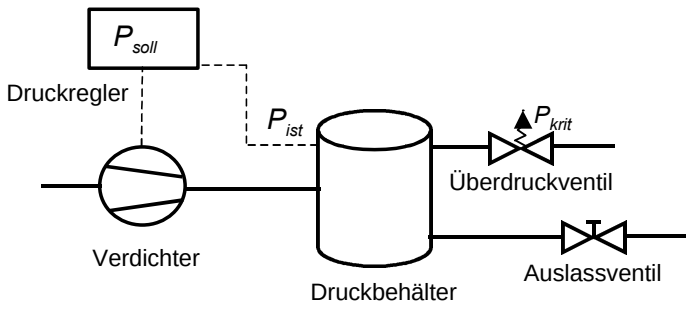


Abbildung 3.1: Schematische Darstellung des Systems „Hochdruckanlage“

3.1.1 PAAG-Verfahren

Das Verfahren zur Prognose, Auffinden der Ursachen, Abschätzen der Auswirkungen und Gegenmaßnahmen (PAAG) ist im englischen Sprachraum auch als HAZOP bekannt und wird insbesondere in Deutschland für die Sicherheitsanalyse von verfahrenstechnischen Prozessen angewendet [BHR90]. Durch die Anwendung eines festen Satzes von Leitwörtern[1] auf Soll-Funktionen werden unerwünschte Abweichungen und deren Auswirkungen ermittelt. Auf diese Weise können neue sicherheitskritische Systemsituationen ohne Fokussierung auf ein bestimmtes Ereignis aufgefunden werden [Dech87]. Für Abweichungen, die relevante unerwünschte Auswirkungen haben, müssen wirksame Gegenmaßnahmen entwickelt werden. Tabelle 3.1 zeigt beispielhaft die Anwendung des PAAG-Verfahrens für die Hochdruckanlage aus Abbildung 3.1 anhand des Leitwortes MEHR.

Tabelle 3.1: Beispielhafte Dokumentation beim PAAG-Verfahren

System		Hochdruckanlage	
Funktionsteil		Verdichter	
Unterfunktion		Druck halten	
Sollfunktion		Druckluft erzeugen	
Leitwort	Ursachen	Auswirkungen	Maßnahmen
MEHR kompressieren	Pumpe schaltet nicht ab	Druck steigt	Überdruckventil installieren

Der Einsatz des PAAG-Verfahrens wird als sehr zeitintensiv und aufwändig bewertet [Bish90]. Die Qualität des Ergebnisses ist in großem Maße abhängig von dem Fachwissen und der Erfahrung der Anwender, da die Analyse auf Brainstorming beruht. Mit diesem Verfahren werden hauptsächlich die Bauelemente des technischen Systems analysiert.

3.1.2 FMEA-Verfahren

Die Fehlermöglichkeits- und Einfluss-Analyse (FMEA) ist, insbesondere in der Automobilindustrie, ein weit verbreitetes Verfahren zur Untersuchung der Zuverlässigkeit und Sicherheit eines Systems [BEE96]. Die Ziele der FMEA sind eine frühestmögliche Erkennung von kritischen Systemelementen und Schwachstellen, eine Einschätzung der Risiken und eine daraus abgeleitete Risikominimierung sowie eine Erhöhung der Klarheit über den Aufbau eines Systems bzw. Produktes [MüTi00]. Das Vorgehen ist in DIN 25448 standardisiert.

---

[1] PAAG-Leitwörter: NEIN, JA, MEHR, WENIGER, SOWOHL ALS AUCH, TEILWEISE, ANDERS ALS, UMKEHRUNG.

Besonderes Merkmal der FMEA ist die Strukturanalyse in Form eines Top-Down-Vorgehens. Die Soll-Funktionen von identifizierten Subsystemen bzw. Systemelementen werden in einem Hierarchiebaum schriftlich festgehalten. Basierend auf dieser Funktionsbeschreibung werden Ausfälle oder verschiedene Abweichungen von den Soll-Funktionen beschrieben. Die möglichen Fehlerfolgen sind die sich ergebenden Fehlfunktionen der übergeordneten Systemelemente. Diese werden durch ein Bottom-Up-Vorgehen ermittelt. Die potenziellen Ursachen sind die denkbaren Fehlfunktionen der untergeordneten Systemelemente. Das in Abbildung 3.2 dargestellte Beispiel vereint Struktur-, Funktions- und Fehlerbaum einer FMEA-Analyse. Die Ergebnisse der Analyse werden in einem Formblatt dargestellt [DIN 25448].

Die Strukturanalyse des FMEA-Verfahrens erlaubt eine systematische Vorgehensweise. Dadurch wird, wie beim PAAG-Verfahren, das Systemverständnis gefördert. Bei der Sicherheitsanalyse werden allerdings nur die Auswirkungen von einzelnen Fehlern betrachtet. Mit Hilfe des FMEA-Verfahrens werden in erster Linie Elemente des technischen Systems untersucht. Bedienfehler werden meistens nicht in die Analyse einbezogen [Leve95]. Der Analyseweg selbst wird nicht festgehalten, daher ist es für nichtbeteiligte Personen oft schwierig, das Ergebnis nachzuvollziehen bzw. zu reproduzieren. Das FMEA-Verfahren ist aufwändig und zeitintensiv [Bish90].

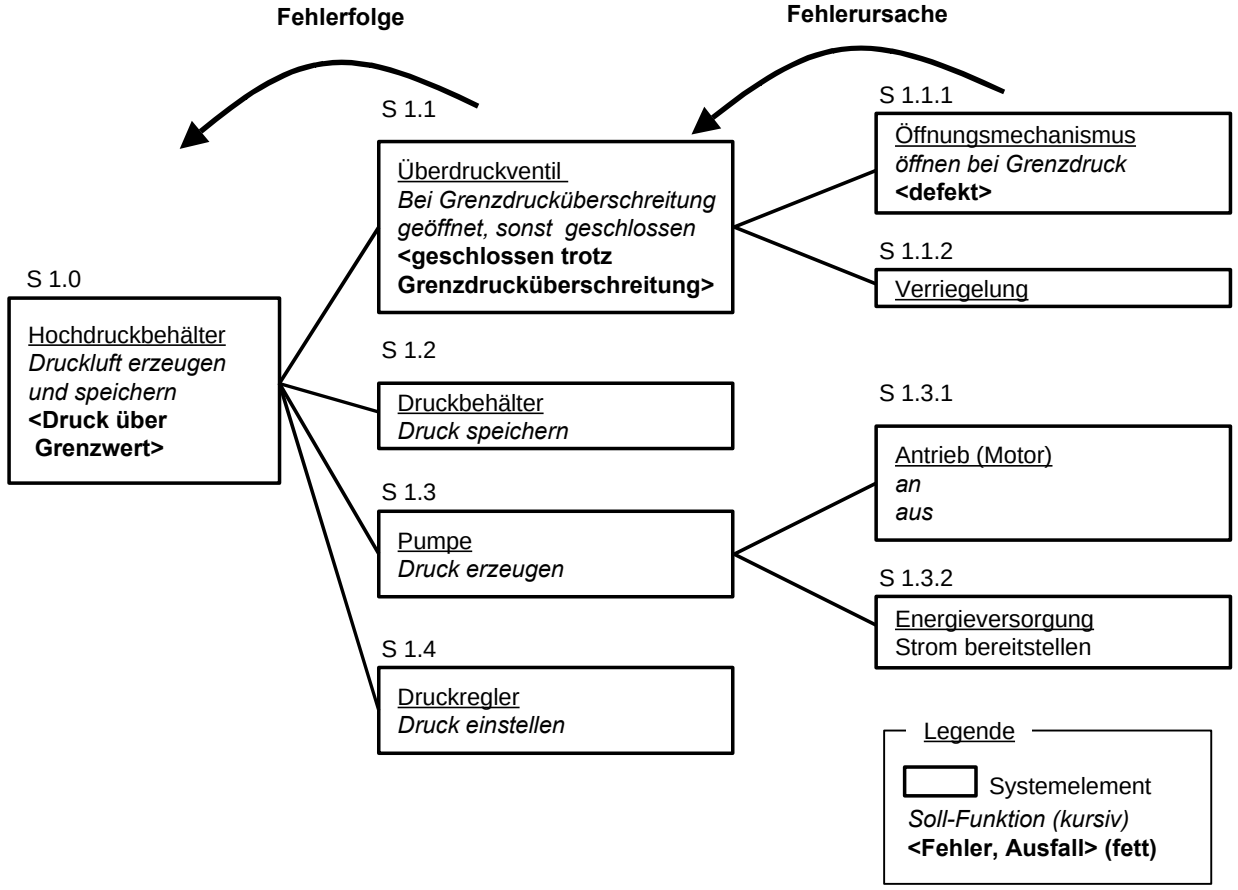


Abbildung 3.2: Beispielhafte Anwendung des FMEA-Verfahrens

3.1.3 FTA-Verfahren

Ziel der Fehlerbaumanalyse (FTA) ist das Auffinden von Ereigniskombinationen, die zu einem Fehlerereignis führen [DIN 25424]. Die Konstruktion eines Fehlerbaums dient als Hilfsmittel. Dabei handelt es sich um einen gerichteten Graphen, der die Zusammenhänge zwischen einem Top-Ereignis und seinen Ursachen aufzeigt. Das Top-Ereignis stellt in der Regel einen unerwünschten Vorgang bzw. eine Gefahr dar und bestimmt das Analyseziel des Verfahrens. Der Fehlerbaum besteht aus Schichten von Ereignissen, die durch verschiedene logische Operationen verknüpft werden. Zu Ereignissen zählen hierbei Fehler, die von einander unabhängig sein müssen. Der Fehlerbaum wird solange erweitert, bis an seinen Spitzen nur noch Grundereignisse stehen. Es existieren keinerlei Vorgaben, wie detailliert eine Fehlerbaumanalyse vorgenommen werden soll. In Abbildung 3.3 wird ein Beispiel für einen Fehlerbaum aufgezeigt.

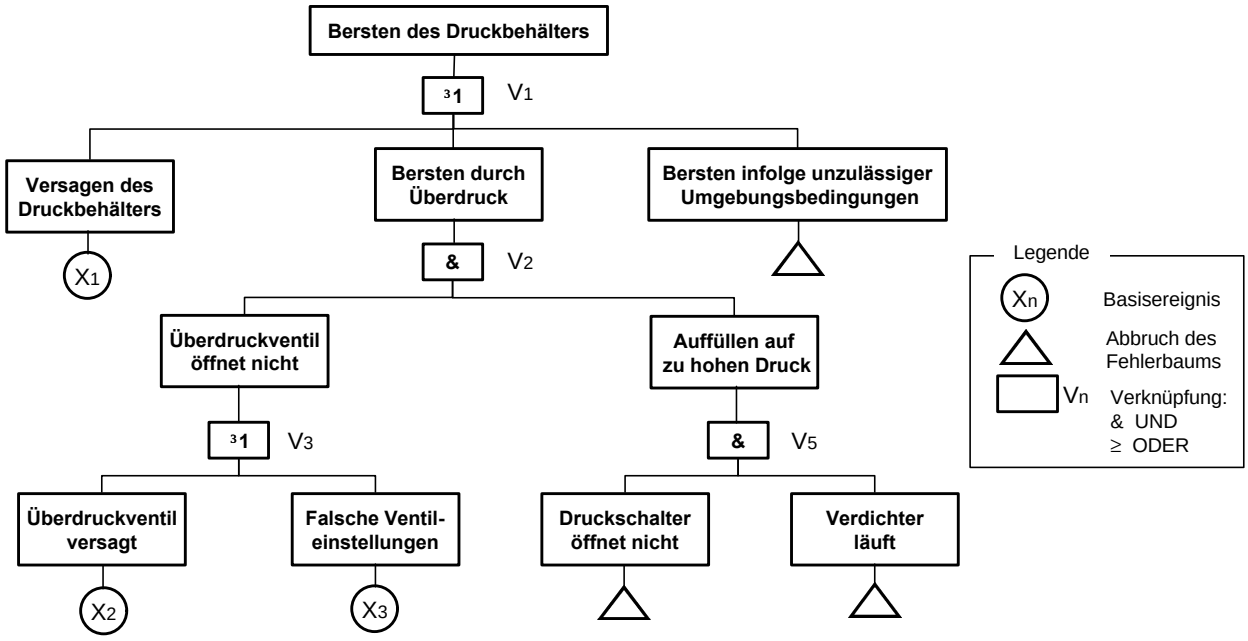


Abbildung 3.3: Anwendung eines Fehlerbaums nach DIN 25424

Bei der qualitativen Auswertung des Fehlerbaums werden Minimalschnitte (cutsets) ermittelt. Ein Schnitt ist definiert als eine Menge von Ereignissen, die zum Top-Ereignis führen. Aus dieser Menge leiten sich die Minimalschnitte ab, die nur die unbedingt notwendige Anzahl von Ereignissen zum Erreichen des Top-Ereignisses enthalten. Der kritische Pfad ist der kürzeste Minimalschnitt und stellt zumeist die Schwachstelle des Systems dar.

Bei der quantitativen Analyse des Fehlerbaums wird die Auftrittswahrscheinlichkeit eines Top-Ereignisses ermittelt. Dazu werden die Minimalschnitte betrachtet und alle Auftrittwahrscheinlichkeiten der Ereignismengen summiert, um damit die Auftrittwahrscheinlichkeit des Top-Ereignisses zu bestimmen.

Das Verfahren der Fehlerbaumanalyse dient nur dazu, die Fehlerursachen und die Wahrscheinlichkeit für das Eintreten eines gut bekannten Ereignisses zu bestimmen. Die Analyse ist nicht zur Bestimmung der allgemeinen Fehleranfälligkeit eines Systems geeignet [Bish90]. Die Vorteile dieser Methode liegen in der übersichtlichen graphischen Aufbereitung der Zusammenhänge von Ursachen eines Fehlers. Sie eignet sich gut, um Szenarien (Snapshots) einer Anlage zu beschreiben [Leve95]. Allerdings ist für einen effektiven Einsatz der Methode detailliertes Wissen über den Entwurf der Anlage notwendig. Daher kann die Fehlerbaumanalyse erst in späten Entwicklungsphasen eingesetzt werden. Der Fehlerbaum kann prinzipiell für alle Bestandteile eines Prozessautomatisierungssystems erstellt werden. Das komplexe Zusammenspiel zwischen den Bestandteilen muss dem Anwender genau bekannt sein, um Zusammenhänge in Baumform darzustellen. Eine Beschreibung der Struktur eines Prozessautomatisierungssystems ist nicht möglich.

3.1.4 ETA-Verfahren

Während die Fehlerbaumanalyse Ursachen für ein bestimmtes Ereignis ermittelt, verfolgt die Ereignisablaufanalyse (Event Tree Analysis – ETA) genau den umgekehrten Weg. Für ein spezielles Ereignis werden die Auswirkungen bzw. die Folgen analysiert [DIN 25419].

Die Analyseergebnisse werden in einem Ereignisablaufdiagramm graphisch dargestellt. Zuerst wird das Anfangsereignis definiert. Das Anfangsereignis kann z. B. ein Ausfall eines Systemelements oder eine Fehlbedienung in einem technischen System sein. Dieses Anfangsereignis hat eine bestimmte Auswirkung auf das betrachtete System. Die Auswirkungen werden im Ereignisdiagramm durch Linien symbolisch dargestellt und rufen eventuell weitere kausale Folgewirkungen hervor, z. B. den Ausfall eines weiteren Systemelements, siehe Abbildung 3.4.

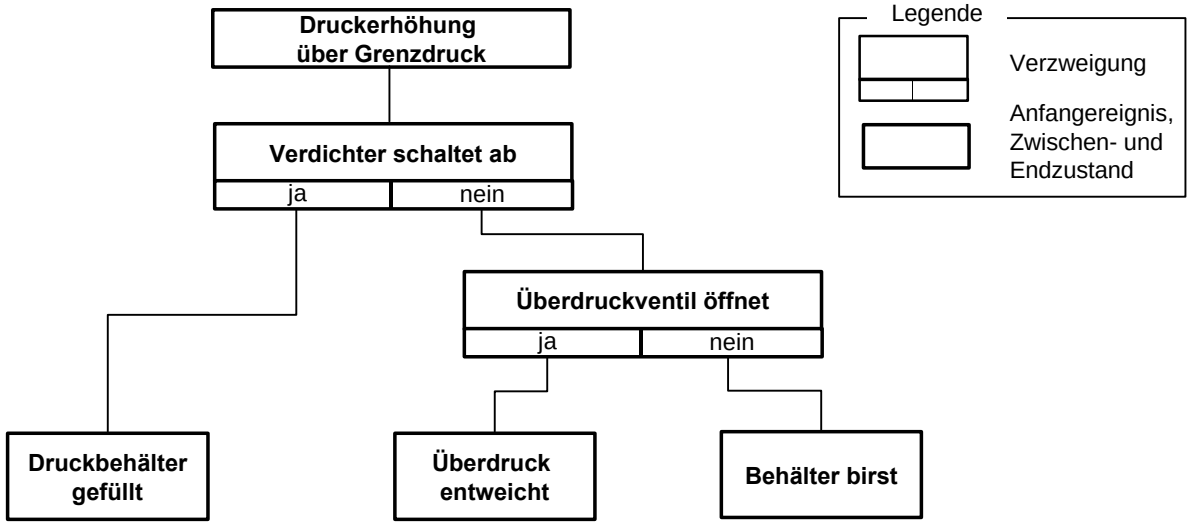


Abbildung 3.4: Anwendung eines Ereignisablaufdiagramms nach DIN 25419

Die Folgewirkung auf weitere Systemelemente wird in Form von Verzweigungen dargestellt. Diese Folgewirkung stellt wiederum das Eingangsereignis für das nächste Systemelement dar. Die Wirkungen werden soweit verfolgt, bis alle Systemelemente abgefragt sind. Für jeden Pfad des Ereignisablaufdiagramms kann eine Wahrscheinlichkeitsbewertung vorgenommen werden. Dabei handelt es sich in der Regel um Schätzwerte [Cros92]. Die Berechnung der Wahrscheinlichkeit eines Pfades erfolgt über die Multiplikation aller Wahrscheinlichkeitswerte der betroffenen Zweige.

Das Ereignisablaufdiagramm kann schon für kleine Systeme sehr schnell unübersichtlich werden [Cros92]. Die Anordnung der Ereignisse ist meist nicht eindeutig. Für die Konstruktion des Ereignisablaufdiagramms ist ein detailliertes Wissen über die Funktionsweise aller Systemelemente und deren mögliche Interaktionen notwendig. Der Bearbeiter muss sich alle Interaktionen vor Augen halten. Nicht selten werden bei der Konstruktion des Ereignisablaufdiagramms Fehler gemacht und wichtige Ereignisse vergessen [NoRa90].

3.1.5 THERP-Verfahren

THERP ist ein verbreitetes Verfahren zur Abschätzung und Bewertung des menschlichen Verhaltens in technischen Anlagen [Strä00]. Die wichtigste Voraussetzung für eine effektive Anwendung dieses Verfahrens liegt in einer gründlichen System- und Aufgabenanalyse. In der Systemanalyse werden die relevanten Systemfunktionen und die möglichen Handlungen des Bedienungspersonals ermittelt. Anschließend folgt eine Schätzung der entsprechenden Wahrscheinlichkeiten für menschliches Fehlverhalten und dessen Auswirkung auf die Sicherheit und Zuverlässigkeit des gesamten Systems.

Als analytisches Werkzeug findet ein Wahrscheinlichkeitsbaum Verwendung. Die Äste stellen darin die binären Entscheidungspunkte dar, an denen nur zwischen der korrekten oder der fehlerhaften Ausführung gewählt werden kann. Jeder Ast ist eine Kombination aus menschlichen Tätigkeiten und deren mutmaßlichen Einflüssen.

Das Herzstück von THERP sind 27 Tabellen mit möglichem menschlichen Fehlverhalten, siehe Tabelle 3.2. Die Werte in den Tabellen beziehen sich auf nominelle menschliche Fehlerwahrscheinlichkeiten. Es handelt sich um generische Werte, die auf Expertenmeinungen beruhen.

Die Stärken des Verfahrens liegen in einer gründlichen System- und Aufgabenanalyse und in einem guten Dokumentationskonzept. THERP eignet sich gut für die Abschätzung menschlicher Zuverlässigkeit bei Routinearbeiten [Reas94].

Der verwendete Wahrscheinlichkeitsbaum lässt allerdings nur eine stark vereinfachte Beschreibung menschlicher Bedieneingriffe zu. Außerdem müssen Handlungen in einzelne Aufgabenelemente aufgespaltet werden. Die Darstellung der Abhängigkeiten dieser Aufgabenelemente untereinander ist oft schwer erkennbar. Beträchtliche Unsicherheiten werden zusätzlich durch

den Mangel an Daten aus realer Beobachtung verursacht. Die Methode kann nur von Spezialisten sinnvoll angewandt werden [Reas94], [Bubb92].

Tabelle 3.2: Auszug aus THERP-Tabellen

Aufgabenart	**Fehlerwahrscheinlichkeit**
Ablesen von Zahlenwerten	
einstellige Zahlen	0,0002
zweistellige Zahlen	0,0006
Analoganzeige	0,003
Werten einer grafischen Darstellung	0,01
Erkennen, dass eine Instrumentenanzeige fehlerhaft ist	0,1
Einfachreaktion auf ein blinkendes Licht	0,0001

3.1.6 SQMA-Verfahren

SQMA wurde für die Gefahrenanalyse und Überwachung von technischen Systemen entwickelt [Lauf96], [Fröh97]. Charakteristisch für SQMA ist die Erstellung eines qualitativen Modells für das technische System [LaGö99a]. Dazu werden dessen Bauelemente separat und unabhängig von ihrer speziellen Verwendung modelliert. Wichtige Prozessgrößen der Bauelemente werden mit Hilfe von qualitativen Intervallvariablen beschrieben. Den Intervallen können Kommentare zugeordnet werden. Anhand der Struktur des technischen Systems werden die beschriebenen Bauelemente zu einem qualitativen Modell zusammengefügt. Mit Hilfe des Modells kann ermittelt werden, welche möglichen Situationen das technische System einnehmen kann und welche Übergänge zwischen diesen Situationen bestehen. Situationen sind bezüglich ihrer Bedeutung für die Sicherheit klassifiziert. Das Ergebnis wird in Form einer Tabelle ausgegeben, siehe Tabelle 3.3.

Tabelle 3.3: Ausschnitt aus der Ergebnisdarstellung des SQMA-Verfahrens

Verdichter	**Druckbehälter**	**Überdruckventil**	**Status**
..	..	..	..
an	drucklos	zu	bestimmungsgemäß
an	überdruck	verklemmt	*kritisch*
an	überdruck	auf	bestimmungsgemäß
..	..	..	

Der Vorteil des SQMA-Verfahrens liegt in der Beschreibung des Prozessgeschehens in Form eines analytischen, qualitativen Modells. Mit Hilfe von Computern können die Auswirkungen von mehreren Fehlern untersucht werden. Das Anwendungsgebiet von SQMA ist auf technische Systeme begrenzt und vernachlässigt das Zusammenspiel der Bestandteile.

3.1.7 Formale Methoden

Unter einer formalen Methode wird die eindeutige Beschreibung von Systemen mit Hilfe von mathematischen Methoden verstanden. Ziel ist die Überprüfung der Konsistenz zwischen streng formal beschriebenen Sicherheitsanforderungen und streng formal beschriebenen Systemverhalten in Form eines mathematischen Beweises. Das Vorgehen ist bei allen formalen Methoden einheitlich. Es müssen zum einen die Sicherheitsanforderungen mit einer mathematischen Sprache (Logik) und zum anderen das zu erwartende bzw. aus dem Entwurf abgeleitete Systemverhalten beschrieben werden. Die Beweisführung, d.h. die Konsistenzprüfung zwischen den Sicherheitsanforderungen und dem Systemverhalten, erfolgt rechnergestützt durch mathematische Beweisführung und wird „formale Verifikation“ genannt.

Das Systemverhalten wird bei den meisten formalen Methoden in Form von endlichen Automaten bzw. Automatenvarianten modelliert. Einige formale Methoden besitzen höhere Beschreibungssprachen, wie z. B. PROMELA [Holz97] und VSE-SPEC [VSE95].

Formale Beschreibungen der Sicherheitsanforderungen bestehen in der Regel aus einem Satz logisch verknüpfter Bedingungen. Ein typisches Beispiel für eine derartige Bedingung ist: „Falls der Versorgungsdruck über einen Grenzdruck steigt, dann ist der Kompressor auszuschalten oder das System komplett zu entlüften“. Die wichtigste Erweiterung von logikbasierten Sprachen zur Beschreibung von Sicherheitsanforderungen ist die Compution Tree Logik (CTL) [ClKu96]. Mit der CTL lassen sich Zusammenhänge zwischen verschiedenen Systemzuständen und damit ganze Abläufe oder auch nur einzelne Sequenzen beschreiben. In Abbildung 3.5 ist die Überführung eines Zustandsautomaten in einen CTL-Baum zu sehen.

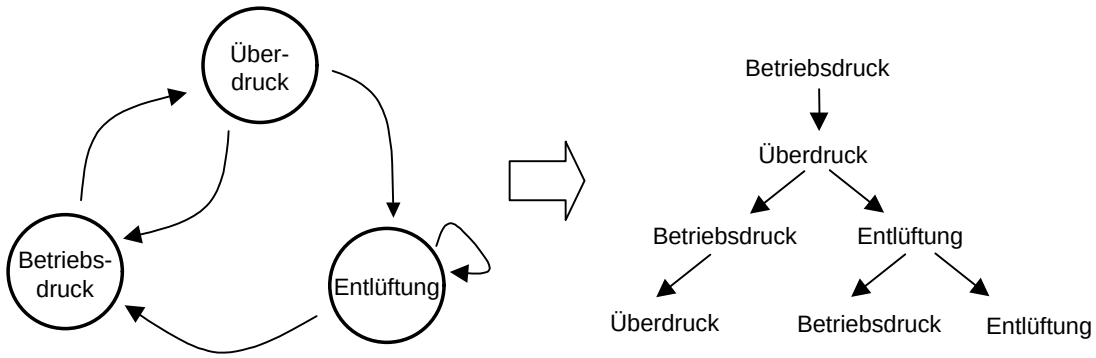


Abbildung 3.5: Exploration eines Zustandsautomaten

Die Beweisführung erfolgt rechnergestützt. Zu den wichtigsten formalen Techniken gehören die deduktive und die algorithmische Verifikation. Bei ersterem handelt es sich um deduktive Schlussfolgerungen, bei denen die Gültigkeit von Sicherheitseigenschaften anhand aufgestellter Regeln geprüft wird. Zu den Vertretern gehört der Theorem-Beweiser VSE [Göhn95]. Beim Modelchecking wird ein Algorithmus zur automatischen Verifikation von

Sicherheitsanforderungen eingesetzt. Zu den wichtigsten Vertretern gehören SMV [McMi92], KRONOS [Yovi97], HyTech [HHW97], UPPAAL [LPY97] und SPIN [Holz97].

Der richtige Einsatz von formalen Methoden gewährleistet eine mathematisch korrekte Überprüfung von Sicherheitsanforderungen. Der Einsatz von komplexen mathematischen Sprachen erscheint vielen Anwendern als schwierig bzw. als „nicht-ingenieurgerecht“ [Moik97], [Leve95]. Der Aufwand für die formale Verifikation ist schon bei relativ kleinen Programmen sehr hoch [GeCr94]. Im Gegensatz zu anderen Modellierungsmethoden sind Modellierungsfehler wahrscheinlicher. Erklärt wird dies durch die Komplexität der anspruchsvollen Beschreibungssprachen [Rush93]. Für Sicherheitsanalysen reicht es nicht aus, nur die Konsistenz zwischen den Sicherheitsanforderungen und dem Programmverhalten nachzuweisen, da viele sicherheitsrelevante Fehler schon in den Anforderungen enthalten sind [Leve95].

3.1.8 Zusammenfassung der diskutierten Verfahren zur Sicherheitsanalyse

In Tabelle 3.4 sind die diskutierten Verfahren zur Sicherheitsanalyse zusammenfassend dargestellt. Die Tabelle enthält folgende Vergleichskriterien:

- Das *Analyseziel* gibt den Analyseschwerpunkt des Verfahrens an, da die Sicherheit eines Systems aus unterschiedlichen Sichtweisen oder Randbedingungen untersucht werden kann.

- Beim *Vorgehen* des Verfahrens wird zwischen induktivem und deduktivem Vorgehen unterschieden. Beim induktiven Vorgehen handelt es sich um eine Vorwärtssuche, d.h. ausgehend von einem speziellen Systemzustand oder Ereignis werden mögliche Auswirkungen ermittelt und bewertet. Beim deduktiven Vorgehen wird in der Regel von einem kritischen Systemzustand oder Ereignis ausgegangen und die dazugehörige Ursache ermittelt. Zusätzlich werden Art und Weise, wie die Analyseergebnisse gewonnen werden, angegeben. Die Mehrzahl der Verfahren verwendet zur Analyse einen Brainstorming-Prozess; nur in wenigen Fällen findet eine analytische modellbasierte Ermittlung der Ergebnisse statt.

- Die *Bewertungsart* unterscheidet zwischen rein qualitativen und quantitativen Ergebnissen. Bei einer quantitativen Bewertung ist die entsprechende Kenngröße angegeben.

- Liegt einem Verfahren zur Durchführung oder Beschreibung ein Modell zu Grunde, so wird die Art des Modells unter dem Vergleichspunkt *Modelltyp* aufgeführt.

- Bei den eingesetzten Modellen ist der *Modellzweck* zu berücksichtigen. Modelle werden zum einen eingesetzt, um die Ergebnisse einer Analyse übersichtlich festzuhalten und quantitativ analysieren zu können. Zum anderen können zur Durchführung der eigentlichen Analyse Modelle eingesetzt werden, die das Verhalten des zu untersuchenden Systems wiedergeben.

- Der *Anwendungsbereich* des Verfahrens bezieht sich auf die unterschiedlichen Bestandteile eines Prozessautomatisierungssystems. Das primäre Anwendungsgebiet ist durch ein Pluszeichen („+“) gekennzeichnet. Ein Minuszeichen („-“) gibt an, dass das Verfahren für diesen Anwendungsbereich ungeeignet ist. Eine Null symbolisiert, dass der Einsatz mit wesentlichen Einschränkungen, wie z. B. starke Abstraktion, verbunden ist.

Tabelle 3.4: Übersicht über Methoden zur Sicherheitsanalyse

Legende: X trifft zu - triff nicht zu	**PAAG**	**FMEA**	**FTA**	**ETA**	**THERP**	**SQMA**	**Formale Methoden**
Analyseziel							
Ermittlung von gefährlichen Betriebssituationen	X	-	-	X	X	X	-
Ermittlung von gefährlichen Ereignissen	X	X	X	-	X	-	-
Ermittlung von kritischen Systemelementen	-	X	-	-	-	X	-
Prüfung von Sicherheitsanforderungen	-	-	-	-	-	-	X
Analyse von Einzelfehlern	X	X	X	X	X	X	-
Analyse von Mehrfachfehlern	-	-	-	-	-	X	-
Vorgehen							
induktiv	X	X	-	X	-	X	-
deduktiv	-	-	X	-	X	-	X
Brainstorming Prozess	X	X	X	X	X	-	-
Modellbasierte Analyse	-	-	-	-	-	X	X
Analyseergebnis							
quantitativ	-	X	X	X	X	-	X
Wahrscheinlichkeitsangaben	-	X	X	X	X	-	-
Risikobewertung	-	X	-	-	-	-	-
qualitativ	X	X	X	X	X	X	-
Verwendetes Modell							
Baumstruktur (Graph)	-	X	X	X	X	-	-
Zustandsautomat	-	-	-	-	-	-	X
Qualitative Modellierung	-	-	-	-	-	X	-
Modellzweck							
Ergebnisrepräsentation	-	X	X	X	X	-	-
Quantitative Auswertung	-	X	X	X	X	-	-
Durchführung der Analyse	-	-	-	-	-	X	X
Anwendungsbereich							
Technisches System	+	+	+	+	-	+	-
Automatisierungssoftware	-	0	0	0	-	0	+
Menschliche Bedieneingriffe	0	0	+	0	+	0	-

Aus der Tabelle 3.4 geht hervor, dass eine ganzheitliche Analyse eines Prozessautomatisierungssystems nur durch Anwendung verschiedener Verfahren möglich ist. Das Analyseergebnis setzt sich dabei aus der Summe der Ergebnisse von Einzelanalysen zusammen. Komplexe Abhängigkeiten zwischen technischem System, Automatisierungssoftware und menschlichen Be-

dieneingriffen können auf diese Weise nur unzureichend erfasst werden. Kombinationen von Fehlern in den unterschiedlichen Bestandteilen eines Prozessautomatisierungssystems stellen aber gerade die Hauptursachen vieler schwerer Unfälle dar [Leve95], [Mont00] und [Mock01]. Häufig wird eine Sicherheitsanalyse des technischen Systems lediglich unter Betrachtung der vorhandenen Schnittstellen durchgeführt [Leve95].

Wünschenswert ist eine ganzheitliche Sicherheitsanalyse von Prozessautomatisierungssystemen, um das dynamische Zusammenspiel der Bestandteile auch unter der Annahme von einem oder mehreren Fehlern zu untersuchen. Bei den meisten Verfahren wird Brainstorming als qualitative Analysetechnik verwendet und die Ergebnisse in Form einer Baumstruktur dargestellt. Aufgrund dieser Tatsache können solche Verfahren nur für kleinere Probleme angewandt werden und sind für die Betrachtung eines ganzheitlichen Ansatzes ohne Bedeutung. Dennoch ist festzustellen, dass Baumstrukturen (bzw. Graphen) bei vielen Verfahren zur Sicherheitsanalyse zum Einsatz kommen. Rein modellbasierte Verfahren, wie das Verfahren SQMA oder allgemein formale Methoden, verwenden Modelle, die für die Verhaltensbeschreibung eines speziellen Bestandteils entwickelt wurden. Es stellen sich folgende Fragen:

- Eignen sich die verwendeten Modelle ebenfalls zur Realisierung einer ganzheitlichen Sicherheitsanalyse?
- Welche Anforderungen muss ein Modell generell für dieses Ziel erfüllen?
- Warum werden Graphen derart häufig bei Sicherheitsanalysen verwendet?
- Eignen sich diese ebenfalls für eine ganzheitliche Beschreibung eines Prozessautomatisierungssystems?

Zur Beantwortung dieser Fragen, wird im nächsten Kapitel auf die unterschiedlichen Eigenschaften und Zwecke von Modellen für die Sicherheitsanalyse und darüber hinaus auf Modelle zur Beschreibung von Automatisierungssystemen eingegangen.

3.2 Modellierungsverfahren zur einheitlichen Beschreibung von Prozessautomatisierungssystemen

3.2.1 Allgemeine Modelleigenschaften und verwendete Modelle für die Sicherheitsanalyse

Ein *Modell* ist die vereinfachende Darstellung der Realität unter einer bestimmten Sicht. Im Allgemeinen werden abstrakte Modelle verwendet, die unterschiedliche Abstraktionsgrade besitzen können. Da die Realität meistens nicht detailgetreu beschrieben werden soll und kann, werden für den Modellzweck unwichtige Aspekte ignoriert – das Modell ist dadurch einfacher zu verstehen als die Wirklichkeit selbst.

Der Aufbau von Modellen wird durch verschiedene Modellelemente geprägt, die entweder grafisch oder textuell beschrieben werden. Die Definition dieser Modellelemente muss so erfolgen, dass das Modell als Ganzes leicht zu verstehen und anwendbar ist. Wichtig für eindeutige Aussagen eines Modells ist der Grad der Formalisierung. Hierbei wird die Struktur der Modellelemente sowie das Regelwerk (Syntax), mit dessen Hilfe die Modellelemente untereinander koppelbar sind, definiert. Den einzelnen Modellelementen, ihren Kombinationen und Zuordnungen werden Bedingungen oder Konzepte aus einem speziellen fachlichen Kontext zugeordnet, die mehr oder weniger detailliert und formal spezifiziert sind (Semantik) [SCJ98].

Bei Sicherheitsanalysen ist es wichtig, dass Zustände eines Prozessautomatisierungssystems unterschieden und bewertet werden können. Zur Durchführung einer Sicherheitsanalyse muss daher ein Modell folgende Aspekte berücksichtigen:

- Klassifizierung von unterschiedlichen Betriebsmodi, wie z. B. ein fehlerhafter oder gefährlicher Vorgang.
- Ermittlung des potenziellen Systemverhaltens unter Annahme verschiedener Fehler.
- Beschreibung von Ursache (Fehler) und Wirkung (Gefahren, Unfälle).

Wie im Kapitel 2.1 beschrieben, besitzen Prozessautomatisierungssysteme ganz unterschiedliche Bestandteile. Zur Realisierung einer ganzheitlichen Betrachtung eines Prozessautomatisierungssystems sind dabei folgende Punkte zu beachten:

- Die unterschiedlichen Eigenschaften und Verhaltensmuster der Bestandteile müssen anhand der Syntax und Semantik des Modells beschreibbar sein. Hierzu zählt explizit die Modellierung von sequenziellen und parallelen Abläufen, kontinuierlichen und diskreten Verhaltensmustern sowie allgemeinen Objektstrukturen.
- Die Struktur eines Prozessautomatisierungssystems kann sich aus den in Abbildung 2.2 vorgestellten Grundstrukturen (vom Offline Betrieb bis zum autonomen Betrieb des Rechnersystems) beliebig zusammensetzen. Die Struktur bestimmt das Zusammenspiel des technischen Systems, der Automatisierungssoftware und der menschlichen Bedieneingriffe maßgeblich. Die Struktur muss daher anhand der Modellelemente abgebildet werden können.
- Aufgrund der Komplexität von Prozessautomatisierungssystemen müssen geeignete Mechanismen zur Datenreduktion bzw. Abstraktion vorhanden sein. Hierarchische Ansätze (schrittweise Dekomposition) oder komponentenbasierte Modellierungskonzepte (Bildung neuer Komponenten aus bestehenden) sind weitere Mechanismen zur Bewältigung der Komplexität.

Ein sehr guter Überblick über die in der Prozessautomatisierungstechnik üblichen Modellkonzepte ist in [LaGö99b], [Schn99] und [SCJ98] zu finden. Im Folgenden werden die verwendeten Modelle für Sicherheitsanalysen untersucht. Wie in Tabelle 3.4 dargestellt, werden für die Si-

cherheitsanalyse im Wesentlichen Graphen, zustandsorientierte und analytisch-qualitative Beschreibungsmittel verwendet.

3.2.2 Graphen

Graphen bestehen aus Knoten und Kanten. Knoten können beliebig über Kanten verbunden werden. Dabei werden gerichtete und ungerichtete Graphen unterschieden. Bei gerichteten Graphen ist der Startknoten und Endknoten der Kanten genau definiert, siehe Abbildung 3.6. Die Baumstruktur stellt einen Sonderfall eines Graphen dar: Bäume sind Graphen mit minimaler Verbindung zwischen allen Knoten. Sie besitzen keine Schleifen [Kühn93].

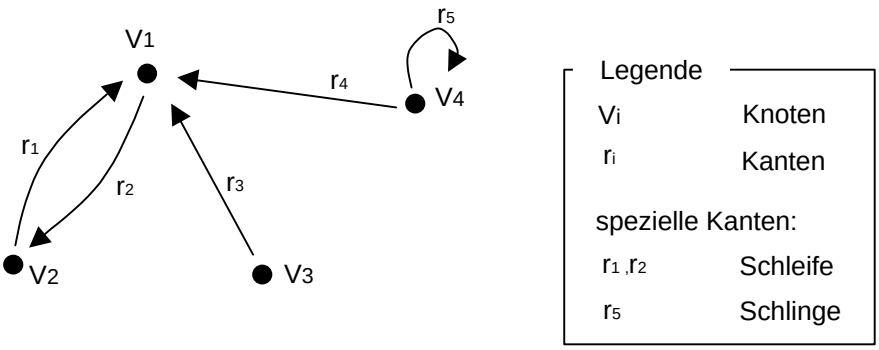


Abbildung 3.6: Beispiel für einen gerichteten Graphen

Bäume bzw. Graphen eignen sich gut, um Zusammenhänge strukturiert wiederzugeben. Die Knoten können als Platzhalter für Aktionen, Ereignisse, menschliche Handlungsweisen aber auch für Bauelemente herangezogen werden. Mit Hilfe von Kanten kann ausgedrückt werden, wie diese Aktionen, Ereignisse, etc. miteinander in Verbindung stehen bzw. voneinander abhängen. Die Erstellung eines Graphen ist leicht und die Struktur kann mathematisch überprüft werden. Bei vielen Graphen werden die Knoten und Kanten quantitativ gewichtet. Typische Beispiele für solche Graphen sind Wahrscheinlichkeitsbäume und Markov-Ketten [SSG+90]. Damit lassen sich Wege durch den Graphen quantifizieren.

Eine Erweiterung der Graphen stellen Logikbäume mit Logikelementen, wie UND, ODER, NICHT u.a., dar (Fehlerbäume, Ereignisablaufdiagramme). Damit lassen sich Zusammenhänge und insbesondere Abhängigkeiten zwischen verschiedenen Knoten (also Ereignissen, Aktionen) präziser beschreiben. Die den Knoten zugeordneten Assoziationen können allerdings nicht näher definiert bzw. modelliert werden.

Der Anwender muss das System gut kennen und wissen, wie die einzelnen Elemente oder Aktionen in Verbindung stehen. Dieses Wissen stellt er grafisch mit Hilfe von Bäumen dar. Der Aufbau eines Baums wird schon für relativ kleine Probleme oder Systeme umfangreich und unübersichtlich. Bei der Erstellung von Bäumen sind Abstraktionsfehler häufig, da in der Regel

zwischen den Systemelementen komplexe Zusammenhänge bestehen, die nicht direkt auf Baumstrukturen übertragen werden können.

3.2.3 Zustandsautomaten

Unter Automat wird ein Objekt verstanden, das zu einer Eingabe ein bestimmtes Ergebnis ausgibt. Ein endlicher Automat besteht aus einer Anzahl von internen Konfigurationen, die Zustände genannt werden [Balz96]. Die Automatentheorie umfasst die mathematische Grundlage und Beschreibung der Zustandsautomaten [SSH92], die durch drei verschiedene Größen definiert werden: den Zustand, die Eingangsgrößen und die Ausgangsgrößen [LaGö99b]. Ereignisse lösen gewöhnlich einen Zustandsübergang (Transition) aus. Transitionen ihrerseits haben eine bestimmte Aktion zur Folge, siehe Abbildung 3.7. Ereignisse können beispielsweise die Änderung von Eingangsgrößen und typische Aktionen, wie z. B. das Setzen von Ausgangsgrößen, sein.

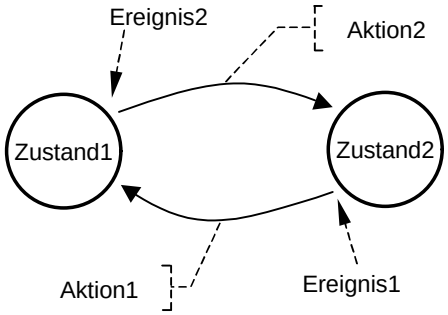


Abbildung 3.7: Grafische Darstellung eines Zustandsautomaten

Wichtige Varianten von Zustandsautomaten sind *Moore-Automaten*, *Statecharts* [Hare87] und *zeitbehaftete Zustandsautomaten* [AlDi94].

Bei der Beschreibung von diskreten Prozessen und bei der Entwicklung von Automatisierungssoftware ist die Anwendung von Zustandsautomaten weit verbreitet. Ebenfalls finden Zustandsautomaten bei formalen Methoden, wie z. B. den Modelcheckern SMV, HyTECH oder UPPAAL, zur funktionalen Systembeschreibung Verwendung. Mit Hilfe von Zustandsautomaten lassen sich komplexe Abläufe kompakt darstellen und simulieren. Der Einsatz von Zustandsautomaten für die Sicherheitsanalyse wird als sehr zeitaufwändig und schwierig bewertet [Lev95]. Ähnlich wie beim Einsatz von Baumstrukturen muss der Anwender das Modell gedanklich schon entwickelt haben und das System gut kennen, bevor er Zustandsautomaten zur detaillierten Beschreibung einsetzen kann. Die physikalische Struktur eines Systems lässt sich anhand von Zustandsautomaten nicht beschreiben, sequenzielle Vorgänge dagegen sehr gut. Kontinuierliche Vorgänge können nur diskretisiert beschrieben werden.

3.2.4 Petri-Netze

Ein Petri-Netz ist ein gerichteter Graph mit zwei Arten von Knoten, den Stellen und den Transitionen [Petri62]. Stellen dienen zur Beschreibung von Zuständen oder Bedingungen, während Transitionen in der Regel Aktionen oder Ereignisse darstellen. Kanten dürfen nur zwischen unterschiedlichen Knotenarten, d.h. zwischen Stellen und Transitionen bestehen. Zur Beschreibung von Vorgängen in einem Petri-Netz werden die Stellen mit *Marken* belegt. Die Marken zeigen den aktuellen Systemzustand an. Ein Wechsel der Marken in andere Stellen des Netzes erfolgt beim Schalten von Transitionen nach definierten Schaltregeln [BiKo00].

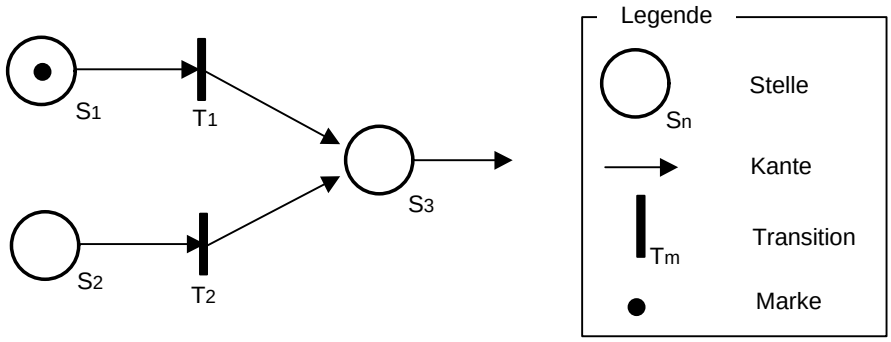


Abbildung 3.8: Grafische Darstellung eines Petri-Netzes

Petri-Netze besitzen wie die Zustandsautomaten eine fundierte mathematische Beschreibung. Mit Hilfe der Inzidenzmatrix kann die Struktur des Netzes wiedergegeben werden. Mit Hilfe von Markierungsvektoren, die die aktuelle Markenbelegung eines Petri-Netzes wiedergeben, und der Inzidenzmatrix lassen sich beliebig viele Schaltvorgänge mathematisch berechnen [Abel90].

Die wichtigsten Varianten von Petri-Netzen für die Automatisierungstechnik sind farbige Petri-Netze, Prädikat-Transitions-Netze, zeitbewertete Petri-Netze und unscharfe (Fuzzy) Netze [LaGö99b].

Petri-Netze eignen sich zur Modellierung, Analyse und Simulation von dynamischen Systemen mit nebenläufigen und deterministischen Vorgängen. Die besondere Stärke von Petri-Netzen liegt allerdings in der Beschreibung von Synchronisationsproblemen und der Analyse von Verklemmungsproblemen (*deadlock*). Ähnlich wie Zustandsautomaten eignen sich Petri-Netze für die Beschreibung von sequenziellen Abläufen, die durch diskrete Prozessgrößen charakterisiert sind. Die Darstellung und Verarbeitung von Prozessgrößen, Informationsgrößen und Sollwertvorgaben können mit Petri-Netzen nur sehr abstrakt beschrieben werden [Stra97]. Die Struktur eines Prozessautomatisierungssystems lässt sich mit Hilfe von Petri-Netzen nicht wiedergeben. Der Weg der Modellerstellung kann durch Dritte nicht nachvollzogen werden, da sich ein und dasselbe Problem meistens mit unterschiedlichen Petri-Netzen beschreiben lässt.

3.2.5 Qualitative Modelle

Qualitative Aussagen sind dem Menschen vertraut und enthalten implizit die Bewertung quantitativer Fakten. Das Wort „qualitativ“ besitzt viele Bedeutungen und heißt mehr als „nichtnumerisch“, siehe Tabelle 3.5. Im Gegensatz zu traditionellen Modellierungstechniken stellt die konzeptionelle Modellierung den zentralen Mittelpunkt bei qualitativen Modellen dar. Die Schritte, die notwendig sind, um aus der Analyse von physikalischen Phänomenen oder Vorgängen allgemeiner Art ein Modell herzuleiten, Beobachtungen und numerische Daten zu interpretieren, gehören zur Kunst des Modellierens und erfordern viel Erfahrung und Kenntnis. Bei diesen Schritten sollen qualitative Modelle den Anwender entlasten und unterstützen.

Tabelle 3.5: Verschiedene Beispiele für qualitative Ausdrücke

Sprachliche Begriffe	**qualitativer Ausdruck**	**quantitativer Ausdruck**
Nicht numerisch	zu schnell	210 km/h
Bereiche	zwischen 15°C und 20°C	17,5 °C
Vagheit	fast nichts	0,001
Wertebereiche	{langsam, moderat, schnell, sehr schnell}	$\exists v \in \Re$
Richtungen, Neigungen	Zunahme	dx/dt = - 4
Relative Werte	zu schnell (Auto) und zu langsam (Flugzeug)	210 km/h
Nicht geordnete Werte	fest oder flüssig	-2° C < +2 °C
Topologien	Normalbetrieb	1,7 < x < 3
Klassifikationen	eine lineare Funktion	9x + 3
Verhalten	periodisch	a sin(t)
Formen	symmetrisch	$x^2+y^2=r^2$
Strukturen	gefroren	-2 °C

„*Qualitative Modelle bilden die Grundlage für eine Formalisierung und Automatisierung des Modellierungsschrittes selbst. Dies stellt eine wichtige Voraussetzung für Computersysteme dar, die sich in einem Weltausschnitt intelligent verhalten, d.h. insbesondere neue oder nicht exakt beschriebene Situationen analysieren und menschliche Benutzer in natürlicher Weise vermitteln*.“ [MoSt95].

Einen guten Überblick über qualitative Modelle geben [Fröh97], [Lauf96], [LaGö99b], [MoSt95]. Die grundlegenden Ansätze sind ENVISION [KlBr84], QPT [Forb84] und QSIM [Kuip94].

Mitte der neunziger Jahre wurden auf Basis von ENVISION, QPT und QSIM weitere qualitative Modelle entwickelt. Bei probabilistischen Modellen, wie [Schil97] und [Lunz98], erfolgt die qualitative Beschreibung von Prozessgrößen durch Symptome. Beispielsweise kann die Wirkleistung Pw eines Generators durch drei Symptome qualitativ beschrieben werden: niedriger Leistungsbereich *pw**n*** für Pw < 0,5 MW, mittlerer Leistungsbereich *pw**m*** für 0,5 MW < Pw <

0,55 MW und hoher Leistungsbereich *pw**h*** für Pw > 0,55 MW. Entsprechend wird jedes Symptom mit einem Wahrheitswert belegt, so dass die qualitative Beschreibung für Pw = 0,517 MW folgendermaßen lautet:

$$Pw = \begin{pmatrix} B(pwn) = FALSCH \\ B(pwm) = WAHR \\ B(pwh) = FALSCH \end{pmatrix}$$

Den Werten größer 0,5 MW und kleiner 0,55 MW ist das Symptom pw**m** zugeordnet und wird daher auf wahr gesetzt. Symptome lassen sich mit Wahrscheinlichkeiten verknüpfen. Beispielsweise bedeutet

```
P(pwm) = P(B(pwm)=WAHR) = 0,98
```

dass die Wirkleistung zu 98% im Bereich 0,5 MW bis 0,55 MW liegt.

In Kapitel 3.1.6 wurde das SQMA-Verfahren erläutert. Das qualitative Modell von SQMA beruht auf dem komponentenorientierten Konzept von ENVISION und verwendet zusätzlich, wie QSIM, Intervallvariablen als qualitative Beschreibungsmittel. Intervallvariablen ermöglichen in Verbindung mit qualitativen Ausdrücken nicht nur ein großes Anwendungsspektrum, sondern ebenfalls eine kompakte Darstellung von Sachverhalten. Abbildung 3.9 zeigt die Beschreibung einer Prozessgröße mit Intervallen und Symptomen [Schil97]. Eine Darstellung mit Symptomen bringt den Vorteil, dass sich zur Berechnung des qualitativen Modells die Boolesche Algebra mit bewährten Algorithmen anwenden lässt. Die Berechnung eines qualitativen Modells basierend auf Intervallvariablen ist aufwändiger, da diese auf der komplexeren Intervallarithmetik beruhen. Der große Vorteil liegt darin, dass neue Werte bei der Modellerstellung berechnet werden können.

Qualitative Modelle werden insbesondere für die Prozessdiagnose und dort vor allem bei der Fehlerlokalisation eingesetzt [Lunz98, Schi97, Fröh97]. [Lauf96] zeigte, dass sich qualitative Modelle ebenfalls zur Unterstützung von Gefahren- bzw. Sicherheitsanalysen eignen. Viele Vorgänge sind mit Hilfe von quantitativen Modellen nur sehr schwer oder gar nicht beschreibbar, während eine qualitative Modellierung in der Regel durchführbar ist [Fröh97]. Eine qualitative Beschreibung ist schon auf der Basis von wenigen ungenauen Informationen möglich [Kuip94]. Der eigentliche konzeptionelle Vorgang der Modellerstellung wird durch qualitative Modelle unterstützt und formalisiert. Qualitative Modelle sind leicht zu interpretieren, da diese implizit die Bewertung von quantitativen Fakten enthalten.

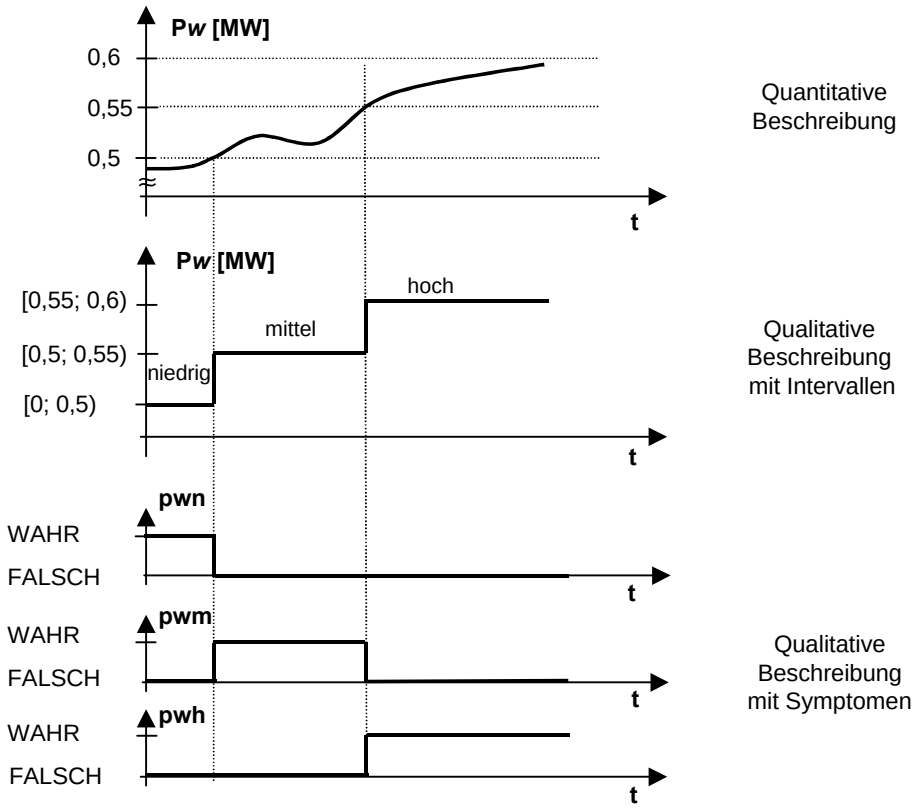


Abbildung 3.9: Qualitative Beschreibung von Prozessgrößen mit Intervallen und Symptomen

3.2.6 Bewertung der Modellansätze

Die behandelten Modellansätze sind in Tabelle 3.6 dargestellt und nach den in Kapitel 3.2.1 erläuterten Kriterien bewertet. Zusätzlich werden die in Kapitel 2.1 beschriebenen Eigenschaften von technischem System, Automatisierungssoftware und menschlichen Bedieneingriffen berücksichtigt.

Die Bewertung stützt sich auf eigene Untersuchungen und auf Ergebnissen der Arbeit „Klassifikation und Bewertung von Beschreibungsmitteln für die Automatisierungstechnik“ [SCJ98].

Zur Realisierung einer ganzheitlichen modellbasierten Sicherheitsanalyse stellen qualitative Methoden einen geeigneten Ansatz dar. Prinzipiell sind damit die Eigenschaften von technischem System, Automatisierungssoftware und menschlichen Bedieneingriffen beschreibbar. Der wesentliche Vorteil von qualitativen Modellen besteht in dem vorgegebenen Abstraktionsmechanismus, der den Anwender bei der konzeptionellen Modellerstellung unterstützt. Dabei wird der Anwender gezwungen, komplexe Vorgänge qualitativ zu beschreiben und damit zu simplifizieren.

Qualitative Ausdrücke sind einfach zu verstehen, da sie der menschlichen Denkweise entsprechen. Im Hinblick auf eine Sicherheitsanalyse ist ebenfalls von großem Vorteil, dass qualitative Ausdrücke implizit die Bewertung von quantitativen Fakten enthalten.

Tabelle 3.6: Übersicht über potenzielle Modelle für die Sicherheitsanalyse

Kategorie	**Kriterien** Legende: + geeignet, 0 möglich, - nicht möglich	**Graphen**	**Zustands-automaten**	**Petri-Netze**	**Qualitative Methoden**
Sicherheitsanalyse	Klassifizieren von Betriebsmodi	0	+	+	+
	Zusammenhang: Ursache und Wirkung	+	+	+	+
	Ermittlung des potenziellen Systemverhaltens	-	0	+	+
Abbildung von Strukturen	Physikalische Strukturen	-	-	-	+
	Objektstrukturen	-	-	-	0
	Komponentenstrukturen	-	-	-	+
Verhaltensbeschreibung	Dynamik	0	+	+	+
	Parallele Abläufe	-	-	+	+
	Sequenzielle Abläufe	+	+	+	+
	Kontinuierliche Größen / Vorgänge	-	-	-	0
	Diskrete Größen / Vorgänge	-	+	+	0
	Steueralgorithmen	-	-	0	0
	Fähigkeitsbasierte Entscheidungsfindung	+	+	+	+
	Regelbasierte Entscheidungsfindung	+	+	+	+
	Wissensbasierte Entscheidungsfindung	-	-	-	0
Modellierung	Modularität	-	0	0	+
	Abstraktionsmechanismus	-	-	-	+
	Unterstützung der Modellierung	-	-	0	+

Eine qualitative Modellierung ist selbst bei unvollständigen Informationen möglich [Kuip94]. Gerade in den frühen Entwicklungsphasen liegt in der Regel wenig Detailwissen bzw. unvollständige Informationen über das zu entwickelnde System vor. Intervallbasierte Ansätze, wie z. B. SQMA verknüpfen qualitative Ausdrücke mit analytischen (quantitativen) Fakten. Diese Kombination ermöglicht die Modellierung von unterschiedlichen Größen und Vorgängen. Basierend auf der Intervallarithmetik ist die Berechnung von neuen signifikanten Werten möglich. Dies stellt eine Grundvoraussetzung zur Modellierung der unterschiedlichen Eigenschaften der Bestandteile dar. Zur Modellierung der Struktur eines Prozessautomatisierungssystems und zur Komplexitätsbewältigung ist ein komponentenbasierter Ansatz erforderlich. Bei Graphen und zustandsbasierten Modellierungsverfahren ist zusätzlich eine Abstraktion der Prozessstruktur, z. B. in Form von Kanten oder Transitionen, notwendig. Dabei ist die Gefahr hoch, dass das Zusammenspiel der Bestandteile zu stark vereinfacht wird und Wechselwirkungen zwischen den Systemelementen für eine Sicherheitsanalyse nicht ausreichend wiedergegeben werden. Das qualitative Modellierungsverfahren SQMA ist hinsichtlich der geschilderten Gründe für eine ganzheitliche Sicherheitsanalyse am besten geeignet, da die qualitative Modellierung anhand von Intervallen erfolgt und es sich um einen komponentenbasierten Ansatz handelt.

3.3 Zusammenfassung der Erkenntnisse und Folgerungen

Die im ersten Teil dieses Kapitels vorgestellten Verfahren zur Analyse von Gefahren bzw. der Sicherheit sind in der Regel für die Untersuchung eines speziellen Bestandteils eines Prozessautomatisierungssystems konzipiert. Verfahren mit dem Fokus auf einen speziellen Bestandteil können nicht einfach für die Betrachtung von anderen Systembestandteilen übertragen werden, da Voraussetzungen und Eigenschaften zu unterschiedlich sind [Leve95], [Bieg98a].

Viele klassische Verfahren zur Analyse der Sicherheit sind inzwischen zehn bis fünfzehn Jahre alt und werden dem Stand heutiger Prozessautomatisierungssysteme nicht mehr gerecht. Insbesondere werden Prozessautomatisierungssysteme durch den zunehmenden Einsatz von Software immer schwieriger zu analysieren. Hinzu kommt, dass die meisten Verfahren Modelle nur zur Ergebnisdarstellung heranziehen, die Analyse beruht auf der subjektiven Bewertung von Sachverständigen und findet in Form eines Denkprozesses (Brainstorming) statt [Bieg97a]. Die Interpretation des komplexen Zusammenspiels zwischen den Bestandteilen heutiger Prozessautomatisierungssysteme übersteigt die Leistungsfähigkeit des Menschen. Die Auswirkungen von Fehlerkombinationen aus verschiedenen Systembestandteilen können nur unzureichend bzw. überhaupt nicht untersucht werden. Gerade systembestandteilübergreifende Fehlerkombinationen stellen die häufigste Unfallursache dar [MeRe99], [Leve95], [Mont00].

Formale Methoden verwenden zur Überprüfung von Sicherheitsanforderungen einen modellbasierten Ansatz. Schon allein die Analyse der Automatisierungssoftware motiviert modellbasierte Ansätze. Die Automatisierungssoftware ist in der Regel komplex aufgebaut, deren Subsysteme sind stark voneinander und deren Funktionen von zahlreichen Randbedingungen abhängig [CGN98].

Das vorgestellte SQMA-Verfahren verwendet zur Analyse des technischen Systems ebenfalls einen modellbasierten Ansatz. Allerdings ist das Verfahren auf die Analyse von technischen Systemen beschränkt.

Die Untersuchung potenzieller Verfahren zur Systemanalyse von Prozessautomatisierungssystemen kann mit folgender Aussage zusammengefasst werden: Es existieren derzeit für Sicherheitsanalysen keine modellbasierten Ansätze, die eine ganzheitliche Betrachtungsweise von Prozessautomatisierungssystemen fördern. Unter ganzheitlicher Betrachtungsweise wird die Beschreibung aller Bestandteile eines Prozessautomatisierungssystems und deren Zusammenspiel in Form eines geeigneten Modells verstanden.

Erst ein modellbasierter Ansatz ermöglicht eine rechnergestützte Durchführung von Sicherheitsanalysen. Der Rechner ist dem Menschen in kombinatorischen Aufgaben weit überlegen. Nachteile, die durch den Brainstorming-Prozess entstehen (subjektive Bewertungen, nicht reproduzierbare Analyseergebnisse und Einschränkung des Analysebereichs) entfallen bei modellbasierten Ansätzen [KES99], [Bieg00c]. Benötigt wird allerdings ein geeignetes Modell.

Dieses Modell muss nicht nur von einem Computer interpretiert werden können, sondern auch vom Menschen (Anwender) leicht anwendbar und verständlich sein [Moik97]. Die Anwender sind in der Regel Ingenieure und eher selten Informatiker oder Mathematiker.

Um Anforderungen an eine geeignete Modellierungsmethode zu spezifizieren, wurden im zweiten Teil dieses Kapitels die wichtigsten Modellierungskonzepte und ihre Varianten diskutiert, die bei herkömmlichen Sicherheitsanalysen und in der Automatisierungstechnik eingesetzt werden. Als Nachteil hat sich dabei die mangelnde Unterstützung des Anwenders bei der eigentlichen Modellerstellung erwiesen. Der Anwender muss die Zusammenhänge der Systembestandteile genau kennen, bevor er diese mit den geschilderten Modellierungskonzepten darstellen kann. Gerade aber die Analyse dieser Zusammenhänge sollte im Rahmen einer modellbasierten Sicherheitsanalyse durch das Verfahren selbst erfolgen. Ansätze mit Petri-Netzen und verschiedenen Arten von Zustandsautomaten haben, neben dem genannten Problem der Modellerstellung, den Nachteil, dass sich diese nur zur Beschreibung des Kontrollflusses der Automatisierungssoftware und diskreter Prozesse sinnvoll einsetzen lassen. Gerade aber verfahrenstechnische Prozesse (kontinuierliche Prozesse) gehören zu den häufigsten Prozessen, die im Rahmen von Sicherheitsanalysen untersucht werden, siehe Kapitel 2.1.1. Die Modellierung von Algorithmen zur Datenverarbeitung ist mit Hilfe von Petri-Netzen und Zustandsautomaten aber nur unzureichend möglich. Mit Hilfe von Graphen kann das Verhalten eines Systemelements nicht modelliert werden. Außerdem ist nur eine hierarchische, keine topologische Strukturbeschreibung des Prozessautomatisierungssystems möglich.

Qualitative Modelle stellen einen geeigneten Ansatz dar, um die geschilderte Problematik zu beheben. Ein großer Vorteil der qualitativen Modellierungstechnik besteht darin, dass eine Erstellung des Modells auch bei wenigen oder ungenauen Informationen möglich ist. In [Lauf96] und [HBB+97] wurde die Eignung von qualitativen Modellen zur Durchführung von Gefahrenanalysen mit einem positiven Ergebnis untersucht. Diese Arbeiten befassten sich allerdings nur mit der Beschreibung des technischen Systems.

Es stellt sich daher die Frage, ob sich mit Hilfe von qualitativen Modellen nicht ebenfalls das Verhalten der Automatisierungssoftware und der menschlichen Bedieneingriffe beschreiben lässt. Somit könnte eine modellbasierte Sicherheitsanalyse zur ganzheitlichen Betrachtung von Prozessautomatisierungssystemen realisiert werden.

3.4 Anforderungen an eine modellbasierte Sicherheitsanalyse von Prozessautomatisierungssystemen

Der vorherige Abschnitt hat deutlich gemacht, dass eine Sicherheitsanalyse von Prozessautomatisierungssystemen auf Basis eines zugrunde liegenden Modells die meisten Vorteile mit sich bringt. Im Folgenden werden nun, basierend auf den gemachten Erkenntnissen, Anforderungen an eine modellbasierte Sicherheitsanalyse formuliert:

- *Anforderung 1*
  Der Einfluss und das Zusammenspiel des technischen Systems, der Automatisierungssoftware und der menschlichen Bedieneingriffe muss analysierbar sein. Das Prozessautomatisierungssystem muss im Rahmen einer Sicherheitsanalyse ganzheitlich betrachtet werden können.

- *Anforderung 2*
  Mögliche Fehler von Bauelementen, Fehler in der Automatisierungssoftware und falsche menschliche Bedieneingriffe sowie deren Auswirkung müssen bei der Sicherheitsanalyse berücksichtigt werden. Insbesondere sollen die Auswirkungen von Fehlerkombinationen aus diesen verschiedenen Systembestandteilen ermittelt werden können.

- *Anforderung 3*
  Es muss eine umfassende Untersuchung des Prozessautomatisierungssystems erfolgen. Die Sicherheitsanalyse darf sich nicht auf ein bestimmtes Betriebsszenarium (Prozesszustand) beschränken, sondern soll sämtliche Randbedingungen berücksichtigen.

- *Anforderung 4*
  Der Analyseweg soll nachvollziehbar und die Ergebnisse der Sicherheitsanalyse reproduzierbar sein.

- *Anforderung 5*
  Das Konzept der Sicherheitsanalyse muss schon bei der Entwicklung (insbesondere in den frühen Entwicklungsphasen) eines Prozessautomatisierungssystems anwendbar sein.

- *Anforderung 6*
  Die Sicherheitsanalyse soll rechnergestützt durchgeführt werden. Hierzu ist ein modellbasierter Ansatz notwendig. Das Modell muss die gestellten Anforderungen an die Sicherheitsanalyse berücksichtigen. Insbesondere muss es in der Lage sein, alle Systembestandteile und die Struktur eines Gesamtsystems zu beschreiben.

- *Anforderung 7*
  Das Modell soll einfach zu erstellen und zu verstehen sein. Einfach wird im Sinne von „ingenieurgerecht“ [Moik97] verstanden.

- *Anforderung 8*
  Die Erstellung des Modells soll weitgehend automatisiert erfolgen.

Zusammenfassend sei erwähnt, dass die Untersuchung von verwendeten Sicherheitsanalysen zeigt, dass kein Konzept bzw. Vorgehen existiert, um ein Prozessautomatisierungssystem ganzheitlich zu untersuchen. Die Diskussion von verwendeten Modellen zeigt, dass diese in der Regel nur Hilfsmittel darstellen, um die Ergebnisse der Sicherheitsanalyse zu dokumentieren. Der Anwender muss den Analysevorgang bereits vollzogen haben, um anschließend seine Ergebnisse mit den beschriebenen Modellen darstellen und bewerten zu können. Qualitative Modelle bieten die Möglichkeit, den Analysevorgang durch das eigentliche Modell selbst zu realisieren. Der qualitative Modellierungsansatz SQMA stellt die Basis für die Realisierung eines ganzheitlichen modellbasierten Ansatzes dar. Das Modellierungsprinzip wird im nächsten Kapitel vorgestellt.

4 Qualitativer Modellierungsansatz nach SQMA

Bei SQMA werden Systemgrößen anhand von Intervallvariablen beschrieben. SQMA verwendet einen komponentenbasierten Modellierungsansatz. Zuerst wird das allgemeine Verhalten jeder Komponente modelliert - unabhängig von ihrer speziellen Funktion im System. Anschließend werden die modellierten Komponenten entsprechend der physikalischen Struktur verknüpft. Auf diese Weise werden Situationen ermittelt, die das potenzielle Verhalten des gesamten Systems widerspiegeln. Ein besonderes Merkmal von SQMA ist die Berücksichtigung von empirischem Wissen (Erfahrungen). Dieses wird mit Hilfe von Kommentaren beschrieben.

Im Folgenden wird gezeigt, wie sich das Verhalten eines Systems durch die Verwendung von SQMA analysiert wird.

4.1 Qualitative Modellierung von Komponenten

Das Modell einer Komponente besteht aus folgenden Elementen:

- **Schnittstellen (Terminals)**. Schnittstellen beschreiben die Möglichkeit einer Komponente mit ihrer Umgebung zu kommunizieren.
- **Qualitative Intervallvariablen (Quantities).** Die physikalischen Größen, die an einer Schnittstelle in Erscheinung treten, werden mit Intervallvariablen beschrieben. Jeder qualitativen Variablen wird ein Werteraum zugewiesen. Mit Intervallen lässt sich dieser Werteraum in verschiedene, für die Funktion der Komponente wichtige, Bereiche unterteilen.
- **Situationen (Situations)**. Das Verhalten einer Komponente wird mit Situationen beschrieben. Diese ergeben sich aus den definierten Intervallvariablen. Eine Situation besteht aus einem gültigen Satz aller qualitativen Variablen mit einem speziellen Wertebereich (Intervall). Als gültig wird eine Situation bezeichnet, die alle Situationsbedingungen (Modellgleichungen) des Komponentenmodells erfüllt.
- **Situationsbedingungen (SituationRules).** Anhand von Wenn-Dann Regeln wird das physikalische und funktionale Verhalten einer Komponente definiert. Diese Regeln bestehen aus einem Satz von arithmetischen und logischen Ausdrücken und schränken den theoretischen Situationsraum einer Komponente ein. Als theoretischer Situationsraum wird der kombinatorisch mögliche Situationsraum bezeichnet. Er kann aus der Anzahl der qualitativen Variablen und deren unterschiedlichen Ausprägungen (Intervalle) berechnet werden.
- **Komentarregeln (CommentRules).** Eine Gruppe von Situationen (d.h. eine Kombination von Intervallvariablen mit bestimmten Ausprägungen) kann mit Hilfe eines Kommentars (qualitativen Ausdrucks) zusammengefasst und klassifiziert werden. Auf diese Weise lässt sich empirisches Wissen berücksichtigen. Darüber hinaus lassen sich ebenfalls Kombinatio-

nen von Wertebereichen verschiedener Größen mit einem qualitativen Ausdruck verknüpfen. Die Kommentare dienen zur Veranschaulichung der Modellergebnisse.

- **Übergänge (Transitions).** Der Übergang einer Situation in eine andere ist in Form einer Transitionsmatrix beschrieben. Die möglichen Übergänge beschreiben das dynamische Verhalten einer Komponente.
- **Transitionsregeln (TransitionRules).** Größen, die kennzeichnend für die in Energiespeichern enthaltene Energie sind, können sich nur stetig ändern. Änderungen sind zudem von anderen Größen abhängig. Zur Beschreibung solcher Sachverhalte dienen Transitionsregeln. Die möglichen Situationsübergänge werden aus den Transitionsregeln berechnet und als Transitionsmatrix angegeben.

Zur Erläuterung der SQMA-Modellelemente ist in Abbildung 4.1 ein System zu sehen, welches aus einem Wassertank und zwei Ventilen besteht. Es dient zur Sammlung von Regenwasser. Qualitativ lässt sich das Verhalten dieses Systems schnell erfassen: Falls es regnet und das Ventil gesperrt ist, sammelt sich Regenwasser im Wassertank. Wird eines der Ventile geöffnet, so kann dem Wassertank Regenwasser entnommen werden.

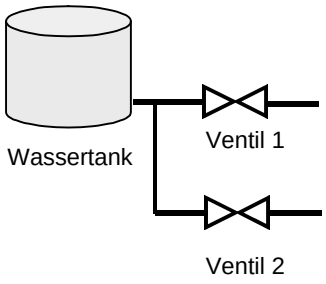


Abbildung 4.1: System zur Sammlung von Regenwasser

Die qualitative Modellierung einer Komponente soll am Beispiel des Wassertanks gezeigt werden. Der Wassertank wird aus seiner Systemumgebung herausgelöst und besitzt zwei unterschiedliche **Schnittstellen,** siehe Abbildung 4.2. An der ersten Schnittstelle tritt die Größe Q_{Zu} auf, wobei Q_{Zu} den Zufluss des Regenwassers repräsentiert. Der Abfluss des Regenwassers aus dem Tank ist mit Q_{Ab} bezeichnet und tritt an der zweiten Schnittstelle des Wassertanks auf. Falls sich Wasser im Tank befindet, tritt an der zweiten Schnittstelle zusätzlich ein Druck P_{Ab} auf. Dessen Wert ist abhängig vom aktuellen Füllstand. Die Größen Q_{Zu}, Q_{Ab} und P_{Ab} werden durch **qualitative Intervallvariablen** beschrieben, der Wertebereich wird in Intervalle eingeteilt. In Abbildung 4.2 ist beispielhaft die normierte Darstellung des Drucks P_{Ab} zu sehen.

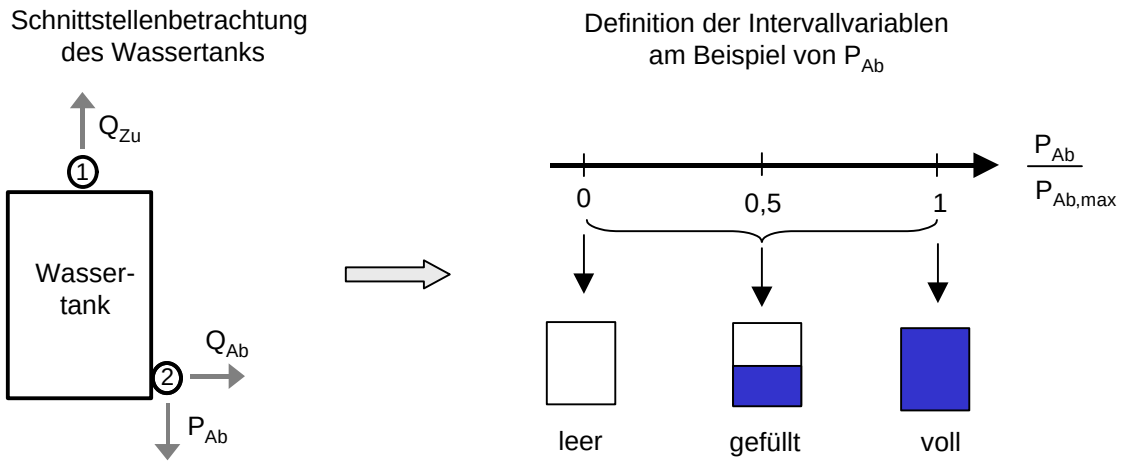


Abbildung 4.2: Definition einer qualitativen Intervallvariablen

Der Referenzwert ist der maximale Druck $P_{Ab,max}$, der bei einem vollen Wassertank auftritt. Qualitativ kann der Wassertank leer, gefüllt oder voll sein. Dieses „Wissen“ wird in Form von Kommentaren angegeben. Über die Größen bzw. den Zusammenhang der Größen ist weiterhin bekannt:

- Falls es regnet, so ist der Zufluss Q_{Zu} kleiner als null (Richtung von Q_{zu} ist zu beachten).
- Falls es nicht regnet, ist Q_{Zu} gleich null.
- Falls Regenwasser entnommen wird, ist der Abfluss Q_{Ab} größer als null.
- Aus einem leeren Wassertank kann kein Regenwasser abfließen.
- Falls es regnet ($Q_{Zu} < 0$) und der Wassertank voll ist ($P_{Ab}=1$), dann läuft der Wassertank über.

Diese Angaben werden mit **Situationsregeln** und **Kommentarregeln** beschrieben. Aus diesen Angaben werden alle möglichen Situationen des Wassertanks ermittelt. Über das dynamische Verhalten der Komponente ist folgendes bekannt:

- Der Füllstand des Wassertanks kann sich nur stetig ändern (P_{Ab} ist stetig).
- Die Änderung ist von Zu- und Abfluss abhängig.

Aus den angegebenen Transitionsregeln können die möglichen Übergänge zwischen den Situationen ermittelt werden. Das qualitative Modell des Wassertanks ist in Tabelle 4.1 dargestellt. Im Allgemeinen ist der Überlauf eines Wassertanks nicht erwünscht und wird im Modell mit einem „U“ klassifiziert. Alle anderen Angaben spiegeln bestimmungsgemäße, d.h. normale Situationen eines Wassertanks wider und erhalten daher das Attribut „B“.

Für die baugleichen Komponenten „Ventil 1“ und „Ventil 2“ reicht es aus, ein allgemeines qualitatives Modell eines Ventils zu erstellen. Dieses Modell kann im Systemmodell mehrmals verwendet werden.

Tabelle 4.1: Situationen und Transitionen der Komponente Wassertank

1:	leer/zufluss	B
2:	leer	B
3:	gefuellt/zufluss	B
4:	gefuellt	B
5:	gefuellt/abfluss/zufluss	B
6:	gefuellt/abfluss	B
7:	ueberlauf/voll/zufluss	U
8:	voll	B
9:	voll/abfluss/zufluss	B
10:	voll/abfluss	B

	1	2	3	4	5	6	7	8	9	10
1	.		x	x	x	x	.	.	.	.
2	x	x	.	.	.	.	.	.	.	.
3	.	.	.	.	.	.	x	x	x	.
4	.	.	x	x	x	x	.	.	.	.
5	x	x	x	x	.	x	x	x	x	x
6	x	x	.	.	.	.	.	.	.	.
7	.	.	.	.	.	.	.	x	x	x
8	.	.	.	.	.	.	x	x	x	x
9	.	.	x	x	x	x	x	x	x	x
10	.	.	x	x	x	x	.	.	.	.

4.2 Kopplung von Komponenten

Die Modelle der Komponenten werden durch Angabe einer **Netzliste** verknüpft. Eine Netzliste legt die Verbindungen zwischen den Schnittstellen der modellierten Komponenten fest. Eine Netzliste ist eine Liste aus Knoten, wobei jeder Knoten aus zwei oder mehreren Schnittstellen von Komponenten besteht. Für das System aus Abbildung 4.1 lautet die Netzliste: der Abfluss des Wassertanks ist mit der Eingangsseite von Ventil 1 und Ventil 2 verbunden:

Knoten: (Wassertank,1
Ventil1,1
Ventil2,1).

Aus der Netzliste lassen sich **Systemgleichungen** ableiten. Diese Systemgleichungen repräsentieren die Wechselwirkung zwischen den einzelnen Komponentenmodellen, indem die an den – in Verbindung stehenden – Schnittstellen definierten Größen unter Verwendung der Kirchhoff'schen Regeln verknüpft werden. Dabei unterscheidet SQMA Potenzialgrößen (Druck) und Flussgrößen (Strom). Für die Verknüpfung von Potenzialgrößen wird der Maschensatz und für Flussgrößen der Knotensatz angewandt. Für Abbildung 4.1 ergibt sich nach dem obigen Knoten:

Druck: $Wassertank.P_{Ab} = Ventil1.P_A = Ventil2.P_A$

Strom: $Wassertank.Q_{Ab} = Ventil1.Q_A + Ventil2.Q_A$

Anhand der Systemstruktur (beschrieben durch die Netzliste) und den Informationen über die Komponenten (beschrieben durch die Komponentenmodelle) werden Systemsituationen abgeleitet. Eine Systemsituation ist ein Satz von Situationen der einzelnen Komponentenmodelle, bei dem alle Systemgleichungen erfüllt sind. Beispiel: Eine Situation des Wassertanks wird mit einer Situation des Ventils 1 und des Ventils 2 verknüpft, falls ...

1. ... in den entsprechenden Situationen die Intervalle des Drucks eine gemeinsame Schnittstelle besitzen und ...
2. ... die Bedingung erfüllt ist, dass die Summe der Flüsse in Ventil 1 und Ventil 2 gleich dem Fluss aus dem Wassertank ist.

Verletzt eine Kombination von Situationen diese Systemgleichungen, so wird diese verworfen und eine neue Kombination überprüft. Dieser Vorgang wird ausgeführt, bis alle kombinatorisch möglichen Sätze von Situationen der Komponenten behandelt wurden. Das Ergebnis ist eine Menge von Systemsituationen, die das mögliche Verhalten des technischen Systems beschreiben.

Aus den Transitionen der Komponentenmodelle werden die möglichen Übergänge zwischen den Systemsituationen bestimmt. Ein Übergang zwischen zwei Systemsituationen ist möglich, falls zwischen den Situationen der Komponentenmodelle ebenfalls ein Übergang definiert ist.

4.3 Verhaltensbeschreibung mit Situationen und Transitionen

Systemsituationen sind Szenarien für den möglichen Betrieb eines Systems, die zu einem beliebigen Zeitpunkt auftreten können. Sie geben Auskunft über das mögliche statische Verhalten eines Systems. Systemsituationen sind mit Momentaufnahmen vergleichbar, bei denen die Werte der Systemgrößen in einem speziellen Bereich (Intervall) liegen. Zur Veranschaulichung und Interpretation des Modells werden die den Intervallen bzw. den Intervallkombinationen zugeordneten Kommentare als Ergebnisdarstellung verwendet.

In Tabelle 4.2 sind alle ermittelten Systemsituationen des Systems zur Sammlung von Regenwasser dargestellt. Die Systemsituation mit der Nummer 11 ist als unerwünscht („U“) gekennzeichnet, da diese einen Überlauf des Wassertanks darstellt. Das Modell sagt demnach aus, dass im Betrieb der Anlage ein Überlauf des Wassertanks auftreten kann. Anhand des allgemeinen Verhaltens der einzelnen Komponenten und der Systemstruktur wurde qualitativ vorhergesagt, wie sich das Gesamtsystem verhalten wird. Aus der Matrix in Tabelle 4.3 können weiterhin Informationen über das dynamische Systemverhalten entnommen werden.

Aus dem qualitativen Modell ist zu erkennen, dass sich mindestens ein Ventil in der Situation „offen“ befinden muss, damit ein Abfluss aus dem Wassertank stattfinden kann. Hingegen kann ein Zufluss durch Regen zu einem beliebigen Zeitpunkt erfolgen. Die daraus möglichen Situationen des gesamten Systems gibt Tabelle 4.2 wieder. So läuft z. B. der Behälter über, falls ein Zufluss stattfindet (Regen), der Wassertank voll und kein Ventil geöffnet ist. Aus der Transitionsmatrix kann entnommen werden, dass ein Wechsel in die Situation Nr. 11 (Überlauf) nur aus Situationen möglich ist, in denen beide Ventile geschlossen sind. Nur aus Situation Nr. 9 und Nr. 12 ist ein Übergang in die Situation Nr. 11 möglich, siehe Tabelle 4.3. Situationen, in denen mindestens ein Ventil geöffnet ist, gehen gemäß dem qualitativen Modell nicht in Situation Nr. 11 über. Sobald ein Ventil geöffnet ist, sinkt der Wasserstand des Wassertanks auch bei gleichzeitigem Zufluss. Ein Anstieg des Wasserstands ist nur bei einem Zufluss und bei geschlossenen Ventilen möglich.

Tabelle 4.2: Ergebnisdarstellung in Form von Systemsituationen

	Wassertank		Ventil1		Ventil2	
1	leer/zufluss	B	zu	B	zu	B
2	leer/zufluss	B	offen	B	zu	B
3	leer/zufluss	B	zu	B	offen	B
4	leer/zufluss	B	offen	B	offen	B
5	leer	B	zu	B	zu	B
6	leer	B	offen	B	zu	B
7	leer	B	zu	B	offen	B
8	leer	B	offen	B	offen	B
9	gefuellt/zufluss	B	zu	B	zu	B
10	gefuellt	B	zu	B	zu	B
11	voll/ueberlauf/zufluss	U	zu	B	zu	B
12	voll	B	zu	B	zu	B
13	gefuellt/abfluss/zufluss	B	zu	B	offen	B
14	gefuellt/abfluss	B	zu	B	offen	B
15	voll/abfluss/zufluss	B	zu	B	offen	B
16	voll/abfluss	B	zu	B	offen	B
17	gefuellt/abfluss/zufluss	B	offen	B	zu	B
18	gefuellt/abfluss	B	offen	B	zu	B
19	voll/abfluss/zufluss	B	offen	B	zu	B
20	voll/abfluss	B	offen	B	zu	B
21	gefuellt/abfluss/zufluss	B	offen	B	offen	B
22	gefuellt/abfluss	B	offen	B	offen	B
23	voll/abfluss/zufluss	B	offen	B	offen	B
24	voll/abfluss	B	offen	B	offen	B

Tabelle 4.3: Transitionen zwischen den Systemsituationen als Matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	.	x	x	x	.	.	.	.	.	.	.	.	x	x	.	.	x	x	.	.	x	x	.	.
2	x	x	x	x	x	x	x	x	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
3	x	x	x	x	x	x	x	x	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
4	x	x	x	x	x	x	x	x	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
5	x	x	x	x	x	x	x	x	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
6	x	x	x	x	x	x	x	x	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
7	x	x	x	x	x	x	x	x	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
8	x	x	x	x	x	x	x	x	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
9	.	.	.	.	.	.	.	.	.	.	x	x	x	x	x	x	x	x	x	x	x	x	x	x
10	.	.	.	.	.	.	.	.	x	x	.	.	x	x	.	.	x	x	.	.	x	x	.	.
11	.	.	.	.	.	.	.	.	.	.	x	.	.	.	x	x	.	.	x	x	.	.	x	x
12	.	.	.	.	.	.	.	.	.	.	x	x	.	.	x	x	.	.	x	x	.	.	x	x
13	x	x	x	x	x	x	x	x	x	x	.	.	x	x	.	.	x	x	.	.	x	x	.	.
14	x	x	x	x	x	x	x	x	x	x	.	.	.	.	.	.	x	x	.	.	x	x	.	.
15	.	.	.	.	.	.	.	.	x	x	.	.	x	x	x	x	x	x	x	x	x	x	x	x
16	.	.	.	.	.	.	.	.	x	x	.	.	x	x	.	x	x	x	x	x	x	x	x	x
17	x	x	x	x	x	x	x	x	x	x	.	.	x	x	.	.	x	x	.	.	x	x	.	.
18	x	x	x	x	x	x	x	x	x	x	.	.	x	x	.	.	x	x	.	.	x	x	.	.
19	.	.	.	.	.	.	.	.	x	x	.	.	x	x	x	x	x	x	x	x	x	x	x	x
20	.	.	.	.	.	.	.	.	x	x	.	.	x	x	.	x	x	x	x	x	x	x	x	x
21	x	x	x	x	x	x	x	x	x	x	.	.	x	x	.	.	x	x	.	.	x	x	.	.
22	x	x	x	x	x	x	x	x	x	x	.	.	x	x	.	.	x	x	.	.	x	x	.	.
23	.	.	.	.	.	.	.	.	x	x	.	.	x	x	x	x	x	x	x	x	x	x	x	x
24	.	.	.	.	.	.	.	.	x	x	.	.	x	x	x	x	x	x	x	x	x	x	x	x

4.4 Anmerkungen zu qualitativen Beschreibungsmitteln

Die Beschreibung der Systemelemente sowie die Erstellung des gesamten Modells erfolgt durch Auswertung von Intervallgleichungen. Das Ergebnis von Intervallgleichungen ist nicht immer eindeutig. Nach [BSM+00] werden bei Addition und Multiplikation die Intervallbereiche gedehnt. Das führt gegebenenfalls zu nicht eindeutigen bzw. unscharfen Mengenbereichen. Weiterhin tritt nach [Stru90] ein Selektionsproblem auf, wenn eine Intervallvariable innerhalb einer Gleichung mehrfach auftritt, z. B.:

$$Q1 - Q1 = 0 \text{ aber } [1;2] - [1;2] = [-1;1]$$

Deshalb müssen die Intervallgleichungen dahingehend optimiert werden, dass in jeder Gleichung jede Variable nur einmal vorkommt.

Die qualitative Modellierung von technischen Systemen wurde in den Arbeiten [Lauf96] und [Fröh97] beschrieben. In diesen Arbeiten stand die Entwicklung der qualitativen Beschreibungsmittel im Vordergrund. Richtlinien und Festlegungen für deren Anwendung wurden hingegen nicht gegeben. Bei der Modellierung existieren somit viele Freiheitsgrade. Schnittstellen können beliebig klassifiziert, Intervallbereiche und qualitative Ausdrücke beliebig gewählt werden. Weiterhin existieren keine Richtlinien zur Bewertung einer Situation anhand ihrer Attribute, die wiederum frei gewählt werden können. Diese Modellierungsfreiheiten können bei gleicher Problemstellung zu verschiedenen Realisierungen führen.

Im Rahmen dieser Arbeit liegt der Schwerpunkt auf der Anwendung und Ergänzung der Beschreibungsmittel von SQMA zur Realisierung einer ganzheitlichen Betrachtungsweise von Prozessautomatisierungssystemen. Der Anwender muss konkrete Richtlinien für die Modellierung erhalten und durch die Modellierungsarbeit geführt werden. In Bezug auf die Realisierung eines ganzheitlichen Modells zur Sicherheitsanalyse ist dies notwendig, damit die verschiedenen Systembestandteile zu einem Gesamtmodell zusammengefügt werden können.

5 Konzept der ganzheitlichen modellbasierten Sicherheitsanalyse

In diesem Kapitel wird der Ansatz für die ganzheitliche Betrachtung von Prozessautomatisierungssystemen zur Durchführung einer modellbasierten Sicherheitsanalyse hergeleitet. Anschließend erfolgt ein Überblick über den Aufbau und die Durchführung des entwickelten Verfahrens.

5.1 Ansatz der ganzheitlichen modellbasierten Sicherheitsanalyse

Wie im zweiten Kapitel erläutert, basiert der Begriff Sicherheit auf der Definition des Grenzrisikos. Das Risiko selbst ist definiert als Produkt der Eintrittswahrscheinlichkeit eines zum Schaden führenden Ereignisses und dem Ausmaß des Schadens selbst. Das Ereignis, das zu einem Schaden führt, wird Ursache genannt. Der Schaden selbst ist die Folge oder die Auswirkung des Ereignisses. Die meisten Sicherheitsanalysen stützen sich auf diese Ursache-Folge-Beziehung. Je nach Vorgehensweise werden diese in induktive und deduktive Verfahren eingeteilt, siehe Kapitel 3.1.8. Bei beiden Techniken wird anhand von Ursache-Folge-Beziehungen das entsprechende Risiko bestimmt. Damit sind Aussagen zur Sicherheit möglich.

Diese bewährte Beurteilung soll ebenfalls bei der ganzheitlichen modellbasierten Sicherheitsanalyse realisiert werden. Die Beziehungen zwischen möglichen Ursachen und Folgen müssen dem Anwender anhand des modellierten Systemverhaltens rechnergestützt und systematisch aufgezeigt werden. Zum Verständnis des Ansatzes ist es zweckdienlich, die Ursache-Folge-Beziehung hinsichtlich der Systembestandteile zu untersuchen. Ursachen, die zu einem Schaden führen, sind Fehler. Wie in Abbildung 5.1 ersichtlich, können Fehler grundsätzlich in jedem Systembestandteil auftreten. Fehler können aber auch im Informationsaustausch zwischen Systembestandteilen auftreten. Diese Fehler haben Folgen, die sich in sicherheitskritischen Situationen des Prozessautomatisierungssystems äußern können. Wie in Kapitel 2.1.1 dargestellt, besitzt lediglich der technische Prozess ein Gefahrenpotenzial. Diese Überlegungen führen zu folgender Aussage: Ein Schaden für Mensch und Umwelt kann nur vom technischen System (in dem der technische Prozess abläuft) ausgehen. Die Ursachen für einen eingetretenen Schaden können hingegen im gesamten Prozessautomatisierungssystem lokalisiert sein.

Bei dem ganzheitlichen Ansatz werden die kausalen Zusammenhänge zwischen Systembestandteilen untersucht. Der große Vorteil liegt darin, dass durch die Verknüpfung verschiedener Sys-

tembestandteile das Zusammenwirken von physikalischen Vorgängen, Automatisierungskonzept und menschlichem Verhalten betrachtet wird.

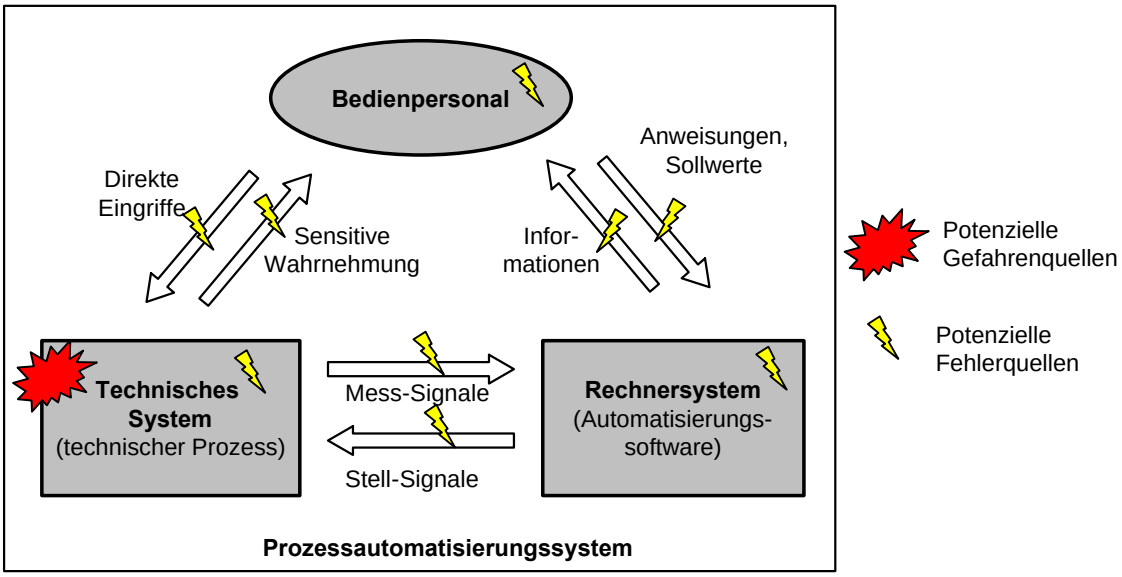


Abbildung 5.1: Fehler- und Gefahrenquellen eines Prozessautomatisierungssystems

Um dieses Zusammenwirken zu beschreiben, muss zuerst das Verhalten von technischem System, Automatisierungssoftware und Bedienpersonal separat in einem entsprechenden Modell festgehalten werden:

- Das Modell des technischen Systems soll (separat betrachtet) das physikalische Verhalten des technischen Prozesses im unkontrollierten, d.h. nicht-angesteuerten Zustand beschreiben. Bei einem Prozess mit Gefahrenpotenzial enthält das Modell auch sicherheitskritische Situationen.

- Das Modell der Automatisierungssoftware muss wiedergeben, wie der technische Prozess anzusteuern ist, damit erstens die gewünschten Vorgänge automatisiert ablaufen und zweitens dabei keine unvorhergesehenen sicherheitskritischen Zustände auftreten.

- Das Modell der Bedieneingriffe beschreibt die Möglichkeiten des Bedienpersonals, um Prozessvorgänge positiv oder negativ zu beeinflussen.

Werden die einzelnen Modelle der drei Systembestandteile zusammengefügt, dann sollten bei einem sicheren Prozessautomatisierungssystem keine sicherheitskritischen Situationen im Gesamtmodell auftreten, bzw. nur solche, deren Risiko als sehr gering eingestuft werden kann. Im Normalfall muss die Automatisierungssoftware so entworfen sein, dass im Betrieb des Prozessautomatisierungssystems keine sicherheitskritischen Situationen auftreten können. Im Idealfall sollte die Automatisierungssoftware daher keine Fehler enthalten, die bei der Steuerung des technischen Prozesses gefährliche Vorgänge auslösen können. Darüber hinaus sollte die ideale

Automatisierungssoftware Fehler im technischen System und falsche Bedieneingriffe erkennen und entsprechende Maßnahmen ausführen, damit daraus keine sicherheitskritischen Situationen entstehen können. Zum Beispiel muss die Automatisierungssoftware beim Überschreiten einer Grenztemperatur die „Heizung“ abschalten, um eine Überhitzung zu vermeiden. Wurden im Modell des technischen Systems Szenarien für eine Überhitzung modelliert, so dürfen diese im Gesamtmodell, d.h. im kontrollierten (bestimmungsgemäßen) Betrieb, nicht mehr vorhanden sein. Auch im nicht-bestimmungsgemäßen Betrieb, z. B. bei Ausfall eines Temperatursensors, ist zu gewährleisten, dass keine sicherheitskritischen Situationen auftreten. Verfügt die Heizanlage über einen Notausschalter, dann kann das Bedienpersonal bei Rauchentwicklung ebenfalls eine Überhitzung oder einen Brand vermeiden. In diesem Fall existieren zwei Mechanismen für das Vermeiden einer Überhitzung: einmal durch Eingriff der Automatisierungssoftware (Abschalten der Heizung) und durch Betätigen eines Notausschalters. Anhand des Gesamtmodells kann der Anwender prüfen, ob oder unter welchen Umständen sicherheitskritische Situationen vorhanden sind. Treten im Gesamtmodell zum Beispiel Szenarien für eine Überhitzung auf, so können gleichzeitig die Ursachen untersucht werden (z. B. Doppelfehler: Ausfall des Temperatursensors und Nichtbetätigen des Notausschalters).

Der ganzheitliche Ansatz bietet also den Vorteil, dass nicht nur verschiedene Fehlerarten der Systembestandteile berücksichtigt werden, sondern dass auch die Auswirkung von Kombinationen dieser Fehler analysiert werden können. Dazu müssen die Modelle der Systembestandteile nicht nur das bestimmungsgemäße Verhalten wiedergeben, sondern ebenfalls die möglichen Fehler (Ursachen) und Gefahren bzw. Schadenszenarien (Folgen) berücksichtigen. Der Zusammenhang zwischen möglichen Ursachen und Folgen soll dem Anwender anhand von Szenarien systematisch aufgezeigt und verdeutlicht werden. Die Auswertung zwischen Ursachen und ihren Folgen wird anhand des ganzheitlichen Modells realisiert.

Die Bedeutung der verschiedenen Fehlerarten ist für den ganzheitlichen Ansatz wichtig. Bei klassischen Sicherheitsanalysen stellt die Annahme von möglichen Fehlern (hypothetische Fehler) eine gängige Technik dar. Diese Art von Fehlern treten nach der Inbetriebnahme – z. B hervorgerufen durch Verschleiß oder Bedienfehler – auf. Diese Technik ist für die Betrachtung des technischen Systems und die menschlichen Bedieneingriffe sinnvoll, allerdings bei der Untersuchung der Automatisierungssoftware nicht anwendbar. Wie in Kapitel 2.1.2 erläutert, handelt es sich bei Softwarefehlern ausschließlich um inhärente Fehlerarten (Anforderungsfehler, Entwurfs- und Implementierungsfehler), die von Beginn an im System vorhanden sind, siehe Abbildung 5.2. Solche Fehler können z. B. im bestimmungsgemäßen Betrieb der Anlage keine Folgen haben und erst im nicht-bestimmungsgemäßen Betrieb sicherheitskritische Situationen hervorrufen. Im Rahmen des ganzheitlichen Ansatzes können inhärente Softwarefehler anhand ihrer Auswirkungen (Folgen) ermittelt werden.

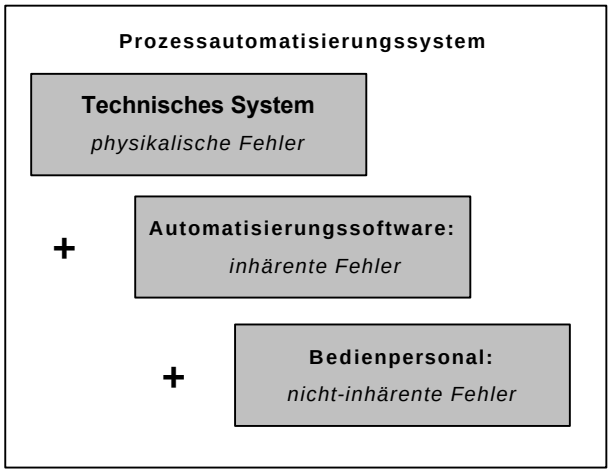


Abbildung 5.2: Fehlerarten eines Prozessautomatisierungssystems

Wird beachtet, dass die Systembestandteile wiederum aus mehreren Subsystemen aufgebaut sind und das Bedienpersonal gegebenenfalls aus mehreren Personen mit unterschiedlichen Aufgabenbereichen bestehen kann, so ist ersichtlich, dass sich die Beziehungen zwischen Ursachen und Folgen weitaus komplexer gestalten als in Abbildung 5.1 dargestellt. Erst mit einem ganzheitlichen modellbasierten Ansatz können komplexe Beziehungen von unterschiedlichen Fehlerarten systematisch untersucht werden.

5.2 Ganzheitliche modellbasierte Sicherheitsanalyse

Die Durchführung der modellbasierten Sicherheitsanalyse zur Behandlung von Prozessautomatisierungssystemen kann in zwei Tätigkeitsbereiche, nämlich in die *Modellerstellung* und in die *Modellanalyse*, unterteilt werden. In Abbildung 5.3 sind diese beiden Bereiche dargestellt. Darüber hinaus ist das Vorgehen in vier einzelne Schritte eingeteilt.

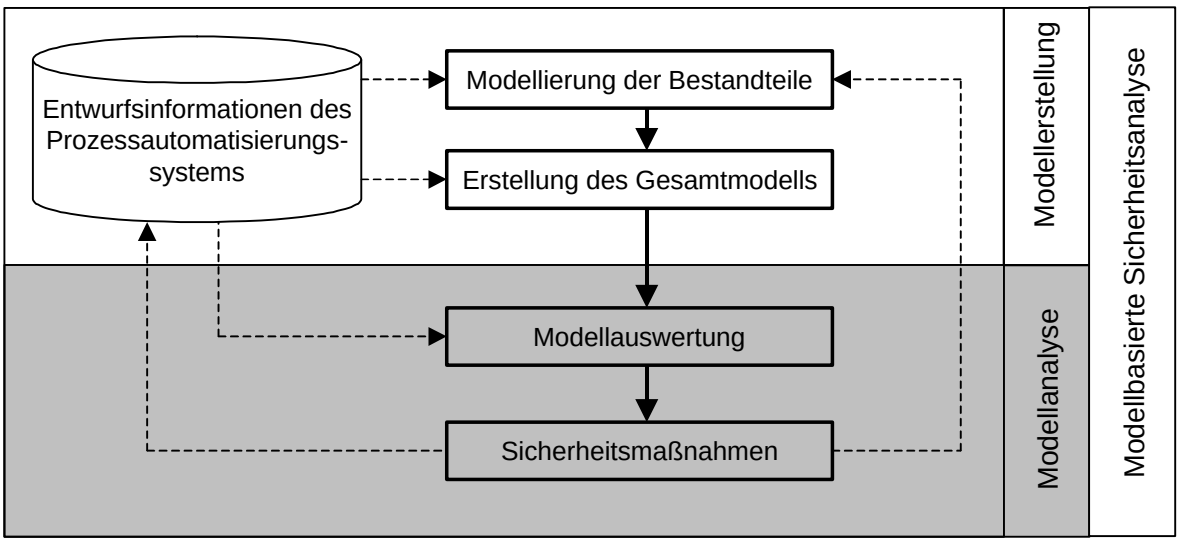


Abbildung 5.3: Durchführung der modellbasierten Sicherheitsanalyse

Der erste Schritt besteht in der *Modellierung der Bestandteile*. Für jeden Bestandteil wird ein entsprechendes Modell angefertigt. *Die Erstellung des Gesamtmodells* erfolgt im zweiten Schritt durch Kopplung der erstellten Teilmodelle. Die Basis für diesen Vorgang bildet die Struktur des Prozessautomatisierungssystems. Das Gesamtmodell soll das mögliche Verhalten des Prozessautomatisierungssystems sowohl im bestimmungsgemäßen Betrieb wie auch im Fehlerfall beschreiben.

Der zentrale Schritt der Sicherheitsanalyse ist die Auswertung des Gesamtmodells und die Überprüfung der bestehenden Sicherheitsvorschriften in Form von Sicherheitsanforderungen. Das Gesamtmodell wird auf sicherheitskritische Situationen hin untersucht und durch den Anwender bewertet. Falls das Gesamtmodell sicherheitskritische Situationen besitzt, müssen diese vom Anwender bezüglich ihres Risikos bewertet werden. Es ist zu unterscheiden, ob diese auf hypothetische Fehler zurückzuführen sind oder sich aus dem Zusammenspiel der Systembestandteile ergeben oder vorhandenen Sicherheitsanforderungen widersprechen:

- Sicherheitskritische Situationen, die auf hypothetische Fehler oder auch Fehlerkombinationen zurückzuführen sind, geben Hinweise auf Schwachstellen des Prozessautomatisierungssystems. Ist deren Eintreten wahrscheinlich oder das Schadenausmaß sehr hoch, so müssen im nächsten Schritt Sicherheitsmaßnahmen gefunden werden, um die vorhandenen sicherheitskritischen Situationen zu vermeiden.

- Sicherheitskritische Situationen, die im bestimmungsgemäßen Betrieb des Systems (d.h. ohne Annahme von Fehlern) auftreten, haben als Ursachen inhärente Fehler (z. B. Entwurfsfehler). Es handelt sich dabei um tatsächliche Fehler, die zu beheben sind.

- Sicherheitsanforderungen werden ebenfalls anhand des Gesamtmodells überprüft. Stehen die Aussagen der Sicherheitsanforderungen nicht im Einklang mit denjenigen des Gesamtmodells, so muss davon ausgegangen werden, dass diese nicht erfüllt sind. In diesem Fall müssen ebenfalls entsprechende Sicherheitsmaßnahmen getroffen werden.

Im letzten Schritt der modellbasierten Sicherheitsanalyse werden, sofern notwendig, *Sicherheitsmaßnahmen* definiert, um sicherheitskritische Situationen des Prozessautomatisierungssystems zu verhindern. Das Ziel von Sicherheitsmaßnahmen ist zum einen die Vermeidung von Fehlern oder das Senken ihrer Eintrittswahrscheinlichkeit (ursachenorientierte Maßnahmen), zum anderen das Reduzieren des Schadenausmaßes oder die Abwendung der Gefahr (folgenorientierte Maßnahmen). Definierte Sicherheitsmaßnahmen werden im Schritt *Modellerstellung* erneut berücksichtigt, um deren Effizienz zu prüfen. Bei einer erneuten Erstellung des Gesamtmodells sollten die entsprechenden sicherheitskritischen Situationen nicht mehr existent sein. Ein kleines Beispiel soll diesen Sachverhalt verdeutlichen: Wurde seither bei einem System zur Vermeidung einer Überhitzung nur ein Schutz in Form eines Notausschalters vorgesehen, so könnte eine entsprechende Sicherheitsmaßnahme folgendermaßen lauten: „Die Temperatur ist

zusätzlich durch die Automatisierungssoftware zu überwachen und beim Erreichen der Grenztemperatur T_{max} ist die Heizung abzuschalten.“ Wird diese zusätzliche Funktion im Teilmodell der Automatisierungssoftware berücksichtigt, so kann durch erneute Durchführung der modellbasierten Sicherheitsanalyse die Wirkung der Sicherheitsmaßnahme überprüft werden. Im Gesamtmodell sollten dann für den bestimmungsgemäßen Betrieb keine Szenarien mehr für den Fall „Überhitzung“ vorhanden sein.

Der ganzheitliche Ansatz berücksichtigt das Zusammenspiel der Systembestandteile im bestimmungsgemäßen sowie auch im nicht-bestimmungsgemäßen Betrieb. Es lassen sich tatsächlich vorhandene sicherheitskritische Situationen des Systems sowie auch mögliche Sicherheitslücken ermitteln. Die Durchführung der modellbasierten Sicherheitsanalyse besteht aus zwei Tätigkeitsbereichen, der Modellerstellung und der Modellanalyse. In den nächsten beiden Kapiteln werden diese Tätigkeitsbereiche detailliert betrachtet.

6 Modellierung eines Prozessautomatisierungssystems

Dieses Kapitel befasst sich zu Beginn mit Kriterien, die bei der Modellierung von komplexen Systemen wichtig sind. Anschließend wird die qualitative Modellierung des technischen Systems, der Automatisierungssoftware und der menschlichen Bedieneingriffe diskutiert. Die Kombination der einzelnen Modelle zu einem Gesamtmodell eines Prozessautomatisierungssystems wird dargestellt.

6.1 Modellierungsprinzip für komplexe Systeme

Bei der Entwicklung von komplexen und umfangreichen Systemen wird im Allgemeinen der Ansatz der hierarchischen Dekomposition verfolgt [Marq95], siehe Abbildung 6.1. Bei der Systementwicklung wird ein geplantes System generell in kleine und überschaubare Subsysteme bis hin zu Systemelementen zerlegt. Dieser Vorgang wird als hierarchische Dekomposition oder als „top-down“ Zerlegung bezeichnet. Der Aufbau und das Verhalten dieser Systemelemente werden in der Entwurfsdokumentation des Prozessautomatisierungssystems spezifiziert. Genau an dieser Stelle setzt die Modellerstellung an. Sie muss im Rahmen der ganzheitlichen modellbasierten Sicherheitsanalyse entwickelt werden und ihr Ziel ist die Aggregation der modellierten Systemelemente zu einem Gesamtsystem. Dabei handelt es sich um eine kopplungsorientierte „bottom-up“ Verknüpfung.

Die Entwurfsdokumentation des Systems dient als Ausgangspunkt, um das Verhalten der Systemelemente mit Hilfe eines geeigneten Modellierungskonzeptes zu beschreiben. Bei der geplanten Modellaggregation ist es notwendig, dass die Verhaltensbeschreibung der Systemelemente auf Basis einer Schnittstellenbetrachtung erfolgt. Nur so ist eine bottom-up Verknüpfung zu einem System möglich. Dazu müssen die *Schnittstellen* und das *Verhalten* der Systemelemente modelliert werden. Die Verhaltensbeschreibung umfasst sämtliche Eigenschaften, die das Innere eines Systemelements charakterisieren. Die Schnittstellen dagegen beschreiben alle diejenigen Eigenschaftsattribute, mittels denen ein Systemelement mit seiner Umgebung Informationen austauschen kann. Dabei werden die Kommunikationswege durch die Struktur des Systems festgelegt. Schnittstellen stellen somit das Bindeglied zwischen dem Inneren und dem Äußeren eines Systems dar und sind Bestandteile einer Systembeschreibung auf allen Hierarchieebenen.

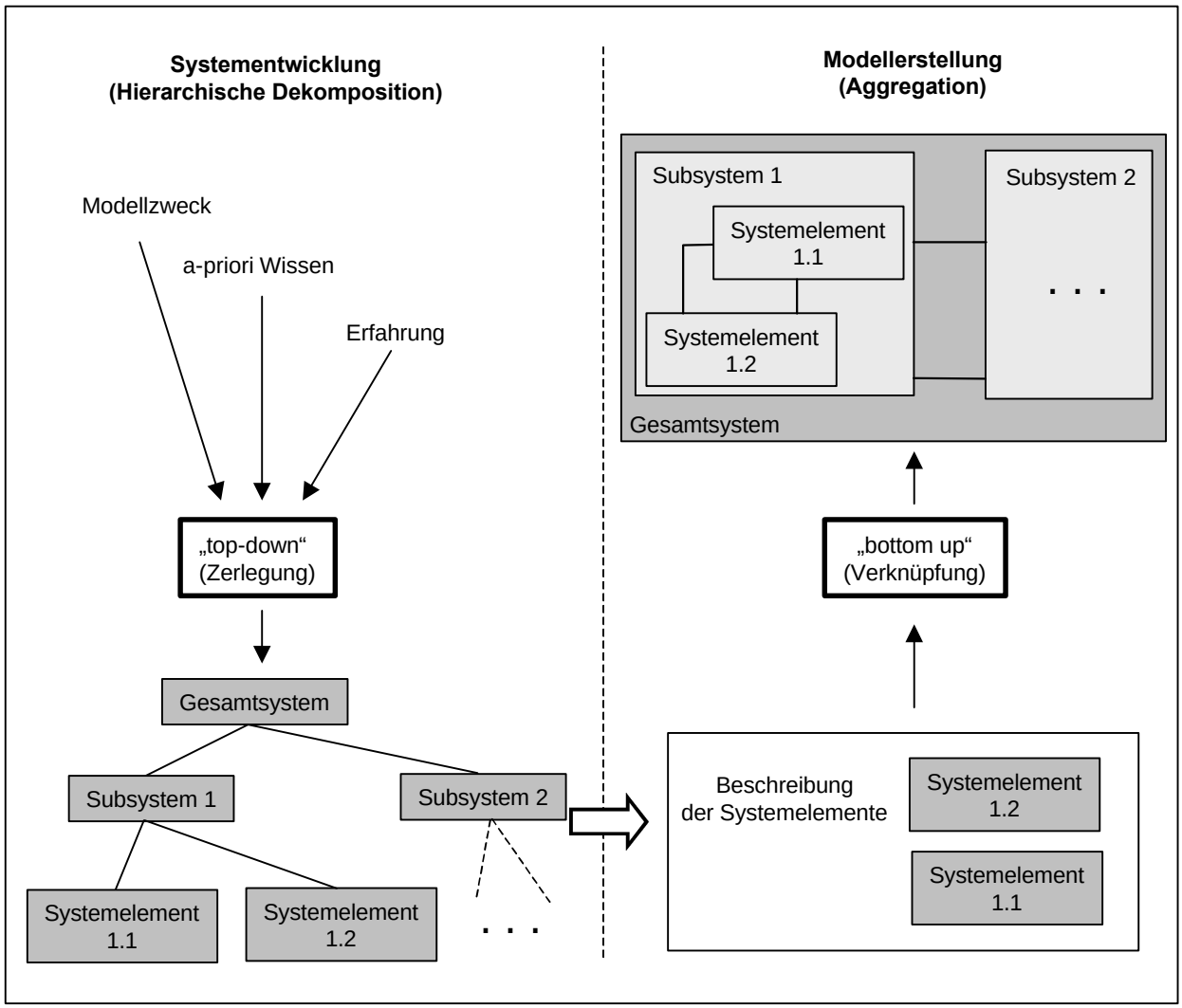


Abbildung 6.1: Vorgehen bei der Modellierung komplexer Systeme

Die Teilmodelle des technischen Systems, der Automatisierungssoftware und der menschlichen Bedieneingriffe müssen zu einem ganzheitlichen Modell zusammengefügt werden. Die Teilmodelle werden wie Subsysteme behandelt und gemäß der Systemstruktur des Prozessautomatisierungssystems verknüpft. In Kapitel 2.1.4 wurden die grundlegenden Strukturen vorgestellt. Bei realen Systemen existieren häufig gemischte Strukturen. Teilen sich die Automatisierungssoftware und das Bedienpersonal Aufgaben bei der Prozessführung, so treten sowohl *open-loop* als auch *closed-loop* Verbindungen auf. Die vorhandene bzw. geplante Sensorik und Aktorik bestimmt den Umfang des Informationsaustauschs zwischen technischem System und der Automatisierungssoftware. Meldungen bzw. erkennbare Prozessereignisse, die einem zuständigen Operator zur Verfügung stehen, sind auch anhand von Schnittstellen und Verbindungen zu abstrahieren. Besitzt z. B. ein Gasbrenner ein Sichtfenster zur Kontrolle des Brennvorgangs, so ist das Sichtfenster als Schnittstelle des technischen Bauelements Gasbrenner zu realisieren. Die verfügbare Information für einen Operator, der den Brennvorgang kontrolliert, hat binären Charakter (Brennvorgang an oder aus). Anweisung und Vorgaben des Operators an die

Automatisierungssoftware geben Aufschluss zwischen den Verbindungen der entsprechenden Teilmodelle.

Die Verbindungen der Systembestandteile wird in dieser Arbeit als *Systemstruktur* eines Prozessautomatisierungssystems bezeichnet. Ein in der Modellierungstechnik synonym verwendeter Begriff ist die Systemtopologie. Auf Basis der Systemstruktur erfolgt die Kopplung der Teilmodelle zu einem Gesamtmodell eines Prozessautomatisierungssystems. Das Gesamtmodell beschreibt, basierend auf dem Zusammenspiel von technischem System, Automatisierungssoftware und menschlichen Bedieneingriffen, das Verhalten des Prozessautomatisierungssystems im bestimmungsgemäßen und nicht-bestimmungsgemäßen Betrieb.

6.2 Qualitative Modellierung des technischen Systems

Die Modellierung von technischen Systemen mit SQMA wurde in den Arbeiten [Fröh97] und [Lauf97] untersucht. Im Folgenden werden die für eine ganzheitliche Modellierung notwendigen Ergänzungen betrachtet. Das technische System enthält im allgemeinen Fall Schnittstellen zu Bedienpersonal und Automatisierungssoftware. Da SQMA bisher nur eine Art von Schnittstellen kennt, müssen neue Schnittstellenarten eingeführt werden.

6.2.1 Qualitative Modellierung von technischen Bauelementen

Technische Bauelemente und Baugruppen

Beim technischen System können die zu modellierenden Systemelemente meistens eins zu eins aus der vorhandenen Dokumentation entnommen werden. Die Systemelemente des technischen Systems werden im Verlauf dieser Arbeit als *technische Bauelemente* und ein Subsystem (bestehend aus mehreren technischen Bauelementen) als *technische Baugruppe* bezeichnet. Die physikalischen Vorgänge in technischen Bauelementen bestimmen in erster Linie ihre Funktion und damit ihr Verhalten. Das Verhalten kann durch Fehler des technischen Bauelements maßgeblich von der normalen Funktionsweise abweichen. Je nach Art des technischen Prozesses kann der physikalische Vorgang im technischen Bauelement ein Gefahrenpotenzial besitzen. Geht ein möglicher Schaden für Mensch und Umwelt von einem technischen Bauelement aus, so muss dies bei der Modellierung eines Bauelements berücksichtigt werden.

Der Name eines technischen Bauelements wird durch die vorangestellten Buchstaben „*ts_*“ gekennzeichnet, zum Beispiel „ts_Behälter“. Die Bezeichnung „ts_“ steht für technisches System und charakterisiert das Systemelement als technisches Bauelement. Dies vereinfacht eine rechnergestützte Identifikation von technischen Bauelementen im Gesamtmodell des Prozessautomatisierungssystems.

Schnittstellen von technischen Bauelementen

Im Hinblick auf die Schnittstellenbeschreibung von technischen Bauelementen ist es sinnvoll, zwischen *aktiven Bauelementen* und *passiven Bauelementen* zu differenzieren. Aktive technische Bauelemente sind Aktoren oder Sensoren und besitzen mindestens eine Schnittstelle zu Systemelementen anderer Systembestandteile. Beispiele für aktive Bauelemente sind Ventile und Füllstandsmelder. Passive Bauelemente zeichnen sich dadurch aus, dass sie nur Schnittstellen zu anderen technischen Bauelementen besitzen. Rohre und Behälter sind typische Beispiele für passive Bauelemente.

Die genaue Anzahl der Schnittstellen ist vom speziellen Bauelement abhängig. Beispielsweise verfügt ein Sensor zur Erfassung einer entsprechenden Messgröße über eine physikalische Schnittstelle und zur weiteren Verarbeitung über eine Schnittstelle, an der die äquivalente Informationsgröße abgegriffen werden kann. Ein aktives Bauelement kann daher im Unterschied zu einem passiven Bauelement, welches nur Schnittstellen für physikalische Größen enthält, zusätzlich beliebig viele Schnittstellen für Informationsgrößen und Bedieneingriffe besitzen, siehe Abbildung 6.2.

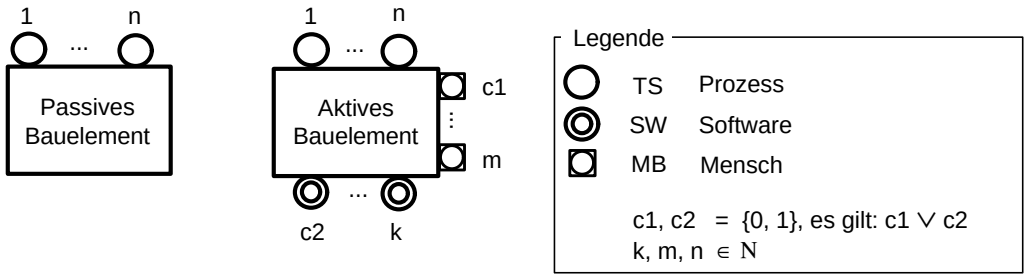


Abbildung 6.2: Schnittstellenbetrachtung von passiven und aktiven Bauelementen

Ein technisches Bauelement kann daher im Rahmen der ganzheitlichen Betrachtungsweise drei verschiedene Arten von Schnittstellen besitzen. Je nach Art der anliegenden Größen werden Schnittstellen vom Typ Prozess (TS), Software (SW) und Mensch (MB) definiert.

Modellgrößen von technischen Bauelementen

Die Definition der Modellgrößen erfolgt wie in Kapitel 4.1 beschrieben. Bei der Verhaltensbeschreibung von aktiven Bauelementen ist jedoch zusätzlich die Kopplung zwischen Ansteuerung bzw. Bedienung und physikalischem Verhalten zu berücksichtigen. Bei Aktoren beeinflussen Informationsgrößen bzw. Bedieneingriffe die physikalischen Größen und bei Sensoren beeinflussen die Werte der technischen Größen den Inhalt von Informationsgrößen oder Meldungen. Die allgemeine Form einer solchen Verknüpfung lautet für Sensoren:

```
// informelle Erklärung der Situationsregel;
logischerAusdruck(technische Größe),dann logischerAusdruck(Informationsgröße)
```

Beispiel „Grenzdruck-Melder“

```
// Wenn Grenzdruck überschritten, gib Alarm
      GrenzDruck > 100mbar,dann Alarm = "ein";
```

und für Aktoren:

```
// informelle Erklärung der Situationsregel
logischerAusdruck(Informationsgröße),dann logischerAusdruck(technische Größe)
```

Beispiel „Ventil“

```
// Ist die Stellgröße gesetzt, dann ist Druckabfall im Ventil gleich null
      Stellgröße = 1, dann Ventildruck = 0;
```

Die Möglichkeit, Wertebereiche oder auch Kombinationen von Modellgrößen mit qualitativen Ausdrücken zu assoziieren, stellt die eigentliche Stärke des SQMA-Verfahrens dar. Die Qualität des Modells steigt und sinkt allerdings auch mit den gewählten qualitativen Ausdrücken. Erschwerend kommt hinzu, dass bei SQMA keinerlei Vorschriften bzw. Richtlinien zur Bildung von qualitativen Ausdrücken bestehen. Das angestrebte Gesamtmodell soll anhand von qualitativen Ausdrücken spezielle Betriebsarten des Prozessautomatisierungssystems eindeutig wiedergeben und muss ebenfalls das Zusammenspiel der Bestandteile verständlich darstellen. Somit ist der Umgang mit qualitativen Ausdrücken näher zu betrachten.

Bei der informellen Beschreibung technischer Systeme muss nach [Ortn97] zwischen zwei Kategorien unterschieden werden. Die erste Kategorie beschäftigt sich mit dem Aufbau technischer Systeme bzw. ihrer Struktur und besitzt bezüglich der Kommentarregeln keine Relevanz. Die zweite Kategorie betrifft die Verhaltensbeschreibung: Hier wird die logisch-kausale bzw. zeitlich-kausale Beschreibung von Handlungen und Abläufen behandelt. Dabei werden drei unterschiedliche Prädikationen betrachtet, nämlich

- die *Dingprädikation* (Substantive),
- die *Geschehnisprädikation* (Verben) und
- die *Eigenschaftsprädikation* (Adjektive, Adverbien).

Die Prädikation beschäftigt sich mit der Relation zwischen Subjekt und Prädikat bzw. mit der Zuordnung von Eigenschaften zu Objekten oder Sachverhalten [Bußn00]. Die Prädikation ist bei einem Objekt im Deutschen eindeutig, hingegen kann bei vielen Objekten die Interpretation nicht mehr eindeutig sein [Hack00].

Bei der Verhaltensbeschreibung lassen sich zwei Arten von Kommentaren unterscheiden: die Beschreibung eines Vorgangs und die Beschreibung einer Eigenschaft. Die Größen eines technischen Bauelements stellen entweder eine bestimmte Eigenschaft oder Vorgang dar. Die Beschreibung einer Eigenschaft erfolgt in der Regel durch Adjektive (z. B. offen, geschlossen) und die eines Vorgangs durch Verben (z. B. zufließen, abfließen).

In der situationsbasierten Ergebnisdarstellung, siehe auch Tabelle 4.2, sind die Namen der technischen Bauelemente in der Tabellenüberschrift aufgeführt. Die Prädikation erfolgt in Form eines *ist*-Bezugs und es besteht eine Relation zwischen den qualitativen Ausdrücken und dem Namen des Bauelements. Der Ausdruck *Wassertank: {„voll“, „leer“, „gefüllt“}* ist folgendermaßen zu lesen: der Wassertank ist voll, leer oder gefüllt. Dieser Bezug ist auch bei Stoffeigenschaften gewährleistet, falls diese aus der Funktion des technischen Bauelements oder aus dessen Name eindeutig hervorgehen. Zum Beispiel ist der Ausdruck *Wassertank: {„heiß“, „kalt“, „warm“}* folgendermaßen zu interpretieren: das Wasser im Wassertank ist heiß, kalt oder warm. Bei komplexeren technischen Bauelementen, die mehrere Bauteile enthalten können, fehlt häufig ein solcher Bezug. Dieser muss durch einen Dingprädikator[2] hergestellt werden. Zum Beispiel ergibt *Wassertank: {„an“, „aus“}* auf den ersten Blick keinen Sinn, allerdings *Wassertank: {„Heizung_an“, „Heizung_aus“}* schon. Diese Überlegungen führen zu folgendem Syntaxdiagramm (Abbildung 6.3). Anstelle eines Leerzeichens wird ein Unterstrich („_“) verwendet, um ein Substantiv mit einem Adjektiv zu verbinden.

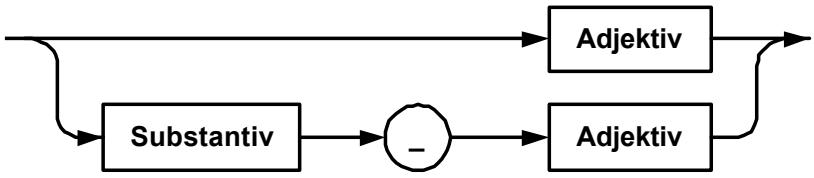


Abbildung 6.3: Syntaktischer Aufbau eines qualitativen Ausdrucks zur Beschreibung von Eigenschaften

Ebenfalls sollen zur besseren Verständlichkeit Vorgänge in einem technischen Bauelement nicht, wie in [Ortn97] vorgeschlagen, durch ein Verb charakterisiert werden, sondern durch die Substantivierung des Verbs. Anstatt der Beschreibung *Wassertank: {zufließen, abfließen}* ist die Beschreibung *Wassertank: {Zufluss, Abfluss}* vorzuziehen. Mit Hilfe des Wortes „Kein“ kann explizit ausgedrückt werden, dass ein Vorgang nicht stattfindet. In der Regel beziehen sich qualitative Ausdrücke mit „Kein“ auf das entartete Intervall [0,0] einer Größe. Der Vorgang kann mit Hilfe eines Adjektivs präziser beschrieben werden, wie z. B. „starker_Zufluss“. Abbildung 6.4 zeigt den syntaktischen Aufbau von qualitativen Ausdrücken zur Beschreibung von Vorgängen.

[2] Anmerkung zur Definition nach [Ortn97]: besser ist Objektprädikator

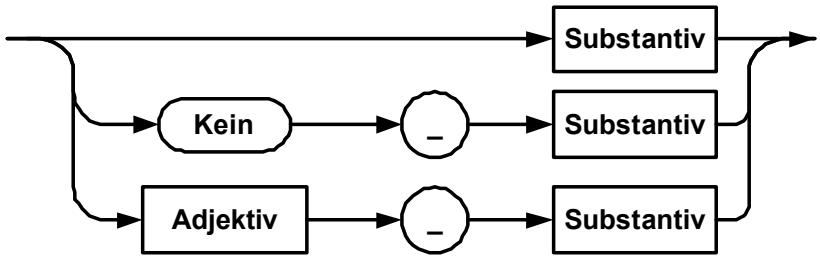


Abbildung 6.4: Syntaktischer Aufbau eines qualitativen Ausdrucks zur Beschreibung von Vorgängen

Zur Realisierung der modellbasierten Sicherheitsanalyse ist es wichtig, die verschiedenen Vorgänge in einem Bauelement oder die Eigenschaften des darin enthaltenen Stoffes gemäß ihrer Bedeutung für die Sicherheit zu klassifizieren. Dies stellt den eigentlichen Mechanismus zur rechnergestützten Identifikation von sicherheitskritischen Situationen dar. Bei der Verhaltensbeschreibung von technischen Bauelementen lässt sich grundsätzlich zwischen *bestimmungsgemäßem* und *nicht-bestimmungsgemäßem* Verhalten unterscheiden.

- Das *bestimmungsgemäße Verhalten* stellt den normalen bzw. den erwarteten Betrieb[3] eines technischen Bauelements dar. Die technischen Größen befinden sich alle im vorgesehenen Arbeitsbereich (Intervall). Das bestimmungsgemäße Verhalten ist für eine Sicherheitsanalyse nur von sekundärem Interesse. Daher sollte dieser Arbeitsbereich anhand eines Intervalls zusammengefasst und entsprechend seiner Bedeutung kommentiert und klassifiziert werden.
- Das *nicht-bestimmungsgemäße Verhalten* umfasst alle Abweichungen vom Normalbetrieb. Es findet weiterhin eine Unterteilung in *fehlerhaftes Verhalten* und *gefährliches Verhalten* statt, siehe Abbildung 6.5.

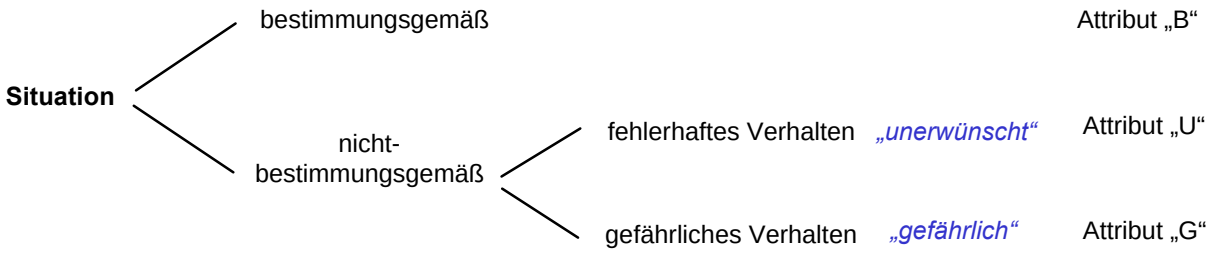


Abbildung 6.5: Klassifizierungsschema von qualitativen Ausdrücken

- *Unerwünschtes Verhalten* wird durch einen oder mehrere Fehler im technischen Bauelement charakterisiert. Es findet eine Abweichung vom bestimmungsgemäßen Betrieb statt. Diese Tatsache wird durch das Attribut „U“ gekennzeichnet. Bei der Betrachtung des Zusammenspiels der Systembestandteile sind mögliche Fehler bei aktiven Bauelementen von besonde-

[3] In der Literatur wird auch vom Normalbetrieb gesprochen.

rem Interesse. So kann z. B. ein verklemmtes Ventil unabhängig vom Stell-Signal entweder nicht mehr geschlossen oder nicht mehr geöffnet werden. Ein verklemmtes Ventil stellt ein unerwünschtes Verhalten dar. Ebenfalls liefern defekte Sensoren unabhängig von realen Vorgaben entweder keine oder falsche Werte. Ein fehlerhaftes aktives technisches Bauelement besitzt daher im allgemeinen Fehlerfall ein *nicht-deterministisches* Verhalten. Derartige Fehler müssen im Modell eines aktiven technischen Bauelements nicht explizit berücksichtigt werden. Die Untersuchung des Gesamtsystems bei nicht-deterministischem Verhalten von einzelnen oder mehreren Aktoren bzw. Sensoren wird in Kapitel 7.2.4 erläutert.

- *Gefährliches Verhalten* ergibt sich entweder aus einem gefährlichen Vorgang oder einer gefährlichen Eigenschaft (Stoffeigenschaft), mit der ein Mensch oder die Umwelt unmittelbar in Berührung kommt. Der Anwender muss prüfen, ob vom betrachteten technischen Bauelement eine Gefahr für Mensch und Umwelt ausgehen kann. So stellt zum Beispiel heißes Wasser im Wassertank noch keine unmittelbare Gefahr dar. Heißes Wasser, das durch einen Fehler oder durch unsachgemäße Bedienung bzw. Ansteuerung aus dem Tank entweichen kann, hingegen schon. Gefährliche Vorgänge sind im Wesentlichen durch deren Energie (z. B. kinetische Energie, elektrische Energie) gekennzeichnet, die auf Mensch und Umwelt einwirken kann. Bei direktem Umgang mit Gefahrgütern bestimmen deren spezifische Stoffeigenschaften die potenziellen Gefahren. Das Gefahrenpotenzial und die Eigenschaften von konkreten Stoffen sind in [GefS99] zusammengestellt.

6.2.2 Modellerstellung des technischen Systems

Die Modellierung des technischen Systems erfolgt im Wesentlichen, wie in Kapitel 4.2 beschrieben, durch Kombination der qualitativen Modelle einzelner Bauelemente. Die Schnittstellen der aktiven Bauelemente vom Typ „Mensch“ (MB) und „Software“ (SW) werden bei der Kombination ignoriert. Das bedeutet, dass nur Schnittstellen vom Typ „Prozess" (TS) untereinander verbunden werden. Die Schnittstellen vom Typ SW und MB bleiben offen. Bei der Komposition der modellierten Bauelemente werden für offene Schnittstellen keine Systemgleichungen erstellt. Das bedeutet, dass die anliegenden Größen an diesen offenen Schnittstellen auch keiner Bedingung unterstellt sind. Das Modell gibt daher das mögliche Verhalten des technischen Systems im unkontrolliertem (nicht angesteuerten) Zustand wieder.

Die in [Pola98], [Fehr98], [Manz97] gemachten Erfahrungen zeigen, dass Modellierungsfehler hauptsächlich durch inkonsistente Beschreibungen innerhalb einer Systemelementdefinition oder durch syntaktische Fehler bei der Verwendung der Beschreibungsmittel begangen wurden. Diese Fehler lassen sich durch eine Unterstützung bei der Modellierung gut minimieren, da die gemachten Annahmen von einander abhängen. So bestimmt beispielsweise der Schnittstellentyp die Art der anliegenden Intervallgrößen. Die Festlegung des Wertebereiches der Intervallgrößen

beeinflusst wiederum die mögliche Formulierung der Situations- und Kommentarregeln. Diese Abhängigkeiten lassen sich durch Computerprogramme gut überprüfen, siehe Kapitel 7.5.

6.3 Qualitative Modellierung der Automatisierungssoftware

6.3.1 Allgemeine Aspekte zur Modellierung der Automatisierungssoftware

Die Systemelemente der Automatisierungssoftware werden im weiteren Verlauf der Arbeit als *Softwarebausteine* bezeichnet. Mehrere Softwarebausteine bilden ein *Softwaremodul*. Im Gegensatz zu technischen Bauelementen existieren für Softwarebausteine eine Vielzahl an unterschiedlichen Varianten, deren konkreter Aufbau letztendlich von der eingesetzten Entwurfstechnik geprägt ist. Softwarebausteine müssen aus den Entwurfsunterlagen der Automatisierungssoftware identifiziert werden können.

Da Softwarebausteine meistens für jedes Prozessautomatisierungssystem individuell entwickelt werden, d.h. dass in der Praxis keine Standard-Softwarebausteine[4] existieren und der Entwurf der Automatisierungssoftware dem des technischen Systems an Umfang und Komplexität übersteigt, ist eine rechnergestützte Modellerstellung dieses Systembestandteils anzustreben. Ebenfalls soll dabei Fehlern bei der Modellerstellung vorgebeugt werden. Ein weiterer Aspekt der rechnergestützten Umsetzung eines Softwareentwurfs in ein entsprechendes Modell besteht darin, dass auf diese Weise im Entwurf enthaltene Konzeptionsfehler (inhärente Fehler) ebenfalls im Modell berücksichtigt werden.

Um Modellierungsfehlern vorzubeugen und Zeit bei der Modellierung einzusparen, ist eine rechnergestützte Transformation des Softwareentwurfs in eine situationsbasierte Modellbeschreibung zu realisieren (siehe Anforderung 8, in Kapitel 3.4). Die Reproduktion der Modellierung ist durch eine Transformationsvorschrift somit ebenfalls durch Dritte nachzuvollziehen (Anforderung 4).

Die Erstellung einer Transformationsvorschrift ist durch den Vorgang der Modellaggregation geprägt. Damit die modellierten Softwarebausteine verknüpft werden können, müssen die Schnittstellen des Softwarebausteins qualitativ definiert werden. Es ist festzustellen, welche Informationsgrößen vom Softwarebaustein erzeugt (Ausgangsgrößen) und welche zur Verarbeitung verwendet werden (Eingangsgrößen). Relationen zwischen Ausgangsgrößen und Eingangsgrößen sind durch das Verhalten des Softwarebausteins festgelegt.

---

[4] Bemerkung: Die Standardisierung von Softwarebausteinen stellt ein wichtiges Forschungsgebiet der Softwaretechnik, aber auch der Automatisierungstechnik dar. Vielversprechend ist die Standardisierung von Softwarebausteinen auf der Basis einer komponentenbasierten Entwurfstechnologie [EEK00], [Göhn98] etc.

Um eine entsprechende Transformationsvorschrift zu erstellen, müssen im ersten Schritt die eigentlichen Softwarebausteine im Entwurf identifiziert werden. In Tabelle 6.1 sind die wichtigsten Entwurfstechniken (siehe auch Kapitel 2.1.2) aufgeführt. Daraus geht hervor, dass Softwarebausteine Objekte, Subsysteme oder selbst Komponenten sein können. Diese Softwarebausteine werden als qualitative SQMA-Komponenten modelliert. Die Informationsgrößen, die an Schnittstellen auftreten, sind im zweiten Schritt mit qualitativen Intervallvariablen zu beschreiben. Nach der Schnittstellenbetrachtung kann gemäß SQMA-Verfahren ein vollständiger Situationsraum eines Softwarebausteins aufgestellt werden. Dieser setzt sich aus Kombinationen aller möglichen Werte der Informationsgrößen zusammen und enthält daher auch alle Kombination zwischen Ein- und Ausgangsgrößen.

Tabelle 6.1: Eigenschaften verschiedener Entwurfstechniken

	Datenfluss / Kontrollfluss-orientierte Technik	**Objektorientierte Technik**	**Komponentenorientierte Technik**
Beispiel für Entwurfs-methoden	SA, SA RT, SD	UML, OMT	UML-RT, ACPLT
Softwarebaustein	Systeme, Subsysteme	Objekte, Module	Komponenten
Schnittstellen	über Daten- & Signalaustausch	definierter Zugriff (über Methoden) auf innere Objekt-Größen (Attribute)	definierte Ein- und Ausgangsgrößen (Protokolle)
Verhaltensbeschreibung	Entscheidungstabellen Zustandsdiagramme Prozessaktivierungstabellen	Zustandsdiagramme	Zustandsdiagramme
Struktur/Verknüpfung	über Datenfluss und z. T auch Kontrollfluss (Datenfluss- und Kontroll-flussdiagramme)	durch Relation der Objekte (Objektdiagramme)	Kommunikationskanäle (Strukturdiagramme)

Je nach Entwurfstechnik kann die Verhaltensbeschreibung in Form von Entscheidungstabellen, Zustandsdiagrammen, Petri Netzen oder Pseudocode vorliegen. Die Verhaltensbeschreibung eines Softwareentwurfs legt definierte Beziehungen zwischen Ein- und Ausgangsgrößen fest. Situationen, die der Verhaltensbeschreibung widersprechen, werden aus dem Situationsraum entfernt. Für diesen Vorgang ist es notwendig, die Mittel zur Verhaltensbeschreibung einer Entwurfstechnik auf die Situations-, Kommentar- und Transitionsregeln von SQMA abzubilden.

Abbildung 6.6 soll diesen Vorgang erläutern. Das Subsystem „Vergleicher" steht mit den Subsystemen „PrimärWertErfassung“ und „SekundärWertErfassung“ in Verbindung. Das Verhalten ist in Form einer Entscheidungstabelle definiert. Der Vergleicher besitzt die binären Eingangsgrößen A, B und die Ausgangsgröße C. Der vollständige Situationsraum besteht daher aus 2 * 2 * 2 = 8 Situationen. Die Aussage der Entscheidungstabelle wird in Situationsregeln überführt. Kommentarregeln erläutern die Funktionsweise des Vergleichers. Situationen, die der Verhal-

tensbeschreibung widersprechen entfallen aus dem Situationsraum (z. B. A = 0 & B = 0 & C = 0). Die verbleibenden Situationen werden kommentiert und sind mögliche Szenarien für den Softwarebaustein „Vergleicher“.

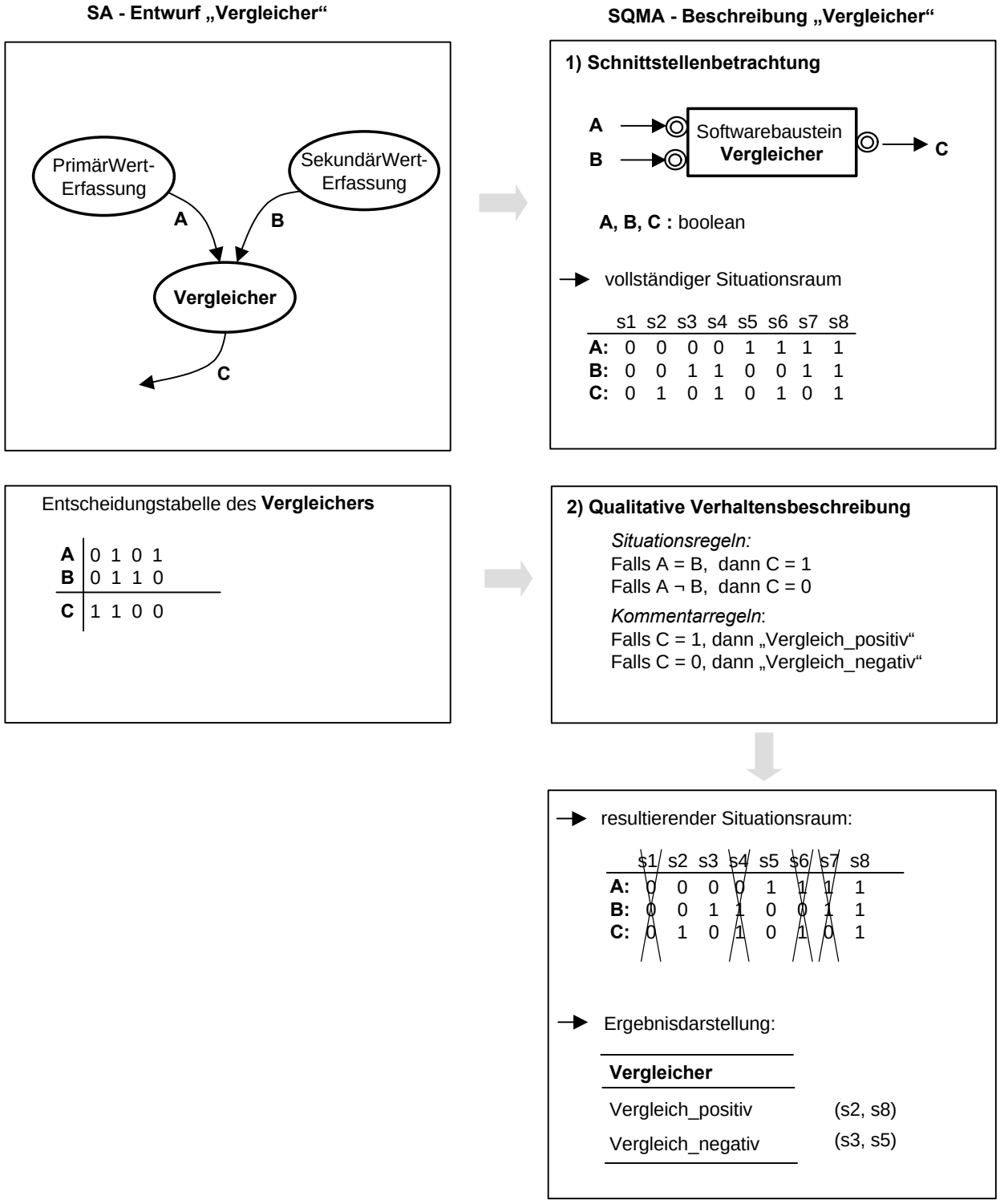


Abbildung 6.6: Beispiel einer Transformation in ein SQMA Modell

Zur Verknüpfung der modellierten Softwarebausteine müssen die Informationen, die zur Erstellung einer Netzliste notwendig sind, ebenfalls aus dem Entwurf abgeleitet werden, siehe Tabelle 6.1 (letzte Zeile). Aus der Netzliste werden Kopplungsbedingungen erstellt. Die Kopplungsbedingungen beschreiben die vorhandenen Kommunikationsmöglichkeiten der Softwarebausteine bzw. Softwaremodule. Analog zum Modell des technischen Systems, gibt das Modell der Automatisierungssoftware das Verhalten in Form von Situationen an, die sich aus der Kombination von Situationen der enthaltenen Softwarebausteine ergeben.

Für die Entwurfsmethode UML-RT wird im Folgenden eine konkrete Transformationsvorschrift entwickelt. Diese ermöglicht eine Überführung eines UML-RT Softwareentwurfs in ein äquivalentes qualitatives Modell nach SQMA. Zur Modellierung von Informationsgrößen müssen die Beschreibungsmittel von SQMA erweitert werden.

6.3.2 Qualitative Modellierung der Automatisierungssoftware auf der Basis von UML-RT

6.3.2.1 Einführung in UML-RT

UML-RT entstand gegen Ende der neunziger Jahre durch die Integration der Methode ROOM [SGW94] in den UML-Standard. UML und UML-RT verfolgen allerdings zwei unterschiedlichen Entwurfstechniken [Lyon98]. Während bei UML eine objektorientierte Softwareentwicklung unterstützt wird [BRJ99], handelt es sich bei UML-RT um eine komponentenorientierte Entwicklungsmethode. Es wird sowohl die Entwicklung von neuen Komponenten als auch die Verwendung und Parametrisierung von vorhandenen Komponenten unterstützt. Auf Basis der Darstellungsmittel von UML lassen sich Komponenten in Form von zusammengesetzten Objekten beschreiben. Bei UML-RT heißen die Komponenten *Kapseln* (*Capsule*) und besitzen eindeutige Funktionen. Kapseln stellen nach außen vollständige und abgeschlossene Einheiten dar [Rati00]. Die Beschreibung einer Kapsel umfasst ein Strukturmodell und eine Verhaltensmodellierung.

Kapseln können mit ihrer Umgebung nur über fest definierte Schnittstellen – die *Ports*[5] (*ports*) – kommunizieren. Jeder Port ist einem Protokoll (*protocol*) eindeutig zugeordnet. Protokolle bestimmen die Informationsgrößen, die gesendet oder empfangen werden können. Dabei werden im Wesentlichen zwei elementare Informationsgrößen unterschieden: Signale (*signals*) und Daten (*data*). Signale haben binären Charakter und lösen im Allgemeinen Ereignisse aus. Daten sind beliebige Größen und werden für den Austausch von zusätzlichen Informationen oder Bedingungen für das Auslösen von Ereignissen benötigt. Die Elemente einer Kapsel können mit den Darstellungsmitteln von UML als Objekte definiert werden. In Abbildung 6.7 ist eine Kap-

[5] Der englische Begriff „port“ wurde in der deutschen Literatur übernommen.

sel mit dem Namen „capsule1“ dargestellt. Diese Kapsel besitzt zwei Ports („port1“ und „port2“). Aus der Objektdarstellung von „capsule1“ ist zu sehen, dass dem „port1“ das Protokoll „protocol1“ und dem port2 das Protokoll „protocol2“ zugeordnet ist. Die Definition eines Protokolls geht ebenfalls aus Abbildung 6.7 hervor. Es wird zwischen Eingangsgrößen (incoming) und Ausgangsgrößen (outgoing) unterschieden.

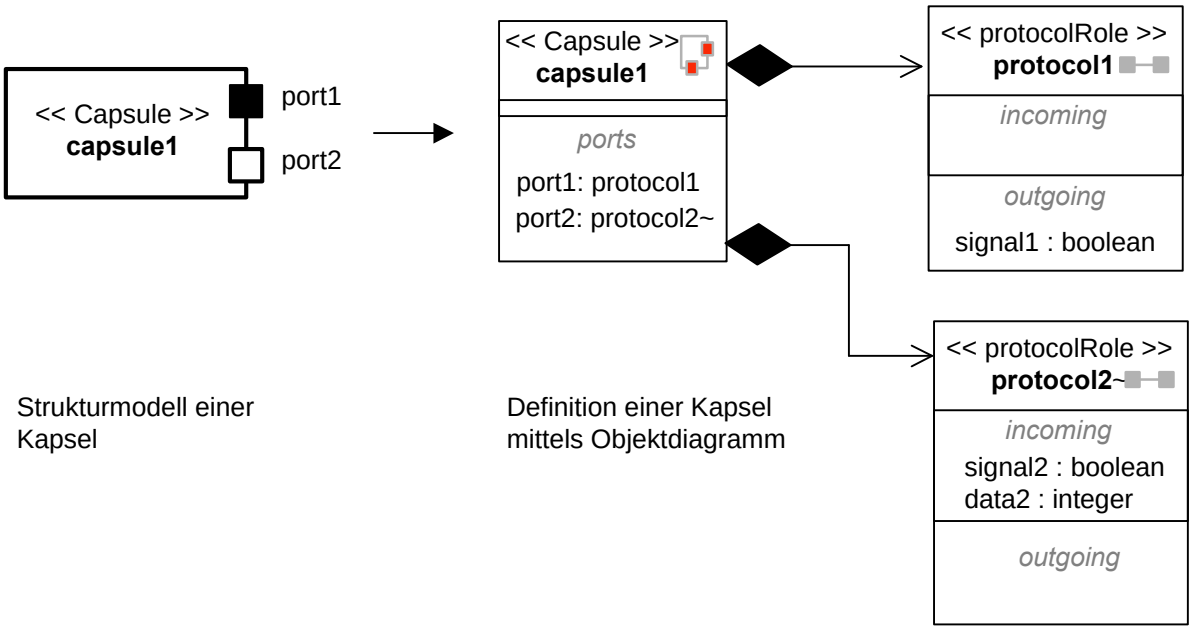


Abbildung 6.7: Definition einer Kapsel mittels Strukturmodell und Objektdiagramm

Die *Verhaltensmodellierung* einer Kapsel erfolgt anhand eines Zustandsdiagramms, der im Wesentlichen auf der erweiterten Form nach Harel basiert, siehe Kapitel 3.2.3. Das Auslösen eines Zustandswechsels ist nur durch Eingangsnachrichten an den definierten Ports möglich. Jeder Zustand kann eine Aktion auslösen. Eine Aktion wirkt sich beim Versenden einer Nachricht aus. In Abbildung 6.8 ist beispielhaft ein Zustandsdiagramm für die Kapsel „capsule1“ aus Abbildung 6.7 dargestellt.

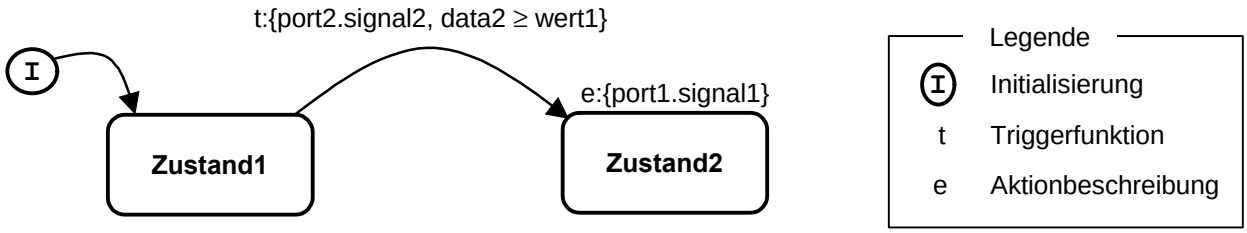


Abbildung 6.8: UML-RT Zustandsdiagramm

Der Zustand „Zustand1“ ist der Startzustand der Kapsel. Dies ist durch das Initialisierungssymbol dargestellt. Ein Übergang in einen anderen Zustand ist nur möglich, falls am Port „port2“ die Nachricht „signal2“ anliegt. Weiterhin muss die Bedingung „data2 ≥ wert1“ erfüllt werden, da-

mit ein Wechsel in „Zustand2“ erfolgen kann. Wird „Zustand2“ erreicht, erfolgt die Ausführung einer Aktion (Versenden des Signals „Signal1“).

Kapseln werden untereinander mit Nachrichtenkanälen (*connectors*) verbunden. Der Verbindungsaufbau wird im Strukturdiagramm beschrieben (Abbildung 6.9). Das Diagramm stellt den möglichen Informationsaustausch unter den Kapseln dar. Der Informationsaustausch wird dabei durch Protokolle bestimmt, die konjungierbar sind. Bei einer Konjunktion werden die Eingangsnachrichten mit den Ausgangsnachrichten vertauscht. Empfangene Nachrichten werden somit zu gesendeten und umgekehrt. Es können nur Schnittstellen mit identischen Protokollen verbunden werden. In Abbildung 6.7 ist das Protokoll „protocol2“ konjungiert. Bezeichner von konjungierten Protokolle besitzen das Symbol „~“ und ihre Ports sind durch weiße Rechtecke dargestellt. Die Empfänger müssen das konjungierte Protokoll der Sender besitzen.

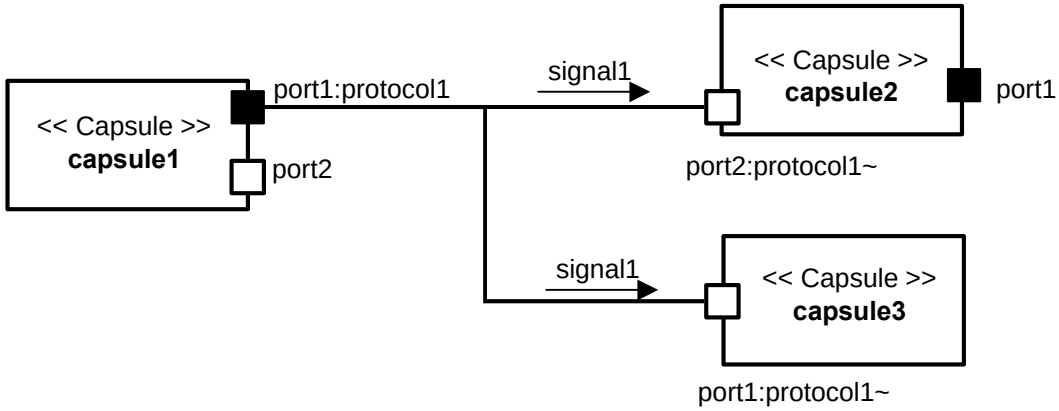


Abbildung 6.9: UML-RT Strukturdiagramm

UML-RT unterstützt ein hierarchisches Entwurfkonzept: Eine Kapsel kann wieder aus mehreren Kapseln aufgebaut sein [Rati00]

6.3.2.2 Qualitative Modellierung einer UML-RT Kapsel

Die für die Transformationsvorschrift wichtigen Beziehungen der Beschreibungsmittel von UML-RT und SQMA sind in Tabelle 6.2 aufgeführt. Kapseln sind die Softwarebausteine eines UML-RT Entwurfs. Jede Kapsel wird durch eine SQMA Komponente beschrieben. Der Name der Kapsel wird aus dem Entwurf übernommen und die Bezeichnung „sw_“ vorangestellt, um das Systemelement als Softwarebaustein zu kennzeichnen.

Die *Schnittstellenbetrachtung* umfasst konkret die Beschreibung der Kapseln mit ihren Ports, der verwendeten Protokolle und Nachrichten. Die Aufgabe der *Verhaltensmodellierung* ist die Zerlegung des Zustandsdiagramms in entsprechende Modellgleichungen (Situationsregeln, Kommentarregeln und Transitionsregeln).

Tabelle 6.2: Beschreibungsmittel von UML-RT und SQMA

Allgemeine Bezeichnung	**UML-RT**	**SQMA**
Softwarebaustein	Kapsel	Component
Schnittstellen	Ports	Terminals
Informationsgrößen	Nachrichten	Quantities
Verhaltensbeschreibung	Zustandsdiagramm	Modellgleichungen (Situations-, Kommentar- und Transitionsregeln)

Schnittstellenmodellierung von UML-RT Kapseln

Jedem Port einer Kapsel wird genau eine SQMA-Schnittstelle (Terminal) zugeordnet. Die Schnittstellen werden entsprechend der Reihenfolge der definierten Ports durchnummeriert (*portNr)*. Der Name des jeweiligen Ports wird als Schnittstellenname übernommen.

Die Beschreibung der Informationsgrößen erfolgt für jeden Port bzw. jedes Protokoll getrennt. Zu beachten ist, dass an einem Port mehrere Signale oder Daten anliegen können. Um zu kennzeichnen, ob es sich um Ein- oder Ausgangsgrößen handelt, wird an den Namen der Informationsgröße die Abkürzungen „_In“ bzw. „_Out“ angehängt.

Die Ermittlung der entsprechenden Intervallbereiche ist bei Signalen, die binäre Größen darstellen, einfach. Hingegen muss der Wertebereich von Daten in Intervalle zerlegt werden. Für deren Ermittlung ist es notwendig, das Zustandsdiagramm der betrachteten Kapsel zu analysieren. Aus den Angaben der Transitionsbedingungen und der Aktionsbeschreibungen eines Zustands werden die signifikanten Werte ermittelt. Diese Werte stellen Intervallgrenzen dar. Dabei bestimmt der verwendete Vergleichsoperator in Transitionsbedingungen oder in Aktionsbeschreibungen die Art des Intervalltyps, siehe Tabelle 6.3.

Tabelle 6.3: Intervallgrenzen in Abhängigkeit eines Vergleichsoperators

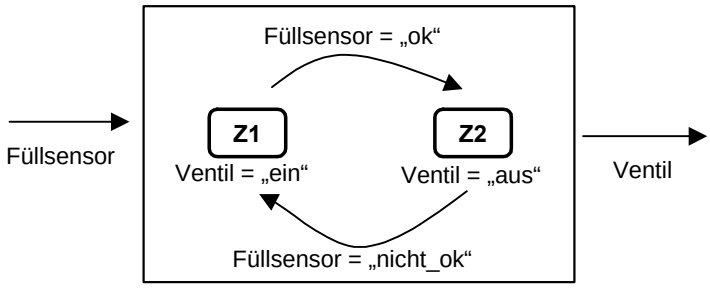
Vergleichsoperator	**Intervalldarstellung**
Variable < Wert	(.., Wert) [Wert, ..)
Variable ≤ Wert	(.., Wert] (Wert, ..)
Variable = Wert	(.., Wert) [Wert, Wert] (Wert..)
Variable ≥ Wert	(.., Wert) [Wert, ..)
Variable > Wert	(.., Wert] (Wert, ..)

Bei natürlichen und reellen Zahlen wird grundsätzlich vom gesamten Werteraum ausgegangen d.h. von [0, ∞) bzw. (-∞,∞). Die ermittelten Intervallgrenzen verfeinern die Aufteilung der Wertebereiche. Ein Beispiel für die Ermittlung der Intervallgrenzen ist in Abbildung 6.13 dargestellt. Die Größe „data2“ wird in die qualitative Größe „data2_In“ überführt. Da es sich um eine natürliche Zahl handelt, lautet das Ausgangsintervall [0, ∞). Aus dem Zustandsdiagramm kann der

signifikante Schwellenwert „wert1“ der Größe „data2“ ermittelt werden. Damit wird das Ausgangsintervall, unter Beachtung von Tabelle 6.3, in die Wertebereiche [0, wert1) [wert1, ∞) verfeinert.

Verhaltensmodellierung von UML-RT Kapseln

Der Ausgangspunkt für die Verhaltensmodellierung stellt der kombinatorisch mögliche Situationsraum dar. Das Ziel ist, Situationen, also Kombinationen von Ein- und Ausgangsgrößen, entsprechenden Zuständen des UML-RT Zustandsdiagramms zuzuordnen. Das bedeutet, dass ein Zustand durch ein oder mehrere Szenarien (Auswirkung des Zustands) beschrieben wird. Nicht im Zustandsdiagramm definierte Kombinationen von Ein- und Ausgangsgrößen sind aus dem Situationsraum zu streichen. Dazu werden Transitionen auf ihre entsprechenden Zustände abgebildet. Ein kleines Beispiel soll die Ausgangssituation verdeutlichen. Das in Abbildung 6.10 dargestellte Zustandsdiagramm bestimmt das Verhalten des Softwarebausteins „Füllstandwächter“, der ein Auslassventil ansteuert. Er verfügt über eine Eingangsgröße „Füllsensor“ und eine Ausgangsgröße „Ventil“. Das Zustandsdiagramm des Softwarebausteins beschreibt sein Verhalten folgendermaßen: Falls die Eingangsgröße Füllsensor den Wert „ok“ annimmt, dann ist die Ausgangsgröße Ventil = „aus“ – dies wird dem Zustand Z2 zugeordnet. Falls der Füllsensor „nicht_ok“ meldet, dann ist die Ausgangsgröße Ventil = „ein“ – diese Sachlage wird dem Zustand Z1 zugeordnet. Da es sich um binäre Größen handelt, lässt sich damit nach SQMA ein kombinatorischer Situationsraum von $2 \cdot 2 = 4$ Situationen erzeugen. Die Situationen 1 und 4 entsprechen nicht dem Verhalten der Zustandsbeschreibung bzw. stellen keine gültigen Szenarien für einen definierten Zustand dar. Wie ersichtlich, sind nicht alle Kombinationen von Ein- und Ausgangsgrößen zulässig.



Situations-nummer	Eingangsgröße: Füllsensor	Ausgangsgröße: Ventil	Bemerkung
1	ok	ein	Nicht definiert
2	ok	aus	Zustand **Z2**
3	nicht_ok	ein	Zustand **Z1**
4	nicht_ok	aus	Nicht definiert

Abbildung 6.10: Kombinationen von Ein- und Ausgangsgrößen für ein Zustandsdiagramm

Die Historie bzw. die Abfolgemöglichkeiten eines Zustandsdiagramms muss bei der Szenarienzuordnung ebenfalls beachtet werden. Zum Beispiel gibt es für den Zustand Z3 in Abbildung

6.11 zwei unterschiedliche Szenarien. Zum einen kann der Zustand Z3 über Z1 und Z2, zum anderen über Z1, Z4, und Z2 erreicht werden. Die Szenarien unterscheiden sich im Wert der Größe „data1“.

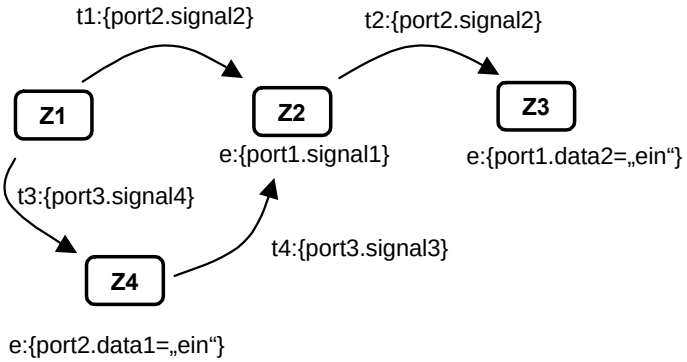


Abbildung 6.11: Informationsgrößen in Abhängigkeit der Historie eines Zustandsdiagramms

Die Transitionsbedingungen eines UML-RT Zustandsdiagramms betreffen Informationsgrößen, die empfangen werden. Dies sind die Eingangsgrößen des UML-RT Softwarebausteins. Aktionsbeschreibungen beziehen sich auf Informationsgrößen, die gesendet werden (Ausgangsgrößen). Das Zustandsdiagramm beschreibt daher das Zusammenwirken von eingehenden und ausgehenden Informationsgrößen einer Kapsel. Ausschlaggebend für die Szenarienzuordnung sind somit folgende Angaben eines UML-RT Zustandsdiagramms [Bieg00b], [Schl99]:

1. *Transitionsbedingungen*
2. *Aktionsbeschreibungen*
3. *Historie*

1. Die *Transitionsbedingungen* geben an, unter welchen Voraussetzungen ein Übergang in einen Zustand Z erfolgt. Falls eine Situation den Zustand Z beschreibt, dann muss grundsätzlich gelten, dass die qualitativ modellierten Eingangsgrößen mindestens eine Transitionsbedingung für den Zustand Z erfüllen müssen. Zu Abbildung 6.11 soll eine Transitionsbedingung beispielhaft dargestellt werden: Um den Zustand Z2 zu beschreiben, müssen die in der Situation festgelegten Werte der Eingangsgrößen die Transitionsbedingung t1 (signal2 = true, bzw. signal2_In = [1,1]) oder t4 (signal3 = true, bzw. signal3_In = [1,1]) erfüllen.

2. *Aktionsbeschreibungen* bestimmen in erster Linie einen Zustand. Der Zustand Z2 aus Abbildung 6.11 zeichnet sich dadurch aus, dass die Ausgangsgröße signal1_Out gesetzt ist (signal1_out = [1,1]). Situationen, bei denen die Werte der Informationsgrößen mindestens eine (vollständige) Transitionsbedingung und *alle* Aktionsbeschreibungen eines Zustands Z erfüllen, stellen mögliche Szenarien des Zustands Z dar, sofern diese die Historie des Zustandsdiagramms nicht verletzen.

3. Zuletzt muss die *Historie* des Zustandsdiagramms betrachtet werden. Im Zustand „Z2“ aus Abbildung 6.11 wird die Ausgangsgröße "signal1" auf "true" gesetzt. Diese Ausgangsgröße bleibt solange gesetzt, bis diese durch irgendeinen anderen Zustand zurückgesetzt wird. Zur Betrachtung der Zustandshistorie werden die unterschiedlichen Ausführungspfade des Zustandsdiagramms ermittelt. Für das Zustandsdiagramm aus Abbildung 6.11 ergibt sich der in Abbildung 6.12 dargestellte Ausführungspfad.

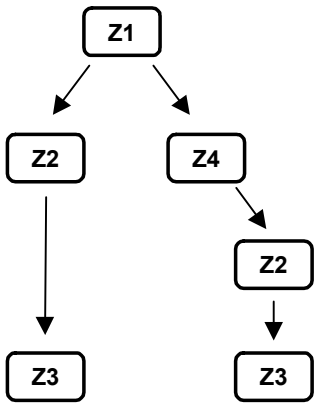


Abbildung 6.12: Ausführungspfade eines Zustandsdiagramms

Beispielsweise müssen die Situation für Zustand Z3 nicht nur mit dessen Transitionsbedingungen und Aktionsbeschreibungen übereinstimmen (d.h. signal2 = "true" und data2 = "ein") sondern ebenfalls mit den Aktionsbeschreibungen von Z1 und Z2 bzw. von Z1, Z4 und Z2 (= Historie) konsistent sein, sofern diese nicht durch den Zustand Z3 selbst zurückgesetzt werden.

Eine Situationsregel für einen Zustand hat demnach folgenden prinzipiellen Aufbau:

```
Transitionsregel erfüllt und Historie erfüllt, dann Aktionsbeschreibung
```

genauer:

```
logischerAusdruck(Eingangsgrößen)= wahr und logischerAusdruck(Ausgangsgrößen
vorheriger Zustände)= wahr, dann logischerAusdruck(Ausgangsgrößen)
```

Ein Zustand wird somit durch eine Menge von Situationen beschrieben, die alle möglichen Werte[6] der modellierten Informationsgrößen für diesen Zustand enthält. Alle Situationen, die eine Situationsgleichung eines Zustands erfüllen, erhalten als qualitativen Ausdruck den Namen des Zustands. Der Name des Zustands wird in Großbuchstaben dargestellt. Dies soll explizit darauf hinweisen, dass es sich um einen Zustand eines Softwarebausteins handelt. Die Zuweisung des Zustandsnamens erfolgt mit Hilfe der Kommentarregeln. Über die Aktionsbeschreibung des Zustands wird der entsprechende Zustandsname mit den Situationen verknüpft, die diese Aktionsbeschreibung erfüllen. Eine Kommentarregel hat demnach den allgemeinen Aufbau:

---

[6] Bezüglich der Modellgrößen an den Schnittstellen.

```
logischerAusdruck(Eingangsgrößen)= wahr und logischerAusdruck(Ausgangsgrößen
vorheriger Zustände)= wahr und logischerAusdruck(Ausgangsgrößen)= wahr, dann
„ZUSTANDSNAME“
```

Falls jeder Zustand eindeutig durch seine Aktionsbeschreibung charakterisiert ist, dann ist folgender Aufbau einer Kommentarregel ausreichend:

logischerAusdruck(Ausgangsgrößen) = wahr, ***dann*** *„ZUSTANDSNAME“*

Nachdem der Zustandautomat mit Hilfe passender Situationen beschrieben wurde, werden alle Situationen, die keinem Zustand zugeordnet werden konnten bzw. nicht dem Zustandsdiagramm entsprechen, aus dem vollständigen Situationsraum der Kapsel entfernt.

Die Definitionen der Situationsregeln und Kommentarregeln haben nur für das Konzept von UML-RT bzw. UML-RT Zustandsdiagrammen Gültigkeit. Der Grund liegt darin, dass bei UML-RT eine strikte Trennung zwischen Ein- und Ausgangsgrößen einer Kapsel vorliegt.

Unter Angabe der Transitionsregeln werden entsprechend dem Zustandsdiagramm mögliche Übergänge zwischen den Situationen angegeben. Hierzu wurden neue Transitionsregeln eingeführt. Diese erlauben die Angabe von möglichen Übergängen unter Verwendung eines qualitativen Ausdruckes.

```
ZUSTANDSNAME1 -> ZUSTANDSNAME2
```

Alle Situationen, die dem Zustand „`ZUSTANDSNAME1`“ zugeordnet sind, können nur in diejenigen Situationen übergehen, die dem Zustand „`ZUSTANDSNAME2`“ angehören.

Mit diesen Angaben ist die Modellierung einer Kapsel vollständig. Die Situationen einer modellierten Kapsel sind Szenarien für deren Verhalten. Eine Situation gibt genau eine mögliche Kombination der Ein- und Ausgangsgrößen einer Kapsel wieder, die aufgrund des definierten Verhaltens möglich ist. Situationen stellen ein geeignetes Mittel zur Interpretation eines Zustandsdiagramms in Form unterschiedlicher Szenarien dar. Die Transitionen zeigen, welche Szenarien ineinander überführbar sind. Ein abstraktes Beispiel einer Transformation von einer UML-RT Kapsel in ein SQMA-Modell ist in Abbildung 6.13, ein konkretes in Kapitel 8 aufgeführt.

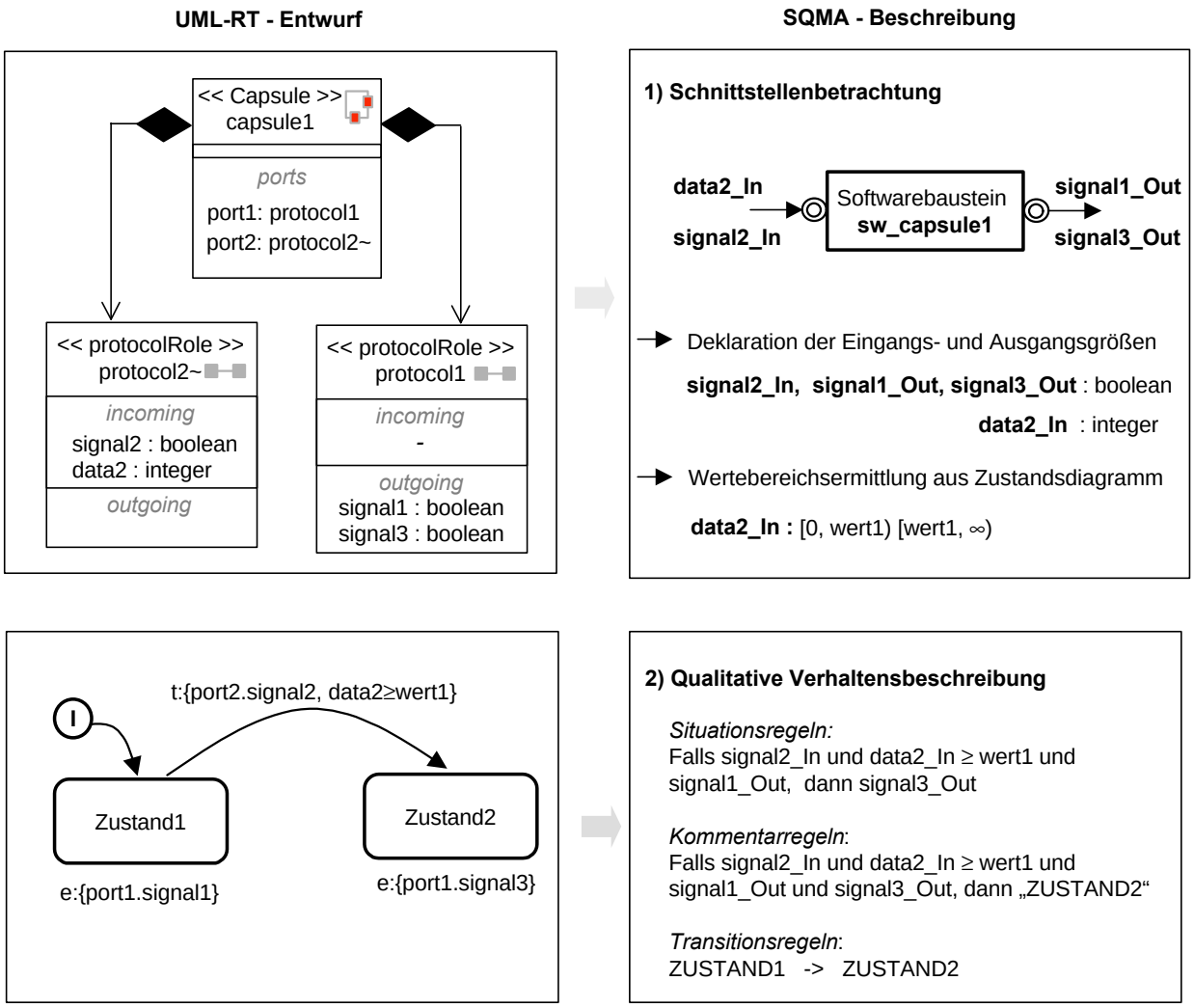


Abbildung 6.13: Transformation UML-RT nach SQMA

6.3.2.3 Erstellung des Modells der Automatisierungssoftware

Nachdem alle Kapseln qualitativ beschreiben wurden, erfolgt die Beschreibung des potentiellen Informationsaustauschs zwischen diesen. Aus dem UML-RT Strukturdiagramm sind alle benötigten Informationen für die Erstellung der Kopplungsbedingungen verfügbar. Die Kopplungsbedingungen bestehen aus einem Gleichsetzen von gesendeten Informationsgrößen mit empfangenen. Aus den UML-RT Verbindungskonzept leiten sich folgende Rahmenbedingungen ab:

- An einem Terminal können beliebig viele Signale und Daten auftreten.
- Es können nur Eingangsgrößen mit gleichnamigen Ausgangsgrößen gleichgesetzt werden.

Für das in Abbildung 6.9 dargestellte Strukturdiagramm ergibt sich die nachstehende Kopplungsbedingung.

```
Capsule1.signal1_Out = Capsule2.signal1_In = Capsule3.signal1_In
```

Mit Hilfe der Kopplungsbedingungen werden wiederum Situationen der einzelnen Kapseln kombiniert, die als Summe das mögliche Verhalten der gesamten Automatisierungssoftware gemäß UML-RT Entwurf beschreiben. Eine Situation des qualitativen Modells der Automatisierungssoftware ist im Grunde ein gültiger Satz von Ein- und Ausgangsgrößen aller Softwarebausteine, der entsprechend der definierten Zustände mit qualitativen Ausdrücken gekennzeichnet ist.

Das vorgestellte Konzept erlaubt eine rechnergestützte Erstellung des qualitativen Modells der Automatisierungssoftware, basierend auf einem UML-RT bzw. ROOM Entwurf. Das im Rahmen der Arbeit entwickelte Werkzeug wird in Kapitel 7.5 vorgestellt. Bei einer rechnergestützten Erstellung wird nicht nur Modellierungsfehlern vorgebeugt, sondern ebenfalls erheblicher Arbeitsaufwand eingespart. Der Aufwand besteht in der einmaligen Entwicklung einer Transformationsvorschrift für eine entsprechende Entwurfsmethode. Eine qualitative Modellierung ist manuell erheblich aufwändiger und kann zu einem fehlerhaften Modell führen [Fehr98].

6.4 Qualitative Modellierung menschlicher Bedieneingriffe

6.4.1 Allgemeine Aspekte zur qualitativen Modellierung menschlicher Bedieneingriffe

Menschliche Bedieneingriffe können nicht personenunabhängig betrachtet werden. Daher bildet eine Person ein nicht weiter zerlegbares Systemelement und wird im Weiteren als *Operator* bezeichnet. Mehrere Operatoren, die in einem speziellen Aufgabenbereich zusammenarbeiten, werden in dieser Arbeit als *Team* definiert. Das Bedienpersonal eines Prozessautomatisierungssystems kann demnach aus mehreren Teams und diese wiederum aus mehreren Operatoren bestehen. Das Verhalten eines Operators ist durch sein Aufgabengebiet geprägt. Dabei kann menschliches Fehlverhalten mit Hilfe eines Klassifikationsschemas bestimmt werden. Die Schnittstellen des Systemelements Operator sind potenzielle Bedieneingriffe und Meldungen. Der Operator besitzt im allgemeinen Fall Schnittstellen zum technischen System und zur Automatisierungssoftware. Nur falls er in einem Team arbeitet, besitzt der Operator Schnittstellen zu anderen Operatoren.

Abbildung 6.14 zeigt einen Operator in Relation zum restlichen System. Nach [Bubb92] erhält ein Operator eine Aufgabenstellung in Form von unterschiedlichen Meldungen, entweder auf optischem, akustischem oder haptischem Weg. Das Mittel zur Aufgabenerfüllung besteht in der Ausführung von Handlungen in Form von Bedieneingriffen. Hierzu stehen dem Menschen Hände, Füße und Sprache zur Verfügung. Ob eine Aufgabe erfüllt wurde, lässt sich nur am erzielten Ergebnis messen, welches nach erfolgreichen Eingriffen die Aufgabenstellung erfüllt haben sollte.

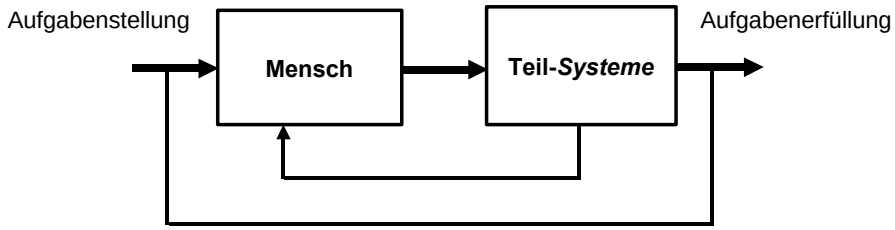


Abbildung 6.14: Aufgabenstellung und Aufgabenerfüllung eines Operators

Eine Schnittstellenbetrachtung des Systemelements „Operator“ kann daher in Form von Meldungen und Bedieneingriffen abstrahiert werden, siehe Abbildung 6.15.

Abbildung 6.15: Mensch als Systemelement „Operator“

Meldungen werden entweder vom technischen System, von der Automatisierungssoftware oder einem weiteren Menschen an den Operator „gesendet“. Meldungen stellen in diesem Zusammenhang alle Ereignisse dar, die der Operator über seine Sinne wahrnehmen kann.

Für die Sicherheitsanalyse ist der Zusammenhang zwischen Informationsgehalt einer Meldung und der entsprechenden ausgelösten Aktion eines Bedieneingriffes von Interesse.

Bedieneingriffe sind im Speziellen von den konkreten Aufgaben des Operators abhängig und von System zu System verschieden, dennoch können allgemeine Aspekte berücksichtigt werden. Aufgrund seiner Arbeitsumgebung und seiner Möglichkeiten kann ein Operator zum Beispiel mehrere Eingriffe gleichzeitig betätigen. Weiterhin können Bedieneingriffe auch losgelöst von Meldungen erfolgen. In dieser Arbeit konnten daher folgende Aspekte von Bedieneingriffen differenziert werden:

- *Simultane und sequenzielle Bedieneingriffe*
  Simultane Eingriffe beschreiben parallele Bedieneingriffe, die zur selben Zeit ausgeführt werden können. Können Bedieneingriffe nur nacheinander ausgeführt werden, stellen diese sequenzielle Bedieneingriffe dar.

- *Reaktive und aktive Bedieneingriffe*
  Als reaktive Bedieneingriffe werden in dieser Arbeit alle Ausführungen des Menschen verstanden, die als Reaktion auf eine bestimmte Meldung erfolgen. Überwachungsaufgaben

werden in Form von reaktiven Bedieneingriffen bearbeitet. Sind die Handlungen des Menschen unabhängig von Meldungen, werden die Bedieneingriffe als aktive Bedieneingriffe bezeichnet.

- *Bediensequenzen*
  Eine Folge von Bedieneingriffen wird als Bediensequenz bezeichnet.

Menschliche Fehlleistungen lassen sich ausschließlich in Bedieneingriffen beobachten. Wahrnehmungs- oder Denkfehler spiegeln sich lediglich in der Verknüpfung von Meldungen und Bedieneingriffen wider [Reas94]. Entsprechend dieser Betrachtungsweise existieren in der Literatur zahlreiche Klassifizierungsansätze menschlicher Fehlleistungen, die in zwei große Bereiche unterteilt werden: die *auftretensorientierte* Klassifizierung und die *ursachenorientierte* Klassifizierung. Bei der auftretensorientierten Betrachtungsweise steht der verhaltenspsychologische Aspekte im Vordergrund. Es wird versucht, menschliche Fehlleistungen unabhängig von speziellen Aufgaben und Handlungen zu strukturieren. Es wird die Frage nach dem „was“, „wie“, „wann“ oder „wo“, aber nicht die Frage nach dem „warum“ der aufgetretenen Fehler gestellt. Als fundamentale auftretensorientierte Klassifikationsansätze sind [Rigb70], [Meis77] und [Swai80] zu nennen. Bei der ursachenorientierten Klassifikation wird versucht, menschliche Arbeitsfehler hinsichtlich ihrer Entstehungsursache zu strukturieren. Definiert wurden wichtige Klassifikationsansätze von [Hack87], [Norm86] und [Zimo90].

Der für die vorliegende Arbeit ausgewählte Klassifikationsansatz stammt von Rouse & Rouse [RoRo83]. Es handelt sich um einen systemorientierten Ansatz, der sowohl auftretensorientierte wie auch ursachenorientierte Sichtweisen berücksichtigt. Dabei bildet das von Rasmussen [Rasm90] entwickelte 3-Ebenen Verhaltensmodell die Grundlage des Ansatzes, siehe Kapitel 2.1.3. Aufgabenstellungen auf der wissensbasierten Ebene erfordern eine aufmerksamere Kontrolle: Sie können nur nacheinander ausgeführt werden und sind für einen Operator stark beanspruchend. Aufgabenstellungen auf der fähigkeitsbasierten Ebene laufen hingegen automatisch ab, sie können parallel abgearbeitet werden und sind für einen Operator weniger anstrengend. Auf allen Bearbeitungsebenen existieren unterschiedliche Möglichkeiten für das Auftreten von Bedienfehlern.

Ausgehend von diesen Vorbetrachtungen kann nun im Folgenden auf das Modellierungskonzept der menschlichen Bedieneingriffe näher eingegangen werden.

6.4.2 Iterative Modellierung menschlicher Bedieneingriffe

6.4.2.1 Prinzip der iterativen Modellierung

Jedes Bedienpersonal wird in Form eines „menschlichen“ Systemelements beschrieben. Die Modellierung der menschlichen Tätigkeiten stützt sich auf Bedienhandbücher. Weitere Informationen sind aus den Entwurfsdokumentationen des technischen Systems (z. B. aus RI-Fließbildern) und der Automatisierungssoftware zu erhalten. Die Schnittstellen zum Bedienpersonal (Anwender) sind in der Regel exakt spezifiziert. Insbesondere werden bei klassenorientierten Entwicklungskonzepten für die Automatisierungssoftware Anwendungsszenarien (use cases) ermittelt. Diese Anwendungsszenarien stellen detaillierte Interaktionen zwischen Bedienpersonal und Softwaresystem dar. Die iterative Modellierung ist in Abbildung 6.16 dargestellt.

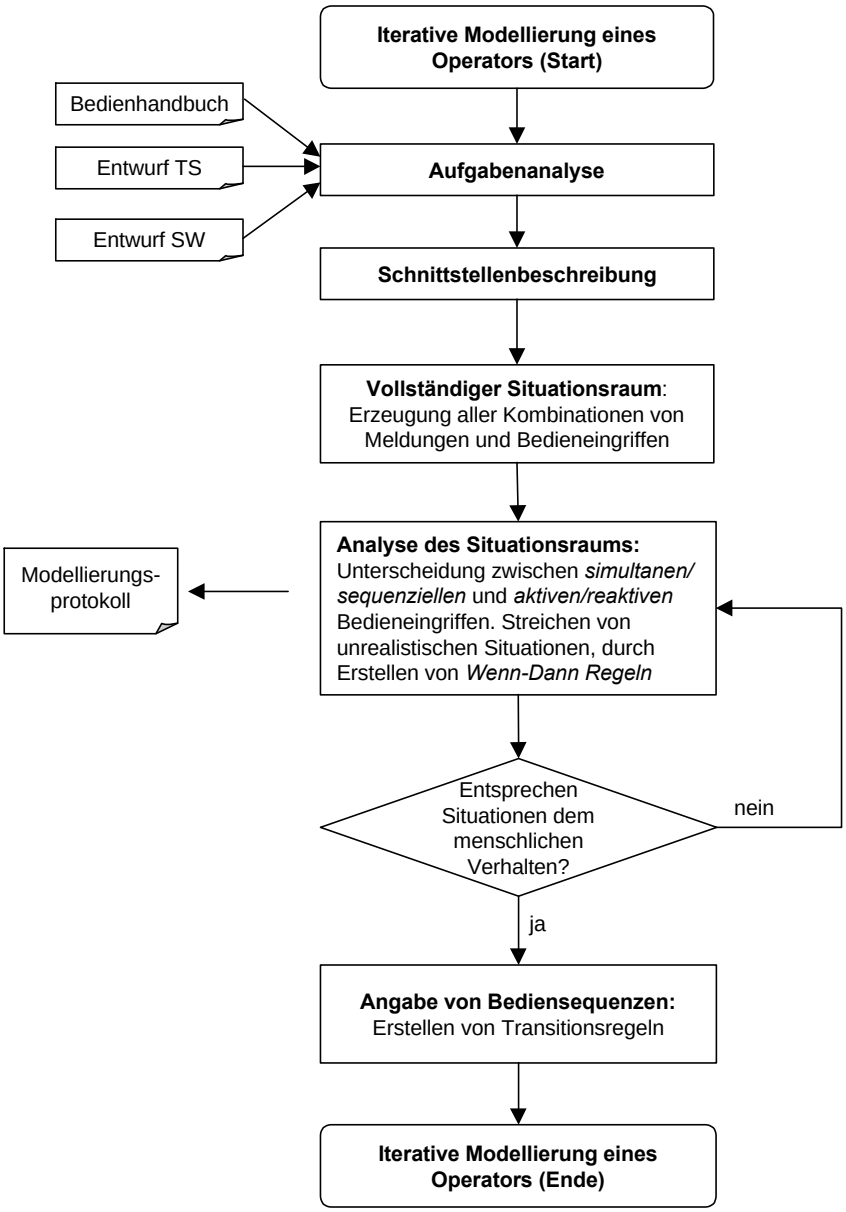


Abbildung 6.16: Durchführung der iterativen Modellierung von Operatoren.

Für jeden zu modellierenden Operator müssen daher im ersten Schritt die zugänglichen Meldungen (Systeminformationen) ermittelt und darüber hinaus sein Aufgabenfeld in Form von möglichen Bedieneingriffen bestimmt werden (Aufgabenanalyse), siehe Abbildung 6.16. Das Ziel der Modellierung ist die Darstellung von Meldungen und Bedieneingriffen in Form von Situationen und die Beschreibung der Bediensequenzen in Form von Transitionen. Während bei der Modellierung des technischen Systems auf physikalische Gesetze und bei der Modellierung der Automatisierungssoftware auf ein entsprechendes Entwurfskonzept zurückgegriffen werden kann, bestehen keine grundsätzlichen Gesetzmäßigkeiten für menschliches Verhalten. Aus diesem Grund wird eine iterative Modellierung angewandt.

Das Prinzip der iterativen Modellierung basiert auf der konzeptionellen Unterstützung von SQMA bei der Modellierung selbst. Dabei wird folgender Gedanke verfolgt: Werden alle Meldungen und Bedieneingriffe eines Operators in Form von qualitativen Größen beschrieben (Schnittstellenbetrachtung), dann stellen alle Kombinationen dieser Größen – der vollständige Situationsraum – auch alle möglichen Bedieneingriffe dar, die der Operator ausführen kann, siehe Abbildung 6.17.

Der vollständige Situationsraum umfasst dabei falsche, richtige und unmögliche Bedieneingriffe. In Abbildung 6.17 ist eine Schnittstellenbetrachtung eines Operators und der entsprechende vollständige Situationsraum zu sehen. In diesem Bespiel besitzt der Operator die Eingangsgröße „Meldung1“ und kann zwischen zwei verschiedenen Bedieneingriffen wählen, die als Ausgangsgrößen definiert werden. Kombinatorisch ergeben sich 8 verschiedene Situationen. Beispielsweise besagt die achte Situation: Falls „Meldung1“ auftritt, werden „Bedieneingriffe1“ und „Bedieneingriffe2“ gleichzeitig ausgeführt.

	Eingangsgrößen	Ausgangsgrößen
1	-	-
2	Meldung1	-
3	-	Bedieneingriff1
4	Meldung1	Bedieneingriff1
5	-	Bedieneingriff2
6	Meldung1	Bedieneingriff2
7	-	Bedieneingriff1, Bedieneingriff2
8	Meldung1	Bedieneingriff1, Bedieneingriff2

Abbildung 6.17: Beispielhafte Ausgangssituation der iterativen Modellierung eines Operators

Bei der weiteren Modellierung müssen simultane, sequenzielle, aktive und reaktive Bedieneingriffe beachtet werden. Bedieneingriffe, die z. B. simultan nicht möglich sind, werden mit Hilfe

von *Situations*-Regeln ausgeschlossen, d.h. die entsprechenden Situationen werden aus dem vollständigen Situationsraum entfernt. Angenommen, die Bedieneingriffe „Bedieneingriff1“ und „Bedieneingriff2“ sind nicht gleichzeitig (simultan) möglich, dann muss dies mit Hilfe einer entsprechenden Situationsregel ausgedrückt werden. Die Regel führt dazu, dass bei einer erneuten Berechnung des Situationsraums die Situationen Nr. 7 und Nr. 8 aus dem vollständigen Situationsraum gestrichen werden.

Höchst unwahrscheinliche Bedienfehler können auf die gleiche Weise ausgeschlossen werden. Wahrscheinliche Bedienfehler werden zugelassen und entsprechend klassifiziert. Jeder Ausschluss von Situationen muss allerdings im Modellierungsprotokoll kommentiert und begründet werden. Damit wird die Reproduktion der Modelle durch Dritte gewährleistet. Der Ausschluss erfolgt unter Verwendung von Situationsregeln und besitzt den Vorteil, dass die Ausgangssituationen nicht einzeln manuell gestrichen werden müssen. Nach der Aufstellung einer Ausschlussregel werden die Situationen der menschlichen Bedieneingriffe erneut berechnet. Es bleiben nur Situationen übrig, die von der aufgestellten Ausschlussregel nicht betroffen sind. Diese Situationen müssen nochmals durchgegangen werden bis eine neue Ausschlussregel erstellt und kommentiert wurde. Dieser Vorgang wird solange iteriert, bis nur noch Situationen existieren, die entweder die Handlungen des Operators wiedergeben oder aber realistische Bedienfehler darstellen.

Dieses Vorgehen besitzt den Vorteil, dass kein Verhaltensmuster eines Operators vergessen werden kann. Das Modell ist in Bezug auf die definierten Größen an den Schnittstellen immer vollständig. Modellierungsfehler wirken sich lediglich in unsinnigen Situationen aus, die durch Kontrolle des Modells identifizierbar sind. Im Folgenden wird auf die zentralen Schritte der iterativen Modellierung eines Operators im Einzelnen eingegangen.

6.4.2.2 Schnittstellenbetrachtung eines Operators

Jeder Operator erhält eine eindeutige Bezeichnung. Diese setzt sich aus einem vorangestellten „op_“ und einem beliebigen Namen zusammen (Beispiel: „op_Zugführer“). Ein Operator kann direkt in das technische System auf ein entsprechendes technisches Bauelement zugreifen (bzw. dieses bedienen). Umgekehrt kann ein Operator von einem bestimmten Bauelement Informationen erhalten. Das Gleiche gilt für die Automatisierungssoftware bzw. für Softwarebausteine. Die Kommunikation zwischen Operatoren muss ebenfalls durch Schnittstellen beschrieben werden. Jede Informationsquelle und Bedienmöglichkeit stellt eine Schnittstelle des zu modellierenden Operators dar.

Meldungen und Bedieneingriffe werden in Form von qualitativen Größen dargestellt. Meldungen sind die Eingangsgrößen und die Bedieneingriffe die Ausgangsgrößen des Systemelements Operator. Meldungen erhalten zur Identifikation die Bezeichnung „_In“ und Bedieneingriffe den Ausdruck „_Out“. Meldungen und Bedieneingriffe setzen sich aus den gleichen Grundelementen zusammen, siehe Tabelle 6.4. Bei der Modellierung der verschiedenen Meldungen und Be-

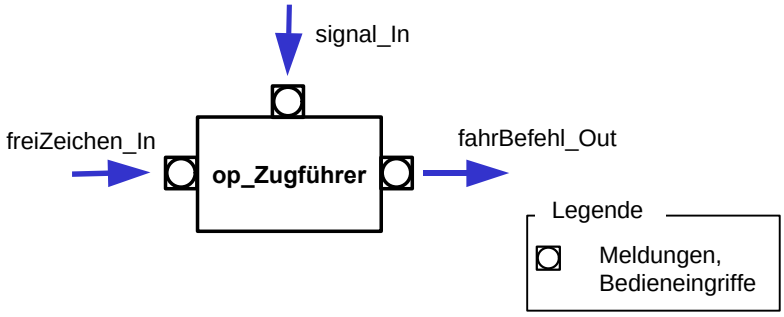
dieneingriffe müssen zusätzlich die Fälle „keine Meldung“ und „kein Bedieneingriff“ berücksichtigt werden.

Tabelle 6.4: Elementare Meldungsarten und Bedieneingriffe

Typ	**Meldungen**	**Bedieneingriffe**	**Beispiel**
boolesche Größe skalare Größe	Information über Prozesszustände	einfache Aktion Befehle	zu / auf -/ leer / halbvoll / voll
natürliche Zahl reelle Zahl	Informationen über Prozessgrößen	Sollwertvorgaben	Füllstand = 5 Liter Reifendruck = 1.5 pa

Zur Veranschaulichung erhalten Intervallgrößen mit Hilfe der Kommentarregeln einen entsprechenden qualitativen Ausdruck. Bei Bedieneingriffen handelt es sich um Vorgänge, die mit Hilfe eines Verbs zu beschreiben sind. Ebenfalls werden Prozessgrößen (als Meldungen oder Vorgaben) implizit durch qualitative Ausdrücke bewertet. Bei der Angabe von Zahlenwerten bestimmen signifikante Werte die Grenzen der Intervalle (z. B. maximale Sollwertvorgabe eines Füllniveaus).

Zusammenfassend ist für eine Schnittstellenbetrachtung eines Operators in Abbildung 6.18 ein Beispiel aufgeführt. Als Meldungen verfügt der Operator „op_Zugführer“ über den Status „signal_In“ einer Signalanlage, zusätzlich muss er das Freizeichen „freiZeichen_In“ eines Schaffners beachten. Mit Hilfe des Bedieneingriffs „fahrBefehl_Out“ kann er den Zug in Bewegung setzen. Alle Größen an den Schnittstellen sind boolesche Intervalle, die einem entsprechenden qualitativen Ausdruck zugeordnet sind.



Bezeichner	**Typ**	**Klasse**	**Intervallbereiche**	**Qualitative Ausdrücke**
freiZeichen_In	Meldung	bool	[0, 0] [1, 1]	1 = „freizeichen“ (0 = "nicht freizeichen")
signal_In	Meldung	scalar	[0, 0] [1, 1]	0 = „rot“ 1 = „grün“
fahrBefehl_Out	Bedieneingriff	bool	[0, 0] [1, 1]	1 = „fahren“ (0 = "nicht fahren")

Abbildung 6.18: Schnittstellenbetrachtung eines Zugführers

6.4.2.3 Modellgleichungen zur Verhaltensbeschreibung eines Operators

Der Ausschluss von unwahrscheinlichen sowie unmöglichen Bedieneingriffen erfolgt unter der Anwendung von Situationsregeln. Um Anforderung 4 (Nachvollziehbarkeit) gerecht zu werden, muss jede aufgestellte Situationsregel genau begründet werden. Viele Bedieneingriffe können vom Operator nicht gleichzeitig ausgeführt werden, da die Bedienelemente zu weit von einander entfernt sind. Diese sequenziellen Bedieneingriffe müssen bei der ersten Iteration mit Hilfe von Situationsregeln angegeben werden. Bei der Angabe aktiver und reaktiver Bedieneingriffe muss der Anwender entscheiden, welche Bedienfehler im Modell des Operators enthalten sein sollen und welche nicht. Die Situationsregeln besitzen folgenden allgemeinen Aufbau:

```
Situationsregel
      sequenzielle Bediengriffe:
      Ausdruck(Bedieneingriffe)     ->    Nicht(Ausdruck(Bedieneingriffe))
      reaktive Bediengriffe:
      Ausdruck(Meldungen)           ->    Ausdruck(Bedieneingriffe))
      aktive Bediengriffe:
      Ausdruck(Meldungen)           ->    Nicht(Ausdruck(Bedieneingriffe))
```

Die verbleibenden Situationen werden in zwei Klassen eingeteilt: in bestimmungsgemäße Situationen (Attribut „B“) und Situationen, die falsche Bedieneingriffe wiedergeben (Attribut „U“). Basierend auf [RoRo83] äußern sich fehlerhafte Bedieneingriffe durch:

- *Keine Reaktion oder falsche Reaktion auf ein Ereignis (omission)*
  Ein notwendiger Bedieneingriff auf ein Ereignis bleibt aus oder es werden nicht die dafür vorgesehenen reaktiven Bedieneingriffe ausgeführt.
- *Unnötige Aktion (commission)*
  Es erfolgt ein reaktiver Bedieneingriff, ohne dass eine dazugehörende Meldung vorliegt.

Der Fahrbefehl des Zugführers, beispielhaft in Abbildung 6.18 dargestellt, ist ein reaktiver Bedieneingriff, da dieser von den Meldungen „fahrZeichen_In“ und „signal_In“ abhängig ist. Nur falls beide Meldungen die Anfahrt des Zugs freigeben, darf der Zugführer den Fahrbefehl erteilen. Die möglichen Fehler eines Zugführers bestehen daher in:

- Erteilen des Fahrbefehls ohne entsprechende Meldungen (commission) und
- nicht erteilen des Fahrbefehls, obwohl die entsprechenden Meldungen vorliegen (ommission).

Bediensequenzen setzen sich aus einer Reihe von einzelnen aktiven Bedieneingriffen zusammen. Die Reihenfolge der Bedieneingriffe wird mit Hilfe von Transitionsregeln festgelegt. Fehlerhafte Bediensequenzen sind Auslassen, falsche Reihenfolge oder unnötige Wiederholungen von Bedieneingriffen [RoRo83]. Um Sequenzen entsprechend beschreiben und klassifizieren zu können, müssen die Transitionsregeln von SQMA erweitert werden. Dies ist notwendig, da SQMA nur die Angabe von Situationsübergänge unterstützt, die auf physikalischen Gesetzmä-

ßigkeiten beruhen. Um Bediensequenzen zu beschreiben, wurden Gruppenübergänge von Situationen ermöglicht. Die erweiterten Transitionsregeln, wie sie auch zur Modellierung der Automatisierungssoftware eingesetzt werden, erlauben die Angabe möglicher Übergänge unter Verwendung eines qualitativen Ausdruckes und besitzen daher folgenden Aufbau:

```
Transitionsregel:
      QUALITATIVER AUSDRUCK1 -> QUALITATIVER AUSDRUCK2      [U|B]
```

Alle Situationen, die dem Ausdruck „`QUALITATIVER AUSDRUCK1`“ zugeordnet sind, können nur in diejenigen Situationen übergehen, die dem Zustand „`QUALITATIVER AUSDRUCK2`“ angehören. Die erweiterten Transitionsregeln unterstützen darüber hinaus die Klassifizierung von Situationsübergängen, um falsche Bediensequenzen zu spezifizieren. Dies erfolgt analog zur Klassifizierung der Situationen mit Hilfe von Attributen (bestimmungsgemäße (B) oder unerwünschte Bediensequenz (U)).

Reaktive Bedieneingriffe sind von Meldungen abhängig, das Auftreten dieser Meldungen wiederum vom technischen System oder von der Automatisierungssoftware. Auf die Reihenfolge der Meldungen hat der Operator keinen Einfluss.

6.4.3 Erstellung eines Modells menschlicher Bedieneingriffe

Während das qualitative Modell des technischen Systems und der Automatisierungssoftware aus technischen Bauelementen bzw. Softwarebausteinen aufgebaut wird, ist eine Zusammenarbeit mehrerer Operatoren nicht immer gegeben. Falls zwei oder mehrere Operatoren untereinander in Kontakt stehen und deren Bedieneingriffe von Entscheidungen anderer Operatoren abhängen, müssen diese zu einem Modell bzw. zu einem Team verbunden werden.

Interaktionen zwischen Bedienpersonal werden durch entsprechende Relationen wiedergegeben. In Abbildung 6.19 ist ein Beispiel aufgezeigt. Ein Zugteam besteht aus einem Schaffner und einem Zugführer. Der Schaffner kommuniziert mit dem Zugführer und beeinflusst dessen Handlungen.

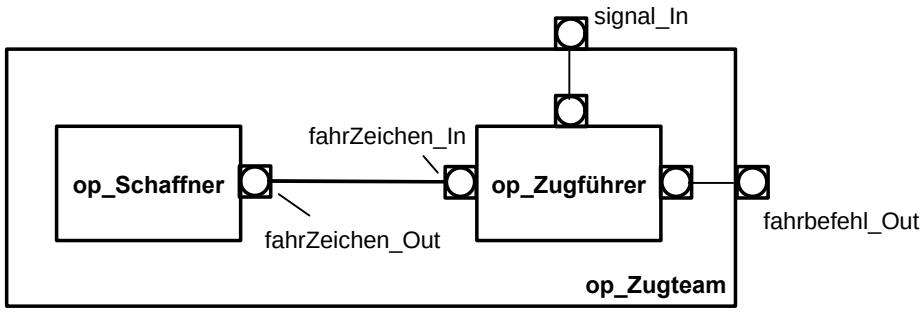


Abbildung 6.19: Zusammenspiel eines Teams am Beispiel Schaffner und Zugführer

Die Bedieneingriffe des Schaffners wirken sich als entsprechende Meldungen für den Zugführer aus. Der Bedieneingriff „fahrBefehl_Out“ des Zugführers ist abhängig vom „fahrZeichen_In“ des Schaffners und von einem entsprechenden Signal einer Signalanlage.

Die Berechnung des Zusammenspiels erfolgt gemäß SQMA-Verfahren. Die Kommunikation zwischen dem Bedienpersonal werden als Maschengleichungen wiedergegeben, d.h. an allen verbundenen Terminals finden sich die gleichen Informationen in Form von Meldungen oder Bedieneingriffen. Für eine gültige Verknüpfung zwischen Operatoren muss gewährleistet sein, dass mindestens ein Bedieneingriff („_Out“) mit mindestens einer Meldung („_In“) verbunden ist. Die Kommunikationswege werden in Form von Kopplungsbedingungen festgehalten. Für das Zugteam ergibt sich folgende Kopplungsbedingung:

```
Schaffner.fahrZeichen_Out = Zugführer.fahrZeichen_In
```

Um das Verhalten des Teams zu beschreiben, werden Situationen von einzelnen Operatoren mit Situationen anderer Operatoren entsprechend den Kopplungsbedingungen kombiniert. Im obigen Beispiel werden nur Situationen des Schaffners und des Zugführers verknüpft, falls bei beiden Personen der Inhalt der Nachricht „fahrZeichen“ identisch ist.

6.5 Erstellung des Gesamtmodells eines Prozessautomatisierungssystems

6.5.1 Ausgangsbasis

Für die Erstellung des ganzheitlichen Modells bilden alle qualitativ beschriebenen Systembestandteile die Ausgangsbasis. Zur Erläuterung des Vorgehens ist es zweckdienlich, die wesentlichen Eigenschaften und Aussagen der Teilmodelle zusammenzufassen.

- Das *Modell des technischen Systems* kann bestimmungsgemäße, unerwünschte und gefährliche Situationen enthalten. Die Situationen stellen das mögliche physikalische Verhalten des technischen Prozesses dar, das aufgrund des technischen Systems möglich ist. Es handelt sich um das allgemeine Verhalten im unkontrollierten Zustand. Das Modell besitzt explizit gekennzeichnete Schnittstellen zur Automatisierungssoftware und zum Bedienpersonal.

- Das *Modell der Automatisierungssoftware* enthält Situationen, die keine Klassifikation besitzen. Die Situationen beschreiben das mögliche Verhalten der Automatisierungssoftware, das auf Basis ihres Entwurfs möglich ist. Das qualitative Modell der Automatisierungssoftware enthält Schnittstellen zum technischen System und zum Bedienpersonal.

- Das *Modell der menschlichen Bedieneingriffe* enthält im Allgemeinen bestimmungsgemäße und unerwünschte Situationen. Die Situationen umfassen das mögliche Verhalten des Bedienpersonals, basierend auf Meldungen und Bedieneingriffen. Das Modell der menschlichen Bedieneingriffe besitzt Schnittstellen zum technischen System und zur Automatisierungssoftware.

Das Verhalten eines Prozessautomatisierungssystems ergibt sich aus dem Zusammenspiel der Systembestandteile. Die Verknüpfung zum Gesamtsystem erfolgt nach dem SQMA-Verfahren. Es werden daher alle Situationen der Systembestandteile untereinander kombiniert. Anhand der Struktur des Prozessautomatisierungssystems und der Modellgrößen wird geprüft, ob diese „zusammenpassen“. Eine gültige Kombination beschreibt ein mögliches Szenarium des Prozessautomatisierungssystems, wie es im Betrieb zu beliebigen Zeitpunkten auftreten kann. Bei der Kombination werden daher Szenarien für das physikalische Verhalten mit Szenarien für das Verhalten der Automatisierungssoftware und Szenarien für das menschliche Verhalten verknüpft. Dabei wird unter anderem geprüft, ob sich gefährliche Situationen des technischen Systems mit Situationen anderer Teilmodelle kombinieren lassen.

Die Erstellung des Gesamtmodells erfolgt durch die nachstehenden Schritte, auf die im Einzelnen eingegangen werden soll.

1. Strukturbeschreibung eines Prozessautomatisierungssystems
2. Konsistenz- und Plausibilitätsprüfungen
3. Erstellung der Kopplungsbedingungen
4. Kombination von Situationen der Teilmodelle

6.5.2 Strukturbeschreibung

Bei der Strukturbeschreibung des Prozessautomatisierungssystems gibt der Anwender an, über welche *offenen* Schnittstellen die Modelle der Systembestandteile miteinander verbunden werden. Bei allen Systembestandteilen stellen *offene* Schnittstellen, d.h. Schnittstellen, die keine Verbindungen aufweisen, die Verbindung zur Außenwelt dar, Abbildung 6.20. Ziel ist die Erstellung einer Netzliste, die die Verbindungen zwischen den offenen Schnittstellen der Systembestandteile festlegt.

Die Erstellung der Netzliste beginnt mit dem technischen System, da dieses über explizit gekennzeichnete Schnittstellenarten verfügt. Es werden nacheinander alle offenen Schnittstellen durchgegangen. Die Art der Schnittstelle weist auf das entsprechende Systembestandteil hin. Der Name der Schnittstelle gibt dem Anwender Auskunft über die entsprechende Zielschnittstelle. Zum Beispiel weist die offene Schnittstelle „`id HandventilA type MB`“ des technischen Systems auf eine Verbindung zu einer Schnittstelle des Teilmodells der menschlichen Bedieneingriffe (MB) hin, die den Eingriff auf das Handventil A beschreibt.

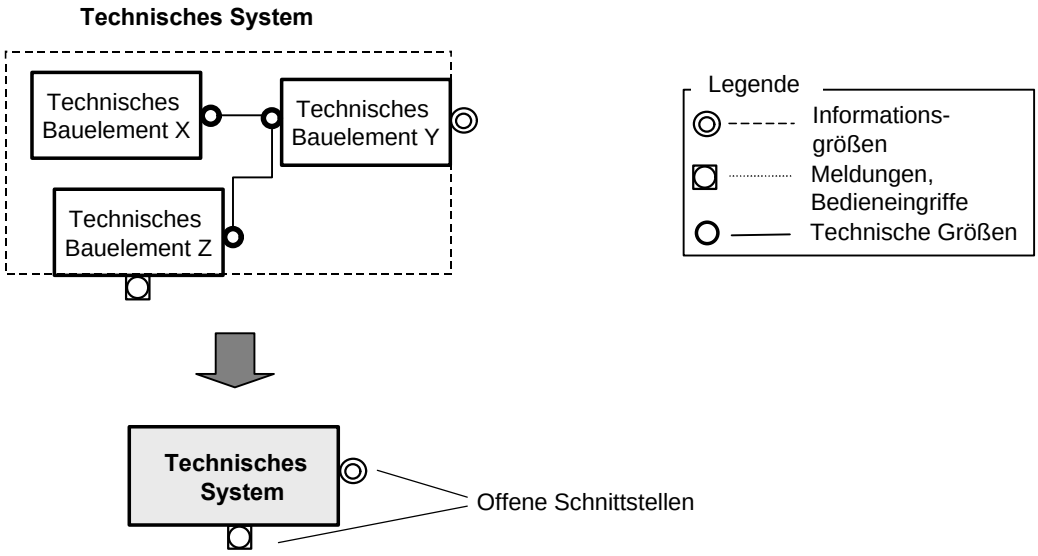


Abbildung 6.20: Offene Schnittstellen eines technischen Systems

Das technische System kann ebenfalls über offene Schnittstellen vom Typ „TS“ verfügen. Diese stellen Schnittstellen des Prozessautomatisierungssystems zu seiner Umwelt dar. Es handelt sich dabei um Anschlüsse für Quellen und Senken, wie z. B. Zu- und Abwasser.

Nach der Behandlung des technischen Systems verbleiben offene Schnittstellen und damit fehlende Verbindungen zwischen dem Modell der Automatisierungssoftware und dem Modell der menschlichen Bedieneingriffe. Ausgehend von den offenen Schnittstellen der menschlichen Bedieneingriffe werden Verbindungen zu entsprechenden Schnittstellen der Automatisierungssoftware vorgenommen. Die Erstellung der Netzliste ist abgeschlossen, falls alle offenen Schnittstellen im Modell der menschlichen Bedieneingriffe verbunden sind, siehe Abbildung 6.21.

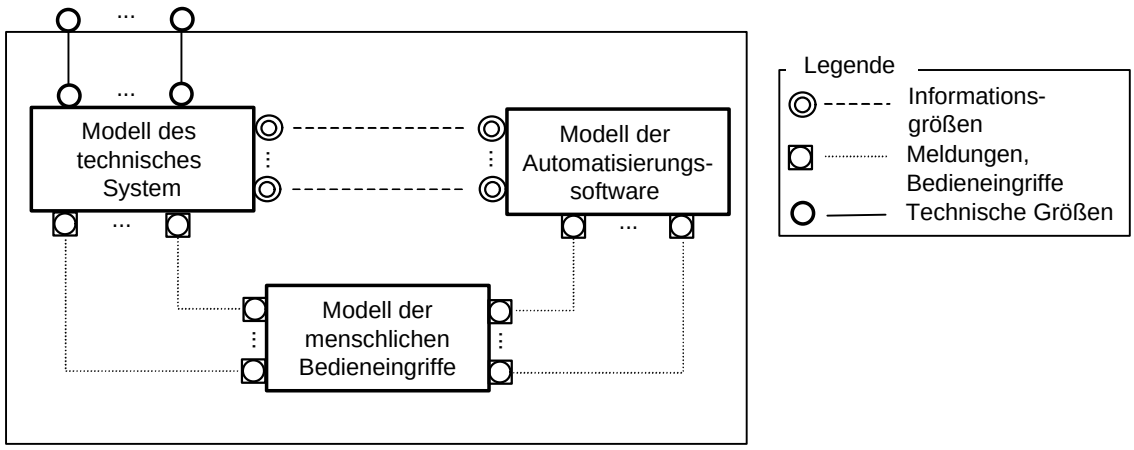


Abbildung 6.21: Verknüpfung der Systembestandteile über definierte Schnittstellen

6.5.3 Konsistenz- und Plausibilitätsprüfung

Um Modellierungsfehlern vorzubeugen, findet eine Überprüfung der Netzliste statt. Diese Überprüfung unterteilt sich in eine Konsistenz- und eine Plausibilitätsprüfung.

Die Konsistenzprüfung beinhaltet die Untersuchung folgender Kriterien:

- *Vollständigkeit der Verbindungen*
  Falls offene Schnittstellen existieren, werden diese identifiziert. Manchmal werden Verbindungen oder Schnittstellen eines Systembestandteils vergessen.

- *Schnittstellentyp*
  Die Überprüfung des Schnittstellentyps garantiert die Verbindungen gleichartiger Modellgrößen[7]. Auf diese Weise wird garantiert, dass nur gleichartige Modellgrößen verbunden werden.

- *Anzahl der verbundenen Modellgrößen*
  Die Anzahl der Modellgrößen, die durch das Verbinden der Schnittstellen verknüpft werden, muss sich entsprechen. Die Ursache für Abweichungen liegt entweder in der unterschiedlichen Definition oder im Vergessen einer Modellgröße.

Die Plausibilitätsprüfung betrachtet zusätzlich den Informationsgehalt der zu verbindenden Modellgrößen:

- *Übereinstimmung des Wertebereichs*
  Die Ursache für widersprüchliche Wertebereiche liegt in Vorzeichenfehlern bei der Definition von Modellgrößen.

- *Quelle- und Senke*-Prinzip
  Bei einer gültigen Verbindung muss mindestens eine ausgehende Modellgröße („_Out“) mit einer eingehenden Modellgröße („_In“) verknüpft sein.

- *Widersprüchliche qualitative Ausdrücke*
  Bei booleschen oder skalaren Modellgrößen empfiehlt sich die Überprüfung der qualitativen Ausdrücke, um sich widersprechende Angaben aufzudecken.

---

[7] Technische Größen, Informationsgrößen, Meldungen und Bedieneingriffe.

6.5.4 Erstellung von Kopplungsbedingungen

Das Zusammenspiel von technischem System, Automatisierungssoftware und menschlichen Bedieneingriffen beruht auf dem Austausch von Nachrichten (Informationen). Ein Informationsaustausch beruht nach [Panr99] auf einem signalfluss-orientierten Verknüpfungskonzept.

Aus der erstellten Netzliste und den Teilmodellen werden die zu verkoppelnden Modellgrößen durch den Intervallschnitt-Operator „=“ verknüpft. Die Verknüpfungsvorschrift besteht darin, dass sich die Intervalle der zu verknüpfenden Intervallgrößen überlappen bzw. eine gemeinsame Schnittmenge besitzen müssen.

6.5.5 Kombination der Teilmodelle zum Gesamtmodell

Alle Situationen der Teilmodelle werden untereinander kombiniert und auf Gültigkeit anhand der aufgestellten Kopplungsbedingungen geprüft. Hierzu werden alle Größen an den offenen Schnittstellen der Systembestandteile in die Kopplungsbedingungen eingesetzt. Besitzen die eingesetzten Werte eine gemeinsame Schnittmenge, dann ist die entsprechende Kopplungsbedingung erfüllt. Eine Kombination ist gültig, falls dabei die verknüpften Modellgrößen alle Kopplungsbedingungen erfüllen. Bei einem positiven Ergebnis wird diese Kombination als Systemsituation bezeichnet und stellt ein mögliches Betriebsszenarium des Prozessautomatisierungssystems dar.

Eine Systemsituation besteht daher aus einer Situation des technischen Systems, einer Situation der Automatisierungssoftware und einer Situation der menschlichen Bedieneingriffe. Eine Situation dieser Systembestandteile ist aus Modellen der Systemelemente oder Subsysteme aufgebaut und entsprechend kommentiert und klassifiziert.

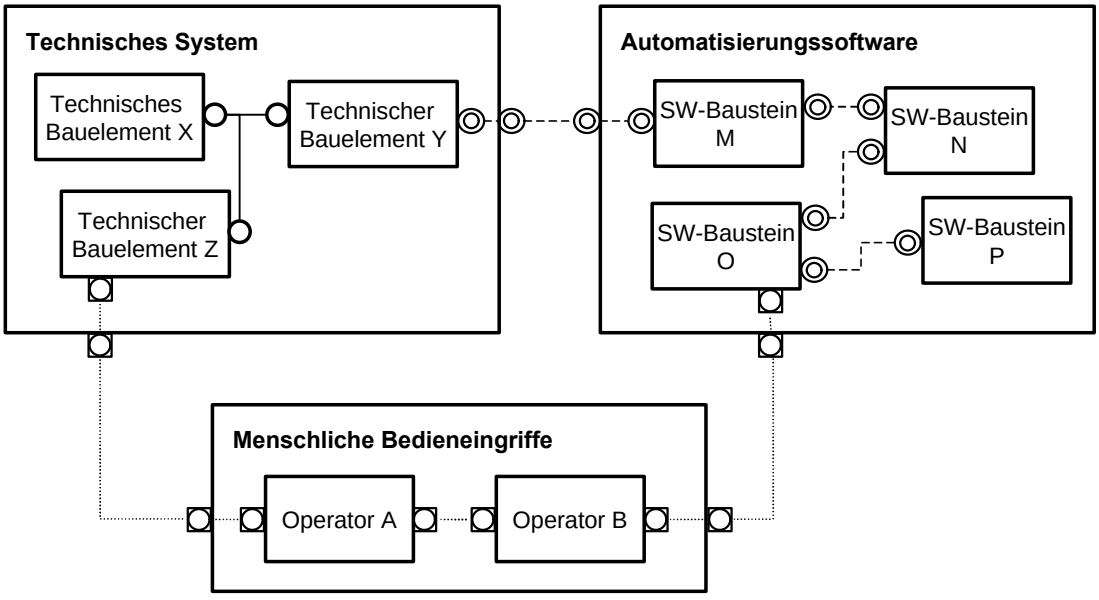


Abbildung 6.22: Beispielhafte Verknüpfung der Systembestandteile eines Prozessautomatisierungssystems

Zu einer Systemsituation lassen sich somit die entsprechenden Situationen von Systemelementen eindeutig zuordnen. Die Ergebnisdarstellung des Gesamtmodells erfolgt in einer Tabelle, die alle Systemsituationen umfasst. In Tabelle 6.5 ist die grundsätzliche Form der Ergebnisdarstellung zu sehen. Die Struktur der Tabelle basiert auf Abbildung 6.22. Die Anordnung der Bestandteile menschlicher Bedieneingriffe, Automatisierungssoftware und technisches System ist frei wählbar.

Tabelle 6.5: Prinzipieller Aufbau einer Systemsituation aus einzelnen Situationen

MB		*ASW*				*TS*			*Status*
A	*B*	*M*	*N*	*O*	*P*	*X*	*Y*	*Z*	
..	..	..	..	..	..	..	..	..	..
Sit9	Sit9	Sit3	Sit3	Sit2	Sit1	Sit1	Sit3	Sit2	G
..	..	..	..	..	..	..	..	..	..

Die empfohlene, dargestellte Anordnung orientiert sich an folgender Überlegung: Der Mensch unternimmt Bedieneingriffe an der Automatisierungssoftware oder direkt am technischen System. Er bestimmt durch sein Verhalten maßgeblich den Ablauf des Prozessautomatisierungssystems. Die Software führt die Befehle des Menschen aus und nimmt entsprechende Änderungen im technischen System vor. Das Ergebnis spiegelt sich in Situationen des technischen Systems wieder. Bei automatisierten Vorgängen prägt die Automatisierungssoftware in erster Linie das Verhalten des technischen Systems. Somit lassen sich Ursache und Auswirkung in der Ergebnisdarstellung von links nach rechts ablesen.

Die erste Zeile der Tabelle 6.5 enthält die Bezeichnung der Systembestandteile, wobei „MB“ für menschliche Bedieneingriffe, „ASW“ für Automatisierungssoftware und „TS“ für technisches System stehen. Die zweite Zeile enthält die Namen der darin enthaltenen Systemelemente bzw. Subsysteme. Die aufgeführte Systemsituation drückt beispielsweise aus, dass das technische Bauelement „Y“ in der Situation Nr. 3 und der Softwarebausteine „N“ in der Situation Nr. 3 sein kann, während der Operator A sich gemäß Situation Nr. 9 seines Modells verhält. Anstatt der Situationsnummern werden die den Situationen zugeordneten qualitativen Ausdrücke dargestellt. Dies ermöglicht eine einfache Interpretation einer Systemsituation. Anhand der zugeordneten Attribute lassen sich fehlerhafte und gefährliche Situationen der Systemelemente identifizieren und einer Systemsituation einen Gesamtstatus zuordnen. Der Gesamtstatus ermöglicht dem Anwender einen schnellen Überblick über nicht-bestimmungsgemäße Systemsituationen. Wie erläutert, sind alle qualitativen Ausdrücke durch Attribute bewertet. Der Gesamtstatus einer Systemsituation entspricht dem dominantesten Klassifizierungsattribut. Dabei gilt die Reihenfolge gefährlich (G) vor unerwünscht (U) vor bestimmungsgemäß (B). Eine Systemsituation wird daher nur als bestimmungsgemäß (B) gekennzeichnet, falls alle enthaltenen Ausdrücke als bestimmungsgemäß klassifiziert sind. Besitzen z. B. eine oder mehrere qualitative Ausdrücke einer Systemsituation das Attribut U, so erhält diese ebenfalls den Gesamtstatus (U). Falls nur ein Ausdruck als gefährlich (G) klassifiziert wurde, so ist der Gesamtstatus der entsprechenden Sys-

temsituation ebenfalls gefährlich (G). Neben dem eingeführten Status ist in der Ergebnisdarstellung weiterhin für jeden qualitativen Ausdruck die zugeordnete Klassifizierung abrufbar. Somit lassen sich die Anzahl von unerwünschten und gefährlichen Vorgängen bzw. Eigenschaften in einer Systemsituation bestimmen.

Die Übergänge von Situationen sind auf Ebene der Teilmodelle durch Transitionen beschrieben. Anhand dieser Transitionen lassen sich Übergänge zwischen Systemsituationen des Gesamtmodells bestimmen. Dabei gilt folgende Regel: *Ein Übergang (Transition) zwischen zwei Systemsituationen existiert genau dann, wenn zwischen den darin festgelegten Situationen der Teilmodelle ebenfalls ein Übergang definiert ist.*

Zusammenfassend ist festzustellen, dass bei der Erstellung der Teilmodelle nicht nur das bestimmungsgemäße Verhalten beachtet wird, sondern ebenfalls sicherheitskritische und unerwünschte Situationen des technischen Systems, Fehler in der Automatisierungssoftware und falsche Bedieneingriffe berücksichtigt werden. Das Gesamtmodell des Prozessautomatisierungssystems wird durch Komposition der drei Teilmodelle gewonnen. Die Idee liegt darin, zu prüfen, ob und unter welchen Voraussetzungen sicherheitskritische Situationen des technischen Systems im Gesamtmodell auftreten bzw. auftreten können. Das Verhalten des Gesamtmodells wird anhand von Systemsituationen und Transitionen dargestellt. Die Durchführung der eigentlichen Sicherheitsanalyse besteht in der Auswertung der ermittelten Systemsituationen und Transitionen, die aufgrund des Zusammenspiels der Systembestandteile möglich sind. Auf diesen Aspekt wird im nächsten Kapitel eingegangen.

7 Modellauswertung der modellbasierten Sicherheitsanalyse

In diesem Kapitel werden zu Beginn allgemeine Aspekte und Mechanismen zur Auswertung des qualitativen Modells vorgestellt. Anschließend wird die Analyse von sicherheitskritischen Systemsituationen unter Berücksichtigung verschiedener Fehlerarten behandelt. Weiterhin wird gezeigt, wie die Prüfung von bestehenden Sicherheitsanforderungen anhand des qualitativen Modells erfolgt. Die Definition von Sicherheitsmaßnahmen und die Vorstellung der Werkzeugunterstützung bei der Durchführung der ganzheitlichen modellbasierten Sicherheitsanalyse schließen dieses Kapitel.

7.1 Allgemeine Aspekte zur Auswertung des qualitativen Gesamtmodells

7.1.1 Interpretation einer Systemsituation

Die qualitativen Ausdrücke einer Systemsituation beschreiben das Verhalten des Prozessautomatisierungssystems. Zur Veranschaulichung soll ein kleines Beispiel für eine Tankstellenanlage dienen, siehe Tabelle 7.1. Die dargestellte Systemsituation sagt aus, dass der Operator Tankwart den Bedieneingriff „tanken“ unternimmt, der Softwarebaustein „BenzinRegler“ sich im Zustand „FÜLLEN“ befindet, das Ventil des technischen Systems geöffnet und der Tank „leer“ ist und gleichzeitig ein „Zufluss“ in den Tank stattfindet. Es handelt sich um eine normale Situation einer Tankanlage, daher ist diese mit dem Attribut „B“ für bestimmungsgemäß gekennzeichnet.

Tabelle 7.1: Beispiel zur Interpretation einer Systemsituation

MB	***ASW***	***TS***		*Status*
Tankwart	*BenzinRegler*	*Ventil*	*Tank*	
..	..	..	..	..
tanken	FÜLLEN	offen	leer/Zufluss	B
..	..	..	..	..

Mit Hilfe des Gesamtstatus lassen sich gefährliche Systemsituationen einfach identifizieren. Innerhalb der Einteilung in gefährliche und unerwünschte Systemsituationen erfolgt eine weitere Unterscheidung nach Anzahl der enthaltenen Fehler und Gefahren. Dies ist möglich, da jede Situation eines Systemelements selbst mit einem Attribut klassifiziert ist. So können die Systemsi-

tuationen gemäß ihrer Bedeutung für die Sicherheit des Prozessautomatisierungssystems sortiert werden.

Für die Analyse des Zusammenspiels der Systembestandteile ist die kompakte Ergebnisdarstellung auf Basis der qualitativen Ausdrücke die übersichtlichste. Mit Hilfe der qualitativen Ausdrücke lassen sich einfache Filter- und Suchroutinen anwenden. Der gesamte Situationsraum kann damit auf konkrete Vorgaben hin analysiert und Fragen von folgender Art beantwortet werden: Ist es möglich, dass alle Ventile geöffnet sind? Welche Situationen kann die Automatisierungssoftware einnehmen, wenn z. B. alle Ventile geöffnet sind? Welche Situationen können die Systemelemente X und Y einnehmen, falls Systemelement Z den Fehler A aufweist?

7.1.2 Erreichbarkeitsanalyse

Transitionen zwischen den Systemsituationen zeigen dem Anwender, in welcher Weise Verhaltensänderungen eines Prozessautomatisierungssystems möglich sind. Die Untersuchung der Transitionen erfolgt im Rahmen einer Erreichbarkeitsanalyse. Dabei werden folgende Untersuchungen unterstützt:

- *Erkennung von isolierten Systemsituationen*
  Isolierte Systemsituationen sind Systemsituationen, welche weder erreicht noch verlassen werden können. Sie besitzen keine Übergänge in andere Systemsituationen. In Abbildung 7.1 ist die Systemsituation S8 eine isolierte Situation.

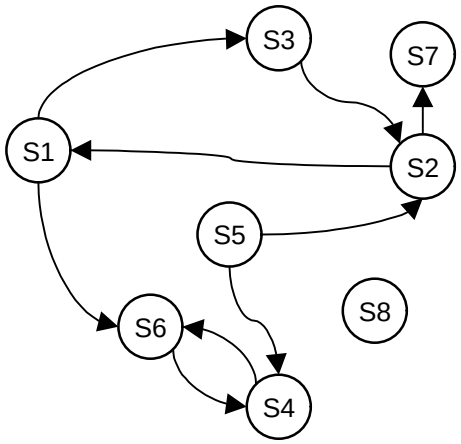


Abbildung 7.1: Beispielhafte Transitionen von Systemsituationen

- *Prüfen von Übergängen*
  Zwischen zwei Systemsituationen A und B kann festgestellt werden, ob ein Übergang besteht. Falls ein oder mehrere Übergänge existieren, erhält der Anwender Auskunft über die Anzahl der benötigten Übergänge. Die Angaben werden nach Anzahl der benötigten Schritte sortiert. Soll geprüft werden, ob zwischen den Systemsituationen S1 und S2 ein Übergang

möglich ist, dann lautet gemäß Abbildung 7.1 das Ergebnis: Ein Übergang zwischen S1 und S2 ist möglich und zwar über S3 in zwei Schritten oder über S6, S4, S5 in vier Schritten.

- *Anzeigen von Vorgängern und Nachfolgern*
  Alle, einer Systemsituation vorangehenden und nachfolgenden Systemsituationen, können dem Anwender aufgezeigt werden. In Abbildung 7.1 besitzt die Systemsituation S2 die Vorgänger S3 und S5 und die Nachfolger S1 und S7.

- *Erkennen von End-Situationen*
  End-Situationen können nicht mehr verlassen werden, d.h. sie besitzen keine Nachfolger, wie z. B. die Systemsituation S7 in Abbildung 7.1.

7.2 Analyse von sicherheitskritischen Systemsituationen

7.2.1 Ursachen von sicherheitskritischen Systemsituationen

Die Basis für die Durchführung der Sicherheitsanalyse sind die klassifizierten und kommentierten Systemsituationen eines Prozessautomatisierungssystems. Dabei ist analysierbar, ob und welche sicherheitskritischen Systemsituationen aufgrund des Zusammenspiels der Systembestandteile möglich sind. Der Anwender muss die vorhandenen sicherheitskritischen Systemsituationen (Gesamtstatus „G“), untersuchen und bewerten. Folgende Fragen stehen bei der Untersuchung im Vordergrund:

- Existieren sicherheitskritische Systemsituationen für das Prozessautomatisierungssystem?
- Was sind die Ursachen für sicherheitskritische Systemsituationen?
- Welche Fehler sind notwendig bzw. was muss passieren, damit es zu sicherheitskritischen Systemsituationen kommen kann?

In Abbildung 7.2 ist das Vorgehen zur Auswertung von Systemsituationen dargestellt. Im ersten Schritt ist zu ermitteln, ob sicherheitskritische Systemsituationen (Gesamtstatus „G“) aufgrund eines hypothetischen Fehlers oder durch Fehlerkombinationen entstehen konnten. Um diese Frage zu beantworten, werden die ermittelten sicherheitskritischen Systemsituationen zusätzlich nach der Anzahl der enthaltenen hypothetischen Fehler oder unerwünschten qualitativen Ausdrücke sortiert. Hypothetische Fehler sind mit dem Attribut „U“ für unerwünscht klassifiziert. In Tabelle 7.2 sind drei Situationen dargestellt, von denen zwei (Systemsituation 4 und 5) den Status „gefährlich“ („G“) besitzen.

Der gefährlichen Systemsituation Nr. 4 liegt der hypothetische Fehler „defektes offenes Ventil“ (Attribut „U“) zugrunde. Diese entspricht Fall 3 in Abbildung 7.2 und wird in Kapitel 7.2.3 genauer beschreiben.

Tabelle 7.2: Beispiel für verschieden klassifizierte Systemsituationen

Nr.	MB	ASW	TS		Status
	Tankwart	BenzinRegler	Ventil	Tank	
..	..	..	..	..	..
3	tanken (B)	FÜLLEN (-)	offen (B)	leer/Zufluss (B)	B
4	stop (B)	BEREIT (-)	offen/defekt (U)	voll/Überlauf(G)	G
5	stop (B)	FÜLLEN (-)	offen (B)	voll/Überlauf (G)	G
..	..	..	..	..	..

Existieren allerdings sicherheitskritische Systemsituationen, die sich nicht auf hypothetische Fehler zurückführen lassen, dann sind die dazugehörigen Situationen des Teilmodells der Automatisierungssoftware zu analysieren, Fall 2 in Abbildung 7.2. Wie im sechsten Kapitel beschrieben, sind die Situationen dieses Teilmodells nicht klassifiziert, da bei Software nur inhärente Fehler vorliegen können. Die Auswirkungen von inhärenten Fehlern können anhand ihrer Folgen ermittelt werden. Genau diese Tatsache wird beim ganzheitlichen Ansatz genutzt. Enthalten daher sicherheitskritische Systemsituationen keine hypothetischen Fehler (also neben dem Attribut „G“ nur noch „B“ Attribute), so kann mit hoher Wahrscheinlichkeit davon ausgegangen werden, dass die Ursache auf einen vorhandenen Fehler im Entwurf der Automatisierungssoftware zurückzuführen ist. Dieser Fall wird in Kapitel 7.2.2 behandelt.

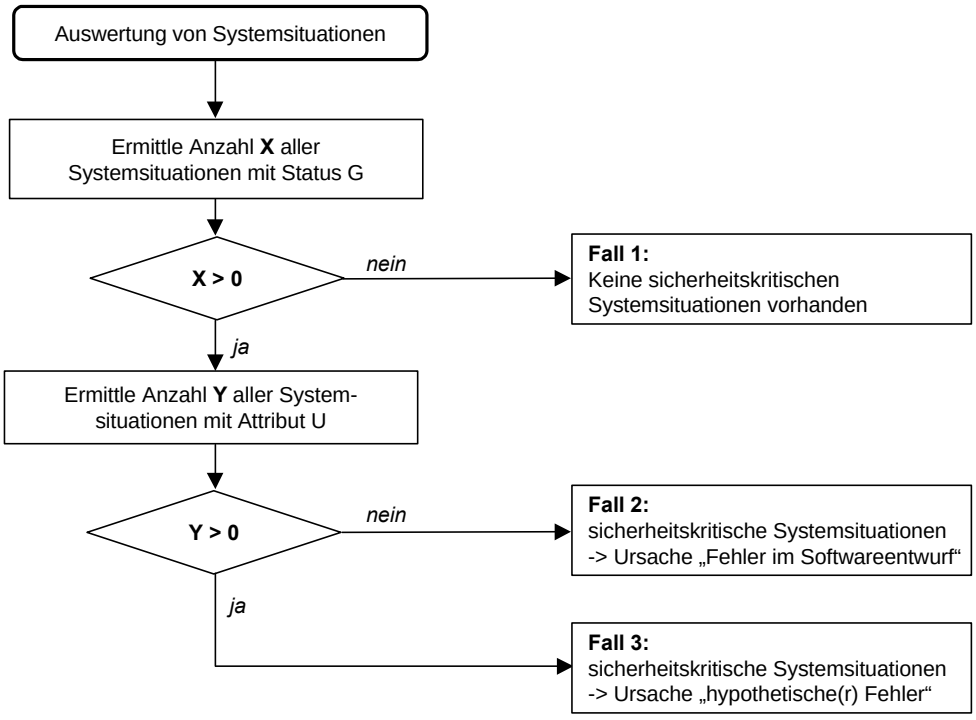


Abbildung 7.2: Vorgehen zur Auswertung von Systemsituationen

7.2.2 Untersuchung von potenziellen Fehlern in der Automatisierungssoftware

Eine Systemsituation setzt sich, wie in Kapitel 6.5.5 beschrieben, aus einer gültigen Kombination von Situationen der drei Systembestandteile zusammen. Enthalten sicherheitskritische Systemsituationen keine Szenarien für mögliche Fehler im technischen System und bei den menschlichen Bedieneingriffen, so ist die Ursache im Entwurf der Automatisierungssoftware zu suchen.

Dabei stellen einzelnen Situationen der Systemelemente, die in der sicherheitskritischen Systemsituation zusammengefasst sind, nützliche Informationen für die Fehlersuche dar. Die Situationen des technischen Systems und der menschlichen Bedieneingriffe sind konkrete Anhaltspunkte zur Ermittlung von Entwurfsfehlern der Automatisierungssoftware.

Die einer sicherheitskritischen Systemsituation zugrunde liegende gefährliche Situation eines technischen Bauelements ist der Ausgangspunkt bei der Ermittlung des Fehlers. Die gefährliche Situation des technischen Bauelements muss durch entsprechende Funktionen der Automatisierungssoftware verhindert werden. Die Lokalisierung eines Softwarefehlers erfolgt anhand der zugehörigen Situationen der Softwarebausteine. Der Anwender der Sicherheitsanalyse muss die Plausibilität der Situationen der Automatisierungssoftware mit Situationen anderer Teilmodelle prüfen, die in der gefährlichen Systemsituation zusammengefasst sind. Die den Situationen zugrunde liegenden Modellgrößen sind eine weitere Informationsquelle. Abbildung 7.3 enthält eine Grafik zur Veranschaulichung. Die Automatisierungssoftware besteht aus drei Softwarebausteinen. Diese befinden sich im Zustand1, Zustand4 und Zustand6. Der Informationsaustausch (Informationsgröße „Sollwert“), zwischen Softwarebaustein in Zustand1 und dem in Zustand6 besitzt einen Wert zwischen 30 und 40. So kann überprüft werden, ob nicht etwa eine falsche Sollwertermittlung, die Ursache für die gefährliche Situation des technischen Systems ist.

Mit Hilfe dieser konkreten Information können Entwickler der Automatisierungssoftware den Entwurf prüfen und verbessern. In Tabelle 7.2 ist ein weiteres Beispiel dargestellt. Der Softwarebaustein „BenzinRegler“ ist im Zustand „FÜLLEN“. Dieser Zustand ist aufgrund des Softwareentwurfs auch beim Bedieneingriff „stop“ des Operators „Tankwart“ möglich. Bestimmungsgemäß müsste beim Bedieneingriff „stop“ der „BezinRegler“ allerdings im Zustand „bereit“ sein. Es handelt sich hierbei um einen Fehler in der Automatisierungssoftware.

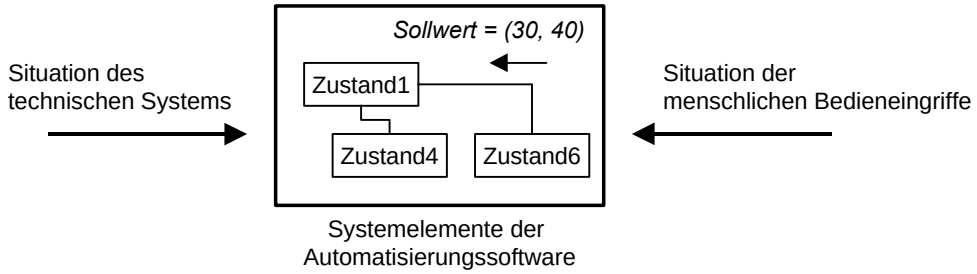


Abbildung 7.3: Verfügbare Informationen zur Lokalisierung potenzieller Softwarefehler bei einer sicherheitskritischen Systemsituation

Konnte ein Fehler in der Automatisierungssoftware als Ursache für die sicherheitskritische Systemsituation identifiziert werden, so stehen dem Anwender zwei Möglichkeiten für die weitere Durchführung der Sicherheitsanalyse zur Verfügung:

- Die Klassifizierung des gefundenen Fehlers in Form des Attributes „F“ oder
- Die Korrektur des Fehlers im Entwurf der Automatisierungssoftware

Die *Klassifizierung* dient zur konkreten Kennzeichnung der ermittelten sicherheitskritischen Systemsituation. Der Fehler im entsprechenden Softwarebaustein wird analog zu den hypothetischen Fehlern im Modell des technischen Systems und der menschlichen Bedieneingriffe gekennzeichnet, allerdings mit dem wichtigen Unterschied, dass es sich hierbei um einen tatsächlich vorhandenen Fehler handelt und nicht um einen hypothetischen. Um dieser Tatsache Rechnung zu tragen, wird das Attribut „F“ eingeführt. Nach Klassifizierung aller identifizierter Softwarefehler wird mit der nächsten Stufe der Sicherheitsanalyse, der Analyse unter Berücksichtigung möglicher (hypothetischer) Fehler, fortgefahren.

Die zweite Alternative sieht die *Korrektur* des Entwurfs der Automatisierungssoftware und damit verbunden die Ursachenbehebung der ermittelten sicherheitskritischen Systemsituation vor. Die Wirksamkeit der Korrekturmaßnahmen kann mit Hilfe des qualitativen Modells erprobt werden. Dazu wird der korrigierte Entwurf wiederum in ein äquivalentes Softwaremodell überführt und das Gesamtmodell erstellt. Bei einer erfolgreichen Korrektur eines Fehlers dürfen die entsprechenden sicherheitskritischen Systemsituationen nicht mehr vorhanden sein.

7.2.3 Untersuchung hypothetischer Fehler

Bei der Betrachtung von hypothetischen Fehlern lässt sich die Frage beantworten: Welche Fehler müssen auftreten, damit sicherheitskritische Systemsituationen entstehen können? - Dabei lassen sich *Einzelfehler* und *Mehrfachfehler* unterscheiden. Die Eintrittswahrscheinlichkeit von Einzelfehlern ist höher als von Mehrfachfehlern. Beispielsweise ist es wahrscheinlicher, dass ein einzelnes Ventil ausfällt als dass zwei Ventile gleichzeitig ausfallen.

Eine sicherheitskritische Systemsituation, die auf einem hypothetischen *Einzelfehler* beruht, besitzt neben dem Attribut „G“ genau einen qualitativen Ausdruck, der mit dem Attribut „U“ klassifiziert ist (siehe Abbildung 7.2: Fall 3 mit Y = 1). Dieser qualitative Ausdruck beschreibt die Fehlerart und kann sowohl im qualitativen Modell des technischen Systems als auch im qualitativen Modell der menschlichen Bedieneingriffe verankert sein. In der Regel gilt, dass Einzelfehler, die direkt eine sicherheitskritische Situation implizieren, durch entsprechende Maßnahmen zu verhindern sind. Im speziellen Fall muss der Anwender prüfen, in wieweit ein hypothetischer Fehler realistisch ist, siehe Kapitel 7.4.

Eine sicherheitskritische Systemsituation, die auf einem hypothetischen *Mehrfachfehler* beruht, besitzt neben dem Attribut „G“ bzw. Gesamtstatus „G“ zwei oder mehrere qualitative Ausdrücke, die mit dem Attribut „U“ klassifiziert sind (siehe Abbildung 7.2: Fall 3 mit Y >1). Wichtig ist, ob eine sicherheitskritische Systemsituation wirklich auf der Kombination von zwei oder mehreren Fehlern beruht oder nur auf das Vorhandensein eines bestimmten, schon behandelten Einzelfehlers zurückzuführen ist. Diese Frage lässt sich mit einem Vergleich der Ergebnisse aus der Einzelfehlerbetrachtung beantworten. Falls neue sicherheitskritische Systemsituationen vorhanden sind, dann sind diese aufgrund von Fehlerkombinationen entstanden. Dabei wird angenommen, dass die Wahrscheinlichkeit von 2-fachen Fehlern im Allgemeinen höher ist als die von n-fachen Fehlern. Somit erhält der Anwender eine im Hinblick auf die Systemsicherheit vorsortierte Ergebnisdarstellung.

In der Ergebnisdarstellung sind ebenfalls die Auswirkungen von Einzelfehlern enthalten, die nicht zu einer sicherheitskritischen Systemsituation führen. Mit Hilfe dieser Informationen können Angaben zum normalen Betrieb unter Fehlereinwirkung gemacht werden, um somit die Zuverlässigkeit eines Systems zu untersuchen, siehe Kapitel „Ausblick“.

7.2.4 Untersuchung von Kommunikationsfehlern zwischen Systembestandteilen

Wie in Abbildung 5.1 dargestellt, können Kommunikationsfehler zwischen den Systembestandteilen ebenfalls mögliche Ursachen für sicherheitskritische Systemsituationen eines Prozessautomatisierungssystems sein. Um die Auswirkungen von Kommunikationsfehler zu analysieren, müssen in der Netzliste des qualitativen Gesamtmodells Verbindungen zwischen den Systembestandteilen entfernt werden. Durch die Trennung einer Verbindung werden offene Schnittstellen geschaffen, d.h. an den davon betroffenen Schnittstellen werden keine konkreten Vorgaben gemacht. Die Trennung einer Verbindung eines aktiven technischen Bauelements hat ein nichtdeterministisches (bzw. unkontrolliertes) Verhalten des betroffenen Bauelements zur Folge, siehe Kapitel 6.2.1, und entspricht der Auswirkung von unterschiedlichen Kommunikationsfehlern (Signalstörung bzw. –verfälschung oder Verbindungsabbruch). Durch die Trennung einer Verbindung existiert mindestens eine Kopplungsbedingung weniger. Anschließend müssen die Sys-

temsituationen des Gesamtmodells neu ermittelt werden. Aus der Ergebnisdarstellung lässt sich anhand des Status der Systemsituationen prüfen, ob durch die Trennung einer Verbindung sicherheitskritische Systemsituationen auftreten können.

Bei der Betrachtung von Kommunikationsfehlern wird ebenfalls zwischen Einzel- und Mehrfachfehler unterschieden. Bei der Analyse von Einzelfehlern werden sukzessiv alle Verbindungen zwischen den Systembestandteilen entfernt. Es wird geprüft, ob dadurch zu sicherheitskritische Systemsituationen entstehen können. Bei Mehrfachfehlern, d.h. bei Kombinationen von Kommunikationsfehlern, kann der Anwender die maximale Anzahl von gleichzeitigen Fehlern bestimmen. Je höher die Anzahl, desto unwahrscheinlicher ist das Auftreten des Mehrfachfehlers in einem realen System.

7.3 Prüfung von Sicherheitsanforderungen

7.3.1 Arten von Sicherheitsanforderungen

Sicherheitsanforderungen an ein Prozessautomatisierungssystem liegen gewöhnlich in textueller Form vor. Grundsätzlich werden zwei Arten von Sicherheitsanforderungen unterschieden:

Invariante Sicherheitsanforderungen betreffen Eigenschaften, die während des gesamten Betriebs erfüllt sein müssen. Zum Beispiel: „Alle Auslassventile des Tanks X dürfen nie zur gleichen Zeit geöffnet werden bzw. offen sein.“

Dynamische Sicherheitsanforderungen beziehen sich auf bestimmte Betriebssituationen des Prozessautomatisierungssystems. Sie lassen sich in eine Vorbedingung und in eine Nachbedingung unterteilen, die über wenn-dann Regeln verknüpft sind. Zum Beispiel: „Wenn Tank X gefüllt wird, dann muss der Tankdeckel geschlossen sein.“

7.3.2 Überprüfung von Sicherheitsanforderungen anhand des qualitativen Gesamtmodells

Die Aussagen von Sicherheitsanforderungen werden mit den Aussagen des qualitativen Gesamtmodells verglichen. Bei widersprüchlichen Aussagen ist die betreffende Sicherheitsanforderung nicht erfüllt. Sicherheitsanforderungen werden ohne die Annahme von hypothetischen Fehlern untersucht. Hierzu werden alle Systemsituationen, die das Attribut „U“ enthalten, aus der Ergebnisdarstellung ausgeblendet. Je nach Art der Sicherheitsanforderung wird unterschiedlich vorgegangen:

Invariante Sicherheitsanforderungen sind durch eine entsprechende Filterung der Ergebnisdarstellung zu überprüfen. Hierzu muss lediglich die Aussage der Sicherheitsanforderung negiert werden. Zum Beispiel entsteht aus der Anforderung „Alle Auslassventile des Tanks X dürfen

nicht gleichzeitig geöffnet werden“ folgende Fragestellung an das qualitative Gesamtmodell: „Existiert eine Systemsituation, in der alle Auslassventile des Tanks X gleichzeitig geöffnet sind?“

Um diese Frage zu beantworten, müssen zuerst die in der Sicherheitsanforderung genannten Systemelemente (oder Subsysteme) im qualitativen Modell identifiziert werden, siehe Abbildung 7.4. Für obiges Beispiel sind alle modellierten technischen Bauelemente relevant, die Auslassventile des Tanks X sind. Die Situationen dieser identifizierten Systemelemente werden entsprechend der Sicherheitsanforderung vorgegeben bzw. eingeschränkt. Falls keine entsprechenden Situationen oder gar Systemelemente mit der Sicherheitsanforderung in Bezug gebracht werden können, dann ist mit dem vorhandenen Modell keine Aussage möglich. Für die identifizierten qualitativ modellierten Auslassventile des Tanks X werden nur Situationen mit dem Kommentar „offen“ zugelassen.

Alternativ können bei konkreten Wertevorgaben bestimmte Intervallbereiche der Modellgrößen vorgegeben werden. Zum Beispiel: „Die Temperatur des Wassers im Tank darf 350K nicht überschreiten“ bzw. die Negation als Frage formuliert: „Existieren Systemsituationen, in denen die Wassertemperatur im Tank größer als 350K ist?“ Somit sind alle Systemsituationen betroffen, in denen die qualitative Größe der Temperatur des Tankinhalts im Wertebereich von (350, ∞) liegen.

Der Anwender muss prüfen, ob für die entsprechenden Vorgaben Systemsituationen bestehen oder nicht. Eine invariante Sicherheitsanforderung ist dann erfüllt, wenn keine Systemsituation für die gemachten Vorgaben vorhanden ist. Das bedeutet, dass sich das Prozessautomatisierungssystem niemals in einem solchen Betriebsszenarium befinden kann. Werden hingegen Systemsituationen gefunden, so ist die entsprechende Sicherheitsanforderung nicht erfüllt. Das Prozessautomatisierungssystem besitzt Betriebszustände, die die Sicherheitsanforderung verletzen. Diese Systemsituationen werden dem Anwender aufgeführt (in der Literatur wird hierbei auch von Gegenbeispiel - engl. counter example- gesprochen).

Das qualitative Modell bietet darüber hinaus die Möglichkeit, zu prüfen, unter welchen Fehlerannahmen eine im bestimmungsgemäßen Betrieb erfüllte Sicherheitsanforderung nicht mehr erfüllt werden kann. Hierzu wird von der vollständigen Ergebnisdarstellung ausgegangen, d.h. es werden auch Systemsituationen mit hypothetischen Fehlern zugelassen. Das Vorgehen ist mit dem oben geschilderten identisch.

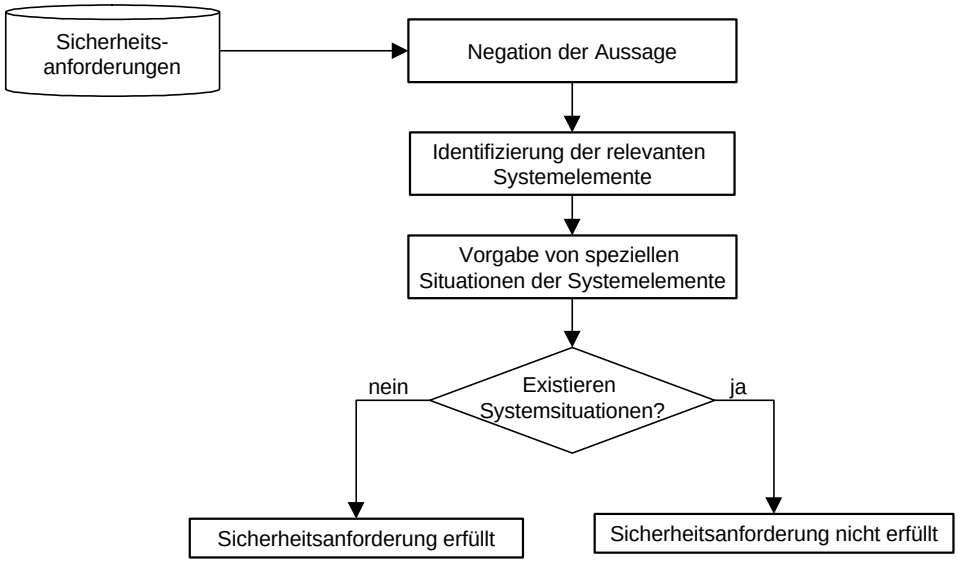


Abbildung 7.4: Vorgehen bei invarianten Sicherheitsanforderungen

Dynamische Sicherheitsanforderungen müssen mit Hilfe der Filtertechnik und der Erreichbarkeitsanalyse geprüft werden. Dynamische Sicherheitsanforderungen, die für das ganze System gelten, werden ähnlich wie invariante Sicherheitsanforderungen behandelt. Der Unterschied liegt darin, dass nur die Nachfolgebedingung, nicht aber die Vorbedingung negiert wird. Für das vorherige Beispiel „wenn Tank X gefüllt wird, dann muss der Tankdeckel geschlossen sein“ folgt die Fragestellung „Existiert eine oder mehrere Systemsituationen, bei denen Tank X gefüllt wird und gleichzeitig der Tankdeckel offen ist?“

Dynamische Sicherheitsanforderungen mit temporalen Aspekten werden mit Hilfe der Erreichbarkeitsanalyse bearbeitet. Die Vorbedingung bestimmt die Ausgangs-Systemsituation, die Nachbedingung die Ziel-Systemsituation. Mit Hilfe der Erreichbarkeitsanalyse wird beantwortet, ob Übergänge von der Ausgangssituation in die Zielsituation möglich sind. Falls Übergänge existieren, kann ausgesagt werden, in wie vielen minimalen Schritten und über welche Systemsituationen die Ziel-Systemsituation erreicht werden kann. Ein kleines Beispiel dient zur Erläuterung: „Ist Behälter X leer, so muss dieser, bevor er gefüllt wird, erst entlüftet werden“. Es erfolgt die Identifizierung der Ausgangs-Systemsituationen, die einen leeren Behälter X darstellen. Die Erreichbarkeitsanalyse gibt alle möglichen Übergänge zu unmittelbar nachfolgenden Systemsituationen an. Falls diese nachfolgenden Systemsituationen z. B. einen „Zufluss in den leeren Behälter“ beschreiben, dann ist die Sicherheitsanforderung nicht erfüllt, da der Behälter erst zu entlüften ist.

7.4 Definition von Sicherheitsmaßnahmen

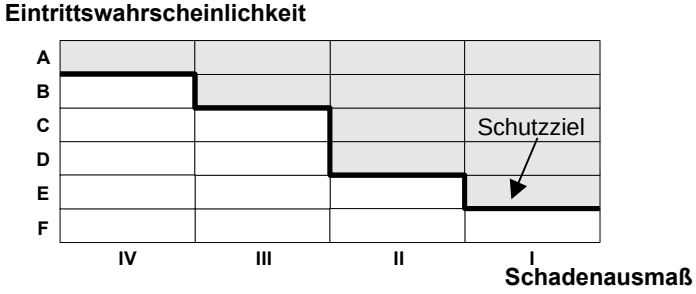
7.4.1 Notwendigkeit von Sicherheitsmaßnahmen

Der Anwender muss aufgrund der gefundenen sicherheitskritischen Systemsituationen entscheiden, ob Sicherheitsmaßnahmen zu erfolgen haben. Unabhängig vom Ergebnis muss er seine Entscheidungen schriftlich begründen. Die Begründung erfolgt direkt in der Ergebnisdarstellung, die für diesen Zweck um die Spalte „Bemerkungen“ ergänzt wird, siehe Tabelle 7.3. Dieses Bemerkungsfeld muss für alle gefährlichen Systemsituationen ausgefüllt werden.

Tabelle 7.3: Ergebnisdarstellung mit zusätzlicher Bemerkungsspalte

MB	***ASW***	***TS***			
Tank-	*BenzinRealer*	*Ventil*	*Tank*	*Status*	*Bemerkungen*
..	..	..	..		..
stop	BEREIT	offen/defekt	voll/Überlauf	G	`Ein defektes Ventil ist unwahrscheinlich. Es werden keine Sicherheitsmaßnahmen ergriffen.`
..	..	..	..		..

Die Entscheidung des Anwenders, Sicherheitsmaßnahmen zu treffen, hängt von vielen Faktoren ab. Gemäß der Risikobewertung nach [VDI/VDE 3542] müssen der in einer sicherheitskritischen Systemsituation dargestellte Schaden (Schadenausmaß) sowie die Auftretenswahrscheinlichkeit seiner Ursache (Fehler) betrachtet werden. Je nach Art des Fehlers (Einzelfehler, Mehrfachfehler) steht dem Anwender die in Abbildung 7.5 dargestellte Empfehlung zur Verfügung.



Eintrittswahrscheinlichkeit		Schadenausmaß	
A	häufig	I	katastrophal
B	oft	II	kritisch
C	gelegentlich	III	begrenzt
D	selten	VI	vernachlässigbar
E	unwahrscheinlich		
F	unmöglich		

Abbildung 7.5: Schutzziel in Abhängigkeit von Eintrittswahrscheinlichkeit und Schadenausmaß

Identifizierte Softwarefehler sind müssen korrigiert werden. Ebenfalls stellen Einzelfehler ein hohes Risiko dar, falls diese die alleinige Ursache einer sicherheitskritischen Systemsituation sind.

In dieser Arbeit wird eine quantitative Berechnung des Risikos nicht verfolgt. Wahrscheinlichkeiten beruhen auf Werten, die durch Laboruntersuchungen, Dauertests oder einfach durch Beobachten des menschlichen Verhaltens ermittelt wurden. Es sind Angaben, die mit größtmöglicher Vorsicht zu behandeln sind und schon in vielen Fällen falsche Entscheidungen beeinflusst haben [Leve95].

7.4.2 Erstellen von Sicherheitsmaßnahmen

Nach [LaGö99a] lassen sich Sicherheitsmaßnahmen nach Perfektions- und Non-Perfektions-Strategien unterscheiden.

Perfektions-Strategien umfassen Maßnahmen zum Ausschluss Fehlern. Mögliche Fehler, die zu sicherheitskritischen Systemsituationen des Prozessautomatisierungssystems führen, werden durch konstruktive Maßnahmen verhindert. Das Ziel ist die Entwicklung eines fehlerfreien Prozessautomatisierungssystems. Prinzipiell stellt die modellbasierte Sicherheitsanalyse selbst einen Mechanismus zur Perfektions-Strategie dar. Vorhandene Entwurfsfehler der Automatisierungssoftware werden ermittelt und behoben.

Non-Perfektions-Strategien umfassen Maßnahmen zur Verhinderung gefährlicher Auswirkungen bei Ausfällen und Fehlern. Viele Fehlerarten, wie z. B. der Ausfall eines technischen Bauelements, können nicht ausgeschlossen werden. Als wichtige Mechanismen für Non-Perfektions-Strategien sind das Fail-Safe-Verhalten und die Ausführung von redundanten Systemelementen zu nennen. Wichtige technische Bauelemente, deren Fehlverhalten unmittelbar eine Katastrophe auslösen, werden oft in redundanter Bauweise ausgeführt. Von Fail-Safe-Verhalten wird gesprochen, wenn beim Auftreten eines kritischen Fehlers das Prozessautomatisierungssystem in einen sicheren Zustand überführt wird. Das Fail-Safe-Verhalten setzt allerdings die Erkennung von (bekannten) Fehlern voraus. Dies kann z. B. durch zusätzliche Überwachungsfunktionen der Automatisierungssoftware erreicht werden. In Abbildung 7.6 sind die unterschiedlichen Aktionsbereiche von Sicherheitsmaßnahmen aufgeführt.

Bei Sicherheitsmaßnahmen, die der Anwender im Rahmen der modellbasierten Sicherheitsanalyse zu treffen hat, handelt es sich um Non-Perfektions-Strategien. Eine nützliche Hilfestellung für die Erstellung von Sicherheitsmaßnahmen ist die Anwendung der Erreichbarkeitsanalyse. Damit lassen sich schon frühzeitig Symptome identifizieren, die zu einer gefährlichen Systemsituation führen können.

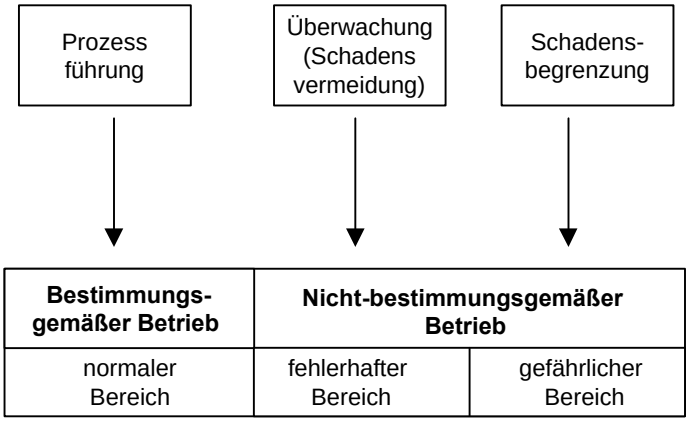


Abbildung 7.6: Aktionsbereiche von Sicherheitsmaßnahmen

Angenommen ein alleiniger Fehler eines Systemelements X besitzt keine Auswirkung auf die Sicherheit eines Prozessautomatisierungssystems, stellt aber in Kombination mit einem anderen Fehler die Ursache einer sicherheitskritischen Systemsituation. Zur Vermeidung dieser sicherheitskritischen Systemsituation ist die Überwachung des Systemelements X zweckdienlich. Beim Auftreten des Fehlers kann z. B. das technische System in einen sicheren Zustand überführt und angehalten werden, bis der Fehler behoben ist.

Überwachungsfunktionen stellen die wichtigste Form von Non-Perfektions-Strategien dar und ermöglichen einen automatischen Schutz des Prozessautomatisierungssystems vor sicherheitskritischen Situationen [Stro92]. Die einfachste Ausführung einer Überwachung ist die Kontrolle des Toleranzbereichs wichtiger Prozessgrößen (signalorientierte Überwachung) [Bieg97b]. Verlassen die Prozessgrößen diesen Bereich, so müssen Maßnahmen automatisch eingeleitet werden. Einen vollständigen Überblick über unterschiedliche Überwachungsstrategien bietet [LaGö99b].

Eine weitere Sicherheitsmaßnahme besteht in einer häufigen Wartung und Kontrolle von sicherheitskritischen Systemelementen. Sicherheitskritische Systemelemente sind Systemelemente, bei deren Ausfall die Sicherheit des gesamten Systems nicht mehr gewährleistet ist. Solche Elemente lassen sich im Rahmen der ganzheitlichen modellbasierten Sicherheitsanalyse in Form der Einzelfehleranalyse identifizieren.

7.4.3 Kontrolle der definierten Sicherheitsmaßnahmen

Die definierten Sicherheitsmaßnahmen oder auch Korrekturen der Automatisierungssoftware können anhand des qualitativen Gesamtmodells überprüft bzw. getestet werden.

Änderungen des Prozessautomatisierungssystems, die durch Sicherheitsmaßnahmen bedingt sind, werden ebenfalls in die entsprechenden qualitativen Modelle der Systembestandteile eingetragen. Anschließend erfolgt eine erneute Ermittlung der möglichen Systemsituationen. Führen die definierten Sicherheitsmaßnahmen zum Erfolg, so dürfen die dadurch betroffenen sicher-

heitskritischen Systemsituationen nicht mehr im Gesamtmodell vorhanden sein. Falls doch, liegt voraussichtlich ein Fehler in den erstellten Sicherheitsmaßnahmen vor.

Werden Fehler im Entwurf der Automatisierungssoftware behoben, so muss das dazugehörige qualitative Modell neu generiert werden. Bei einer anschließenden Neuberechnung der möglichen Systemsituationen ist zu berücksichtigen, dass neue Entwurfsfehler vorhanden sein können. Um dieser Tatsache Rechnung zu tragen, muss zuerst das in Kapitel 7.2.2 beschriebene Vorgehen durchgeführt werden.

7.5 Werkzeugunterstützung bei der Durchführung einer ganzheitlichen modellbasierten Sicherheitsanalyse

Im Rahmen der Arbeit wurden sowohl zur Erstellung qualitativer Modelle sowie zur Modellanalyse Softwarewerkzeuge entwickelt, die den Anwender bei der Durchführung der modellbasierten Sicherheitsanalyse unterstützen.

7.5.1 Modellassistent

Der entwickelte Modellierungsassistent MODAS führt den Anwender durch die qualitative Modellierung und übernimmt aufwändige Routinearbeit. MODAS verfügt über folgende Merkmale:

- *Modellierung von Systemelementen*
  Die Modellierung erfolgt schrittweise über Dialoge mit dem Anwender. Der Anwender muss weder mit dem Aufbau einer SQMA-Modelldefinition noch mit der Syntax der Beschreibungsmittel vertraut sein. Über einen Dialog wird sichergestellt, dass der Anwender nur mit bereits definierten Modelldaten weiterarbeiten kann. Dabei wird automatisch die Konsistenz der Angaben zwischen den verschiedenen Beschreibungsmitteln gewährleistet. Die Dialogführung unterstützt den Anwender, indem Vorschläge für die qualitative Modellierung gemacht werden, die sich aus dem bisherigen Modellierungskontext ableiten lassen.

- *Visuelle Modellierung*
  Für jedes modellierte Systemelement wird stellvertretend ein grafisches Symbol generiert, das den Namen und sämtliche Schnittstellen des Systemelements aufweist. Alle Symbole werden in einer Schablone für die weitere Bearbeitung zur Verfügung gestellt. Per „Drag und Drop“ erfolgt die Übertragung der Symbole aus der Schablone auf ein Zeichenblatt. Der geschilderte Vorgang ist mit einer Instanziierung von Objekten bei objektorientierten Entwurfmethoden zu vergleichen. Jedes qualitativ beschriebene Systemelement kann in einem Modell mehrmals auftreten. Die Verknüpfungen zwischen den Systemelementen werden durch Linien auf der grafischen Ebene angegeben, siehe Abbildung 7.7. Die Verbindungen der Systemelemente werden überprüft und daraus automatisch eine entsprechende Netzliste

erzeugt. MODAS ist ebenfalls in der Lage, aus einer schon bestehenden Netzliste eine grafische Systemstruktur zu generieren.

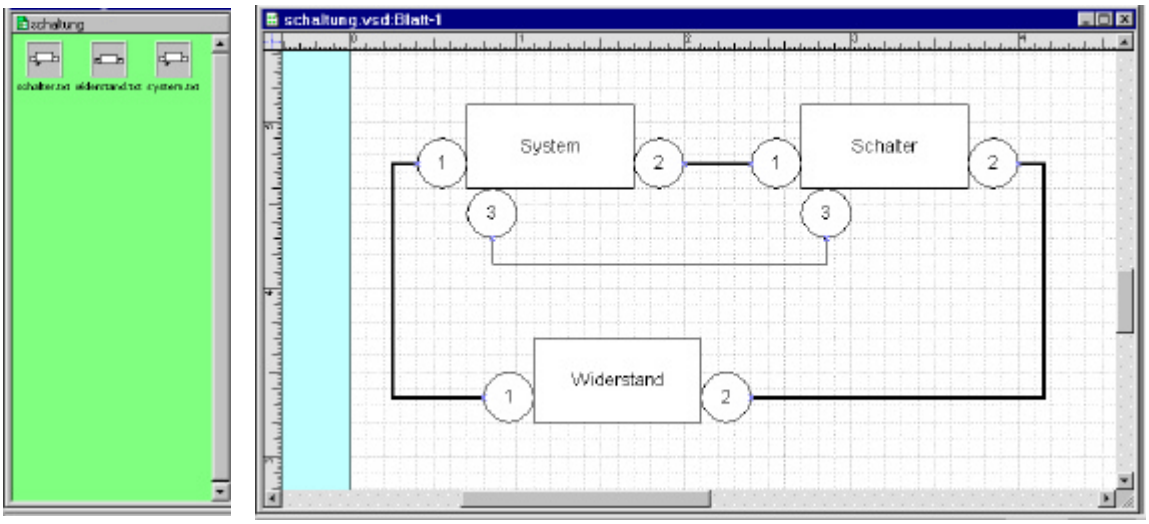


Abbildung 7.7: Visuelle Modellierung

- *Konfigurationsmanagement*
  MODAS organisiert und verwaltet Modelldaten eigenständig. Es wird z. B. gewährleistet, dass vorhandene Situationen konsistent zu den Modelldefinitionen sind. Der Stand der qualitativen Modellierung ist in einem Projektfenster ersichtlich. Die beschriebenen Systemelemente besitzen entsprechend ihrem Modellierungsfortschritt unterschiedliche Attribute.

7.5.2 Konverter für die Automatisierungssoftware

Das in Kapitel 6.3.2.2 vorgestellte Konzept zur rechnergestützten Modellierung der Automatisierungssoftware fand in einem Softwarewerkzeug Anwendung [Schl99]. Dabei werden die Entwurfsdaten ausgehend von einem UML-RT/ROOM Softwareentwurf ausgewertet und in die entsprechenden Softwarebausteine zerlegt. Zu jedem identifizierten Softwarebaustein wird eine SQMA-Modelldatei (qualitatives Modell des Softwarebausteins) erzeugt. Die Verbindungen zwischen den Softwarebausteinen werden ebenfalls automatisch aus dem Softwareentwurf entnommen. Die Verknüpfung zu den Modellen des technischen Systems und der Bedieneingriffe erfolgt mit MODAS.

7.5.3 Modellanalyse

Die Ergebnisdarstellung des qualitativen Modells hat Tabellencharakter. Zur Auswertung von Systemsituationen wurde Microsoft Excel mit Hilfe eines PlugIn[8] erweitert, um folgende Punkte zu automatisieren:

- *Einlesen von Systemsituationen*
  Die Ergebnisdarstellung wird in Excel geöffnet, automatisch formatiert und der Autofilter zur Analyse aktiviert. Der Anwender kann sofort mit der Auswertung der Systemsituationen beginnen.

- *Erstellen statistischer Diagramme*
  Mit der Auswertung der Ergebnisdarstellung in Form von statistischen Diagrammen erhält der Anwender zu Beginn der Sicherheitsanalyse einen schnellen Überblick. Das Diagramm zeigt die Anzahl von gefährlichen, unerwünschten und bestimmungsgemäßen Systemsituationen in Form von Balken an.

- *Filtermethoden*
  Systemsituationen lassen sich nach ihren Attributen (G, U, B), d.h. gemäß ihrer Bedeutung für die Sicherheit des Prozessautomatisierungssystems sortieren. Innerhalb der Einteilung in gefährliche und unerwünschte Systemsituationen erfolgt eine weitere Sortierung nach Anzahl der enthaltenen Fehler oder Gefahren. Darüber hinaus lassen sich Systemsituationen ebenfalls nach den qualitativen Kommentaren einzelner Systemkomponenten ordnen.

- *Einschränkung der Sichtweise auf die Ergebnisdarstellung*
  Systemsituationen, die einer bestimmten Vorgabe nicht entsprechen, können ausgeblendet werden. Dies führt zu einer Einschränkung der Ergebnisdarstellung und stellt ein wichtiges Hilfsmittel zur Prüfung von Sicherheitsanforderungen dar. Der Anwender kann bestimmte Situationen der Ergebnisdarstellung gemäß seinen Ansprüchen ausblenden. Dabei lässt sich das Verhalten des Prozessautomatisierungssystems unter konkreten Randbedingungen prüfen und Fragen folgender Art beantworten. Ist es möglich, dass alle Ventile geöffnet sind? Wie verhält sich das System, wenn z. B. alle Ventile geöffnet sind? Welche Situationen können die Systemelemente X und Y einnehmen, falls Systemelement Z den Fehler A aufweist?

- *Erweiterung der Sichtweise auf die Ergebnisdarstellung*
  Für detailliertere Untersuchungen können zusätzlich die einer Systemsituation zugrunde liegenden Modellgrößen mit ihren entsprechenden Werten in der Ergebnisdarstellung eingeblendet werden.

---

[8] Ein PlugIn enthält eine Ansammlung von Makros (realisiert in VisualBasic for Applikation).

Zusammenfassend für Kapitel 7 kann festgehalten werden, dass bei der Durchführung der modellbasierten Sicherheitsanalyse das qualitative Gesamtmodell des Prozessautomatisierungssystems schrittweise ausgewertet wird. Im ersten Schritt werden sicherheitskritische Systemsituationen untersucht, die bei bestimmungsgemäßem Verhalten des technischen Systems und korrekten menschlichen Bedieneingriffen auftreten. Es wird angenommen, dass in diesem Fall vorhandene sicherheitskritische Systemsituationen auf Fehler in der Automatisierungssoftware zurückzuführen sind. Anschließend erfolgt die Analyse hypothetischer Fehler, um herauszufinden, unter welchen Umständen sicherheitskritische Systemsituationen auftreten können. Die Überprüfung vorhandener Sicherheitsanforderungen erfolgt durch einen Vergleich der Aussage des Modells mit der Aussage der Sicherheitsanforderung. Widersprüchliche Aussagen deuten darauf hin, dass entsprechende Sicherheitsanforderungen nicht erfüllt sind. Bei sicherheitskritischen Systemsituationen ist anhand des Risikos zu entscheiden, ob Sicherheitsmaßnahmen notwendig sind oder nicht. Da es sich nur um eine subjektive Beurteilung handeln kann, muss der Anwender seine Entscheidung begründen.

8 Anwendung der ganzheitlichen modellbasierten Sicherheitsanalyse am Beispiel einer Waschschleudermaschine

Die Anwendung der modellbasierten Sicherheitsanalyse wird in diesem Kapitel am Beispiel einer Produktautomatisierung veranschaulicht. Dazu dient das Prozessautomatisierungssystem „Waschschleudermaschine“. Zuerst werden das technische System, die Automatisierungssoftware und die menschlichen Bedieneingriffe beschrieben. Anschließend erfolgt die Erstellung des qualitativen Modells der Systembestandteile und deren Kombination zu einem Gesamtmodell. Abschließend wird die Durchführung der modellbasierten Sicherheitsanalyse dargestellt.

8.1 Prozessautomatisierungssystem Waschschleudermaschine

8.1.1 Technisches System der Waschschleudermaschine

Die Waschschleudermaschine ist eine kommerzielle Hochleistungs-Waschmaschine, wie sie in Wäschereien, Krankenhäusern und Heimen zu finden ist. Die Waschmaschine (siehe Abbildung 8.1) besitzt die Typenbezeichnung FEX 16 und wird von der Firma Seibt + Kapp hergestellt. Das technische System ist durch einen kontinuierlichen Waschprozess geprägt. Die Trommel hat ein Fassungsvermögen von 16 kg und ein Volumen von 164 Liter. Das Wasser wird durch eine Elektroheizung in der Trommel erhitzt. Ein Drehstromasynchronmotor ist für die Wasch- und Schleuderbewegung der Trommel vorgesehen. Die Trommel ist direkt mit einem Wasseranschluss verbunden. Es besteht die Möglichkeit, das Abwasser über ein Zwei-Wege-Ventil in die Kanalisation oder in einen eingebauten Rückgewinnungstank zu pumpen. Der Rückgewinnungstank umfasst 112 Liter und dient zur Verringerung des Wasser-, Waschmittel- und Energieverbrauchs. Das Wasser des Tanks hat eine Temperatur von ca. 35-40°C und kann für das Vorwaschen und Klarwaschen erneut eingesetzt werden. Die Beladung der Waschtrommel erfolgt über eine elektronisch verriegelbare Tür.

Abbildung 8.1: Waschschleudermaschine

Beim Betreiben des technischen Systems müssen folgende gefährliche Betriebssituationen oder Ereignisse vermieden werden:

S1. Heizen bei leerer Trommel

S2. Heizen über Kochtemperatur der Lauge (Dampfentwicklung)

S3. Heizen bei offener Trommeltür

S4. Abpumpen der Lauge mit sehr hoher Temperatur

S5. Öffnen der Trommel bei hohem Wasserstand oder drehender Trommel

S6. Schleudern mit großer Unwucht bzw. bei gefüllter Trommel

8.1.2 Automatisierungssoftware der Waschschleudermaschine

Die Automatisierungssoftware ermöglicht eine vollautomatische Prozessführung der Waschschleudermaschine.

Wie aus Abbildung 8.2 ersichtlich, basiert der Entwurf der Automatisierungssoftware auf vier Softwarebausteinen. Die Komponente „*Wasserhaushalt*“ ist verantwortlich für alle Steuerungsabläufe, die mit dem Zu- und Ablauf von Frischwasser zusammenhängen. Zusätzlich übernimmt die Komponente die Verriegelung der Trommeltür. Die Wassertemperatur wird von der Komponente „*Heizung*“ geregelt. Die Kapsel „*Waschsteuerung“* steuert den gesamten Waschprozess und bildet somit den Kern der Automatisierung. Sie greift hierzu auf ein individuell vom Anwender gestaltetes Waschprogramm zurück, welches durch eine gleichnamige Kapsel verwaltet und geprüft wird. Ein Waschprogramm besteht aus mehreren Anweisungsschritten, die der

Betreiber der Waschschleudermaschine frei programmieren kann. Die Kapsel „*Waschsteuerung*“ interpretiert die Anweisungen und führt die dazu notwendigen Steuerungsaufgaben aus.

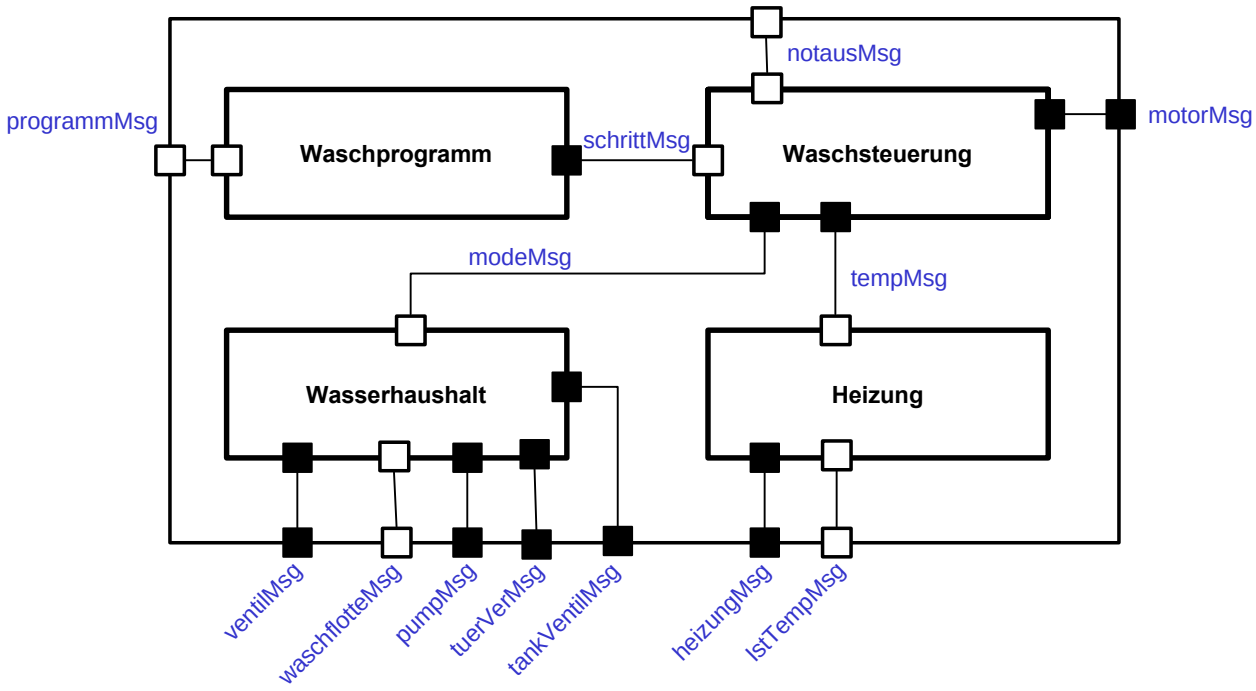


Abbildung 8.2: Strukturdiagramm der Automatisierungssoftware

Beispielhaft für den Entwurf einer Komponente wird die Komponente „*Heizung*“ im Folgenden detaillierter vorgestellt, siehe Abbildung 8.3. Sie besitzt insgesamt drei verschiedene Ports („port1“ bis „port3“), deren Zuordnung zu den jeweiligen Protokollen ebenfalls aus Abbildung 8.2 hervor. So besitzt z. B. das Protokoll „*heizungMsg*“ ein ausgehendes Signal „*heizen*“ vom Typ „*boolean*“.

Damit sind alle Schnittstellen, die zur Kommunikation zur Verfügung stehen, beschrieben. Als nächstes wird dargestellt, nach welchem Muster die Kommunikation stattfindet bzw. wie sich die Kapsel verhalten soll. Die Verhaltensbeschreibung erfolgt, wie in Kapitel 6.3.2.1 erläutert, anhand erweiterter Zustandsdiagramme, siehe Abbildung 8.4.

Die Kapsel „*Heizung*“ besitzt zwei verschiedene Zustände. Der Zustand „*Ruhe*“ wird bei der ersten Initialisierung der Kapsel aufgerufen. Die Kapsel befindet sich auch im Zustand „*Ruhe*“, wenn der Wert der Informationsgröße „istTemp“ größer oder gleich dem Wert von „tempVorgabe“ ist. In Anhängigkeit der Eingangsgrößen „tempVorgabe“ und „istTemp“ wechselt die Kapsel vom Zustand „*Ruhe*“ in den Zustand „*Heizen*“.

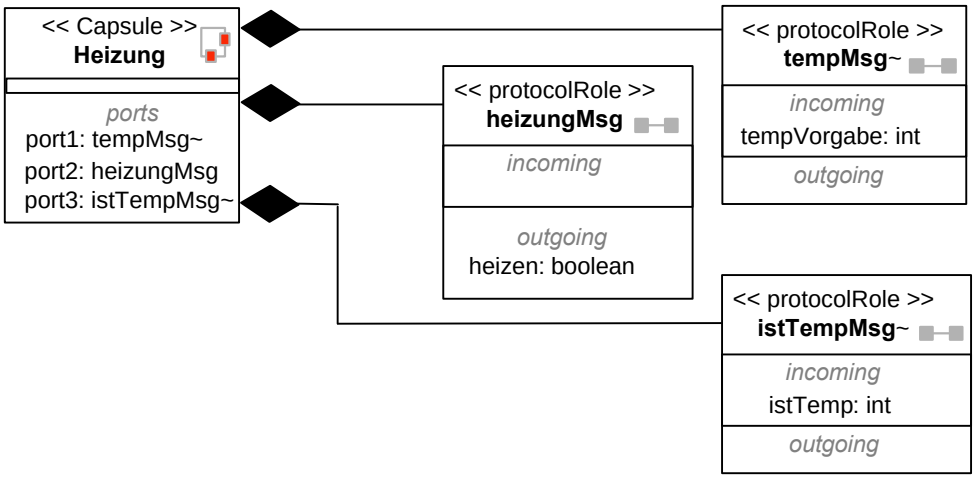


Abbildung 8.3: Spezifikation der Kapsel *Heizung*

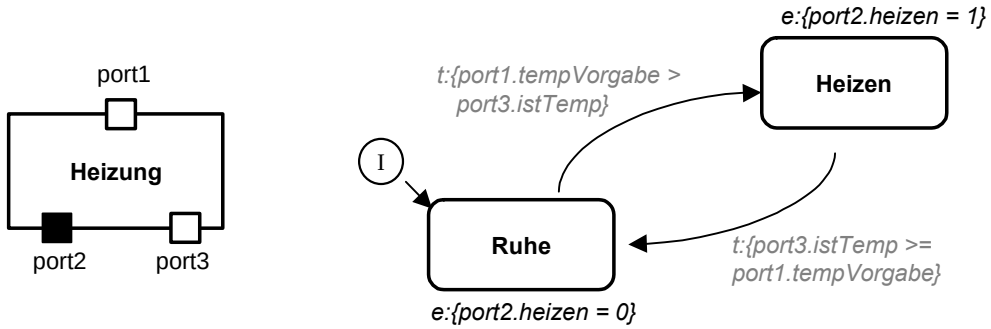


Abbildung 8.4: Zustandsdiagramm der Kapsel *Heizung*.

8.1.3 Menschliche Bedieneingriffe an der Waschschleudermaschine

Der Benutzer der Waschschleudermaschine hat folgende Routineaufgaben zu erledigen:

- Wahl des Waschprogramms
- Beladen und Entladen der Waschtrommel mit Wäsche

Darüber hinaus hat er die Möglichkeit, bestehende Waschprogramme gemäß seinen Bedürfnissen anzupassen. Ein Waschprogramm setzt sich aus mehreren Einzelschritten zusammen, die entsprechend ihrer Reihenfolge ausgeführt werden [Seib99].

Nicht alltägliche Bedieneingriffe sind hingegen

- die Betätigung des Notausschalters und
- Vorgabe einzelner Waschschritte zu Diagnosezwecken.

8.2 Qualitative Modellierung des Prozessautomatisierungs-systems „Waschschleudermaschine“

8.2.1 Qualitative Modellierung des technischen Systems

Das qualitative Modell des technischen Systems umfasst fünf technische Bauelemente. Der Rückgewinnungstank ist das einzige passive technische Bauelement. Die Waschtrommel stellt den Kern des Prozessgeschehens dar und besitzt daher die meisten Aktoren und Sensoren. Der kontinuierliche Waschprozess manipuliert die Medien „Wasser“ und „Wäsche“.

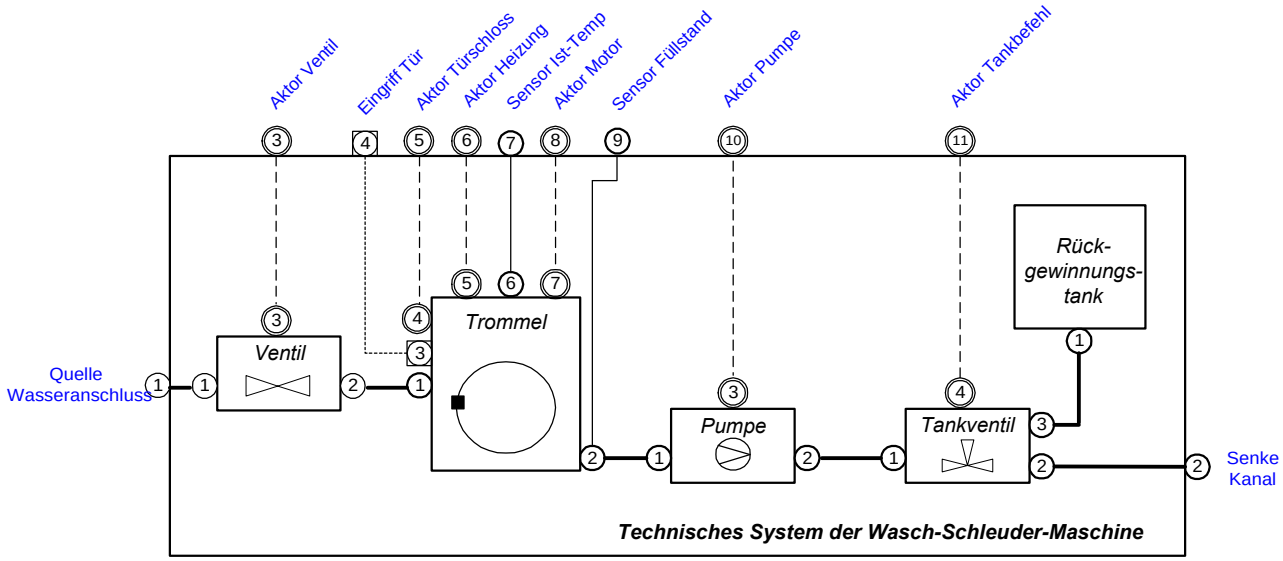


Abbildung 8.5: Übersicht über das qualitative Modell des technischen Systems der Wasch-schleudermaschine.

Auf Systemelementebene sind die in Tabelle 8.1 aufgeführten gefährlichen Vorgänge beschrieben. Insgesamt enthält das qualitative Modell des technischen Systems 43.944 verschiedene Situationen, welche das unkontrollierte Verhalten beschreiben. Die Anzahl der Situationen ergeben sich aus der Anzahl der Intervallvariablen und der Anzahl der vorhandenen Wertebereiche (Intervalle). Diese Situationen unterscheiden sich in den unterschiedlichen Wertebereichen der technischen Prozessgrößen. Werden die Situationen gemäß ihrer Kommentare zusammengefasst, so beschreiben 1.872 Situationen verschiedene Betriebsszenarien das technische System und damit den technischen Prozess. Von diesen 1.872 Situationen sind 1.338 gefährliche, 355 unerwünschte und nur 179 bestimmungsgemäße Betriebsszenarien.

Tabelle 8.1: Berücksichtigte gefährliche Vorgänge der Waschschleudermaschine

Technisches Bauelement	Ausgehende Gefährdung	betroffene Größen
Trommel	Wasseraustritt aus Tür	Füllstand, Türposition
	Rotation bei offener Tür	Stellgröße Motor, Türposition
	Überhitzung der Heizstäbe	Wasserstand, Stellgröße Heizung
	Laugengasentwicklung	Wassertemperatur
Tank	Überlauf	Wasserstand, Zulauf

8.2.2 Qualitative Modellierung der Automatisierungssoftware

Der Entwurf der Automatisierungssoftware wurde gemäß des im Kapitel 6.3.2.2 vorgestellten Konzepts auf ein entsprechendes qualitatives Modell abgebildet. Jede Kapsel findet sich in einem modellierten Softwarebaustein wieder. Für die Komponente „Heizung“ ist in Abbildung 8.6 die äquivalente qualitative Beschreibung dargestellt.

Die Struktur des Automatisierungssoftwaremodells entspricht dem vorliegenden Strukturdiagramm in Abbildung 8.2. Es enthält 71 verschiedene Situationen, die das potenzielle Verhalten wiedergeben. Werden diese 71 Situationen entsprechend den im Entwurf spezifizierten Zuständen zusammengefasst, so ergeben sich 11 verschiedene Betriebsszenarien für die Automatisierungssoftware, siehe Tabelle 8.2. Aus der Tabelle geht weiterhin hervor, wie viele Situation den jeweiligen Betriebsszenarien zugeordnet sind.

Tabelle 8.2: Betriebsszenarien für die Automatisierungssoftware

Waschprogramm	Waschsteuerung	Wasserhaushalt	Heizung	Anzahl der Situationen
RUHE	RUHE	RUHE	RUHE	4
RUHE	NOTAUS	RUHE	RUHE	4
EINGABE_KORREKT	WASCHSCHRITT_AKTIV	FUELLEN	RUHE	8
EINGABE_UNZUL	RUHE	RUHE	RUHE	12
EINGABE_UNZUL	NOTAUS	RUHE	RUHE	12
EINGABE_KORREKT	RUHE	RUHE	RUHE	6
EINGABE_KORREKT	NOTAUS	RUHE	RUHE	6
EINGABE_KORREKT	WASCHSCHRITT_AKTIV	RUHE	HEIZEN	3
EINGABE_KORREKT	WASCHSCHRITT_AKTIV	RUHE	RUHE	8
EINGABE_KORREKT	WASCHSCHRITT_AKTIV	InTANK	RUHE	4
EINGABE_KORREKT	WASCHSCHRITT_AKTIV	InKANAL	RUHE	4

```
Component sw_Heizung

Terminals
            #1 type SW id port1;
            #2 type SW id port2;
            #3 type SW id port3;

Quantities
            id tempVorgabe_In from 1 type potential
            intervals [0,0] (0,?) ;

            id heizen_Out from 2 type potential
            intervals [0,0] [1,1];

            id istTemp_In from 3 type potential
            intervals [0,0] (0,?);

SituationRules
            //Zustand RUHE
            tempVorgabe_In <= istTemp_In -> heizen_Out = 0;

            //Zustand HEIZEN
            istTemp_In < tempVorgabe_In -> heizen_Out = 1;

CommentRules
            //Zustand RUHE
            tempVorgabe_In <= istTemp_In && heizen_Out = 0 => RUHE;

            //Zustand HEIZEN
            istTemp_In < tempVorgabe_In && heizen_Out = 1 => HEIZEN;

TransitionRules
            RUHE -> HEIZEN;
            HEIZEN -> RUHE;
```

Abbildung 8.6: Qualitative Beschreibung der Kapsel *Heizung*

8.2.3 Qualitative Modellierung der menschlichen Bedieneingriffe

Im Rahmen der Sicherheitsanalyse der Waschschleudermaschine werden insbesondere Eingriffe des Menschen untersucht, die nicht zu den alltäglichen Routineaufgaben zählen. Wie in Abschnitt 8.1.3 geschildert, ist es zu Diagnose- und Testzwecke möglich, dass der Mensch einzelne Waschschritte direkt vorgibt. Diese Szenarien werden im qualitativen Modell der menschlichen Bedieneingriffe festgehalten. Das Modell umfasst die Eingriffe „Notaus betätigen“ und „Trommeltür öffnen“ bzw. „Trommeltür schließen“. Darüber hinaus können der Automatisierungssoftware folgende Einzelwaschschritte vorgegeben werden: „füllen“, „waschen“, „schleudern“, „in den Kanal pumpen“ oder „in den Tank pumpen“ und entsprechende Sollwertangaben modifizieren.

Insgesamt beschreiben 33 verschiedene Situationen mögliche Bedieneingriffe des Operators. Diese stellen 19 unterschiedliche Bedienszenarien[9] des Menschen dar, siehe Tabelle 8.3. Die erste Situation steht für „kein Bedieneingriff“. Ein Bedieneingriff mit gleichzeitiger Betätigung des Notausschalters wurde als unerwünscht, also als fehlerhaft deklariert. Es besteht keine Vorgabe an eine bestimmte Bediensequenz. Alle Abfolgen werden bei der Analyse betrachtet.

Tabelle 8.3: Modellierte Bedieneingriffe des Wartungspersonals

	Bedieneingriffe		**Status**
1	-		B
2	tuer_oeffnen		B
3	tuer_schliessen		B
4	notaus		B
5	notaus	tuer_oeffnen	U
6	notaus	tuer_schliessen	U
7	keine_Eingabe	SollwertVorgabe	U
8	fuellen		B
9	fuellen	SollwertVorgabe	B
10	heizen		B
11	heizen	SollwertVorgabe	B
12	waschen		B
13	waschen	SollwertVorgabe	B
14	schleudern		B
15	schleudern	SollwertVorgabe	B
16	leeren_in_Tank		B
17	leeren_in_Tank	SollwertVorgabe	B
18	leeren_in_Kanal		B
19	leeren_in_Kanal	SollwertVorgabe	B

8.2.4 Komposition der Teilmodelle

Die Verbindungen der aktiven technischen Bauelemente zum Rechnersystem bestimmen die möglichen Kommunikationswege zwischen Automatisierungssoftware und technischem System der Waschschleudermaschine, siehe Abbildung 8.8. Die vorgesehenen menschlichen Bedieneingriffe sind in dieser Abbildung durch (gestrichelte) Verbindungen dargestellt. Durch Komposition der Teilmodelle ergeben sich 3.964 verschiedene Systemsituationen, die sich zu 110 unterschiedlich kommentierten Szenarien zusammenfassen lassen. In Abbildung 8.7 ist eine statistische Auswertung der Teilmodelle sowie des Gesamtmodells zu sehen. Aus dem Diagramm geht hervor, dass das Gesamtmodell sechs sicherheitskritische Systemsituationen enthält. Ebenfalls ist zu sehen, dass die Anzahl an gefährlichen Situationen des technischen Systems im Ge-

[9] Für jede „SollwertVorgabe“ existieren 3 verschiedene Situationen.

samtmodell erheblich reduziert ist. Da es sich um eine vollautomatisierte Prozessführung handelt, vermeidet die Automatisierungssoftware den größten Teil der gefährlichen Situationen des technischen Systems.

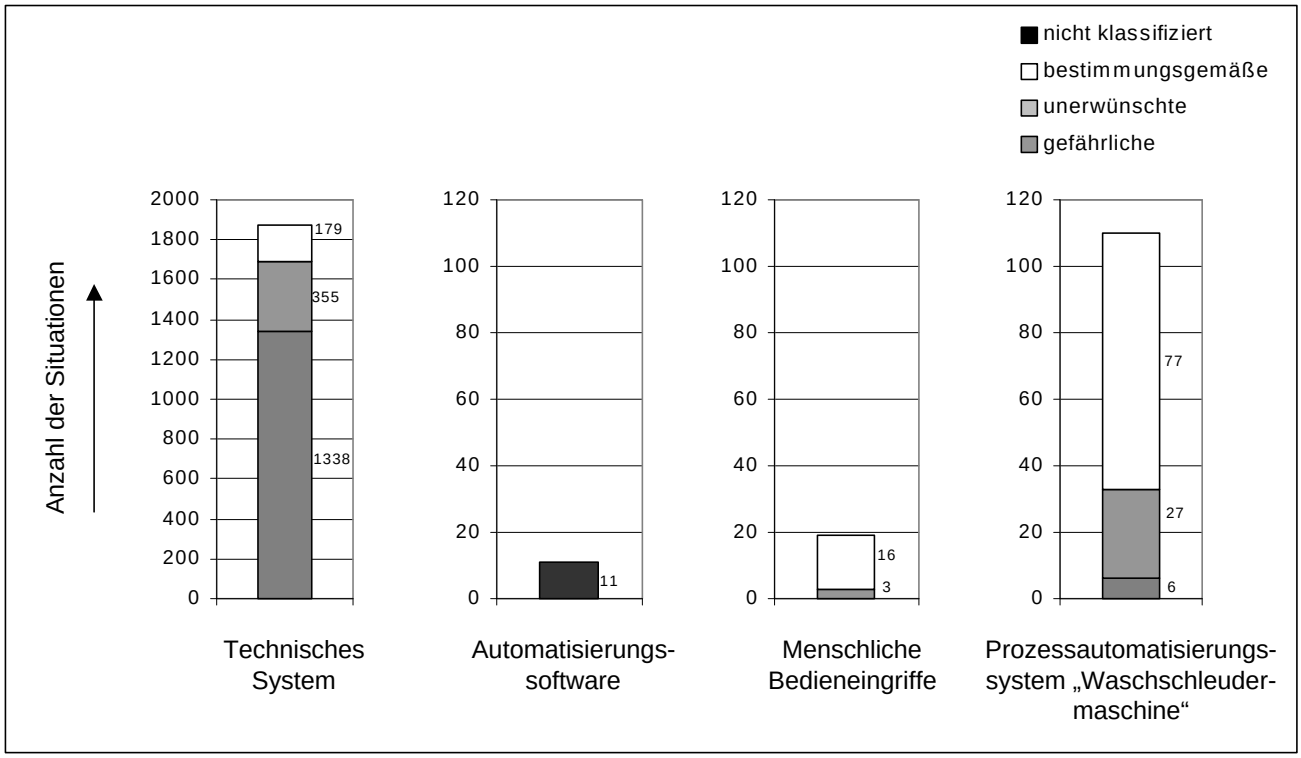


Abbildung 8.7: Statistische Betrachtung der Betriebsszenarien

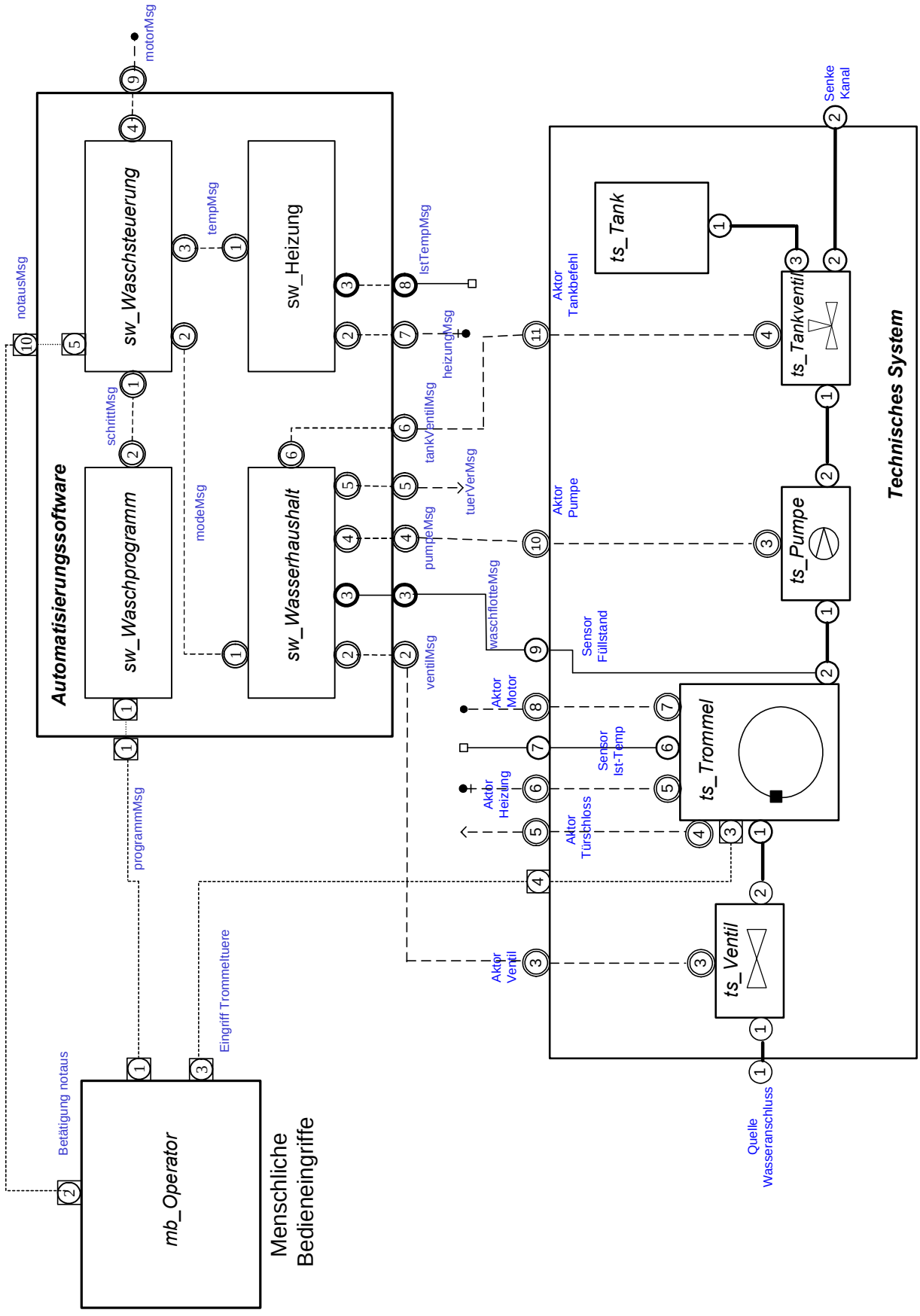


Abbildung 8.8: Qualitatives Gesamtmodell der Waschschleudermaschine (Strukturansicht)

8.3 Sicherheitsanalyse der Waschschleudermaschine

8.3.1 Interpretation von Systemsituationen der Waschschleudermaschine

In Tabelle 8.4 sind stellvertretend drei Betriebsszenarien aus den insgesamt 119 herausgegriffen worden. Aus Gründen der hier begrenzten Darstellungsmöglichkeiten sind die Situationen der einzelnen Systembestandteile nicht in einer Zeile angeordnet, sondern untereinander.

Tabelle 8.4: Drei bestimmungsgemäße Systemsituationen der Waschschleudermaschine

	Status	Menschliche Bedieneingriffe: mb_Operator		Automatisierungssoftware: sw_Waschprogramm	sw_Waschsteuerung	sw_Wasserhaushalt	sw_Heizung
23	B	heizen	SollwertVorgabe	EINGABE_KORREKT	RUHE	RUHE	RUHE
80	B	heizen	SollwertVorgabe	EINGABE_UNZUL	RUHE	RUHE	RUHE
92	B	heizen	SollwertVorgabe	EINGABE_KORREKT	WASCHSCHRITT_AKTIV	RUHE	HEIZEN

Technisches System

	ts_Ventil	ts_Trommel					ts_Pumpe	ts_Tankventil	ts_Tank	
23	zu	leer	tuer_geschlossen	heizung_aus	kalt	motor_aus	aus	in_kanal	gefuellt	kein_ablauf
80	zu	gefuellt	tuer_geschlossen	heizung_aus	kalt	motor_aus	aus	in_kanal	gefuellt	kein_ablauf
92	zu	gefuellt	tuer_geschlossen	heizung_ein	kalt	motor_aus	aus	in_kanal	gefuellt	kein_ablauf

Die bestimmungsgemäßen Systemsituationen (Status „B“) mit den Nummern 23, 80 und 92 stellen unterschiedliche Szenarien für das Heizen der Waschflotte dar. Obwohl der Mensch „heizen“ als Befehl vorgibt, findet nur in Systemsituation 92 tatsächlich ein Heizen des Flottenwassers statt.

In Systemsituation 23 wird der menschliche Bedieneingriff „heizen“ und „SollwertVorgabe“ des Softwarebausteins „Waschprogramm“ als „korrekt“ bewertet. Die Waschsteuerung aktiviert allerdings korrekterweise den Softwarebaustein „Heizung“ nicht, da sich kein Wasser in der Waschtrommel befindet.

Systemsituation 80 zeigt, wie die Automatisierungssoftware auf einen unzulässigen Bedieneingriff reagiert. Der Softwarebaustein „Heizung“ wird aufgrund einer unzulässigen Sollwertvorgabe nicht aktiviert. Die Softwaregrößen „eingaben“, „eingabewerte“ und „w_schritt“ der Komponente „sw_Waschprogramm“ besitzen in dieser Systemsituation die in Tabelle 8.5 aufgeführten Werte. Aus dieser zusätzlichen Information geht hervor, dass die Sollwertvorgabe („eingabewerte“) einen Wert größer als 90°C besitzt.

Tabelle 8.5: Information über Werte von Softwaregröße

	sw_Waschprogramm	
80	EINGABE_UNZUL	
	eingaben	=[2,2]
	eingabewerte	$=(90,\infty)$
	w_schritt	=[0,0]

8.3.2 Auswertung von Systemsituationen

Untersuchung der Automatisierungssoftware

Das Gesamtmodell enthält sechs kritische Systemsituationen, die trotz bestimmungsgemäßem Betrieb des technischen Systems und korrekten menschlichen Bedieneingriffen auftreten können, siehe Tabelle 8.6. Ist in der Waschtrommel kein Wasser enthalten, so kann die Tür der Trommel offen stehen. Diese Tatsache wurde im Entwurf der Automatisierungssoftware nicht berücksichtigt. Außerdem ist es möglich, dass der Operator die Einzelschrittvorgaben „waschen“ und „schleudern“ vorgeben kann. Der Asynchronmotor wird gestartet, obwohl die Trommeltür geöffnet ist. Es besteht also Verletzungsgefahr. Die sicherheitskritischen Systemsituationen unterscheiden sich lediglich darin, dass der Rückgewinnungstank unterschiedlich gefüllt sein kann. Diese identifizierten Systemsituationen sind aufgrund des Entwurfs der Automatisierungssoftware im realen Betrieb der Waschschleudermaschine möglich und müssen verhindert werden.

Tabelle 8.6: Sicherheitskritische Systemsituationen der Waschschleudermaschine

Menschliche Bedieneingriffe

Status			mb_Operator
53	G	53	waschen
54	G	54	waschen
55	G	55	waschen
56	G	56	schleudern
57	G	57	schleudern
58	G	58	schleudern

Automatisierungssoftware

	sw_Waschprogramm	sw_Waschsteuerung	sw_Wasserhaushalt	sw_Heizung
53	EINGABE_KORREKT	WASCHSCHRITT_AKTIV	RUHE	RUHE
54	EINGABE_KORREKT	WASCHSCHRITT_AKTIV	RUHE	RUHE
55	EINGABE_KORREKT	WASCHSCHRITT_AKTIV	RUHE	RUHE
56	EINGABE_KORREKT	WASCHSCHRITT_AKTIV	RUHE	RUHE
57	EINGABE_KORREKT	WASCHSCHRITT_AKTIV	RUHE	RUHE
58	EINGABE_KORREKT	WASCHSCHRITT_AKTIV	RUHE	RUHE

Technisches System

	ts_Ventil	ts_Trommel						ts_Pumpe	ts_Tankventil	ts_Tank	
53	zu	leer	tuer_geoeffnet	heizung_aus	kalt	motor_waschen	*verletzungsgefahr*	aus	in_kanal	gefuellt	kein_ablauf
54	zu	leer	tuer_geoeffnet	heizung_aus	kalt	motor_waschen	*verletzungsgefahr*	aus	in_kanal	voll	kein_ablauf
55	zu	leer	tuer_geoeffnet	heizung_aus	kalt	motor_waschen	*verletzungsgefahr*	aus	in_kanal	leer	kein_ablauf
56	zu	leer	tuer_geoeffnet	heizung_aus	kalt	motor_schleudern	*verletzungsgefahr*	aus	in_kanal	gefuellt	kein_ablauf
57	zu	leer	tuer_geoeffnet	heizung_aus	kalt	motor_schleudern	*verletzungsgefahr*	aus	in_kanal	voll	kein_ablauf
58	zu	leer	tuer_geoeffnet	heizung_aus	kalt	motor_schleudern	*verletzungsgefahr*	aus	in_kanal	leer	kein_ablauf

Die Ursache für das Vorhandensein dieser sicherheitskritischen Systemsituationen liegt in einem Entwurfsfehler der Automatisierungssoftware, da das Einschalten des Motors bei geöffneter Tür nicht verhindert wird bzw. die Türverriegelung bei laufendem Trommelmotor nicht aktiv ist.

Sicherheitsanforderungen

Die Sicherheitsanforderung S1 bis S6 aus Abschnitt 8.1.1 stellen Prüfkriterien zur Analyse der Systemsituationen dar. Wie in Kapitel 7.3 geschildert, wird die Aussage der Sicherheitsanforderungen negiert und entsprechende Filtervorgaben abgeleitet, siehe Tabelle 8.7. Ein Widerspruch zur Sicherheitsanforderung liegt vor, falls zu den jeweiligen Vorgaben Systemsituationen im qualitativen Modell existieren. Beispielsweise lautet das Filterkriterium zur Überprüfung der Sicherheitsanforderung S5 „Tür geöffnet bei drehender Trommel“. Es existieren sechs gefährliche Systemsituationen, die schon im vorangegangenen Abschnitt diskutiert wurden. Diese Sicher-

heitsanforderungen werden nicht erfüllt. Für das Filterkriterium „Schleudern und Trommel gefüllt“ wurden drei Betriebsszenarien gefunden. Dies widerspricht der Sicherheitsanforderung S6.

Tabelle 8.7: Übersicht über die geprüften Sicherheitsanforderungen

Prüfkriterien	**Filterkriterien für technisches System**	**Anzahl gefundener Systemsituationen**
S1: Heizen bei leerer Trommel	Heizung eingeschaltet und Trommel leer	0
S2: Heizen über Kochtemperatur	trommel.temp[10] = (90, ?)	0
S3: Heizen bei geöffneter Trommeltür	Trommeltür geöffnet und Heizung an	0
S4: Abpumpen der Spülflotte mit hoher Temperatur	Pumpe eingeschaltet und trommel.temp > 60	0
S5: Geöffnete Trommeltür bei		
1. Wasserstand	1. Trommeltür geöffnet und Trommel gefüllt	0
2. Drehender Trommel	2. Trommeltür geöffnet und Motor ein (waschen und schleudern)	6
S6: Schleudern mit gefüllter Trommel	Motor schleudert und Trommel gefüllt	3

Untersuchung von Einzelfehlern

Während bisher der bestimmungsgemäße Betrieb der Waschschleudermaschine betrachtet wurde, stellt sich nun die Frage, wie sich das gesamte Prozessautomatisierungssystem verhält, wenn einzelne Fehler auftreten.

Mit Hilfe aktiver Bauelemente, d.h. Aktoren und Sensoren, wird das Prozessgeschehen manipuliert. Fehler führen hierbei zu ungewollten oder sicherheitskritischen Betriebsszenarien. Als Fehler werden nicht-bestimmungsgemäße Verhaltensweisen der aktiven Bauelemente angenommen. Dazu wird in der Modellstruktur die jeweilige Verbindung der aktiven Bauelemente zur Automatisierungssoftware unterbrochen, siehe Kapitel 7.2.4. Die Ergebnisse der Analyse ergaben sich direkt aus dem qualitativen Gesamtmodell der Waschschleudermaschine und sind in Tabelle 8.8 zusammengefasst.

[10] Trommel.temp bezeichnet die gleichnamige physikalische Größe, die im qualitativen Modell der Trommel die Temperatur der Waschflotte darstellt.

Tabelle 8.8: Analyse von Einzelfehlern der Waschschleudermaschine

Nichtbestimmungsgemäßes Verhalten von	**Auswirkung**
Bauelement Ventil	*keine sicherheitskritischen Situationen*
Bauelement Pumpe	*keine sicherheitskritischen Situationen*
Bauelement Tankventil	*keine sicherheitskritischen Situationen*
Trommelaktor Türschloss	*sicherheitskritische Situationen:* - Auslauf Waschflotte durch Trommeltür - Verletzungsgefahr bei rotierender Trommel
Trommelsensor Füllstand	*sicherheitskritische Situationen:* - Auslauf Waschflotte durch Trommeltür - Überhitzung der Heizanlage (ohne Flottenwasser)
Trommelsensor Temperatur	*sicherheitskritische Situationen:* - Laugengasentwicklung (Heizen über Kochtemperatur) - Überhitzung der Heizanlage (ohne Flottenwasser)

Untersuchung von Mehrfachfehlern

Die Prüfung der Sicherheitsanforderungen hat ergeben, dass Schleudern bei gefüllter Trommel möglich ist. Hierbei treten zwei Fehler in unterschiedlichen Systembestandteilen gleichzeitig auf (Mehrfachfehler). Der Bediener der Waschschleudermaschine gibt trotz gefüllter Trommel den Einzelwaschschritt „Schleudern“ vor und die Software führt diesen ohne Prüfung aus.

Falls der Trommelaktor „Türschloss“ und der Trommelsensor „Temperatur“ gleichzeitig ausfallen, werden alle Sicherheitsanforderungen S1 bis S6 verletzt.

8.3.3 Sicherheitsmaßnahmen

Die modellbasierte Sicherheitsanalyse der Waschschleudermaschine führt zu folgenden Aussagen.

1. Das Systembestandteil „Automatisierungssoftware“ enthält zwei Fehler:
   Zum einen kann bei leerer Waschtrommel die Trommeltür geöffnet werden, obwohl diese sich in Rotation befindet. Zum anderen ist bei gefüllter Trommel die Vorgabe des Waschschritts „schleudern“ möglich.

2. Beim Ausfall des Füllstandssensors der Waschtrommel ist ein Wasseraustritt durch die Trommeltür und ein Überhitzen der Heizanlage möglich. Fällt die Türverriegelung der Trommeltür aus, dann können schwerwiegende Unfälle enstehen. Die Trommeltür kann jederzeit geöffnet werden. Bei Ausfall des Temperatursensors kann es zu einer gefährlichen Laugengasentwicklung kommen.

Bei der Definition der Sicherheitsmaßnahmen spielt das akzeptierbare Risiko beim Betrieb der Waschschleudermaschine die ausschlaggebende Rolle. In Kapitel 7.4 wurde beschrieben, dass

bei einer Beurteilung von gefährlichen Situationen, die Eintretenswahrscheinlichkeit und das Schadensausmaß zu beachten sind.

Das Modell der menschlichen Bedieneingriffe wurde für den seltenen Fall der Wartung oder Diagnose der Waschschleudermaschine aufgestellt. Es kann davon ausgegangen werden, dass qualifiziertes Fachpersonal, welches die Wartungsarbeiten durchführt, kaum falsche Bedieneingriffe vornehmen wird. Die Eintrittswahrscheinlichkeit von Bedienfehlern ist als niedrig einzustufen. Das Schadenausmaß ist für identifizierte sicherheitskritische Systemsituationen als hoch anzusehen, da das Bedienpersonal ernsthaft verletzt werden kann. Das Risiko beim Wartungsbetrieb der Waschschleudermaschine überschreitet das zulässige Grenzrisiko[11].

Die identifizierten Fehler der Automatisierungssoftware lassen sich einfach beheben und damit die ermittelten gefährlichen Systemsituationen vermeiden. Anstatt die Verriegelung der Trommeltür mit der Kapsel „*Wasserhaushalt*“ zu realisieren, wird die Türverriegelung von der zentralen Kapsel „*Waschsteuerung*“ übernommen, siehe Abbildung 8.9. Das Zustandsdiagramm der Kapsel „Waschsteuerung“ muss dahingehend ergänzt werden, dass die Waschtrommeltür zusätzlich verriegelt wird, sobald der Motor der Waschtrommel aktiv ist. Eine zusätzliche Überprüfung des Waschflottenstands verhindert ein Schleudern bei gefüllter Trommel. Der Entwurf der Automatisierungssoftware wurde dahingehend korrigiert.

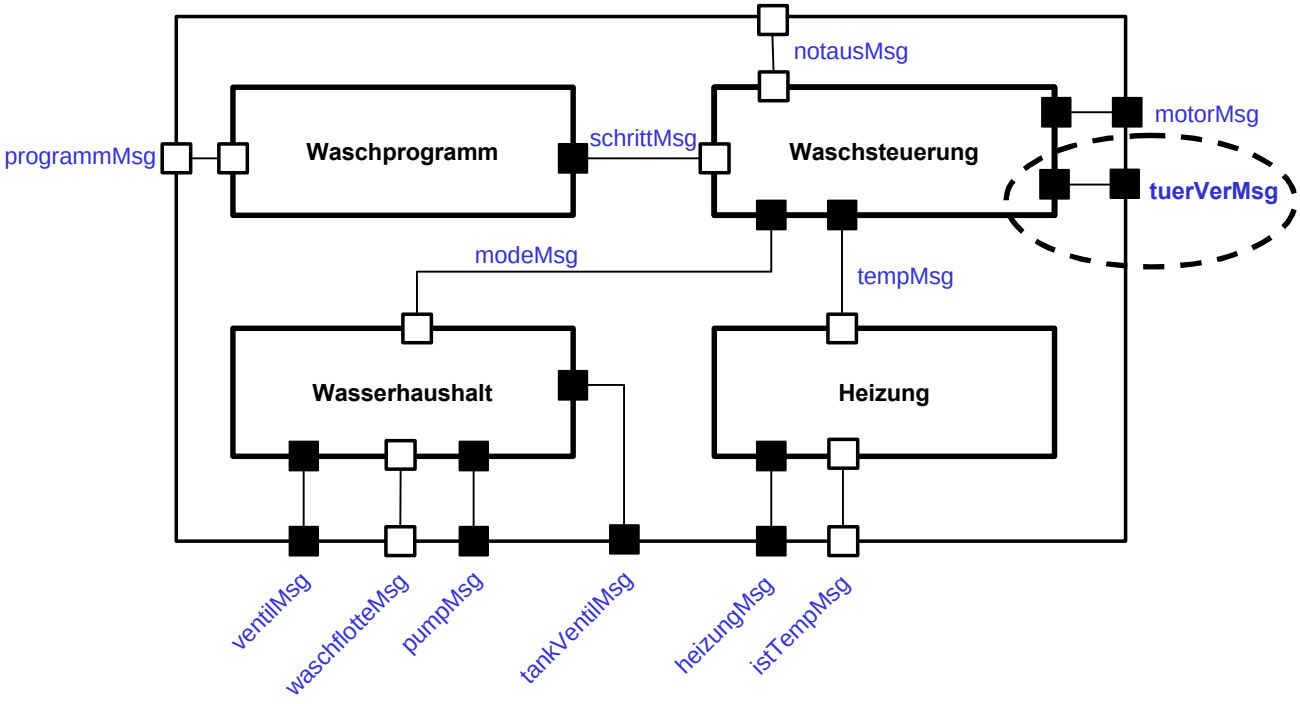


Abbildung 8.9: Korrigiertes Strukturdiagramm der Automatisierungssoftware

[11] Die Beurteilung des Grenzrisikos basiert auf der subjektiven Meinung des Anwenders der Sicherheitsanalyse, in diesem Fall auf der des Autors.

Bei erneuter Erstellung des Gesamtmodells der Waschschleudermaschine (unter Berücksichtigung der korrigierten Automatisierungssoftware) waren keine sicherheitskritischen Systemsituationen für den bestimmungsgemäßen Betrieb zu ermitteln.

Die zweite Aussage der modellbasierten Sicherheitsanalyse deckte durch die Annahme von Fehlern Schwachstellen hinsichtlich der Systemsicherheit auf. Um eine erhöhte Systemsicherheit der Waschschleudermaschine zu realisieren, können z. B. wichtige aktive Bauelemente, wie Temperatursensor oder Füllstandssensor redundant ausgeführt werden. Hierbei ist sicherlich das Risiko aus wirtschaftlichen Gesichtpunkten abzuschätzen. Redundante Bauelemente bedeuten gleichzeitig höhere Kosten des Produkts.

Zusammenfassend sei herausgestellt, dass die vorliegenden Ergebnisse nicht wie bei klassischen Verfahren anhand von Brainstorming, sondern anhand des qualitativen Modells rechnergestützt ermittelt wurden. Das Zusammenspiel von technischem System, Automatisierungssoftware und menschlichen Bedieneingriffen wurde systematisch analysiert. Die Ergebnisse der ganzheitlichen Sicherheitsanalyse sind plausibel. Der Aufwand für die Modellierung des Prozessautomatisierungssystem Waschschleudermaschine und die Auswertung betrug zwei Tage.

9 Zusammenfassung und Ausblick

9.1 Zusammenfassung des Konzepts der ganzheitlichen modellbasierten Sicherheitsanalyse

Das entwickelte Konzept einer modellbasierten Sicherheitsanalyse beruht auf einer ganzheitlichen Betrachtungsweise von Prozessautomatisierungssystemen, bei der insbesondere das kausale Zusammenspiel der Systembestandteile technisches System, Automatisierungssoftware und menschliche Bedieneingriffe systematisch analysiert wird.

Das Modellierungsprinzip beruht auf Modellierung der Systemelemente, die durch Modellaggregation gemäß einer vorliegenden Systemstruktur zu einem System verknüpft werden. Die qualitativen Beschreibungsmittel des SQMA-Verfahrens unterstützen dieses Prinzip. Alle Systembestandteile werden in Form von Situationen qualitativ beschrieben:

- Das Modell des technischen Systems enthält Situationen, die das physikalische Verhalten des Prozessautomatisierungssystems beschreiben. Es werden gefährliche, unerwünschte und bestimmungsgemäße Situationen unterschieden.

- Die Situationen des Modells der Automatisierungssoftware stellen das (geplante) Verhalten der Automatisierung dar und sind hinsichtlich nicht-bestimmungsgemäßer Zustände nicht klassifiziert. Bei der Automatisierungssoftware können nur inhärente Fehler vorhanden sein, nicht aber Fehler, die nach Inbetriebnahme entstehen.[12]

- Die Situationen des Modells der menschlichen Bedieneingriffe beschreiben das mögliche menschliche Verhalten, basierend auf Meldungen und Bedieneingriffen. Es wird zwischen unerwünschten und bestimmungsgemäßen Situationen unterschieden.

Die Situationen der Teilmodelle werden untereinander kombiniert und hinsichtlich ihres möglichen Informationsaustauschs zu Systemsituationen des gesamten Prozessautomatisierungssystems verknüpft. Die Systemsituationen stellen das mögliche Verhalten des Prozessautomatisierungssystems im Betrieb dar, unter Berücksichtigung der enthaltenen Gefahren und Fehler. Anhand des Gesamtmodells kann ermittelt werden, ob gefährliche Situationen des technischen Systems auf den Betrieb des Prozessautomatisierungssystems Einfluss haben oder ob diese durch die Konzeption des Systems verhindert werden können. Gefährliche Vorgänge werden in Form von sicherheitskritischen Systemsituationen dargestellt. Der Anwender interpretiert die

[12] Fehler können sich hingegen erst nach der ersten Inbetriebnahme oder im späteren Betrieb auswirken.

Systemsituationen basierend auf deren qualitativen Ausdrücken und Klassifikationen. Bewertet werden sicherheitskritische Systemsituationen aufgrund ihres Risikos. Es hängt von der subjektiven Entscheidung des einzelnen Anwenders ab, ob Sicherheitsmaßnahmen durchzuführen sind oder nicht. In jedem Fall ist die Entscheidung schriftlich zu begründen.

Bestehen Sicherheitsanforderungen für ein Prozessautomatisierungssystem, so können diese mit den Aussagen des Gesamtmodells verglichen werden. Durch den Vergleich kann geprüft werden, ob sich die Aussagen des Modells mit den Aussagen des Gesamtmodells decken. Falls es zu widersprüchlichen Aussagen kommt, so muss davon ausgegangen werden, dass die entsprechenden Sicherheitsanforderungen nicht erfüllt werden.

Die Idee des ganzheitlichen Ansatzes beruht auf der Annahme, dass eine Gefahr einem technischen Bauelement eindeutig zugeordnet werden kann[13]. Bei Prozessautomatisierungssystemen sind Gefahren latent vorhanden. Die Ursachen, die zur Auslösung einer Gefahr führen, können allerdings im gesamten Prozessautomatisierungssystem – d.h. im technischen System, in der Automatisierungssoftware und in menschlichen Bedieneingriffen – lokalisiert sein. In der Regel stellen Fehler oder Kombinationen von Fehlern die Ursachen dar. Bei der entwickelten modellbasierten Sicherheitsanalyse werden die kausalen Zusammenhänge zwischen Ursache und Folge unter Beachtung der Wechselwirkung der Systembestandteile aufgedeckt und analysierbar.

Die Modellerstellung der Teilmodelle und die Durchführung der Sicherheitsanalyse werden durch Software-Werkzeuge unterstützt. MODAS ist eine Modellierungsumgebung, die nicht nur Modellierungsfehlern vorbeugt, sondern den Anwender mit Hilfe interaktiver Dialoge durch die Modellierung führt. Mit Hilfe eines Konverters können Softwareentwürfe, die auf UML-RT basieren, in ein entsprechendes qualitatives Modell überführt werden. Die Auswertung von Systemsituationen erfolgt unter Anwendung verschiedener Filter und Sortierfunktionen. Ein schneller Überblick über das Ergebnis der ganzheitlichen modellbasierten Sicherheitsanalyse wird durch statistische Diagramme ermöglicht.

9.2 Bewertung und Erkenntnisse

Die Stärke des Konzepts für die Sicherheitsanalyse liegt in der Kombination des ganzheitlichen modellbasierten Ansatzes mit dem rechnergestützten Vorgehen:

- Mit Hilfe des ganzheitlichen modellbasierten Ansatzes zur Sicherheitsanalyse ist es möglich, das Zusammenspiel der Systembestandteile eines Prozessautomatisierungssystems umfassend zu analysieren und zu bewerten. Dadurch lässt sich eine systematische Ursache-Folge-Analyse auf Systemebene realisieren. Das Bedienpersonal und die Automatisierungssoft-

[13] Es wird auch von Gefahrenquellen und Gefahrenherden gesprochen.

ware haben neben der Prozessführung die Aufgabe, potenzielle Unfälle zu verhindern. Mit Hilfe der modellbasierten Sicherheitsanalyse lässt sich nicht nur analysieren, inwieweit das vorgesehene Automatisierungskonzept dies berücksichtigt, sondern ebenfalls unter welchen Umständen sich Gefahren dennoch auswirken können. Mit Hilfe des realisierten Ansatzes können bestehende Sicherheitsanforderungen auch im nicht-bestimmungsgemäßen Betrieb eines Prozessautomatisierungssystems überprüft werden.

- Im Vergleich zu klassischen Verfahren der Sicherheitsanalyse ist ausschlaggebend, dass der Rechner und nicht der Anwender selbst mögliche Situationen des Prozessautomatisierungssystems ermittelt. Der Anwender muss diese lediglich bewerten und eventuell Maßnahmen ableiten. Im Gegensatz zu klassischen Verfahren wie PAAG, FMEA oder FTA sind die Ergebnisse der modellbasierten Sicherheitsanalyse nicht nur für Dritte nachvollziehbar, sondern ebenfalls reproduzierbar. Dank der Rechnerunterstützung lassen sich dabei die Auswirkungen von beliebig vielen Fehlern, insbesondere Fehlerkombinationen, berücksichtigen. Formulierte Sicherheitsmaßnahmen können im Gegensatz zu den meisten klassischen Sicherheitsanalysen mit dem entwickelten ganzheitlichen Ansatz unmittelbar überprüft werden. Die rechnergestützte Ermittlung von sicherheitskritischen Systemsituationen eines Prozessautomatisierungssystems setzt voraus, dass Gefahren im technischen Prozess eindeutig einem oder mehreren technischen Bauelementen zugeordnet werden können.

Das Anwendungsbeispiel „Waschschleudermaschine“ hat gezeigt, wie einfach sich die Interpretation von Systemsituationen gestaltet. Es ist nicht notwendig, dass der Anwender den Zusammenhang aller Bestandteile eines Systems kennt, da dieser in Form ermittelter Systemsituationen dargestellt wird. Die qualitativen Ausdrücke des Modells geben eine solide Grundlage für die Beseitigung von Sicherheitslücken bzw. für die Definition von Sicherheitsmaßnahmen.

Der zeitaufwändigste Schritt der modellbasierten Sicherheitsanalyse besteht im Erstellen der notwendigen Teilmodelle. Hierzu hat sich der qualitative Modellierungsansatz nach SQMA bewährt, der auf Intervallvariablen basiert. Die qualitative Modellerstellung ist schnell erlernbar und einfach durchzuführen [Hett99], [Pola98]. Der erhöhte Aufwand für die Modellierung ist allerdings durch die Möglichkeiten der rechnergestützten Sicherheitsanalyse gerechtfertigt. Der Modellierungsaufwand wird durch den Einsatz der entwickelten Softwarewerkzeuge deutlich gesenkt. Der Modellierungsaufwand für das Anwendungsbeispiel „Waschschleudermaschine“ betrug zwei Tage. Kombinatorisch waren $1{,}25 \cdot 10^9$ Systemsituationen möglich, 3964 Systemsituationen erfüllten die Kopplungsbedingungen. Die Ermittlung der Systemsituation des Gesamtsystems betrug 2,2 Sekunden.

Der limitierende Faktor bei der Berechnung der Gesamtsituationen liegt in der Explosion der kombinatorischen Möglichkeiten. Der Algorithmus wurde in [Bent96] optimiert. Je mehr Kopplungsbedingungen vorhanden sind, desto effizienter arbeitet der Kombinationsalgorithmus. Die Kopplungsbedingungen werden nach Anzahl der enthaltenen Modellvariablen sortiert. Erfüllen beispielsweise zwei Modellvariablen eine bestimmte Kopplungsbedingung nicht, dann kann ein ganzer Ast des Kombinationsbaumes entfallen. Allgemeine Angaben zur Leistungsfähigkeit des Algorithmus können nicht gemacht werden, da diese von der speziellen Systemstruktur abhängig sind.

Wie bei allen modellbasierten Ansätzen bestimmt die Güte des Modells in erster Linie die Qualität des Ergebnisses. Das Zusammenspiel zwischen Systemelementen oder Systembestandteilen muss als Struktur bzw. in Form von Verbindungen beschreibbar oder abstrahierbar sein. Dies entspricht im Wesentlichen dem realen Aufbau von Prozessautomatisierungssystemen. Die Bedieneingriffe bzw. Meldungen eines Operators sind durch Verbindungen zum technischen System und zur Automatisierungssoftware abstrahiert. Es lassen sich daher nur „geplante“ bzw. „vorgesehene“ Bedieneingriffe untersuchen.

Aufgrund der konzeptionellen Modellierungsunterstützung von SQMA wirken sich Modellierungsfehler im Ergebnis durch unrealistische Situationen aus und sind somit im Gegensatz zu anderen Modellierungsmethoden (z. B. Petri-Netze oder Zustandsautomaten) in vielen Fällen einfach zu erkennen. Die Modellierung der Automatisierungssoftware erfolgt rechnergestützt; Modellierungsfehler können nur aufgrund von Fehlern in der entwickelten Transformationsvorschrift vorhanden sein.

9.3 Ausblick

Es liegt nahe, den ganzheitlichen qualitativen Ansatz ebenfalls für Zuverlässigkeitsanalysen einzusetzen. Diese Idee wurde im Rahmen eines DFG-Antrags begutachtet und wird als Forschungsprojekt gefördert. Für die späteren Entwicklungsphasen steht eine Reihe von Methoden für die Zuverlässigkeitsanalyse zur Verfügung, die für einen bestimmten Bestandteil des Systems entwickelt wurden. Mit Hilfe qualitativer Modelle lässt sich das mögliche Verhalten eines Systems prinzipiell auch in sehr frühen Entwicklungsphasen beschreiben. Dabei ist es wichtig, sowohl mögliche Szenarien im bestimmungsgemäßen Fall als auch im Fehlerfall anhand von Situationen zu untersuchen, um das Systemverständnis des Softwareentwicklers zu erhöhen. Das Formulieren von zuverlässigkeitsfördernden Anforderungen könnte somit erleichtert werden. In Abbildung 9.1 ist das Vorgehen anhand des ganzheitlichen Systemmodells zur Identifikation von Anforderungen abstrakt dargestellt.

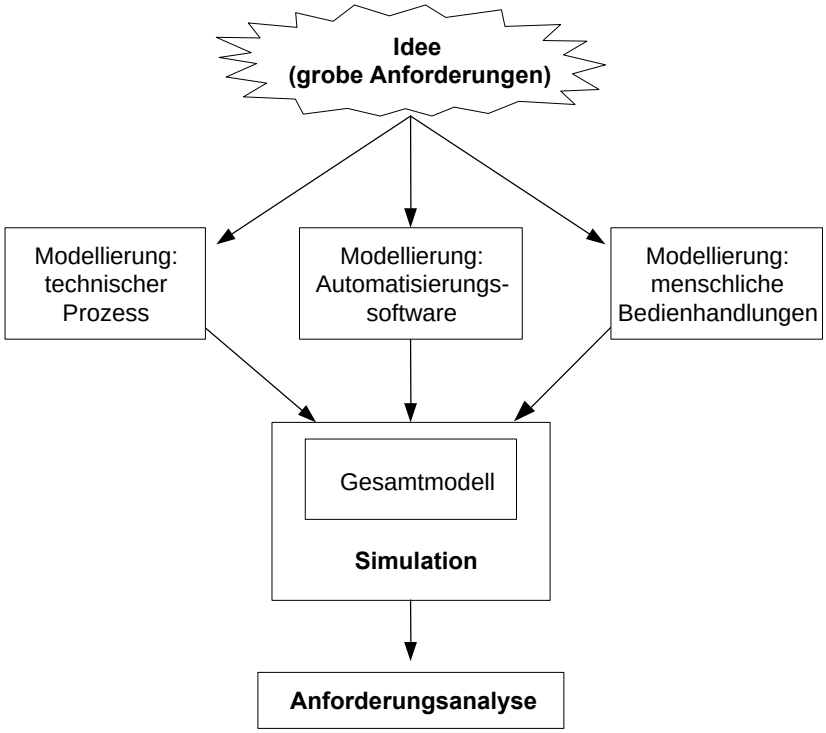


Abbildung 9.1: Qualitative Modelle zur Anforderungsanalyse

Durch qualitative Simulation des geplanten Systems kann ebenfalls der Einfluss verschiedener Zuverlässigkeitsmaßnahmen erprobt, optimiert und bewertet werden. Bei der Ermittlung der Systemzuverlässigkeit müssen Kenngrößen für das technische System, für die Automatisierungssoftware und für menschliche Bedieneingriffe ermittelt und in das qualitative Modell integriert werden. Dies soll im angesprochenen fakultätsübergreifenden Forschungsprojekt erarbeitet werden.

Die Verwendung qualitativer Modelle ist bei der Entwicklung von fehlertoleranten Systemen ebenfalls denkbar. Fehlertolerante Systeme besitzen eine sehr hohe Zuverlässigkeit und funktionieren selbst bei Ausfall mehrerer Systemelemente korrekt. Gerade für Systeme, bei denen das Eingreifen eines Menschen nicht möglich (z. B. Satelliten) oder unerwünscht ist, erhält die Entwicklung fehlertoleranter Software eine immer größere Bedeutung. Dabei gilt es, das Automatisierungskonzept so auszulegen, dass der Betrieb eines Systems auch beim Auftreten von Fehlern – unter gewissen Einschränkungen – aufrechterhalten werden kann. Um solche Systeme entwickeln zu können, muss das Verhalten des Systems unter allen möglichen Randbedingungen bekannt sein, insbesondere dessen Fehler und deren mögliche Auswirkungen. Genau dies ist die Stärke des vorgestellten Modellierungsverfahrens. Anstatt nun das ganzheitliche Modellierungskonzept auf einen bestehenden Entwurf anzuwenden, um dessen Schwachstellen zu lokalisieren, erfolgt die Modellierung entwurfsbegleitend. Somit ist ein langsames „Annähern“ an das Problem möglich und Entwurfsalternativen können anhand des Modells bewertet und verfeinert werden.

Literaturverzeichnis

[Abel90] D. Abel: *Petri-Netze für Ingenieure*, Springer, Aachen, 1990.

[AlDi94] R. Alur and D.L. Dill: *A Theory of Timed Automata*, Theoretical Comp. Science 126, S.183-235, Standford (USA), 1994.

[Bahn00] Forschungs- und Technologie Report der Deutschen Bahn AG: *Flexibel - Ohne Fahrer*, Bahntech 04/00, Berlin, 2000.

[Balz96] H. Balzert: *Lehrbuch der Software-Technik – Software Entwicklung*, Spektrum Akademischer Verlag, Heidelberg, Berlin, Oxford 1996.

[Beck97] Beck: *Umwelt-Recht*, Deutscher Taschenbuch Verlag, München, 10.Auflage 1997.

[BEE96] B. Beltz, J. Edenhofer, J. Eilers u.a.: *Sicherung der Qualität vor Serieneinsatz: System-FMEA,* Verband der Automobilindustrie, Band 4, Teil 2, Frankfurt, 1996.

[Bell98] F. Belli: *Methoden und Hilfsmittel für die systematische Prüfung komplexer Software*, Informatik-Spektrum 21:337-346, Springer-Verlag, 1998.

[Bent96] H. Bentele: *Entwurf und Implementierung eines optimierten Analysealgoritmus für die Situationsbasierte Qualitative Modellierung und Analyse,* Diplomarbeit, Institut für Automatisierungs- und Softwaretechnik, Universität Stuttgart, 1996.

[BFU02] Bundesstelle für Flugunfalluntersuchung, http://www.bfu-web.de/

[BHR90] K. Bareis, H. Hoffmann, L. Rossinelli: *PAAG – Verfahren (HAZOP)*, Risikobegrenzung in der Chemie, Internationale Sektion der IVSS für die Verhütung von Arbeitsunfällen und Berufkrankheiten in der chemischen Industrie, Heidelberg, 1990.

[Bieg00a] U. Biegert: *Sichere Automatisierungssysteme mit Hilfe qualitativer Modellierung und quantitativer Risikobewertung*, 14. Symposium Simulationstechnik, Universität Hamburg, Hamburg, 2000.

[Bieg00b] U. Biegert*: Computer-aided Safety Analysis of Computer-controlled Systems: a case example*, Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje (Polen), 2000.

[Bieg00c] U. Biegert: *Using Qualitative Models for Safety Analysis of Industrial Automation Systems*, 13th Conference on Software & Systems Engineering and their Applications – ICSSEA, Paris (Frankreich), 2000.

[Bieg98a] U. Biegert: *Qualitative Beschreibung von Automatisierungssoftware,* Hamburg Harburg, Überblick in Automatisierungstechnik 46, 7/98, 1998.

[Bieg97a] U. Biegert: *Sicherheitsanalyse für Automatisierungssysteme*, Kolloquium: Software-Entwicklung, TAE, Esslingen, (S.757ff.), 1997.

[Bieg97b] U. Biegert: *Prozessüberwachung auf Basis von qualitativen Modellen*, 11. Symposium Simulationstechnik, Dortmund, 1997.

[BiKo00] U. Biegert, J. Konnertz: *Introduction into modelling with Petri Nets*, Online Symposium for Electronics Engineers, OSEE, Waltham (USA), 2000.

[Bish90] P.G. Bishop: *Dependability of Critical Computer Systems 3*, Techniques Directory, EWICS TC7, Elsevier Applied Science, London, New York, 1990.

[Bits01] F. Bitsch*: Safety Patterns – the Key to Formal Specification of Safety Requirements*, SAFECOMP – Computer Safety, Reliability and Security, 20th International Conference, Budapest (Ungarn), 2001.

[BRJ99] G. Booch, J. Rumbaugh, I. Jacobson: *The Unified Modelling Language User Guide*, Addison Wesley, 1999.

[BSM+00] I. N. Bronstein, K. A. Semendjajew, G. Musiol, H. Mühlig: *Taschenbuch der Mathematik*, 5. Auflage, Harri Deutsch Verlag, Frankfurt, 2000.

[Bubb92] H. Bubb: *Menschliche Zuverlässigkeit*, ecoMed Fachverlag, Landsberg, 1992.

[Bußn90] H. Bußmann: *Lexikon der Sprachwissenschaft*, Alfred Kröner Verlag, Stuttgart, 1990.

[ClKu96] E. M. Clarke und E.M. Kurshan: *Computer-aided Verification*, IEEE Spectrum, S.61-67, Brüssel (Belgien), 1996.

[Cros92] A. Cross: *Fault tree and event tree*, In A.E. Grenne (Editor) , High Risk Saftety Atechnology, S.49 ff., John Wiley & Sons, New York, 1992.

[Dech87] Ausarbeitung des DECHEMA/GVC-Arbeitsausschusses Risiko, Schadenanalyse, Zuverlässigkeit: *Bewertung sicherheitsanalytischer Methoden für chemische und verfahrenstechnische Anlagen*, Chem.-Ing. Tech. 59, 1987.

[DIN 19250] DIN 19250: *Grundlegende Sicherheitsbetrachtungen für MSR-Schutzeinrichtungen*, Beuth Verlag, Berlin 1994.

[DIN 31000] DIN VDE 31000T2: *Allgemeine Leitsätze für das sicherheitsgerechte Gestalten technischer Erzeugnisse, Begriffe der Sicherheitstechnik, Grundbegriffe*. Beuth-Verlag, Berlin, 1987.

[DIN 40041] DIN 40041: *Zuverlässigkeit, Begriffe*, Beuth-Verlag, Berlin, 1987.

[EEK00] U. Epple, U. Enste, M. Kneissl: *Development of standardized process control software – avoiding bugs by using graph grammars*, 3rd Mathmod Vienna, IMACS Symposium on Mathematical Modelling, Wien, 2000.

[EnMü93] M. Engshuber und R. Müller: *Grundlagen der Verfahrenstechnik für Automatisierungsingenieure*, 2. Auflage, Deutscher Verlag für Grundstoffindustrie, Leipzig-Stuttgart 1993.

[Fehr98] B. Fehrenbacher: *Sicherheitsuntersuchungen bei einem EBS-Bremssystem für Nutzfahrzeuge*, Diplomarbeit, Institut für Automatisierungs- und Softwaretechnik, Universität Stuttgart, 1998.

[Forb84] K.D. Forbus: *Qualitative Process Theory*, Artificial Intelligence, New York, 1984.

[Fröh97] P. Fröhlich: *Überwachung verfahrenstechnischer Prozesse unter Verwendung eines qualitativen Modellierungsverfahrens,* Dissertation, Institut für Automatisierungs- und Softwaretechnik, Universität Stuttgart, 1997.

[GeCr94] S. Gerhart, D. Craigen, T. Ralston: *Experience with formal methods in critical systems*, IEEE Software, 11(1), S.21-28, January, 1994.

[GefS99] GefStoffV – Gefahrstoffverordnung:*Verordnung zum Schutz vor gefährlichen Stoffen,* Neufassung vom 15. November 1999, BGBl. I 1999.

[Göhn98] P. Göhner: *Komponentenbasierte Entwicklung von Automatisierungssystemen*. GMA-Kongress 98, Mess- und Automatisierungstechnik, Ludwigsburg, 1998.

[Göhn95] P. Göhner: *Spezifikation und Verifikation von sicheren Softwaresystemen*, atp – Automatisierungstechnische Praxis 37 (24-31), Oldenbourg, 1995.

[Hack00] S. Hackmack: *Prädikation und sekundäre Prädikation*, Bremer Linguistik Workshop, Februar, Bremen, 2000.

[Hack87] W. Hacker: *Fehlhandlungen und Arbeitsfehler,* Schriften zur Arbeitspsychologie, Nr 41, Verlag H. Huber, Stuttgart, 1987.

[HaKo99] W.A. Halang, R. Konakowsky: *Sicherheitsgerichtete Echtzeitsysteme*, Oldenbourg Industrieverlag GmbH, 1999.

[Hare87] D. Harel: *Statecharts: A Visual Formalism for Complex Systems*, In Science of Computer Programming, S.231-274, Elsevier Science Publisher (North Holland), 1987.

[HBB+97] Huber, Burgbacher, Biegert, Billmann: *Qualitative Systemanalyse und computergestützte Gefahrenidentifikation (HAZOP),* Chemie Ingenieur Technik 7/97, S.992, Weinheim, 1997.

[Hett99] F. Hettrich*: Intelligenter Hilfeassistent für die qualitative Modellierungsmethode SQMA*, Studienarbeit, Institut für Automatisierungs- und Softwaretechnik, Universität Stuttgart, 1999.

[HHW97] T.A. Henzinger, P.S. Ho und H. Wong-Toi: *HyTech: A model checker for hybrid Systems,* Software Tools for Technology Transfer Vol.1(1,2) S.110-122, Springer, 1997.

[Holz97] G.J. Holzmann: *The Spin Model Checker,* IEEE Transfer on Software Engineering, Vol. 23, No 5, S.279-295, 1997.

[IEC 61508] International Electronical Commission IEC 61508: *Functional safety of electronical/electronic/programmable electronic safety-related systems*, parts 1-3, 7th Version, Genf (Schweiz), 1999.

[KlBr84] J. De Kleer und J.S. Brown: *A Qualitative Physics Based on Confluences*, Artificial Intelligence, New York (USA), 1984.

[Kühn93] P. Kühn: *Einführung in die Informatik 1 und 2*, begleitendes Manuskript zur gleichnamigen Vorlesung, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1993.

[Kuip94] B. Kuipers: *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*, MIT Press, Cambridge, Massachusetts, 1994.

[LaGö99a] R. Lauber, P. Göhner: *Prozessautomatisierung*, Band 1, 3. Auflage, Springer-Verlag, Berlin, Heidelberg, New York, 1999.

[LaGö99b] R. Lauber, P. Göhner: *Prozessautomatisierung*, Band 2, 1. Auflage, Springer-Verlag, Berlin, Heidelberg, New York, 1999.

[Lauf96] X. Laufenberg: *Ein modellbasiertes qualitatives Verfahren für die Gefahrenanalyse,* Dissertation, Institut für Automatisierungs- und Softwaretechnik, Universität Stuttgart, 1996.

[Leve95] N.G. Leveson: *SAFEWARE – System Safety and Computers,* 1. Aufl., Addison-Wesley, New York, 1995.

[Litz98] L. Litz: *Grundlagen der sicherheitsgerichteten Automatisierungstechnik*, Automatisierungstechnik, at 2/98, Oldenbourg, 1998.

[LPY97] K.G. Larsen, P. Pettersson und W. Yi: *UPPAAL in a nutshell*, Software Tools for Technology Transfer Vol.1, S.134-152, Springer, New York (USA), 1997.

[LRR+98] P. Liggesmeyer, M. Rothfelder, M. Rettelbach und T. Ackermann*: Qualitätssicherung software-basierter technischer Systeme: Problembereiche und Lösungsansätze*, Informatik Spektrum 21, 1998.

[Lunz98] J. Lunze: *Qualitative Modellierung dynamischer Systeme durch stochastische Automaten*, at Automatisierungstechnik 46/98, S.271ff., Oldenbourg Verlag, 1998.

[Lyon98] A. Lyons: *UML for Real-Time Overview*, ObjecTime Publication, 1998.

[Manz00] S. Manz: *Entwicklung qualitativer Modelle zur Prozessüberwachung und Diagnose dynamischer Systeme*, 45. Internationales wissenschaftliches Kolloquium (IWK), TU Ilmenau, 2000.

[Manz97] S. Manz: *Qualitative Modellierung einer Einspritzanlage für Dieselmotoren,* Diplomarbeit, Institut für Automatisierungs- und Softwaretechnik, Universität Stuttgart, 1997.

[Marq95] W. Marquardt: *Modellbildung als Grundlage der Prozess-Simulation*, In Schuler, H. (Hrsg.): Prozess-Simulation. VCH, Weinheim, 1995.

[McMi92] K,L. McMilian: *The SMV System*, Carnegie-Mellon University, 1992

[Meis77] D. Meister: *Human Error in Man-Machine Systems*, Human Aspects of Man-Machine-Systems, Brown, N. T. Ed. Open University Press, Miltore Keyness (England), 1977.

[MeRe99] K. Meffert, D. Reinert: *Mikroprozessoren in sicherheitskritischen Anwendungen*, Elektronik 4 /1999.

[Mock01] R. Mock: *Moderne Methoden der Risikobewertung komplexer Systeme*, DISP 144, Nr. 39, 2001.

[Moik99] A. Moik: *Strukturierte Erstellung von formalen Sicherheitsmodellen für Automatisierungssysteme mit Sicherheitsverantwortung*, Encress, 1999.

[Moik97] A. Moik: *Formale Spezifikation der Steuerung einer Industriewaschmaschine – Erfahrungsbericht*, VDI-Berichte 1336, S. 51-62, VDI-Verlag, Düsseldorf, 1997.

[Mont00] S. Montenegro: *Sichere und fehlertolerante Steuerungen*, Carl Hanser Verlag, 2000.

[MoSt95] M. Montag, P. Struss: *Qualitatives und modellbasiertes Schließen,* München, 1995.

[MüTi00] D. H. Müller, T. Tietjen: *FMEA-Praxis*, Carl Hanser Verlag, München, Wien, 2000.

[NoRa90] C. Norman, J. Rasmuessen: *The application of probabilistic risk assessment techniques to energy technologies*, Readings in Risk, S.195-205, Resources for the Future, New York, 1990.

[Norm86] C. Norman: *New view of information processing: Implications for intelligent decision support systems,* in Hollnagel, Mancini &Woods (Hersg), Intelligent decision support in process environments, Springer-Verlag, Berlin, 1986.

[Obje97] ObjecTime: *ObjecTime Developer 5.1 User's Guide*, ObjecTime Limited, 340 March Road, Kanata, Ontario, 1997.

[Ortn97] E. Ortner: *Methodenneutraler Fachentwurf*, Teuber Verlag, Leipzig, 1997.

[Pala92] E-G. Paland: *Technisches Handbuch*, INA Wälzlager Schaeffler KG, Herzogenaurach, 1992

[Panr99] K. Panreck: *Systembeschreibung zur Modellierung komplexer Systeme*, at – Automatisierungstechnik 47 - 4/99, S.157 ff., Oldenbourg Verlag, 1999.

[Petri62] C.A. Petri: *Kommunikation mit Automaten*, Dissertation, Universität Darmstadt, 1962.

[Pilz85] V. Pilz: *Sicherheitsanalysen zur systematischen Überprüfung von Verfahren und Anlagen – Methoden Nutzen und Grenzen*, Chemie Ingenieur Technik 57, VCH Verlagsgesellschaft mbH, Weinheim, 1985.

[Pola98] A. Polat: *Kritikalitätsanalyse eines elektronischen Steuerungsgerätes für Dieselmotoren,* Diplomarbeit, Institut für Automatisierungs- und Softwaretechnik, Universität Stuttgart, 1998.

[Polk94] M. Polke: *Prozessleittechnik*, 2. Auflage, Oldenbourg, München-Wien, 1994.

[Rasm83] J. Rasmussen: *Skills, rules, knowledge: signals, signs and symbols and other distinctions in human performance models,* IEEE Transactions on Systems, Man and Cybernetics*,* SMC 13(3), S.257-267, 1983.

[Rati00] RationalRose RealTime: *Modeling Language Guide*, Rational Software Corporation, 2000.

[Reas94] J. Reason: *Menschliches Versagen*, Spektrum Akademischer Verlag, 1994.

[ReLe96] J. D. Reese and N. G. Leveson: *Software Deviation Analysis: A „Safeware Technique",* Safeware Engineering Corp., USA, 1996.

[Rigb70] L.V Rigby: *The Nature of Human Error,* Annual Technical Conference Transactions of the ASQC American Society for Quality Control, Milwaukee, WI 1970.

[RoRo83] W.B. Rouse, S.H. Rouse: *Analysis and Classifikation of Human Error,* IEEE Transaction of Systems, Man and Cybernetics, Vol. 13, No 4, S.539-549, 1983.

[Rush93] J. Rushby: *Formal verification of algorihms for critical systems,* IEEE Tansactions on Software Engineering, SE-19(1), S.13-23, January, 1993.

[Schi97] F. Schiller: *Diagnose dynamischer Systeme auf der Grundlage einer qualitativen Prozessbeschreibung*, VDI-Fortschrittberichte, Reihe 8: Meß-, Steuerungs- und Regelungstechnik Nr. 653, 1997.

[Schle99] A. Polat: *Kritikalitätsanalyse eines elektronischen Steuerungsgerätes für Dieselmotoren,* Diplomarbeit, Institut für Automatisierungs- und Softwaretechnik, Universität Stuttgart, 1999.

[Schn99] E. Schnieder: *Methoden der Automatisierung*, Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig, 1999.

[SCJ98] E. Schnieder, M. Chouikha, A. Janhsen: *Klassifikation und Bewertung von Beschreibungsmitteln für die Automatisierungstechnik*, at – Automatisierungstechnik 46/98 (12), S. 582 ff., Oldenburg Verlag, 1998.

[Seib99] Seibt+Kapp GmbH: *Handbuch zur Variofex 16,* Erdmannhausen, 1999.

[SGW94] G. Selic, P. Gullekson, P. Ward: *Real-Time Object-Oriented Modeling,* John Wiley & Sons Inc., New York, 1994.

[SSG+90] J. Serrono, V. Santonja, P.J. Gil, R. Ors: *Reliability and Safety Evaluation Techniques für Components and Processes*, Dept. Ingenieria de Sistemas, Computadores y Automatica, Valencia, 1990.

[SSH92] P. Sander, W. Stucky, R. Herschel: *Automaten, Sprachen, Berechenbarkeit*, 2. Auflage, Teubner Verlag, Stuttgart, 1995.

[Strä00] O. Sträter: *Evaluation of Human Reliability on the Basis of Operational Experience*, Dissertation, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) mbH, 2000.

[Stra97] M. Straßer: *Modifizierung der SQMA-Methode zur Modellierung von Softwarekomponenten*, Diplomarbeit, Institut für Automatisierungs- und Softwaretechnik, Universität Stuttgart, 1997.

[Stro92] G. Strohrmann: *Automatisierungstechnik*: *Grundlagen, analoge und digitale Prozesssysteme*, Band 1, Oldenburg, München, Wien, 1992.

[Stru90] P. Struss: *Problems of interval-based qualitative reasoning*, In Qualitative Reasoning about Physical Systems, 1990.

[Swai80] A.D. Swain: *The human element in system safety: A guide for modern mangement*, Albuquerque, 1980.

[VDI 3542] VDI/VDE 3542: *Sicherheitstechnische Begriffe für Automatisierungssysteme, Anwendungshinweise und Beispiele*, VDI/VDE-Richtlinien, Beuth Verlag GmbH, 1998.

[VDIN 01] VDI-Nachrichten vom 21.09.01 zum Thema: *„Weltsicherheitskongress Saarbrücken*“, zusammengestellt von B. Eusemann, S.8, 2001.

[Well99] T. Weller: *Entwicklung eines Modellierungsassistenten für die qualitative SQMA-Modellierung*, Diplomarbeit, Institut für Automatisierungs- und Softwaretechnik, Universität Stuttgart, 1999.

[WeOs98] E. Westkämper, Osten-Sacken, v.d.D.: *Product Life Cycle Costing Applied to Manufacturing Systems*, In: Annals of the CIRP Vol. 47/1/1998, S.353 - 356, Bern, Hallwag, 1998.

[Yovi97] S. Yovine: *Kronos: A verification tool for real-time systems*, Software Tools for Technology Transfer Vol.1(1,2), S.123-133, Springer, 1997.

[Zimo90] B. Zimolong: *Fehler und Zuverlässigkeit,* Enzyklopädie der Psychologie: Ingenieurspsychologie, S. 313-345, Verlag für Psychologie, Dr. C. J Hogrefe, Göttingen, Toronto, Zürich, 1990.

Lebenslauf

Persönliche Daten:

18.08.1968	geboren in Heilbronn verheiratet, zwei Kinder

Schulbildung:

1975 – 1979	Grundschule in Heilbronn
1979 – 1985	Gymnasium in Heilbronn
1985 – 1988	Technisches Gymnasium in Heilbronn, Abschluss Abitur

Wehrdienst:

1988 – 1989	1./Nachschubbataillon 12, Bad Mergentheim

Studium:

1990 – 1996	Studium der Elektrotechnik an der Universität Stuttgart, Studienmodell: „Technische Elektronik“
1996	Abschluss mit akad. Grad. Diplom-Ingenieur

Berufstätigkeit:

1996 - 2002	Wissenschaftlicher Assistent am Institut für Automatisierungs- und Softwaretechnik der Universität Stuttgart Erziehungsjahr 2001