

Formale Fehlerbaumanalyse, Spezifikation und
Testen von hybriden Echtzeitsystemen in
Anwendung auf die Servicerobotik

Antrag auf Gewährung einer Sachbeihilfe
– Neuantrag –

Antragsteller: Prof. Dr. Bernd Krieg-Brückner,
Prof. Dr. Jan Peleska

Universität Bremen

Bremen, den 10.09.99

1 Allgemeine Angaben

Antrag auf Gewährung einer Sachbeihilfe
– Neuantrag –

1.1 Antragsteller

Prof. Dr. Bernd Krieg-Brückner, Universitätsprofessor,
geboren am 15.02.49, deutscher Staatsangehöriger,

Dienstanschrift:

Bremer Institut für Sichere Systeme

TZI, FB 3, Universität Bremen

Postfach 330440, D-28334 Bremen

Tel.: (+49) 421 218 3660

Fax : (+49) 421 218 3054

e-Mail: bkb@informatik.uni-bremen.de

Prof. Dr. Jan Peleska, Universitätsprofessor,
geboren am 14.02.58, deutscher Staatsangehöriger,

Dienstanschrift:

Bremer Institut für Sichere Systeme

TZI, FB 3, Universität Bremen

Postfach 330440, D-28334 Bremen

Tel.: (+49) 421 218 7092

Fax : (+49) 421 218 3054

e-Mail: jp@informatik.uni-bremen.de

1.2 Thema

Formale Fehlerbaumanalyse, Spezifikation und Testen von hybriden Echtzeitsystemen in Anwendung auf die Servicerobotik

1.3 Kennwort

SafeRobotics

1.4 Fachgebiet und Arbeitsrichtung

Praktische Informatik: Formale Methoden, Sichere Systeme

1.5 Voraussichtliche Gesamtdauer

36 Monate

1.6 Antragszeitraum

36 Monate

1.7 Gewünschter Beginn der Förderung

1. Januar 2000

1.8 Zusammenfassung

In diesem Projekt sollen formale Methoden entwickelt bzw. angepaßt werden, die geeignet sind, eine große Klasse von Servicerobotik-Anwendungen als Sichere Systeme zu gestalten. Das Verfahren läßt sich allgemein nutzen, um Sicherheitsanforderungen an hybride Echtzeitsysteme zu spezifizieren und deren Einhaltung nachzuweisen. Die Anwendung der Methode auf den Bremer Autonomen Rollstuhl ist zentraler Bestandteil des Vorhabens. Auf dem zukunfts-trächtigen Gebiet der Rehabilitationsrobotik ist die Idee des beweisbar korrekten Systemverhaltens bisher nur in Ansätzen vorhanden. Dies gilt insbesondere für Fälle, in denen die Anforderungen der gemeinsamen Steuerung durch Mensch und Maschine zu berücksichtigen sind.

Das Arbeitsprogramm umfaßt die Aufgaben „Formaler Sicherheitsentwurf unter Einbeziehung der Echtzeitanforderungen eines eingebetteten Systems“, „Systematische Behandlung durch gemeinsame Steuerung induzierter Bedrohungen“, die „Spezifikation des Kommunikationsverhaltens“ sowie die Validation und das „Testen des Gesamtsystemverhaltens“.

2 Stand der Forschung, eigene Vorarbeiten

2.1 Stand der Forschung

Es soll hier nur der Stand der Forschung in bezug auf die Anwendung formaler Methoden in der Robotik beschrieben werden. Allgemeinere Informationen zu formalen Methoden finden sich in [4], eine Übersicht verschiedener Bedrohungsanalyseverfahren stellt Storey [24] zusammen. Die systematische Untersuchung der Fehlerbaumtechnik geht zurück auf Vesely et al. [25]; die IEC definierte 1990 in IEC 1025 (s. [10]) eine standardisierte Darstellung von Fehlerbäumen.

Im Gegensatz zu anderen Anwendungsfeldern wie der Luft- und Raumfahrt ist die formale Behandlung von Sicherheitsanforderungen in den recht jungen Forschungsgebieten der Servicerobotik und insbesondere der Rehabilitationsrobotik (noch) nicht etabliert. Obwohl es allgemeiner Konsens ist, daß Roboter wie z.B. Verstärker menschlicher Körperkraft [11] oder intelligente Rollstühle [1, 18] als sicherheitskritische Systeme betrachtet werden sollten, gibt es bisher nur wenige Arbeitsgruppen, die sich mit dem Thema Sicherheit in der (Service-)Robotik beschäftigen.

An der Universität Lancaster wurden verschiedene Ansätze zur Sicherheitsanalyse untersucht und das Konzept eines „Safety Managers“ eingeführt [21, 23]. Als Anwendungsbeispiele in diesen Arbeiten dienen hauptsächlich automatisierte Baumaschinen, wie z.B. ein Bagger-Roboter. Der Schwerpunkt wird auf die Systemanalyse gelegt (Entwicklung der Bedrohungsanalysemethode CLASH), während Verifikations- oder Testansätze fehlen [20, 22, 13]. Einen weitaus formalen Ansatz verfolgt Zhang mit der Entwicklung eines semantischen Modells für dynamische Systeme, den sogenannten „Constraint Nets“ [26]. In dieser Arbeit wird zusätzlich ein formales Verifikationsverfahren vorgeschlagen. Die Arbeitsgruppe „Robots for Hazardous Environments“ an der Rice-University beschäftigt sich mit fehlertoleranten Roboterarchitekturen [15]. Ziel dieser Arbeiten ist es, die Zuverlässigkeit von Robotern durch Verfahren wie Fehlerbaumanalyse und Risikoanalyse zu erhöhen [6]; formale Verifikationen werden durch die entwickelten Methoden nicht abgedeckt. Nicht direkt im Robotikzusammen-

hang, jedoch für die Fehlerbaumanalyse relevant sind die Arbeiten einer polnischen Gruppe um Górski. Zum Beispiel in [5] wird ein Ansatz diskutiert, wie sich Fehlerbäume inklusive der zeitlichen Abhängigkeiten formalisieren lassen und auf Konsistenz überprüfen lassen.

Die oben aufgeführten Arbeiten beschränken sich auf bestimmte Aspekte der Systementwicklung, keines deckt den kompletten Software- bzw. Systemlebenszyklus ab. Im hier beantragten Projekt soll dagegen eine einheitlich Methode entwickelt werden, die Aufgaben des Requirements Engineering genauso erfüllt wie sie die Verifikations- und Testphase abdeckt (s.u.).

2.2 Eigene Vorarbeiten

2.2.1 Kognitive Robotik und der Bremer Autonome Rollstuhl

Die Arbeitsgruppe *AG BKB/Kognitive Robotik* (Leitung: Prof. Krieg-Brückner) im Bremer Institut für Sichere Systeme an der Universität Bremen beschäftigt sich seit über fünf Jahren mit der Entwicklung eines intelligenten Transportmittels für behinderte und ältere Menschen.

Im Rahmen des von der DFG geförderten Graduiertenkollegs „Raumorientierung und Handlungsorganisation Autonomer Systeme“ (Sprecher: Krieg-Brückner) in interdisziplinärer Zusammenarbeit von Neurobiologie, Neurophysik, Psychologie, Informatik und Robotik sowie im Rahmen von zwei jeweils zweijährigen studentischen Projekten mit je ca. 20 Teilnehmern (das dritte läuft seit dem WS 98/99) sind in der AG BKB vier einschlägige Dissertationen und zahlreiche Veröffentlichungen entstanden sowie das Simulationssystem SimRobot (s.u.) und der erste Prototyp des teilautonomen Rollstuhls (u.a. [55, 56, 57, 38]) entwickelt worden. Besonders ausgezeichnet wurde [60] durch die Verleihung des Dissertationspreises 1999 der Arbeitsgemeinschaft der deutschen KI-Institute. Der zweite Prototyp, der Bremer Autonome Rollstuhl „Rolland“ (Abb. 1) ist wie sein Vorgänger mit Sensorik und Steuer-PC ausgestattet, nun auf der Basis des Elektrorollstuhls Meyra Genius 1.522. Er dient in erster Linie als wissenschaftliche Experimentierplattform im Rahmen des Schwerpunktprogramms „Raumkognition“ der Deutschen Forschungsgemeinschaft. Die Forschung im Bereich der Raumkognition konzentriert sich auf Fragen der Navigation. Dabei werden Erkenntnisse aus der Psychologie und der Biologie berücksichtigt, es wird aber auch versucht, in diesen Bereichen noch offene Fragen durch die technische Umsetzung auf einem Roboter zu beantworten. Letzteres ist insbesondere für die kognitive Psychologie interessant, da das Vorwissen eines „Roboterprobanden“, im Gegensatz zu tierischen und menschlichen Versuchskandidaten, sehr leicht exakt definiert werden kann.

Neben der theoretischen Beschäftigung mit den Grundlagen der Navigation werden robuste Techniken untersucht, die es dem Rollstuhl, aufbauend auf verschiedenen Grundverhalten wie Wandverfolgung oder Einbiegen in eine Tür, erlauben, auch komplexe Verhaltensweisen zu erlernen. Durch Kombination der Grundverhalten mit dem Erkennen von Wegmarkenkonstellationen wurde ein Navigationsverfahren entwickelt, das dem autonomen mobilen System das Erlernen von Routen durch Teaching erlaubt, d.h. Vormachen durch einen Lehrer. Die gelernten Routen können danach selbständig von dem mobilen System zurückgelegt werden. Dabei wird insbesondere Wert auf die Erkennung und das Behandeln möglicher Fehler gelegt [43, 59]. Als Wegmarken wurden bisher ein-



Abbildung 1: Der Bremer Autonome Rollstuhl „Rolland“

fache, künstliche Markierungen verwendet; an einer Nutzung von natürlichen Strukturen zur Orientierung wird gegenwärtig gearbeitet ([64], s.a. [34]). In Gängen können zudem auch Ecken als Orientierungspunkte verwendet werden, die sich allein durch die Auswertung der Eigenbewegung des Systems erkennen lassen [62]. Ein weiteres Navigationsverfahren erlaubt das Erlernen von Routen mit Hilfe einer 360°-Zeilenkamera [55, 56, 57, 58, 60]. Dieses Verfahren kommt bereits jetzt ohne künstliche Landmarken aus, ist allerdings stark abhängig von der Qualität der Sensoren und der Beleuchtung.

Im Rahmen des Schwerpunktprogramms „Raumkognition“ werden auch die Grundlagen der sprachlichen Repräsentation von Raum und Bewegung untersucht. Zu diesem Zweck soll die Kontrolle des Rollstuhls durch Spracheingabe realisiert werden. Beim Aufbau des Bremer Autonomen Rollstuhls wurde der am Bremer Institut für Sichere Systeme entstandene Robotersimulator SimRobot [66] eingesetzt und so weiterentwickelt, daß direkt zwischen Simulation und Realität umgeschaltet werden kann [61]. Dieses hat sich als leistungsfähiges und zeitsparendes Konzept bei der Entwicklung erwiesen.

2.2.2 Rollstuhl und Sicherheit

Neben dem Einsatz des Rollstuhls als Experimentierplattform eignet er sich insbesondere auch als Demonstrator für den Einsatz formaler Methoden bei der Entwicklung eingebetteter Systeme. Dies läßt sich mit der bereits erwähnten besonderen Relevanz von Sicherheitsaspekten in der Rehabilitationsrobotik begründen. Es hat sich als sinnvoll erwiesen, den in der Literatur verwendeten Sicherheitsbegriff (Definition von Laprie [14]), der besagt, daß Sicherheit dann erreicht ist, wenn das Nicht-Auftreten von katastrophalen Konsequenzen für

die Umgebung des Systems gewährleistet ist, dahingehend zu erweitern, daß er zusätzlich noch die Befolgung von Benutzeranweisungen einschließt [44], also auch Funktionalität und Verfügbarkeit umfaßt.

Lankenau (AG BKB/Kognitive Robotik) und Meyer (AG BS) entwickelten mit Hilfe formaler Methoden (Fehlerbaumbasierte Bedrohungsanalyse (s. Abb. 2), Ableitung von Sicherheitsbedingungen und –mechanismen, CSP–Spezifikation, Verifikation von Sicherheitseigenschaften durch Modelchecking) eine Sicherheitsschicht für den sensorikbestückten elektrischen Rollstuhl Rolland [44, 46, 47].

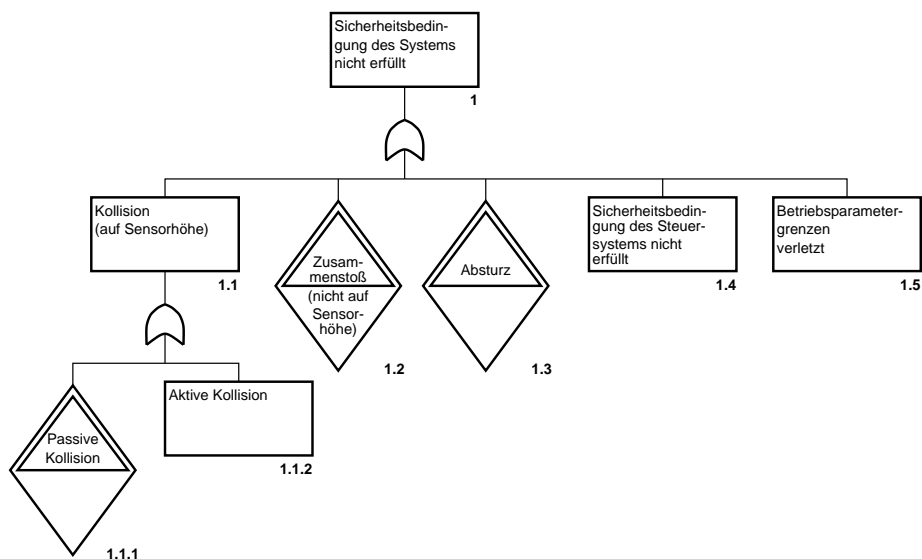


Abbildung 2: Oberste Ebene des Fehlerbaums für den Bremer Autonomen Rollstuhl [44]. Die schlimmste Bedrohung ist eine Verletzung der Sicherheitsbedingungen des Systems. Die Bedrohungen sind nicht formal spezifiziert.

Die Sicherheitsschicht ermöglicht durch die konsequente Berücksichtigung von Worst–Case–Betrachtungen mit Hilfe neuartiger virtueller Sensoren eine kollisionsfreie Navigation in einer geeigneten Umgebung (s.a. [63]). Zusätzlich unterstützt sie mit einem realzeitfähigen Netzwerkprotokoll die Integration unterschiedlicher Rechner und Betriebssysteme. Damit lassen sich sowohl zeitkritische als auch nichtrealzeitfähige Applikationen in einem System gemeinsam betreiben. Der entwickelte Verifikationsansatz basiert auf der Idee, die Spezifikation des Systems gegen eine CSP–Implementierung des Fehlerbaums zu überprüfen, um zu garantieren, daß das System niemals einen katastrophalen Zustand annimmt. Der fehlerbaumbasierte Ansatz liefert außerdem eine Spezifikation der Voraussetzungen, die die Umgebung erfüllen muß, damit das System korrekt arbeitet. Im Gegensatz zu dem hier beantragten Projekt wurde jedoch vollständig auf eine formale Spezifikation der Bedrohungen verzichtet. Erst die zu verifizierenden Sicherheitsbedingungen wurden in CSP notiert.

Als erstes Anwendungsmodul oberhalb der Sicherheitsschicht wurde der sogenannte DoorWizard entwickelt, der dem Benutzer die Filigranarbeit beim Durchfahren einer Tür abnimmt [63].

2.2.3 Formale Methoden und die UniForM Workbench

Die Arbeitsgruppe *AG BKB/Formale Methoden und Werkzeuge* beschäftigt sich seit über 15 Jahren an der Universität Bremen mit der Entwicklung korrekter Software für Sichere Systeme; in diesem Zusammenhang sind zwölf Dissertationen und zahlreiche Veröffentlichungen entstanden. Das hier am meisten relevante neuere Ergebnis ist die Universelle Entwicklungsumgebung für formale Methoden, die UniForM Workbench, die als Integrator für bisherige Arbeiten zur Methodik, vor allem aber Softwaresysteme zur Unterstützung der Entwicklung Sicherer Systeme (auch anderer) dient (vgl. [42, 40, 41, 36, 37, 48]). Für die formale Spezifikation von (abstrakten) Datentypen werden Werkzeuge zur Unterstützung des internationalen Standards CASL der CoFI Initiative der IFIP WG 1.3 (Foundations of System Specification) entwickelt (siehe [32] und [27]). Daneben sind die Werkzeuge zur Entwicklung durch Transformation besonders vielversprechend, da sie auch die Wiederverwendung des Entwicklungsprozesses erlauben [39]. Ein bereits früher entwickelter Satz von Transformationsregeln [35] wird derzeit angepaßt.

Die beiden Antragsteller und ihre Arbeitsgruppen arbeiten im Rahmen des Bremer Instituts für Sichere Systeme (Sprecher: Krieg-Brückner) eng zusammen. In mehreren Kooperationsprojekten zwischen dem Bremer Institut für Sichere Systeme und Industriepartnern wie der DASA, OHB (Bremen), Elpro (Berlin), Airbus, Siemens Verkehrstechnik, Bosch und der Südafrikanischen Staatseisenbahn wurde das Werkzeug RT-Tester (vormals VVT-RT [28, 50, 51, 52, 54]) implementiert und eingesetzt. Das Tool zur formalen Entwicklung, Validation und Test für reaktive Systeme ermöglicht es, auf der Basis einer formalen CSP-Spezifikation automatisch alle relevanten Testfälle zu generieren. Diese werden in einer Hardware-in-the-Loop Testanordnung als Eingabe für das zu testende Zielsystem verwendet. Die Auswertung der durchgeführten Tests geschieht in Echtzeit. Die Vorteile von RT-Tester gegenüber herkömmlichen Testmethoden sind u.a. die Möglichkeit der Überprüfung des Zusammenwirkens von Hardware und Software, die erreichbare extrem hohe Testabdeckung (konvergiert gegen Verifikation), die einfache Testbarkeit des Echtzeitverhaltens sowie der minimale Aufwand bei Regressionstests; besonders interessant für industrielle Anwendungen ist dabei die automatische Auswertung der Tests. RT-Tester wird derzeit in die UniForM Workbench integriert.

In anderen Kooperationsprojekten mit der DASA-Raumfahrt-Infrastruktur wurde für das zentrale fehlertolerante Rechnersystem der International Space Station Alpha der Beweis der Deadlock- und Livelock-Freiheit geführt [30, 31]. Hierzu wurden Abstraktionstransformationen von Hand durchgeführt, die derzeit im Rahmen einer Diplomarbeit in der UniForM Workbench als formale Transformationsregeln implementiert werden sollen.

Für die Abwicklung von Software-Projekten, aber auch die Integration von Software- und Hardware-Entwicklung spielen in der Praxis Vorgehensmodelle eine große Rolle. Zur Unterstützung des V-Modells sind bereits erste Werkzeuge entstanden [29], die auf formale Methoden erweitert und in die UniForM Workbench integriert werden sollen. Ferner wurde in einer Dissertation eine Methodik für die kombinierte Entwicklung von Hardware und Software für sicherheitskritische Systeme entwickelt [65], mit der das V-Modell erweitert werden soll.

3 Ziele und Arbeitsprogramm

3.1 Ziele

3.1.1 Wissenschaftliche Zielsetzung

Am Beispiel des Bremer Autonomen Rollstuhls soll in diesem Projekt eine auf der UniForM Workbench basierende formale Entwicklung von zeit- und sicherheitskritischen eingebetteten hybriden Systemen demonstriert werden. Besonders interessant ist in diesem Fall, daß über die herkömmliche Integration einer Steuereinheit in ein technisches System hinaus die Kooperation zwischen menschlichem Nutzer und eingebettetem System in sicherer Weise erfolgen muß (shared-control system). Es wird der gesamte Software-Entwicklungszyklus von der Problemspezifikation bis zur Verifikation und zum automatisierten Test des Systems abgedeckt, so daß die in den vergangenen Jahren mit den bisherigen Ansätzen im Bereich „Formale Entwicklung Sicherer Systeme“ gewonnenen Erkenntnisse in das Feld der „(Service)-Robotik“ transferiert werden.

Roboter im allgemeinen und Service- bzw. Rehabilitationsroboter im speziellen eignen sich ausgezeichnet als Demonstrationsobjekt für die Anwendbarkeit unterschiedlicher formaler Methoden für die Entwicklung korrekter Software respektive sich spezifikationsgemäß verhaltender technischer Systeme. Sie sind im Vergleich zu einigen Beispielen aus der Luftfahrt noch überschaubar groß, bieten aber dennoch bereits alle relevanten Problemstellungen und Anforderungen, die Entwicklungsmethoden erfüllen sollten. Es handelt sich um eingebettete hybride Systeme, die sehr stark reaktiv ausgelegt sind, da sie unablässig mit der Umgebung kommunizieren (z.B. durch Auswerten von Sensorinformationen). Insbesondere Rehabilitationsrobotikanwendungen unterliegen harten Echtzeitanforderungen, vor allem wenn es sich wie beim Bremer Autonomen Rollstuhl um mobile Systeme handelt. Wegen fehlender Informationen über die Beschaffenheit der Umgebung müssen die Systeme fehlertolerant ausgelegt sein: sie müssen auch unvorhergesehene Situationen meistern; aus Kostengründen kommt allerdings eine Mehrfachauslegung der Hardware nicht in Frage.

Die zusätzlich interessante Komponente der Rehabilitationsroboter gegenüber anderen eingebetteten Systemen ist der hohe Grad an Mensch-Maschine-Interaktion. Existierende formale Entwicklungen eingebetteter Systeme haben im allgemeinen nur die möglichen Eingaben einer physikalisch mehr oder weniger beschreib- und damit modellierbaren Umwelt zu berücksichtigen. Durch geeignete Modelle und Worst-Case-Abschätzungen über das Verhalten der Umwelt lassen sich hier treffende Aussagen über die Zuverlässigkeit der Systeme machen. Die Entscheidungsfindung eines menschlichen Benutzers, der mit der Maschine interagiert, läßt sich dagegen nur schwer in einem (physikalischen) Modell beschreiben, so daß er eine Art „zweite Umwelt“ darstellt, die im formalen Entwicklungsprozeß zu berücksichtigen ist. Diese Integration wird noch wichtiger, wenn es sich wie im Falle des Bremer Autonomen Rollstuhls bei den zu entwickelnden Systemen um Shared-Control Systeme handelt, in denen es zu wechselnden Priorisierungen hinsichtlich der Steuerung zwischen technischem System und Benutzer kommt.

3.1.2 Relevanz für Anwendungen

Wegen der zunehmenden Verschiebung der Altersstruktur in der Bevölkerung steigt der Bedarf an Unterstützung für sensorisch oder motorisch beeinträchtigte Menschen (im wesentlichen Ältere, Kranke und Behinderte) deutlich an. Die Erhaltung beziehungsweise die Wiedergewinnung der eigenen Mobilität beeinflusst gerade bei diesem Personenkreis in hohem Maße die Lebensqualität.

Die Rehabilitationsrobotik als Teilbereich der Servicerobotik beschäftigt sich daher unter anderem mit der Frage, wie sich Menschen im täglichen Leben durch Roboter, wie z.B. intelligente Elektrorollstühle, unterstützen lassen können. Diesbezüglich wird in der Arbeitsgruppe „Kognitive Robotik“ (Leitung: Krieg-Brückner) derzeit im Rahmen des DFG Schwerpunktprogramms „Raumkognition“ Grundlagenforschung in bezug auf die Wahrnehmung der Umgebung und die Navigation im Innen- und Außenraum betrieben (s. Abschnitt 2.2).

Vergegenwärtigt man sich die Entwicklung vom klassischen Industrieroboter, der sehr schnell und sehr präzise sich stets wiederholende Aufgaben stupide ausführt, über herkömmliche Serviceroboter, die z.B. nachts die Reinigung von Fabrikhallen übernehmen, bis hin zu heutigen und zukünftigen Rehabilitationsrobotern, wie dem hier als Demonstrator verwendeten Bremer Autonomen Rollstuhl, so ist die stark steigende Abhängigkeit des menschlichen Benutzers vom technischen System offenkundig. Während zu Lackschäden führende Fehlfunktionen von Fließbandrobotern in Automobilwerken „lediglich“ Geld kosten, können unvorhergesehene Aktionen eines (teil-) autonomen Rollstuhls unter Umständen gesundheitsgefährdende Konsequenzen für den Benutzer oder für Personen in dessen Umgebung haben.

Neben dem steigenden Sicherheitsniveau gilt es, den möglicherweise hohen Einfluß des menschlichen Benutzers auf die Entscheidungsfindung des Systems zu beachten. Für Anwendungen im Bereich der Servicerobotik ist demnach die Zuverlässigkeit des technischen Systems von herausragender Bedeutung. Dies gilt insbesondere auch für Rehabilitationsroboter, da die Benutzer möglicherweise unter sensorischen oder motorischen Einschränkungen leiden, und somit noch stärker auf ein zuverlässiges Verhalten der Maschine angewiesen sind. Dabei sind zum Beispiel bei einem Rollstuhl nicht nur Zusammenstöße des Fahrzeugs mit Gegenständen oder Lebewesen unbedingt zu vermeiden, sondern es ist auch die Funktionsfähigkeit des Rollstuhls sicherzustellen, wenn es etwa darum geht, den Benutzer zum Medizinschrank zu fahren.

Auch wenn die Rollstuhlanwendung in diesem Projekt als Demonstrationsbeispiel im Vordergrund steht, so soll doch die Universalität des hier vorgeschlagenen Vorgehens betont werden. Die entwickelten und zu entwickelnden Methoden lassen sich auf eine große Klasse eingebetteter Systeme anwenden.

3.2 Arbeitsprogramm

Im Rahmen des hier beantragten Projekts werden vier Arbeitspakete in Angriff genommen. Durch die Zusammenarbeit mit anderen Mitgliedern der Forschungsgruppe wird (wie oben angedeutet) die Abdeckung des gesamten Entwicklungszyklus eines sicherheitskritischen eingebetteten Echtzeitsystems gewährleistet — von der Sicherheitsanalyse und der daraus resultierenden formalen Spezifikation der Systemeigenschaften bis zum automatisierten Testen des realen Systems.

3.2.1 Formale Bedrohungsanalyse und Sicherheitsmechanismen (AP 1)

Im Rahmen einer formalen Bedrohungsanalyse (*hazard analysis*) werden in systematischer Weise alle Situationen erfaßt, die ein Versagen des Systems zur Folge haben könnten. Diese Situationen sind formal zu spezifizieren, so daß sie in der Folge zur Definition von Sicherheitsbedingungen herangezogen werden können. Anschließend sind Verfahren zu entwickeln, die gewährleisten, daß diese Sicherheitsbedingungen stets eingehalten werden, so daß das Gesamtsystem niemals versagt. Durch die Verwendung einer formalen Spezifikationsprache für die Repräsentation der Bedrohungen des Systems wird eine mechanische und damit automatisierbare Herleitung der Sicherheitsbedingungen unterstützt. Schließlich erlaubt die formale Spezifikation eine Verifikation der geforderten Sicherheitseigenschaften für zu entwickelnde Sicherheitmechanismen. Im Robotikkontext ist es von besonderem Interesse, daß aus der Bedrohungsanalyse voraussetzende Bedingungen bezüglich der Beschaffenheit der Umgebung abgeleitet werden können.

So soll sich beispielsweise berechnen lassen, mit welcher Geschwindigkeit sich Objekte in der Umgebung des Roboters höchstens bewegen dürfen, ohne daß sie eine Kollisionsgefahr für den Roboter darstellen.

Die bekannten Verfahren für eine systematische Bedrohungsanalyse (vgl. Abschnitt 2.1) lassen sich in zwei Gruppen einteilen: FMEA (Failure modes and effects analysis), FMECA (Failure modes, effects and criticality analysis), HAZOP (Hazard and operability studies), ETA (Event tree analysis) und andere analysieren in einer „bottom-up“-Strategie, welche Konsequenzen das Auftreten eines Fehlers (in einer Komponente) für das Gesamtsystem hat [24]. Die Definition der Bedrohungen erfolgt üblicherweise in natürlicher Sprache. Der Hauptvertreter der zweiten Gruppe von Bedrohungsanalyseverfahren ist die Fehlerbaumanalyse [25]. Sie verfolgt eine „top-down“-Strategie, indem die allgemeinste Bedrohung des Gesamtsystems schrittweise verfeinert wird. Die Definition der jeweiligen Bedrohungen kann sowohl natürlichsprachlich als auch formal geschehen. Die Abhängigkeiten der Bedrohungen untereinander werden meistens durch aussagenlogische Formeln bzw. ihr grafisches Äquivalent ausgedrückt.

Als Erweiterung des von Lankenau und Meyer beschriebenen Ansatzes [44] soll im hier beantragten Projekt eine formale Beschreibungsmethode für die Bedrohungen entworfen werden, die es erlaubt, Echtzeitbedingungen sowie hybride Aspekte zu repräsentieren. Aufbauend auf der von Hansen in [7] vorgestellten Nutzung des Duration Calculus [3] soll nicht nur die Spezifikation der einzelnen Basisbedrohungen eine Echtzeitsemantik erhalten, sondern vielmehr auch die Verknüpfung der verschiedenen Bedrohungen im Fehlerbaum.

Ein erster Vorschlag in dieser Richtung stammt von Górski und Wardziński [5], die sowohl an den Blättern als auch für die Knoten temporallogische Ausdrücke verwenden. Ihr Ansatz der auf Timed Petri Nets basierenden Konsistenzanalyse der Spezifikation erlaubt allerdings keine Aussagen zur Verifikation des Systems gegenüber der Spezifikation der Sicherheitseigenschaften.

Konkret soll die Fehlerbaumanalyse folgende Schritte umfassen:

- Spezifikation der Bedrohungen im Duration Calculus (evtl. im erweiterten Duration Calculus). Dabei sind sowohl die Blätter des Fehlerbaums als auch die inneren Knoten zu modellieren.

- Ableiten von Sicherheitsbedingungen. Dies ist im Fall der herkömmlichen UND/ODER-Fehlerbäume im wesentlichen durch eine Negation der Bedrohungen an den Blättern zu erreichen. Es ist zu untersuchen, wie sich dies auf die hier angestrebte Semantik übertragen läßt. Am Ende existiert eine Spezifikation von Sicherheitsbedingungen im Duration Calculus. Mit Hilfe der Instantiierung des Theorembeweisers Isabelle für den Duration Calculus [8] sollen dann Vollständigkeit und Konsistenz dieser Sicherheitsbedingungen überprüft werden.
- Die spezifizierten Sicherheitsbedingungen sollen in eine Untermenge des Duration Calculus, die DC Implementables ([19]) oder eine Erweiterung davon transformiert werden. Dabei sind die kontinuierlichen Variablen (z.B. der aktuelle Abstand des Rollstuhls zum nächsten Hindernis) geeignet zu diskretisieren.
- Der nicht-hybride Anteil der Spezifikation in DC Implementables soll nach Timed-CSP übersetzt werden.
- Der letzte Transformationsschritt ist eine strukturelle Dekomposition in einen Standard-CSP und einen nur aus Timerprozessen bestehenden Teil. Im Unterschied zu Arbeitspaket AP 3, in dem es um die Spezifikation der Sicherheitsschicht des Bremer Autonomen Rollstuhls (also des Systems) geht, wird hier die Spezifikation der Sicherheitsanforderungen durchgeführt. Trotzdem sollten sich beide Arbeitspakete in diesem Punkt gut ergänzen können.

Schließlich läßt sich ein Fehlerbaum mit Hilfe des hier angestrebten Verfahrens als sequentieller Echtzeitprozeß interpretieren und in CSP [9] implementieren (solange man sich auf eine diskrete Modellierung der Zeit beschränkt). Dies bringt eine Reihe von Vorteilen mit sich: Es erlaubt die Verifikation von Sicherheitseigenschaften, indem das modellierte System im Model-Checker parallel zum implementierten Fehlerbaum betrieben wird [45]. Damit sind im Gegensatz zu [5] nicht nur Aussagen über die Konsistenz der Spezifikation (gegeben durch den Fehlerbaum) möglich. Vielmehr kann verifiziert werden, ob das modellierte System die Spezifikation erfüllt.

Da der Fehlerbaum eine formale Spezifikation des *ungewünschten* Systemverhaltens darstellt, läßt er sich zur automatischen Generierung und Auswertung von Testfällen verwenden (s.u. RT-Tester).

Das Arbeitspaket *Formale Bedrohungsanalyse und Sicherheitsmechanismen* gliedert sich damit zeitlich wie folgt:

Tätigkeit	Umfang in PM
Syntax und Semantikdefinition	4
Ableiten von Sicherheitsbedingungen	6
Spezifikation des ungewünschten Verhaltens	8
Fallstudie Rollstuhl	4
ΣArbeitspaket AP 1	22

Syntax und Semantikdefinition: Da weder die in [10] definierten grafischen Symbole zur Repräsentierung von Fehlerbäumen noch die in [44] verwendete textuelle Darstellung mächtig genug sind, um die oben beschriebenen Eigenschaften spezifizieren zu können, muß eine einheitliche Syntax gefunden werden, mit der sich sowohl die Basisbedrohungen als auch die Abhängigkeiten verschiedener Knoten des Fehlerbaums untereinander modellieren lassen. Dabei sollen sowohl zeitbehaftete als auch hybride Parameter modelliert werden können. Für die neu eingeführten Operatoren (Abhängigkeiten der Knoten untereinander) muß eine Semantik definiert werden, die Aussagen über die zeitlichen Abhängigkeiten von Fehlerbäumen zuläßt. Idealerweise sollte für jeden Knoten im Fehlerbaum eine Zeit bestimmt werden können, während der er „sensitiv“ ist für eine Eingabe unterer Knoten oder Blätter.

Daher bietet sich hier der Duration Calculus bzw. der erweiterte Duration Calculus an. Es wäre besonders elegant, wenn sich der Fehlerbaum werkzeunterstützt erstellen ließe.

Ableiten von Sicherheitsbedingungen: Der Nachweis der Vollständigkeit der Sicherheitsbedingungen ist zu erbringen. Dieser Schritt soll mit Isabelle/DC umgesetzt werden.

Spezifikation des ungewünschten Verhaltens: Es sind die oben aufgezählten Transformationsschritte durchzuführen. Dabei ist zu untersuchen, ob und wenn ja wie die Untermenge DC Implementables erweitert werden muß, um die hier benötigte Sprachmächtigkeit zu liefern.

Am Ende steht die Spezifikation des ungewünschten Verhaltens in Timed-CSP oder in CSP plus Timerprozessen. Sie ist die Basis für das automatisierte Testen (Arbeitspaket AP 4).

Fallstudie Bremer Autonomer Rollstuhl: Das Sensorik-Aktorik Modul des Bremer Autonomen Rollstuhls soll einer formalen Bedrohungsanalyse der hier vorgeschlagenen Art unterzogen werden. Dabei ist insbesondere die Berücksichtigung der harten Echtzeitanforderungen wie auch die Integration der hybriden Parameter (Geschwindigkeit, Distanz zu Hindernissen etc.) zu realisieren. Das Ergebnis dieser Bedrohungsanalyse ist die Spezifikation des *ungewünschten* Systemverhaltens und dient im weiteren VVT-Prozeß als Grundlage für die Testphase.

3.2.2 Gemeinsame Steuerung durch Mensch und Maschine (AP 2)

Sicherheitsentwürfe herkömmlicher eingebetteter Systeme basieren auf der Grundannahme, daß ein Sicherheits-Controller ins System zu integrieren ist, der über Eingabekanäle die relevanten Statusinformation des gesamten Systems erfährt und über Ausgabekanäle in der Lage ist, steuernden Einfluß auf das System zu nehmen [24]. Auf jedem Kanal können Daten eines bestimmten Datentyps kommuniziert werden (z.B. Temperaturen zwischen -40 und $+40^{\circ}\text{C}$). Aufgrund physikalischer Gesetze können sich aufeinanderfolgende Werte auf einem Kanal oder voneinander abhängige Werte auf verschiedenen Kanälen nicht beliebig verändern (z.B. Beschleunigung auf Höchstgeschwindigkeit in minimaler

Zeit, Position im Raum). Die Beschreibung dieser Einschränkungen (physikalische Gesetze, Hypothesen über die Beschaffenheit der Umgebung etc.) fließt in das sogenannte *physikalische Modell* ein.

Werden allerdings zum Beispiel die Vorgaben des Benutzers über Sollgeschwindigkeit und –lenkwinkel oder, allgemein gesprochen, die Kommandos eines menschlichen Systemoperators als Eingabekanäle eines Sicherheits–Controllers betrachtet, so können deutlich weniger einschränkende Annahmen getroffen werden.

Abstrakt gesehen wählt der Mensch seine Kommandos vollkommen nichtdeterministisch aus einer endlichen Menge an Befehlen aus. Es gilt zu klären, inwieweit eine formale Beschreibung des mentalen Modells des Menschen vom technischen System einschränkende Aussagen zuläßt. Dies ist insbesondere im Rahmen einer Risikoanalyse von großem Interesse, da hier nicht Worst–Case Betrachtungen im Vordergrund stehen, sondern quantitative oder evtl. auch qualitative Abschätzungen von Fehlerwahrscheinlichkeiten. In diesem Zusammenhang ist die Theorie der präferierten mentalen Modelle von Interesse (z.B. [12]).

Neben diesem Modellierungsproblem soll untersucht werden, welche prinzipiellen Auswirkungen auf den Sicherheitsentwurf der Einfluß des Menschen auf die Steuerung des Systems hat. Diese sind voraussichtlich dann ziemlich gering, wenn eine eindeutige Hierarchie der „Steuerberechtigten“ derart besteht, daß im Zweifelsfall das technische System den Benutzer überstimmen darf. Falls es jedoch wechselnde Prioritäten gibt, so daß in gewissen Situationen der Benutzer die alleinige Kontrolle über das Fahrzeug hat, so ergeben sich zusätzliche Konflikte, die im Sicherheitsentwurf berücksichtigt werden müssen [17, 16]. Beispielfhaft soll hier die Fehlerklasse „Einschränkung der Entscheidungsfreiheit des Steuernden“ erwähnt werden. Durch eine gemeinsame Steuerung von Mensch und Maschine, bei der das technische System den Menschen überwacht und evtl. korrigiert, werden neue Bedrohungen geschaffen, während die eingeführte Automatisierung bestehende Gefahren abgewendet hat. So ist es beispielsweise in einigen Situationen denkbar, daß der Benutzer daran gehindert wird, bestimmte Manöver auszuführen, obwohl er sich sicher ist, daß diese ungefährlich sind. Derartige Konflikte müssen systematisch erkannt werden können, um sie beim Entwurf der Automatisierung auszuschließen. Daher ist die Bedrohungsanalysemethode um Verfahren zu erweitern, die die Berücksichtigung der durch die gemeinsame Steuerung induzierten Fehlerklassen erlauben.

Das Ziel dieses Arbeitspakets ist die Entwicklung eines Verfahrens, das es ermöglicht, die Spezifikation des technischen Systems und die formale Beschreibung des Verhaltens des menschlichen Bedieners auf Konfliktpotential hin zu untersuchen.

Es bleibt anzumerken, daß dieses Arbeitspaket eine große Herausforderung darstellt, die insbesondere darin besteht, das menschliche Verhalten und das Verhalten einer Maschine mit einem gemeinsamen formalen Beschreibungsmechanismus zu modellieren. Es ist zu diesem Zeitpunkt noch nicht abzusehen, in welchem Maß dieser Ansatz erfolgreich sein wird. Falls das beschriebene Vorhaben jedoch vielversprechende Ergebnisse liefern sollte, so wäre es sehr leicht auf eine große Klasse von Systemen mit gemeinsamer Steuerung wie Flugzeuge, Autos etc. übertragbar.

Tätigkeit	Umfang in PM
Auswahl einer geeigneten Spezifikations- sprache	4
Einbettung in Bedrohungsanalyse	6
Fallstudie Rollstuhl	4
Σ Arbeitspaket AP 2	14

Auswahl einer Spezifikations-*sprache*: Im Hinblick auf die Einbettung der durch die gemeinsame Steuerung induzierten Probleme in die Bedrohungsanalyse ist die formale Methode zur Spezifikation der Bedrohungen zu überprüfen und ggf. zu erweitern. Die Voraussetzung dafür ist ein guter Überblick über den Stand der kognitionswissenschaftlichen Forschung im Bereich der Mentalen Modelle, Kognitiven Prozesse und dem sogenannten *Mode Confusion* Problem.

Einbettung in Bedrohungsanalyse: Die automatische Erkennung von Konfliktpotential in der Systemspezifikation wird durch eine erweiterte Bedrohungsanalyse sichergestellt. Dabei werden die Spezifikation des technischen Systems und die Modellierung des menschlichen Benutzers auf mögliches Konfliktpotential untersucht.

Fallstudie Rollstuhl: Am Beispiel des Sensorik–Aktorik Moduls des Bremer Autonomen Rollstuhls, das die vom Fahrer abgesetzten Fahrkommandos je nach Hindernissituation korrigiert, soll das entwickelte Konzept in der Praxis überprüft werden. Diese Fallstudie ist exemplarisch für eine Vielzahl von Anwendungsgebieten, bei denen das technische System eine Kontrollfunktion über den bedienenden Menschen ausübt, wie z.B. in der Luftfahrt oder dem Eisenbahnbereich.

3.2.3 Spezifikation des Kommunikationsverhaltens der Sicherheits- schicht des Bremer Autonomen Rollstuhls (AP 3)

Der Sicherheitsrechner unterliegt harten Echtzeitanforderungen, d.h. es müssen obere Zeitschranken für die Dauer eines Bearbeitungszyklus der Sicherheits-schicht garantiert werden. Diese Zeitschranken sind zudem sehr niedrig, da schon bei verhältnismäßig geringen Geschwindigkeiten des Rollstuhls von unter 1m/s sehr schnell (in weniger als 32 ms) auf neue Informationen über die Beschaffenheit der Umgebung reagiert werden muß, um ein sicheres Manövrieren zu ermöglichen.

Die relevanten Daten, mit denen die Sicherheitsschicht umgeht, sind hybrider Natur: Während eines rollstuhl-internen Zeittaktes ändern sich die Geschwindigkeit, der Lenkwinkel, die Position im Raum und der Abstand zu Hindernissen nicht in diskreten Schritten sondern kontinuierlich über die Zeit.

Die Betrachtung dieser „Robotikaspekte“ ist *nicht* Gegenstand dieses Arbeitspakets, sie werden unabhängig von dem hier beantragten Projekt in einem beantragten Vorhaben „Abstraktion und Wiederverwendung von formalen Programm-entwicklungen“ untersucht werden.

Hier geht es um das reine Kommunikationsverhalten der Sicherheitsschicht des Rollstuhls. Es ist die Basis für die Einhaltung der in Arbeitspaket AP 1 definierten Sicherheitsbedingungen. Unter „Kommunikationsverhalten“ soll dabei nicht

nur die Kommunikation zwischen Anwendungsmodulen (z.B. Planungsalgorithmus, Selbstlokalisierung) über das echtzeitfähige Netzwerk verstanden werden, sondern insbesondere auch die Kommunikation mit den Sensoren und den Aktoren (vgl. Abb. 3).

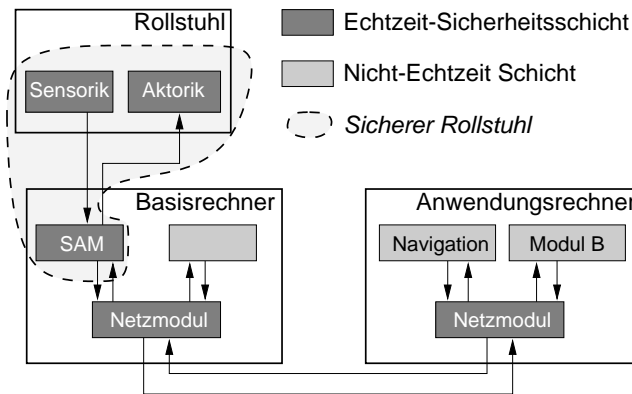


Abbildung 3: Systemarchitektur. Das Sensorik-Aktorik-Modul (SAM) stellt eine „sichere“ Schnittstelle zum eigentlichen Rollstuhl zur Verfügung.

Die Spezifikation des Kommunikationsverhaltens soll in Timed-CSP vorgenommen werden. Da es lediglich auf die Definition oberer Zeitschranken in diskreten Einheiten ankommt, reicht dies an dieser Stelle aus.

Als Beispiel sei eine wichtige Eigenschaft der Ultraschallsensoren hier skizziert. Um das rechtzeitige Anhalten vor Hindernissen gewährleisten zu können, muß eine gewisse Abtastrate der Umgebung sichergestellt sein. Dies läßt sich z.B. erreichen, indem gezeigt wird, daß für jeden Sensor niemals mehr als t Zeiteinheiten zwischen zwei Messungen vergehen.

Nach der Identifikation der Kommunikationskanäle und der Spezifikation des Kommunikationsverhaltens auf diesen Kanälen in Timed-CSP soll mit den für die UniForM Workbench derzeit entwickelten Transformationen eine *Architekturtransformation* in Anteile ohne Zeit (spezifiziert in Standard-CSP) und in Timer-Prozesse durchgeführt werden. Dieser Schritt ist schon aus Arbeitspaket AP 1 bekannt, mit dem Unterschied, daß dort keine Spezifikation des implementierten Systems betrachtet wurde, sondern die Spezifikation der Sicherheitsbedingungen.

Die Architekturtransformation hat das Ziel, eine weiterverwendbare Spezifikation in der um Timer-Prozesse erweiterten CSP-Variante zu liefern, die einerseits zum Model-Checken der kommunikationsspezifischen Sicherheitsbedingungen und andererseits für weitere Verifikationsansätze (s.u.) verwendet werden kann [53].

Als weiterer Teil dieses Arbeitspakets soll der von Buth et al. entwickelte Ansatz [30, 31] erweitert werden. Es handelt sich um eine Verifikation der Freiheit von Verklemmungen (*deadlocks* und *livelocks*) durch Abstraktion des relevanten Kommunikationsverhaltens mit Hilfe von Transformationen.

Tätigkeit	Umfang in PM
Sicherheitsschicht \longleftrightarrow Sensorik, Aktorik	3
Kommunikation zwischen den SW-Modulen via Netzwerk	3
Integration: Kommunikationsverhalten Sicherheitsschicht	3
Architekturtransformation	6
Livelock- und Deadlock-Analyse	6
Σ Arbeitspaket AP 3	21

Sicherheitsschicht \longleftrightarrow Sensorik, Aktorik: Das Kommunikationsverhalten zwischen der Sicherheitsschicht und der Sensorik (Entfernungssensoren, Geschwindigkeits- und Lenkwinkelmessung) sowie der Aktorik (Sollgeschwindigkeit und -lenkwinkel) soll in Timed-CSP spezifiziert werden.

Kommunikation zwischen den SW-Modulen via Netzwerk: Das echtzeitfähige Netzwerkprotokoll, das der Kommunikation innerhalb der Sicherheitsschicht sowie der Anbindung nach außen (z.B. zu Navigationsalgorithmen) zugrunde liegt, wird in Timed-CSP spezifiziert.

Integration: Kommunikationsverhalten Sicherheitsschicht: Die beiden Teilspezifikationen sollen integriert werden zu einer vollständigen Spezifikation des Kommunikationsverhalten der Sicherheitsschicht.

Architekturtransformation: Die Timed-CSP Spezifikation des Kommunikationsverhalten der Sicherheitsschicht wird dekomponiert in einen Standard-CSP Teil und einen nur aus Timerprozessen bestehenden Teil.

Livelock- und Deadlock-Analyse: Es soll eine Abhängigkeitsanalyse [31] der als abstrakte, auf ihr Kommunikationsverhalten reduzierte CSP-Prozesse repräsentierten Komponenten der Sicherheitsschicht durchgeführt werden.

3.2.4 Validation und Test (AP 4)

Die in der UniForM Workbench vorhandenen Methoden und Werkzeuge für Validation und Test basierend auf formalen Spezifikationen (mit automatischer Auswertung) sollen für das Gesamtsystem (einschließlich der nicht formal entwickelten Anteile) eingesetzt werden, ebenso die Simulationsumgebung SimRobot (vgl. Abb. 4). Dabei übernimmt das Testwerkzeug RT-Tester erstens die Rolle eines „nichtdeterministischen Fahrers“, zweitens die Rolle des Sensorikinterfaces und drittens auch noch die Rolle der Steuereinheit der Simulationsumgebung. Während die Simulationsumgebung ideale Sensordaten an RT-Tester liefert, entscheidet das Tool entsprechend der formalen Spezifikation der Umwelt, ob es die korrekten Werte verfälscht oder nicht. So werden reale Sensoren modelliert. RT-Tester kann dann das von der Rollstuhl-Sicherheitsschicht generierte Aktorikkommando auf seine sichere Ausführbarkeit überprüfen. Zusätzlich erzeugt RT-Tester als „Anwendungsmodul“ Fahrkommandos, die der Sicherheitsschicht des Rollstuhls übergeben werden. Mit Hilfe dieser Architektur können durch RT-Tester automatisch generierte Testfälle in einer durch den

Simulator SimRobot modellierten Umgebung virtuell abgespielt werden, womit sich eine wesentlich höhere Testabdeckung erreichen läßt als es in einer realen Umgebung jemals möglich wäre.

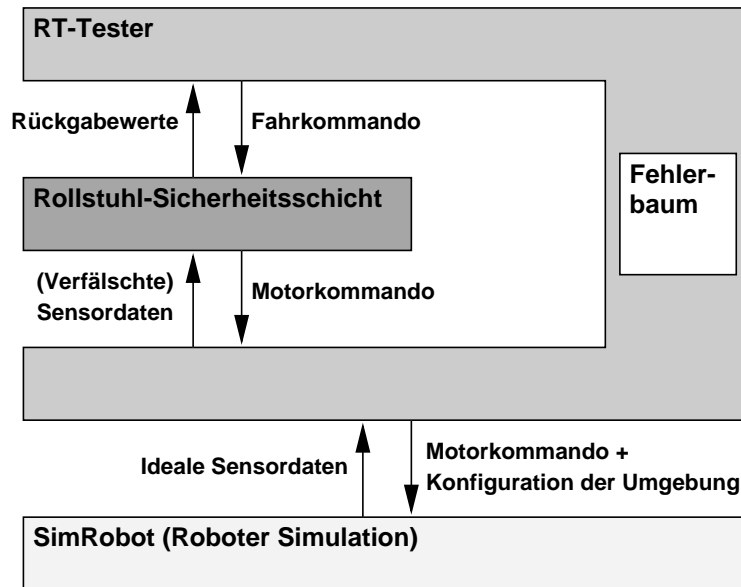


Abbildung 4: Testszenario unter Einsatz des Testtools RT-Tester und dem Robotersimulator SimRobot.

Da sich andererseits die reale Umwelt nicht exakt simulieren läßt, zeigen diese Tests „nur“, daß das verwendete System unter den gewählten Simulationsbedingungen korrekt arbeitet. Diese basieren auf den Hypothesen, die aus der formalen Bedrohungsanalyse abgeleitet werden konnten, wie z.B. *Alle Objekte in der Umwelt haben ihre größte horizontale Ausdehnung in der Höhe der Sensoren des Rollstuhls*. Trotzdem ist das automatisierte Testen ein wichtiger Beitrag zum Nachweis der Sicherheitseigenschaften des Systems, da die Simulation über RT-Tester so einstellbar ist, daß eine möglichst „böswillige“ Umgebung modelliert wird. Dabei ist zu beachten, daß „böswillig“ nicht eine völlige Willkür bedeutet, sondern als „nichtdeterministisch im Rahmen der gegebenen Hypothesen“ zu verstehen ist. Wenn sich jedoch die Umwelt in der Realität entsprechend der aufgestellten Hypothesen verhält, so wird das reale System spezifikationsgemäß arbeiten.

Tätigkeit	Umfang in PM
Anpassungen	1
Modellierung der Umwelt	10
Durchführung der Tests, Analyse der Ergebnisse	4
Σ Arbeitspaket AP 4	15

Anpassungen: Einarbeitung in das Tool RT-Tester. Nötige softwaretechnische Anpassungen bei SimRobot, dem Sensorik/Aktorik-Modul und RT-Tester.

Modellierung der Umwelt: Erstellung einer vollständigen Spezifikation des Verhaltens der Umwelt in CSP. Als Arbeitsgrundlage dient hier die in Arbeitspaket AP 3 erstellte Timed-CSP-Spezifikation, die bereits marginale Ansätze einer Spezifikation der Umwelt enthält. Diese müssen hier deutlich vertieft (Verhalten von Hindernissen etc.) und nach CSP übertragen werden.

Durchführung der Tests: Möglicherweise aufgedeckte Fehler müssen beseitigt werden. Analyse der Ergebnisse.

3.2.5 Zeitlicher Ablauf

In Abbildung 5 ist der zeitliche Ablauf des Arbeitsprogramms und die einzelnen Abhängigkeiten der Arbeitspakete dargestellt.

4 Voraussetzungen für die Durchführung des Vorhabens

4.1 Zusammensetzung der Arbeitsgruppe

Im Rahmen des hier beantragten Projekts wird es eine enge Kooperation mit einigen Mitarbeitern der *AG BKB/Formalen Methoden und Werkzeuge*, *AG BKB/Kognitive Robotik* sowie der *AG BS/Betriebssysteme und Verteilte Systeme* geben.

Folgende Mitarbeiter sind mit einem Teil ihrer Arbeitszeit involviert:

- Dr. Bettina Buth (Wiss. Assistentin)
- Dr. Shi Hui (Wiss. Assistentin)
- Dr. Christoph Lüth (Wiss. Assistent)
- Dr. Holger Schlingloff (Wiss. Mitarbeiter)
- Dipl.-Inf. Oliver Meyer (Wiss. Mitarbeiter)

Es besteht ein Querbezug zum DFG-Schwerpunktprogramm „Raumkognition“ (siehe auch nächsten Abschnitt). Die dort beschäftigten Mitarbeiter

- Dr. Thomas Röfer (Wiss. Assistent)
- Dipl.-Inf. Axel Lankenau (Wiss. Mitarbeiter)
- Dipl.-Ing. (BA) Dipl.-Phys. Rolf Müller (Wiss. Mitarbeiter)

werden dieses Vorhaben unterstützen.

Gleiches gilt für Dipl.-Inf. Stefan Bisanz, der im DFG-Schwerpunktprogramm „Integration von Techniken der Softwarespezifikation für ingenieurwissenschaftliche Anwendungen“ beschäftigt ist.

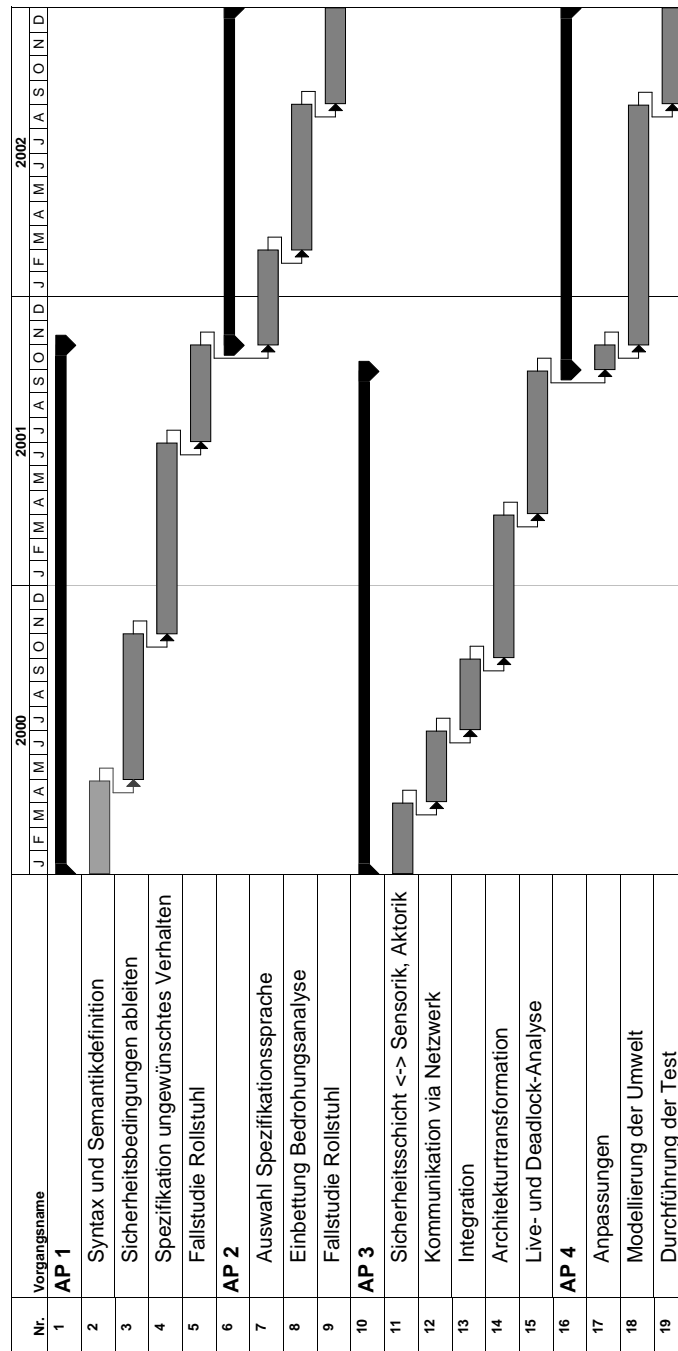


Abbildung 5: Zeitliche Abfolge der einzelnen (Teil-)Arbeitspakete.

4.2 Zusammenarbeit mit anderen Wissenschaftlern

Wie bereits angedeutet, besteht eine enge Zusammenarbeit mit Wissenschaftlern innerhalb des DFG-Schwerpunktprogramms Raumkognition. Während die hier

beantragten Arbeitspakete AP 1, AP 3 und AP 4 in diesem Zusammenhang nur von geringer Bedeutung sind, besteht großes Interesse am Arbeitspaket AP 2 (Formalisierung durch gemeinsame Steuerung induzierter Bedrohungen).

4.3 Auslandsbezug

Die das Model-Checking betreffenden Aspekte der Arbeitspakete AP 1 und AP 3 sollen mit Hilfe von Prof. Dr. Ed Clarke, Carnegie Mellon University, Pittsburgh (USA) sowie von Prof. Dr. Anders Ravn und Prof. Dr. Kim Larsen, Universität Aalborg (Dänemark, Model-Checking Werkzeug Uppaal) untersucht werden.

Dr. Ulrich Nehmzow, Universität Manchester (GB) wird dieses Vorhaben im Rahmen der Arbeitspakete AP 2 (hier: Mode-Confusion Problem) und AP 4 (Modellierung von Roboterverhalten in dynamischen Umgebungen) unterstützen.

4.4 Apparative Ausstattung

Wie bereits erwähnt, stehen der Rollstuhl *Rolland* als Plattform für die Fallstudien sowie das Rechnernetz der Arbeitsgruppen für Dienste wie Drucken und Datensicherung zur Verfügung.

4.5 Laufende Mittel für Sachausgaben

Es stehen Haushaltsmittel für die üblichen laufenden Sachausgaben zur Verfügung.

4.6 Sonstige Voraussetzungen

Keine

5 Erklärungen

5.1 Finanzierung des Vorhabens bei einer anderen Stelle

Ein Antrag auf Finanzierung dieses Vorhabens wurde bei keiner anderen Stelle eingereicht. Wenn wir einen solchen Antrag stellen, werden wir die Deutsche Forschungsgemeinschaft unverzüglich benachrichtigen.

5.2 Vertrauensdozent

Der DFG-Vertrauensdozent der Universität Bremen wird von diesem Antrag informiert.

Literatur

Stand der Forschung

- [1] Bell, D. A., Levine, S. P., Koren, Y., Jaros, L. A., Borenstein, J. (1994). *Design Criteria for Obstacle Avoidance in a Shared Control System*. In: Proceedings of the RESNA '94 Conference. Nashville. 17–24.
- [2] Borenstein, J. und Koren, Y. (1990). *Tele-autonomous Guidance for Mobile Robots*. In: IEEE Transactions on Systems, Man, and Cybernetics. Special issue on unmanned systems and vehicles. Vol. 20. No. 6, Nov/Dez 1990. 1437–1443.
- [3] Chaochen, Z., Hoare, C.A.R., Ravn, A. (1991). *A calculus of durations*. In: Information Processing Letters 40. Elsevier. 269–276
- [4] Clarke, E. und Wing, J. (1996). *Formal Methods: State of the Art and Future Directions*. CMU Computer Science Technical Report. CMU-CS-96-178.
- [5] Górski, J. und Wardziński, A. (1997). *Timing Aspects of Fault Tree Analysis of Safety Critical Systems*. In: Safer Systems — Proceedings of the Fifth Safety-critical Systems Symposium. Springer
- [6] Hamilton, D. L., Visinsky, M. L., Bennett, J. K., Cavallaro, J. R., Walker, I. D. (1994). *Fault-Tolerant Algorithms and Architectures for Robotics*. In: Proceedings of 1994 IEEE Mediterranean Electrotechnical Conference. Antalya, Turkey. 1034–1036.
- [7] Hansen, K. M. (1996). *Linking Safety Analysis to Safety Requirements – Exemplified by Railway Interlocking Systems*. PhD Thesis. Technical University of Denmark.
- [8] Heilmann S. (1999). *Proof Support for Duration Calculus*. Dissertation. Department of Information Technology. Technical University of Denmark.
- [9] Hoare, C. A. R. (1985). *Communication Sequential Processes*. International series in computer science. Prentice Hall
- [10] IEC (1990). *International Standard IEC 1025 Fault Tree Analysis*. International Electrotechnical Commission. Genf
- [11] Kazerooni, H. (1998). *Human Power Extender: An Example of Human-Machine Interaction via the Transfer of Power and Information Signals*. In: Proceedings of AMC 98 – 5th International Workshop on Advanced Motion Control, Coimbra University Portugal. 565–572.
- [12] Knauff, M., Rauh, R., Schlieder, C. (1995). *Preferred Mental Models in Qualitative Spatial Reasoning: A Cognitive Assessment of Allen's Calculus*. In: Proc. 17. Annual Conf. of the Cognitive Science Society.
- [13] Kotonya, G. und Sommerville, I. (1997). *Integrating Safety Analysis and Requirements Engineering*. In: Proceedings of Joint Asia Pacific Software Engineering Conference (APSEC) and International Computer Science Conference (ICSC) 1997. Hongkong. 259–270.

- [14] Laprie, J. C., ed. (1992). *Dependability: Basic Concepts and Terminology*. Springer-Verlag. Berlin, Heidelberg, New York.
- [15] Leuschen, M. (1997). *Robot Reliability Through Fuzzy Markov Models*. MSc Thesis. Rice University. Texas. USA.
- [16] Leveson, N. G. und Palmer, E. (1997). *Designing Automation to Reduce Operator Errors*. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics. Orlando, Florida, USA.
- [17] Leveson, N. G., Pinnel, L. D., Sandys, S. D., Koga, S., Reese, J. D. (1997). Analyzing Software Specifications for Mode Confusion Potential. In: Proceedings of the Workshop on Human Error and System Development. Glasgow, Schottland.
- [18] Pires, G., Araújo, R., Nunes, U., de Almeida, A.T. (1998). *ROBCHAIR – A Powered Wheelchair Using a Behaviour-Based Navigation*. In: Proceedings of AMC 98 – 5th International Workshop on Advanced Motion Control, Coimbra University Portugal. 536–541.
- [19] Schenke, M. und Olderog, E.-R. (1999). *Transformational design of real-time systems. Part I: From requirements to program specifications ?* In: Acta Informatica 36. Seiten 1–65.
- [20] Seward, D. W., Bradley, A., Margrave, F. W. (1994). *Hazard Analysis Techniques for Mobile Construction Robots*. In: Proceedings of the 11th Int. Symp. on Robotics in Construction. Brighton, England. 35–42.
- [21] Seward, D. W., Margrave, F. W., Sommerville, I., Kotonya, G. (1995). *Safe Systems for Mobile Robots – The Safe-SAM project. Achievement and Assurance of Safety*. In: Proceedings of the Third Safety-Critical Systems Symposium, Brighton, England. Veröffentlicht: Springer Verlag. ISBN 3 540 19922 5. 153–170.
- [22] Seward, D. W., Margrave, F. W., Sommerville, I., Morrey, R. (1996). *LU-CIE the Robot Excavator – Design for System Safety*. In: Proceedings of IEEE International Conference on Robotics and Automation. Minneapolis. USA
- [23] Sommerville, I., Seward, D., Morrey, R., Quayle, S. (1997). *Safe Systems Architectures for Autonomous Robots*. In: Proceedings of the Third Safety-Critical Systems Symposium, Brighton, England. Veröffentlicht: Springer Verlag. ISBN 3 540 19922 5.
- [24] Storey, N. (1996). *Safety-Critical Computer Systems*. Addison-Wesley. ISBN 0-201-42787-7.
- [25] Veseley, W. et al. (1981). *Fault Tree Handbook*. US Nuclear Regulatory Commission, NUREG-0492. Washington, DC.
- [26] Zhang, Y. (1994). *A Foundation for the Design and Analysis of Robotic Systems and Behaviors*. PhD Thesis. University of British Columbia, Kanada.

Eigene Vorarbeiten

- [27] Astesiano, E., Bidoit, M., Krieg-Brückner, B., Kirchner, H., Mosses, P., Sannella, D. und Tarlecki, A. *CASL – the Common Algebraic Specification Language (Draft)*. Theoretical Computer Science, Elsevier. (Eingeladener Beitrag).
- [28] Amthor, P. und Dick, S. (1997). *Test eines Bordcomputers für ein dezentrales Zugsteuerungssystem unter Verwendung des Werkzeuges VVT-RT*. 7. Kolloquium Software-Entwicklung Methoden, Werkzeuge, Erfahrungen: Mächtigkeit der Software und ihre Beherrschung, Technische Akademie Esslingen.
- [29] Blank Purper, C. und Westmeier, S. (1998). *A Graphical Development Process Assistant for Formal Methods*. In: Proc. VISUAL'98 (short papers), ETAPS'98, Lisbon. Siehe auch:
<http://www.informatik.uni-bremen.de/~uniform/gdpa>
- [30] Buth, B., Kouvaras, M., Peleska, J., Shi, H. (1997). *Deadlock Analysis for a Fault-Tolerant System*. In: Johnson, M. (ed.), Algebraic Methodology and Software Technology. AMAST 97. LNCS 1349. Springer. 60–75.
- [31] Buth, B., Peleska, J., Shi, H. (1999). *Combining Methods for the Livelock Analysis of a Fault-Tolerant System*. Algebraic Methodology and Software Technology. Proceedings of the 7th International Conference, AMAST'98. LNCS 1548. 124–139. Springer.
- [32] CoFI. The Common Framework Initiative for Algebraic Specification. Siehe auch: <http://www.brics.dk/Projects/CoFI>
- [33] Haxthausen, A. und Peleska, J. (1998). *Formal Development and Verification of a Distributed Railway Control System*. In: Proceedings of the 1st FMERail Workshop. Utrecht. Niederlande. Wird auch erscheinen als: World Congress on Formal Methods 1999. Toulouse, Frankreich. LNCS. Springer
- [34] Herwig, Ch. (1996). *Visual Motion Processing for Active Observers*. Ph.D. thesis. In: Krieg-Brückner, B., Roth, G., Schwegler, H. (Hrsg.): ZKW-Bericht 1/96. ISSN 0947-0204. Zentrum für Kognitionswissenschaften. Universität Bremen.
- [35] Hoffmann, B. und Krieg-Brückner, B. (eds.) (1993). *PROgram Development by Specification and Transformation, The PROSPECTRA Methodology, Language Family, and System*. LNCS 680. Springer. Siehe auch: <http://www.informatik.uni-bremen.de/~prospectra>
- [36] Karlsen, E. W. (1998). *The UniForM WorkBench – a Higher Order Tool Integration Framework*. In: International Workshop on Current Trends in Applied Formal Methods. LNCS. Springer (im Erscheinen).
- [37] Karlsen, E. W. (1998). *Tool Integration in a Functional Setting*. Dissertation. Universität Bremen (im Druck).

- [38] Kollmann, J., Lankenau, A., Bühlmeier, A., Krieg-Brückner, B., Röfer, T. (1997). *Navigation of a Kinematically Restricted Wheelchair by the Parti-Game Algorithm*. In: Spatial Reasoning in Mobile Robots and Animals, AISB-97 Workshop. Manchester University. ISSN 1361-6153. 35-45
- [39] Kolyang, Lüth, C., Meyer, T., Wolff, B. (1997). *TAS and IsaWin: Generic Interfaces for Transformational Program Development and Theorem Proving*. In Bidoit, M., Dauchet, M. (eds.): Theory and Practice of Software Development '97. LNCS 1214. Springer.
- [40] Krieg-Brückner, B. (1998). *UniForM Perspectives for Formal Methods*. In: FM-Trends — International Workshop on Current Trends in Applied Formal Methods. LNCS. Springer (im Erscheinen).
- [41] Krieg-Brückner, B., Peleska, J., Olderog, E.-R., Baer, A. (1999). *The UniForM Workbench, a Universal Development Environment for Formal Methods*. World Congress on Formal Methods 1999. Toulouse, Frankreich. LNCS. Springer (im Erscheinen).
- [42] Krieg-Brückner, B., Peleska, J., Olderog, E.-R., Balzer, D., Baer, A. (1996). *UniForM, Universal Formal Methods Workbench*. In: Grote, U., Wolf, G. (eds.): Statusseminar des BMBF: Softwaretechnologie. Deutsche Forschungsanstalt für Luft- und Raumfahrt, Berlin 337-356. Siehe auch <http://www.informatik.uni-bremen.de/~uniform>
- [43] Krieg-Brückner, B., Röfer, T., Carmesin, H.-O., Müller, R. (1998). *A Taxonomy of Spatial Knowledge for Navigation and its Application to the Bremen Autonomous Wheelchair*. Freksa, Ch., Habel, Ch., Wender, K. F. (Eds.): Spatial Cognition. Lecture Notes in Artificial Intelligence 1404. Springer. 373-397.
- [44] Lankenau, A. und Meyer, O. (1997). *Der autonome Rollstuhl als sicheres eingebettetes System*. Diplomarbeit. Universität Bremen
- [45] Lankenau, A. und Meyer, O. (1999). *Formal Methods in Robotics: Fault Tree Based Verification*. In: Proceedings of Quality Week Europe 1999, Brüssel, Belgien. (im Erscheinen)
- [46] Lankenau, A., Meyer O., Krieg-Brückner, B. (1998). *Safety in Robotics: The Bremen Autonomous Wheelchair*. In: Proceedings of AMC'98 – Coimbra, 5th Int. Workshop on Advanced Motion Control, Coimbra, Portugal. ISBN 0-7803-4484-7. 524-529.
- [47] Lankenau, A. und Röfer, T. (1998). *Architecture of the Bremen Autonomous Wheelchair*. In: Report – Situierete Künstliche Kommunikatoren, SFB 360, Report 98/13. Universität Bielefeld. ISSN 0946-7572
- [48] Lüth, C., Karlsen, E. W., Kolyang, Westmeier, S., Wolff, B. (1998). *HOL-Z in the UniForM WorkBench – a Case Study in Tool Integration for Z*. In: Proc. ZUM'98, 11th Int. Conference of Z Users. LNCS 1493, Springer. 116-134.
- [49] Mossakowski, T. und Roggenbach, M. (1999). *The Datatypes REAL and COMPLEX in CASL*. COFI-Note M-7 in [32].

- [50] Peleska, J. (1996). *Formal Methods and the Development of Dependable Systems*. Bericht 1/96, Universität Bremen, Fachbereich Mathematik und Informatik. Siehe auch:
<http://www.informatik.uni-bremen.de/~jp/papers/depend.ps.gz>
- [51] Peleska, J. und Siegel, M. (1996). *From Testing Theory to Test Driver Implementation*. In: M.-C. Gaudel, J. Woodcock (Eds.): FME'96: Industrial Benefit and Advances in Formal Methods. LNCS 1051. Springer, pp. 538–556.
- [52] Peleska, J. und Siegel, M. (1997). *Test Automation of Safety-Critical Reactive Systems*. South African Computer Journal 19, pp. 53–77.
- [53] Peleska, J. (1998). *Testing Reactive Real-Time Systems*. Tutorial bei der School on Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT) '98. Denmark Technical University, Lyngby.
- [54] Peleska, J., Bisanz, S., Fiss, I. und Endress, M. (1999). *Non-Standard Graphical Simulation Techniques for Test Specification Development*. In: H. Szczerbicka (Ed.): Proceedings of the 13th European Simulation Multiconference Volume 1. Warschau. Polen. 575-580.
- [55] Röfer, T. (1995). *Controlling a robot with image based homing*. In: Krieg-Brückner, B., Herwig, Ch. (Hrsg.): Tagungsband des Workshops „Kognitive Robotik“, ZKW Bericht Nr. 3/95. Universität Bremen.
- [56] Röfer, T. (1995). *Image based homing using a self-organizing feature map*. In: Fogelman-Soulie, F., Gallinari, P. (Eds.): Proc. Int. Conf. Artificial Neural Networks. EC2 & Cie. Vol. 1. 475–480.
- [57] Röfer, T. (1995). *Bildbasierte Navigation mit eindimensionalen 360°-Bildern*. In: Dillmann, R., Rembold, U., Lüth, T. (Hrsg.): Autonome Mobile Systeme 1995. Springer. 193–202.
- [58] Röfer, T. (1997). *Controlling a Wheelchair with Image-based Homing*. Spatial Reasoning in Mobile Robots and Animals, AISB-97 Workshop. Manchester University. 66–75.
- [59] Röfer, T. (1997). *Routemark-Based Navigation of a Wheelchair*. In: Proc. Third ECPD International Conference on Advanced Robotics, Intelligent Automation and Active Systems, Bremen. 333–338.
- [60] Röfer, T. (1998). *Panoramic Image Processing and Route Navigation*. Dissertation. Monographs of the Bremen Institute of Safe Systems 7, Aachen, Shaker.
- [61] Röfer, T. (1998). *Strategies for Using a Simulation in the Development of the Bremen Autonomous Wheelchair*. In: Zobel, R., Moeller, Dietmar (Eds.): Simulation – Past, Present and Future. Society for Computer Simulation International. 460–464.
- [62] Röfer, T. (1998). *Routenbeschreibung durch Odometrie-Scans*. In: Wörn, H., Dillmann, R., Henrich, D. (Eds.): Autonome Mobile Systeme 1998. Informatik aktuell. Springer. 122-129.

- [63] Röfer, T. und Lankenau, A. (1998). Architecture and Applications of the Bremen Autonomous Wheelchair. In: Wang, P. P. (Ed.): Proc. of the Fourth Joint Conference on Information Systems 1. Association for Intelligent Machinery. 365-368.
- [64] Röfer, T. und Müller, R. (1998). *Navigation and Routemark Detection of the Bremen Autonomous Wheelchair*. In: Lüth, T., Dillmann, R., Dario, P., Wörn, H. (Eds.): Distributed Autonomous Robotics Systems. Springer. 183-192.
- [65] Schrönen, M. (1998). *Methodology for the Development of Micro-Processor Based Safety-Critical Systems*. Dissertation. Monographs of the Bremen Institute of Safe Systems 8, Aachen, Shaker (im Druck).
- [66] Siems, U., Herwig, C., Röfer, T. (1994). *SimRobot, ein System zur Simulation sensorbestückter Agenten in einer dreidimensionalen Umwelt*. Krieg-Brückner, Roth, Schwegler (Hrsg.): ZKW Bericht Nr. 1/94. Universität Bremen.