

Konzeption und Evaluation  
eines hybriden, skalierbaren  
Werkzeugs zur mechatronischen  
Systemdiagnose am Beispiel  
eines Diagnosesystems für freie  
Kfz-Werkstätten

Nghia Duc Dang

**Konzeption und Evaluation eines hybriden,  
skalierbaren Werkzeugs zur mechatronischen  
Systemdiagnose am Beispiel eines  
Diagnosesystems für freie Kfz-Werkstätten**

zur Erlangung des akademischen Grades

**Doktor der Naturwissenschaft**

Von der Fakultät für Mathematik und Informatik,  
Lehrstuhl für Künstliche Intelligenz und Angewandte Informatik  
der Universität Würzburg

genehmigte

**Dissertation**

von

**Nghia Duc Dang**

aus Hanoi

Tag der Disputation:	20. April 2012
Erster Gutachter:	Prof. Dr. rer. nat. Frank Puppe
Zweiter Gutachter:	Professor Dr. rer. nat. Klaus-Dieter Althoff

# Danksagung

Die vorliegende Arbeit entstand während meiner dreijährigen Tätigkeit als Doktorand sowie zwei weiteren Jahren als Mitarbeiter in dem Zentralbereich Forschung und Voraentwicklung der Robert Bosch GmbH.

An dieser Stelle möchte ich mich bei Herrn Prof. Dr. Frank Puppe herzlich für die Unterstützung bedanken, die er mir zukommen ließ, und für die Freiheit, die er mir während der Forschungsarbeit sowohl in wissenschaftlicher als auch in zeitlicher Hinsicht gewährte. Darüber hinaus gilt mein Dank allen Mitgliedern der Prüfungskommission, insbesondere Herrn Prof. Dr. Klaus-Dieter Althoff für sein Interesse an meiner Arbeit und für die Übernahme des Co-Referats.

Obwohl eine Dissertation die Befähigung eines Einzelnen zu selbständiger wissenschaftlicher Arbeit unter Beweis stellen soll, wäre diese Arbeit nicht ohne die Mitwirkung von Kollegen und Freunden zustande gekommen: Neben anderen danke ich daher insbesondere Dr. Gerrit de Boer, der mir während unserer engen Zusammenarbeit immer wieder wertvolle Denkanstöße gab, sowie Peter Engel, Alexander Opfermann, Dr. Anes Hodzic, Prof. Dr. Dirk Weidemann, Martin Schaedel, Manfred Kaul und Dieter Hagenlocher, mit denen ich viele fruchtbare Diskussionen führte.

Für die gute Zusammenarbeit und die anregenden Diskussionen während des ersten und zweiten Feldtests gilt mein Dank folgenden Werkstattmitarbeitern sowie Fachexperten: Rolf Endemann, Manfred Kärcher, Volker Kolonko, Uwe Steingass, Uzunkaya Hatec, Michael Dörrer, Daniel Tica, Florian Hiekel, Dominic Ferro, Sven Fisch, Steffen Seyfang, Frank Breganski, Lothar Nagel, Martin Walter, Daniel Gruber, Claus Schulz, Alexander Maier, Martin Schaedel, Dennis Behrendt, Dietrich Pagel, Alex Bader, Oliver Eitle und Thomas Wambera.

Nicht nur hierfür, aber besonders für die Hilfe bei der anstrengenden Arbeit des Korrekturlesens und der Herstellung der Kompatibilität zur deutschen Sprache danke ich Claudia Köhler sowie Susanne Kischitzki.

Meiner Familie danke ich für ihr Verständnis, in der Zeit der Anfertigung dieser Arbeit häufig nur geteilte Aufmerksamkeit erfahren zu haben.

Ludwigsburg, den 24 Oktober 2011

# Kurzfassung

Die Entwicklung eines wissensbasierten Systems, speziell eines Diagnosesystems, ist eine Teildisziplin der künstlichen Intelligenz und angewandten Informatik. Im Laufe der Forschung auf diesem Gebiet wurden verschiedene Lösungsansätze mit unterschiedlichem Erfolg bei der Anwendung in der Kraftfahrzeugdiagnose entwickelt. Diagnosesysteme in Vertragswerkstätten, das heißt in Fahrzeughersteller gebundenen Werkstätten, wenden hauptsächlich die fallbasierte Diagnostik an. Zum einen hält sich hier die Fahrzeugvielfalt in Grenzen und zum anderen besteht eine Meldepflicht bei neuen, nicht im System vorhandenen Fällen. Die freien Werkstätten verfügen nicht über eine solche Datenbank. Somit ist der fallbasierte Ansatz schwer umsetzbar. In freien Werkstätten - Fahrzeughersteller unabhängigen Werkstätten - basiert die Fehlersuche hauptsächlich auf Fehlerbäumen. Wegen der wachsenden Fahrzeugkomplexität, welche wesentlich durch die stark zunehmende Anzahl der durch mechatronische Systeme realisierten Funktionen bedingt ist, und der steigenden Typenvielfalt ist die geführte Fehlersuche in freien Werkstätten nicht immer zielführend.

Um die Unterstützung des Personals von freien Werkstätten bei der zukünftigen Fehlersuche zu gewährleisten, werden neue Generationen von herstellerunabhängigen Diagnosetools benötigt, die die Probleme der Variantenvielfalt und Komplexität lösen. In der vorliegenden Arbeit wird ein Lösungsansatz vorgestellt, der einen qualitativen, modellbasierten Diagnoseansatz mit einem auf heuristischem Diagnosewissen basierenden Ansatz vereint. Neben der Grundlage zur Wissenserhebung werden in dieser Arbeit die theoretische Grundlage zur Beherrschung der Variantenvielfalt sowie die Tests für die erstellten Diagnosemodelle behandelt.

Die Diagnose ist symptom basiert und die Inferenzmechanismen zur Verarbeitung des Diagnosewissens sind eine Kombination aus Propagierung der abweichenden physikalischen Größen im Modell und der Auswertung des heuristischen Wissens. Des Weiteren werden in dieser Arbeit verschiedene Aspekte der Realisierung der entwickelten theoretischen Grundlagen dargestellt, zum Beispiel: Systemarchitektur, Wissenserhebungsprozess, Ablauf des Diagnosevorgangs in den Werkstätten.

Die Evaluierung der entwickelten Lösung bei der Wissenserhebung in Form von Modellerstellungen und Modellierungsworkshops sowie Feldtests dient nicht nur zur Bestätigung des entwickelten Ansatzes, sondern auch zur Ideenfindung für die Integration der entwickelten Tools in die existierende IT-Infrastruktur.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und Lösungsübersicht</b>	<b>1</b>
1.1	Motivation und Zielstellung . . . . .	1
1.2	Verbrennungsmotor . . . . .	3
1.3	Lösungsansatz . . . . .	5
1.3.1	Konzept zur Modellierung des Diagnosewissens . . . . .	6
1.3.2	Beschreibung des Ansatzes des Inferenzmechanismus . . . . .	10
1.3.3	Modellierungsapplikation . . . . .	16
1.3.4	Diagnose-Runtime-System . . . . .	19
1.4	Überblick über die Evaluierungsergebnisse . . . . .	22
1.4.1	Aufwand bei der Modellbildung . . . . .	22
1.4.2	Ergebnisse der Feldtests . . . . .	24
1.5	Gliederung der Arbeit . . . . .	24
<b>2</b>	<b>Werkstatt diagnose im Kraftfahrzeugbereich</b>	<b>26</b>
2.1	Randbedingungen und Anforderungen . . . . .	27
2.1.1	Wissenserhebung für die Werkstatt diagnose . . . . .	27
2.1.2	Das Diagnoselaufzeitsystem in der Werkstatt . . . . .	29
2.2	Stand der Forschung und Technik . . . . .	31
2.2.1	Bestehende Diagnoseverfahren . . . . .	31
2.2.2	Existierende Lösungen der Werkstatt diagnose . . . . .	35
2.3	Lösungsweg . . . . .	40
<b>3</b>	<b>Wissenserhebung</b>	<b>41</b>
3.1	Anforderungsanalyse des Diagnosemodells . . . . .	41
3.1.1	Funktionale Modellierung auf Basis der objektorientierten Methode . . . . .	41
3.1.2	Qualitative Modellierung . . . . .	42
3.1.3	Stationäre Betrachtung . . . . .	43
3.1.4	Berücksichtigung des Fehlermodus bei der Modellierung . . . . .	44

3.1.5	Verwendung zusätzlichen heuristischen Wissens . . . . .	45
3.2	Inhalt des Diagnosemodells . . . . .	46
3.3	Komponentenmodellierung . . . . .	53
3.3.1	Komponentenschnittstelle . . . . .	54
3.3.2	Interne Variablen einer Komponente . . . . .	56
3.3.3	Qualitatives Verhaltensmodell . . . . .	58
3.3.4	Verhaltenstabelle einer Komponente . . . . .	61
3.3.5	Beeinflussungstabelle einer Komponente . . . . .	62
3.3.6	Ein beispielhaftes Komponentenobjektdiagramm . . . . .	65
3.4	Messungsmodellierung . . . . .	65
3.4.1	Messung mittels Systemsensor . . . . .	65
3.4.2	Abbildung der externen Messung . . . . .	67
3.5	Abbildung des Fehlercodes . . . . .	69
3.6	Abbildung der Beanstandung . . . . .	73
3.7	Abbildung der Tests . . . . .	78
3.7.1	Modellierung des Stellgliedtests . . . . .	78
3.7.2	Modellierung des Komponententests . . . . .	79
3.8	Systemmodellierung . . . . .	81
3.8.1	Systemschnittstelle . . . . .	81
3.8.2	Verbindung . . . . .	83
3.9	Parametrisierung und Bedatung des Modells . . . . .	84
3.10	Behandlung der Variantenvielfalt . . . . .	84
3.11	Modellprüfung . . . . .	90
<b>4</b>	<b>Inferenzmechanismen zur Diagnoseerstellung</b>	<b>98</b>
4.1	Aufbereitung der Eingabedaten (Algorithmen E und A) . . . . .	99
4.2	Verdachtsgenerierung . . . . .	103
4.2.1	Ermittlung der verdächtigen Komponenten (Algorithmus P) . . . . .	103
4.2.2	Bewertung der verdächtigen Komponenten (Algorithmus B) . . . . .	108
4.3	Koordination der Messungen und Tests . . . . .	109
4.3.1	Koordinierung der Sensoren-, Messungseinsätze und Tests (Algorithmus K) . . . . .	110
4.3.2	Gruppierung der verdächtigen Komponenten (Algorithmus S) . . . . .	112

4.4	Anwendungsbeispiele . . . . .	114
4.4.1	Anwendungsbeispiel mit existierenden Fehlercodes . . . . .	114
4.4.2	Anwendungsbeispiel mit Kundenbeanstandung, Sichtprüfung und ohne Fehlercode . . . . .	116
4.4.3	Anwendungsbeispiel mit Kundenbeanstandung . . . . .	124
<b>5</b>	<b>Wissenserhebung mit DMB-System</b>	<b>131</b>
5.1	Modellerstellungsprozess, Rollen und Systemarchitektur . . . . .	131
5.1.1	Modellerstellungsprozess . . . . .	131
5.1.2	Rollen bei der Modellerstellung . . . . .	137
5.1.3	Systemarchitektur DMB-System . . . . .	138
5.2	Das entwickelte DMB-System . . . . .	142
5.2.1	Speicherformat des Modells . . . . .	142
5.2.2	Qualitätssicherung . . . . .	143
5.2.3	Variantenmanager . . . . .	146
<b>6</b>	<b>Diagnose mit DRT-System</b>	<b>148</b>
6.1	Diagnoseprozess und Systemarchitektur . . . . .	148
6.1.1	Diagnoseprozess mit DRT-System . . . . .	148
6.1.2	Systemarchitektur DRT-System . . . . .	151
6.2	Diagnose mit DRT-System . . . . .	155
6.3	Migration in das bestehende Laufzeitsystem . . . . .	159
<b>7</b>	<b>Evaluierung des Lösungsansatzes</b>	<b>163</b>
7.1	Evaluierung der Wissenserhebung . . . . .	163
7.2	Evaluierung der Diagnose . . . . .	171
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>184</b>
8.1	Zusammenfassung . . . . .	184
8.2	Ausblick . . . . .	187
8.2.1	Erweiterung des Diagnosewissens . . . . .	187
8.2.2	Weiterführende Projekte . . . . .	190
	<b>Literatur</b>	<b>194</b>

# 1 Einleitung und Lösungsübersicht

Der ansteigende Vernetzungsgrad im Fahrzeug, die stark zunehmende Anzahl der durch Software realisierten Funktionen, die steigende Komplexität hydraulischer, pneumatischer und mechanischer Fahrzeugkomponenten sowie die steigende Typenvielfalt der Kraftfahrzeuge erschweren die Fehlersuche und -behebung in der Werkstatt. Diese stetige Zunahme der Komplexität im Fahrzeug ist bedingt durch Forderungen von Staat und Umwelt, durch Kundenwünsche und technische Angebote. Staat und Umwelt fordern die Reduzierung von Abgasausstoß und Kraftstoffverbrauch und schreiben eine Onboard-Diagnose zur Auswertung dieser Faktoren vor. Die Individualisierung des Fahrzeugs ist durch Kundenwünsche entstanden; hier müssen verschiedene Aspekte betrachtet werden, wie zum Beispiel Sicherheit, Kraftstoffverbrauch, Zuverlässigkeit und Qualität, Verbesserung der Verkehrssituation, Abgas- und Außengeräuschemissionen, Fahrleistung und Transportvolumen, Kauf- und Betriebskosten, Komfort, Emotionalität und Recycling. Die technischen Angebote sind vielseitig und umfassen verschiedene Bereiche des Fahrzeugs von Motor bis Komfort. Bei der Fahrzeugassistenz sind beispielsweise Systeme wie elektronisches Stabilisierungsprogramm (ESP), Antiblockiersystem (ABS) oder Antriebsschlupfregelung (ASR) für schnellere und präzisere Eingriffe in Gefahrensituationen seit längerer Zeit im Einsatz. Fahrerassistenzsysteme, welche den Fahrer entlasten und gleichzeitig seine Konzentration fördern wie Navigation, adaptive Geschwindigkeitsregelung (ACC), Spurwechselassistent (SWA) oder Spurhalteassistent (LDW), etc. sind in Fahrzeugen der Oberklasse eingebaut und werden zukünftig auch in preiswerten Klassen Einzug finden. Konsequenz daraus sind eine größere Verteilung der Fehlerauswirkungen und mehr Fehlermöglichkeiten. Erforderlich sind daher verbesserte Diagnosefunktionalitäten in Entwicklung, Produktion und Service, das heißt im gesamten Lebenszyklus eines Kraftfahrzeugs. Diese Anforderung ist Ziel der Dissertation und wird im ersten Teil dieses Kapitels besprochen. Danach werden die Anwendungsbeispiele sowie die Randbedingungen kurz beschrieben. Der entwickelte Lösungsansatz wird im dritten Teil erörtert. Im Anschluss daran wird ein Überblick der Realisierung und der Evaluierung des Lösungsansatzes präsentiert. Abschließend wird die Gliederung der Arbeit vorgestellt.

## 1.1 Motivation und Zielstellung

In den Werkstätten müssen Personal und Prüftechnik mit der fortschreitenden Entwicklung der Kraftfahrzeuge mithalten. Dies wird zunehmend erschwert durch die oben genannten Gründe wie steigende Typenvielfalt - insbesondere bei den Diagnoseabläufen in freien Werkstätten -, die wachsende Fahrzeugkomplexität und die stark zunehmende Anzahl der durch Software realisierten Funktionen im Fahrzeug.

Die Diagnose von Fahrzeugen in den Werkstätten basiert heute auf dem Einsatz von spezieller Prüftechnik, mit deren Hilfe die in den Steuergeräten des Fahrzeugs abgelegten Diagnoseinformationen („Diagnostic Trouble Codes“, DTC) ausgelesen werden. Ein DTC gibt - aus Sicht des erzeugenden Steuergerätes - Hinweise auf mögliche Fehlerquellen, die aber in der Regel



nicht hinreichend sind. Aus diesem Grund werden in der Werkstatt neben den DTC der Steuergeräte weitergehende relevante Informationen zur Diagnose herangezogen, wie zum Beispiel physikalische Messgrößen (Spannung, Druck, Abgas usw.) und subjektive Beobachtungen (Geräusche, Sichtkontrolle).

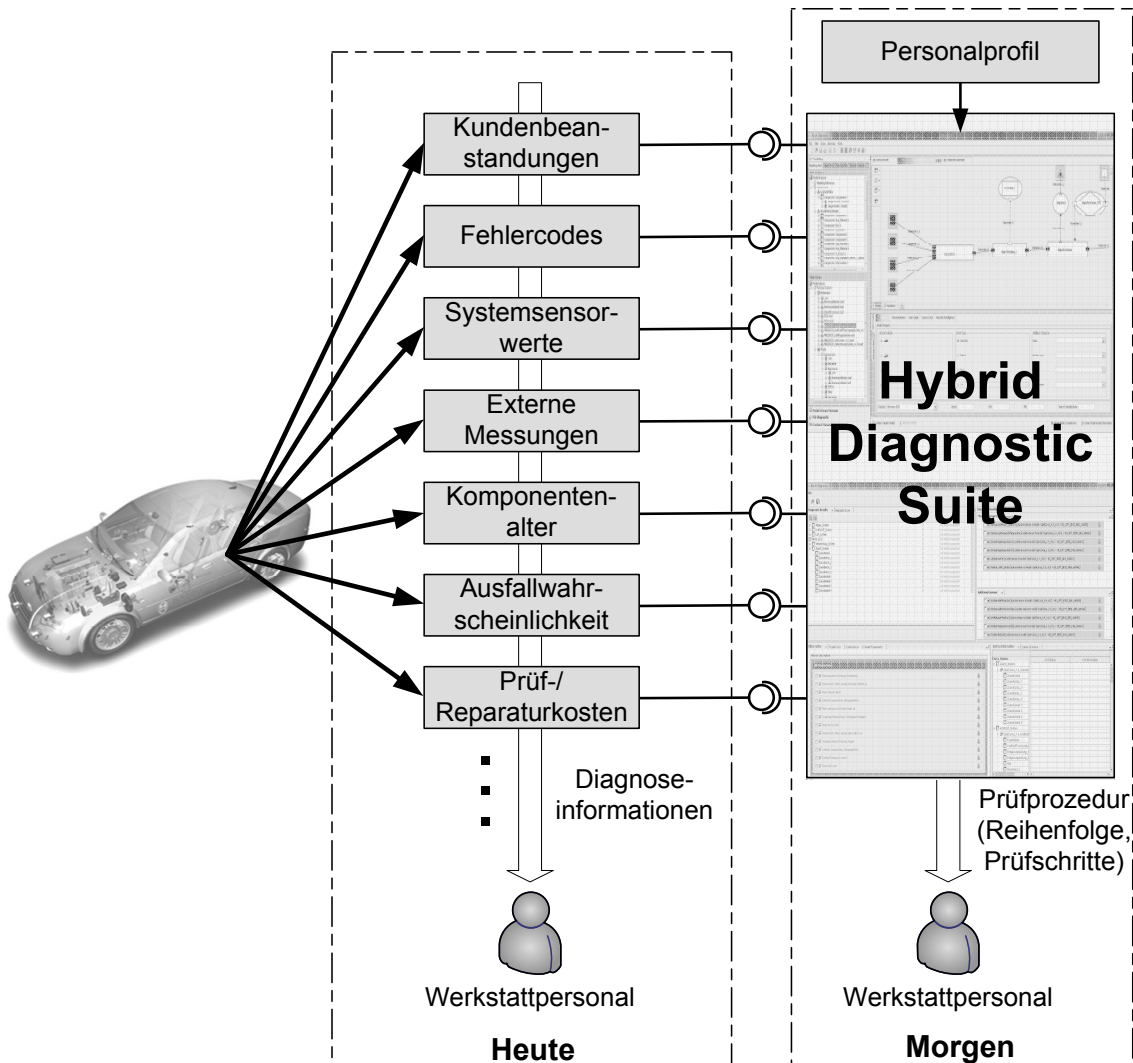


Abbildung 1.1: Verfahren und Strategien für neue Prüftechniksysteme

Bei der Suche nach der defekten Komponente müssen die vorliegenden Diagnoseinformationen zur Bildung von Fehlerhypothesen logisch kombiniert und hieraus weitere Prüfschritte und Messungen abgeleitet werden. Hierbei wenden Diagnosesysteme in Vertragswerkstätten hauptsächlich die fallbasierte Diagnostik an. Zum einen hält sich hier die Fahrzeugvielfalt in Grenzen und zum anderem besteht eine Meldepflicht bei neuen, nicht im System vorhandenen Fällen. Die freien Werkstätten verfügen nicht über eine solche Datenbank. Somit ist der fallbasierte Ansatz schwer umsetzbar. Das Werkstattpersonal der freien Werkstätten muss bei einem Diagnosevorgang die vorliegenden Diagnoseinformationen zur Bildung von Fehlerhypothesen selbst kombinieren und hieraus weitere Prüfschritte und Messungen ableiten. Hierzu gehört auch die Methodik bei der Fehlersuche, das heißt, das Treffen von Entscheidungen über durchzuführende Prüfgänge und deren Reihenfolge („Welches Steuergerät wird zuerst

abgefragt?“, „Welche physikalischen Signale werden als nächstes gemessen?“, „Welche Komponente wird als erstes einem Komponententest unterzogen?“ usw.). Diese Methodik bestimmt erheblich die Kosten bei der Fehlersuche und mitunter deren Erfolg, insbesondere in freien Werkstätten, da die Anzahl der zu diagnostizierenden Fahrzeuge entsprechend hoch und der relative Erfahrungsschatz des Werkstattpersonals entsprechend gering sein kann, verglichen mit dem einer Vertragswerkstatt (Abbildung 1.1, Heute).

Bei der Entwicklung und Produktion von Fahrzeugen wird Diagnosewissen aus Entwicklungsdaten und Testdaten erstellt. Hier ist Wissen in detaillierter Form vorhanden, das nicht öffentlich zugänglich ist. Eine nicht vertraglich gebundene Werkstatt, die dieses Wissen anfordert, muss einen sehr hohen Betrag zahlen. Im Service in der freien Werkstatt ist daher detailliertes Wissen über einen bestimmten Fahrzeugstyp oft nicht verfügbar, insbesondere bei neuen Fahrzeugtypen. Um Fahrzeugtypen bestimmter Hersteller zu diagnostizieren, benötigt eine freie Werkstatt im Allgemeinen das Diagnosetool des jeweiligen Herstellers. Dadurch entstehen enorme Kosten für diese Werkstatt. Eine herstellernunabhängige Diagnosesoftware stellt für diese Werkstätten eine kostengünstige Alternative dar. Um die Unterstützung des Personals von freien Werkstätten bei der zukünftigen Fehlersuche zu sichern, werden neue Generationen von herstellernunabhängigen Diagnosetools, die die Probleme der Variantenvielfalt und Komplexität lösen, benötigt. Diese Arbeit trägt zur Lösung dieser Aufgabe bei, indem sie eine neue Methode der Offboard-Diagnose vorstellt (Abbildung 1.1, Morgen).

Als Anwendungsbeispiel dienen in dieser Arbeit Otto- und Dieselmotor, welche in Kraftfahrzeugen häufig eingesetzt werden. In verschiedenen vorhergehenden Arbeiten wurden oft elektrische Systeme als Anwendungsfall herangezogen. Beim Verbrennungsmotor treffen mehrere physikalische Domänen aufeinander, zum Beispiel Mechanik, Thermodynamik, Elektronik, Hydraulik, außerdem sind die Informations- und Steuerungssysteme (ECU) beteiligt. Bei der Verarbeitung der Diagnoseinformationen sind daher nicht nur Messwerte und Beobachtungen wie bei elektrischen Systemen sondern zusätzliche Informationen, welche aus den ECU stammen, nötig. Alle diagnoserelevanten Informationen in einer Diagnosewissensbasis abzubilden und bei dem Diagnosevorgang zu verwenden ist eine Herausforderung für die zu entwickelnden Wissenserhebungs- und Diagnosetools.

## 1.2 Verbrennungsmotor

Die Anwendungsbeispiele sind ein Common-Rail-System (CP3-System; Versuchsträger: Daimler E220cdi) und ein Motronic-System mit E-Gas (ME 7.x-System; Versuchsträger: Opel-Corsa 1.2). Die Bereitstellung des Moments aus dem Verbrennungsvorgang im Brennraum ist die Hauptfunktion der beiden Systeme. Um das Verbrennungsmoment zu erzeugen, werden bei dem ME 7.x-System die Luftmasse und die verfügbare Kraftstoffmasse gemischt und in den Brennraum eingeführt; der Zündfunke leitet die Verbrennung des Luft-Kraftstoff-Gemischs zu einem bestimmten Zeitpunkt ein. Das Motorsystem kann in verschiedene Subsysteme aufgeteilt werden: Luft, Einspritzung, Brennraum, Abgas, Kühlung und Motorsteuerung (vgl. [122]). Die Aufteilung des Systems bezieht sich auf die einzelnen Funktionen bzw. Aufgaben der Subsysteme. Zum Beispiel ist es Aufgabe des Luftsystems, beim jeweiligen Betriebszustand des Motors (Warmlauf, Leerlauf, Teillast, etc.), ein bestmöglich angepasstes Luft-Kraftstoff-Verhältnis bereitzustellen. Um die zugewiesene Aufgabe auszuführen, sind die Subsysteme im Ottomotor-System wiederum aus mehreren Komponenten aufgebaut. Das

Luftsystem (Abbildung 1.2) zum Beispiel besteht aus mehreren Komponenten wie Drosselklappe, Ansauglufttemperatursensor, Luftmassenmesser, Luftfilter und Luftleitung [107].

Das Motorsystem besitzt damit bezüglich der Diagnoseaufgaben des mechatronischen Systems typische Randbedingungen (vgl. [152]):

- Durch den Dekompositionsansatz wird das System in Subsysteme und Komponenten aufgeteilt und durch den Kompositionsansatz können die Komponenten beziehungsweise Subsysteme umgekehrt zu einem System zusammengefasst werden.
- Das System verfügt über eigene Sensoren; diese Sensoren haben eine beschränkte Messgenauigkeit und werden für die Regelung und Selbstdiagnose verwendet.
- Aufgrund der begrenzten Anzahl der Systemsensoren ist es oft nicht möglich, direkt von dem Fehlercode der Onboard-Diagnostik auf eine mögliche Fehlerursache zu schließen.
- Die Einbringung zusätzlicher Sensoren ist möglich, jedoch aufwendig, da sie direkt an das zu diagnostizierende System angebunden werden müssen.
- Die Ausfallwahrscheinlichkeiten und die entstehenden Kosten bei Reparatur bzw. Austausch der Komponenten in Standardsituationen sind in der Regel bekannt.
- Die Fehlfunktion im mechatronischen System kann meist auf Komponentenfehler zurückgeführt werden.

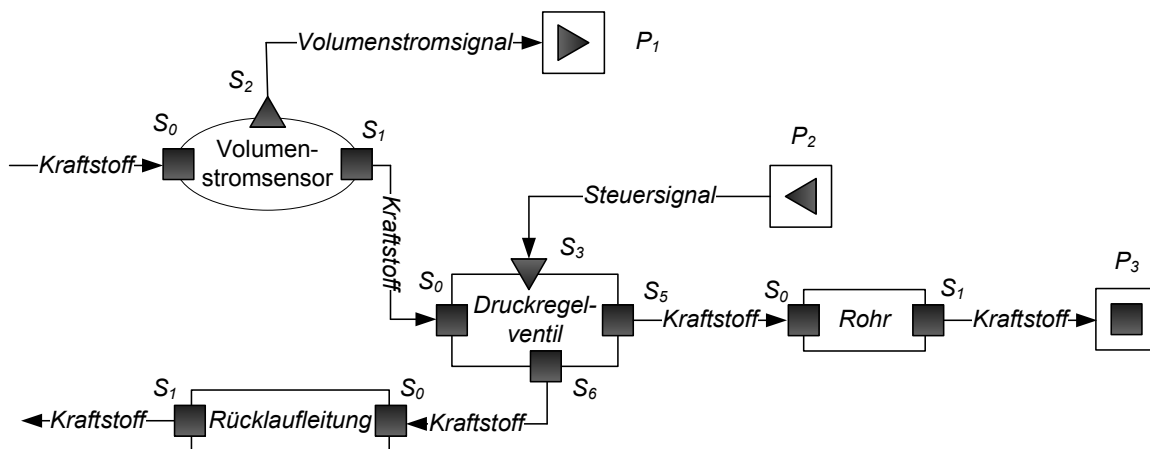


Abbildung 1.2: Teilmodell des Kraftstoffsystems

Des Weiteren existieren typische Merkmale bei der Werkstattdiagnose:

- Es gibt viele Komponenten mit gleichen Funktionen, die sich in vielen Systemen wiederfinden, aber immer wieder in unterschiedlichen Varianten und Kombinationen vorkommen; hinsichtlich der Diagnostik sollten diese gleich behandelt werden.
- Die Vorgänge bei der Diagnostik in der Werkstatt sind vorwiegend stationär, daher ist eine statische Betrachtungsweise möglich.
- Das System funktioniert mit unterschiedlichen Betriebszuständen und wird in verschiedenen Zuständen betrieben.

- Bei der Diagnostik beschreiben die Experten die funktionalen Zusammenhänge durch grobe Wertebereiche und ungefähre Zusammenhänge, zum Beispiel monotone Abhängigkeit. Solche Beziehungen oder Relationen können nur durch qualitative Beschreibungen erfasst werden, welche viele spezielle Ausprägungen zulassen.
- Der Diagnosevorgang muss nicht in Echtzeit durchgeführt werden, er gilt aber weiterhin als zeitkritische Anwendung mit begrenzter Rechenleistung und begrenztem Speicherplatz.

Aufgrund der oben genannten Diagnoserandbedingungen des Verbrennungsmotors ist dieser sehr gut geeignet zu prüfen, ob der zu entwickelnde Diagnoseansatz für mechatronische Systeme geeignet ist. Um ein wissensbasiertes Diagnosesystem zu realisieren, muss das Expertenwissen formalisiert werden. Unter Berücksichtigung der erwähnten Randbedingungen werden im nächsten Abschnitt die Repräsentationsform des Diagnosewissens und das Implementierungskonzept der Inferenzkomponente des Diagnosetools erläutert und ein Konsens für diese Arbeit gebildet.

### 1.3 Lösungsansatz

Um die Herausforderungen der freien Werkstattdiagnose beherrschen zu können, müssen in dieser Arbeit verschiedene Probleme bei der Wissenserhebung sowie den Schlussfolgerungen gelöst werden. Bei der Wissenserhebung soll eine neue Wissensrepräsentation, die die in Abschnitt 1.2 beschriebenen Eigenschaften des mechatronischen Systems und das verfügbare Diagnosewissen in freien Werkstätten abbildet, entwickelt werden. Aus diesem Grund konzentriert sich der Lösungsansatz, der die Wissenserhebungsprobleme bewerkstelligt, auf zwei Hauptthemen:

- Abbildungskonzept des mechatronischen Systems für die Werkstattdiagnose
- Lösung der Variantenvielfalt in freien Werkstätten.

Die Lösungskonzepte zur Wissenserhebung werden in Abschnitt 1.3.1 beschrieben. Analog zur Wissenserhebung soll die Inferenzkomponente in dem Diagnosetool verschiedenen Anforderungen der freien Werkstattdiagnose gerecht werden:

- Ausnutzung aller diagnoserelevanten Informationen
- Ermöglichung der systemübergreifenden Diagnose
- Reduzierung des Rechenaufwandes.

Zur Lösung dieser Probleme bei der Entwicklung der Inferenzkomponente werden in Abschnitt 1.3.2 Konzepte vorgeschlagen.

### 1.3.1 Konzept zur Modellierung des Diagnosewissens

Im Laufe der Forschungen haben sich nacheinander unterschiedliche Ansätze des diagnostischen Schließens entwickelt. Die wichtigste Unterscheidung besteht zwischen erfahrungsbasierter (regel- beziehungsweise fallbasierter) Diagnostik und modellbasierter Diagnostik (vgl. [112]). Alle Ansätze arbeiten mit abgespeichertem Wissen, welches zum Lösen einer Diagnoseaufgabe genutzt wird, wobei die Güte und Aussagekraft der Diagnostik direkt von der Qualität und Relevanz des abgespeicherten Wissens abhängt. Die meisten Diagnosesysteme, die gegenwärtig in freien Werkstätten eingesetzt werden, beruhen auf dem erfahrungsbasierten Ansatz. Bei diesem ist vor allem partielles und unvollständiges Wissen leichter abbildbar als bei dem modellbasierten Ansatz und damit schneller eine Diagnosemöglichkeit lieferbar. Andererseits existieren einige Vorteile der funktionalen modellbasierten gegenüber der erfahrungsbasierten Diagnostik:

- Bei neuen beziehungsweise fremden technischen Systemen ist oft nur funktionales Wissen über einzelne Systemkomponenten und über Komponentenzusammenhänge (Strukturaufbau des Systems) und wenig Erfahrungswissen vorhanden.
- Die Beherrschbarkeit der Variantenvielfalt ist besser, da sich Modellbestandteile, die möglichst kontextfrei modelliert sind, wesentlich besser als Erfahrungswissen wiederverwenden lassen.

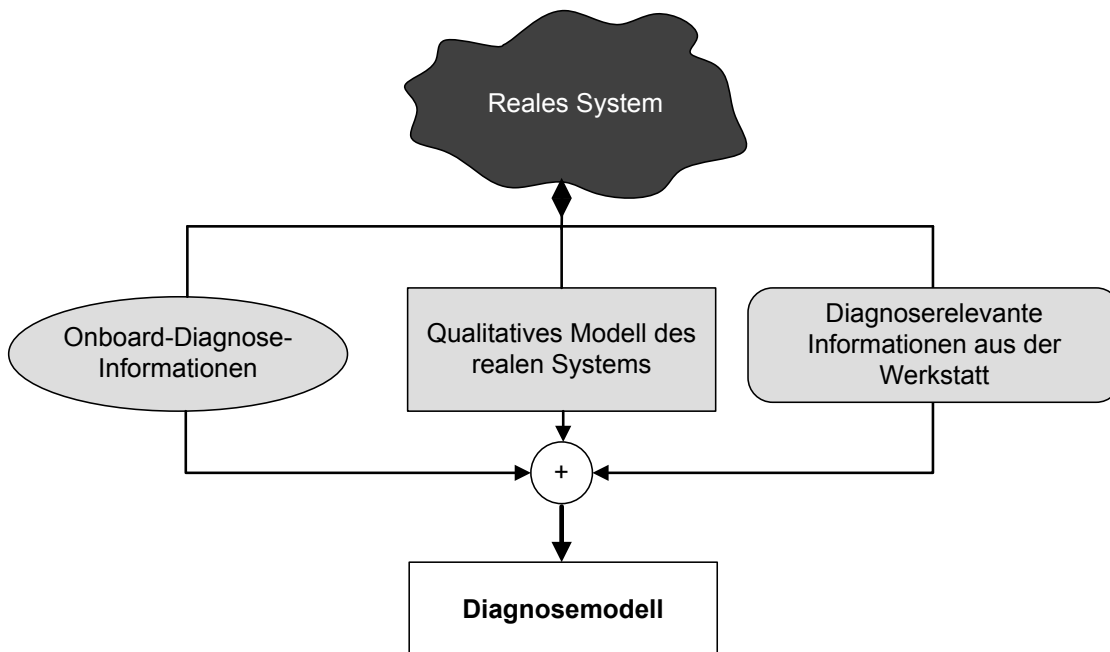


Abbildung 1.3: Grober Aufbau des Diagnosemodells

Aufgrund der oben beschriebenen Abwägungen wurden Konzepte und Implementierung eines hybriden, skalierbaren Lösungsansatzes zur mechatronischen Systemdiagnose entwickelt und realisiert. Der Ansatz ist ein Hybrid, da sowohl der modell- als auch der erfahrungsbasierte Ansatz bei der Wissenserhebung und dem Diagnosevorgang verwendet werden. Damit können bei der Diagnose alle verfügbaren Informationen berücksichtigt werden, da das Diagnosemodell neben dem physikalischen Modell des Systems auch die Informationen der

Onboard-Diagnose sowie diagnoserelevante Informationen aus der Werkstatt beinhaltet. In der Inferenzkomponente des Ansatzes müssen die modellbasierten und die erfahrungsbasierten Schließalgorithmen kombiniert werden. Das Diagnosemodell besteht aus folgenden Bestandteilen (Abbildung 1.3):

- Qualitatives Modell des physikalischen Systems
- Onboard-Diagnoseinformationen
- Diagnoserelevante Informationen aus der Werkstatt.

Das qualitative Modell wird aus einzelnen Komponentenmodellen und deren Verbindungen aufgebaut. Ein Komponentenmodell besteht aus folgenden Teilinformationen:

- Komponentenschnittstellen, mit denen eine Komponente mit seinen benachbarten Komponenten interagiert (vgl. [15], [111]).
- Interne Funktions- und Fehlervariablen, die die internen Komponenteneigenschaften oder Fehlerzustände darstellen (vgl. [137]).
- Die Verhaltensbeschreibung definiert das Nominal- und Fehlverhalten der Komponente in unterschiedlichen Betriebszuständen, da eine Komponente in unterschiedlichen Betriebszuständen unterschiedliche Verhalten aufweisen kann (vgl. [137]). Die Spezifikation des Verhaltens erfolgt in Form von relationalen qualitativen Übergangsfunktionen. Als Beispiel hierfür soll die Verhaltensbeschreibung einer Luftleitung betrachtet werden (vgl. [78]):

$$\begin{aligned} \text{Massenfluss}_{in} &\stackrel{qual}{=} \text{Massenfluss}_{aus} \\ \text{Druck}_{in} &\stackrel{qual}{-} \text{Druck}_{aus} \stackrel{qual}{=} \text{Massenfluss}_{aus} \end{aligned}$$

Aus dieser Spezifikation des Verhaltens wird die Verhaltenstabelle generiert (vereinfachtes Beispiel in Tabelle 1.1). Die Generierung setzt voraus, dass die qualitativen Gleichheitszeichen und die qualitativen Operationen (zum Beispiel qualitative Addition, Subtraktion, Monotonie usw.) vordefiniert sind (vgl. [20], [136], [92]).

$\text{Massenfluss}_{in}$	$\text{Druck}_{in}$	$\text{Massenfluss}_{aus}$	$\text{Druck}_{aus}$
hoch	hoch	normal	hoch
hoch	hoch	niedrig	hoch
...	...	...	...
hoch	niedrig	hoch	niedrig

Tabelle 1.1: Beispiel für Aufbau einer Verhaltenstabelle

Weil Systemsensoren eine weitere Rolle bei dem Diagnosevorgang übernehmen, enthalten sie zusätzliche Messeigenschaften mit folgenden Informationen:

- physikalische Modellgröße, die gemessen wird

- Wertebereich der Modellgröße.

Eine Verbindung beschreibt die Abhängigkeit zwischen zwei Schnittstellen unterschiedlicher Komponenten. Das Verhalten des Gesamtsystems ergibt sich aus der Interaktion der einzelnen Komponenten über die Verbindungen. Nicht zu verwechseln sind die Verbindungen des realen Systems mit den Verbindungen des Modells (vgl. [15], [124]). So wird beispielsweise ein Rohr nicht durch eine Verbindung, sondern durch eine Komponente modelliert.

Zum Grundumfang des Motorsteuergeräts in modernen Kraftfahrzeugen gehört die Onboard-Diagnostik (OBD). Die OBD erfolgt durch den permanenten Abgleich der Aktionen und der Reaktionen des Systems im Betrieb mit den Befehlen des Steuergerätes und der Signale der verschiedenen systemeigenen Sensoren untereinander auf ihre Plausibilität (vgl. [99]). Wenn die Plausibilität nicht gegeben ist, erfolgt ein Eintrag in den Fehlerspeicher des Steuergeräts. Wegen der unterschiedlichen Diagnoseinformationen, die ein Fehlercode beinhaltet, ist es notwendig, dass eine Abbildung des Fehlercodes im Modell in zwei Arten unterschieden wird:

- Fehlercode als virtueller Sensor: Dieser zeigt auf abgebildete physikalische Modellgrößen. Bei der Modellierung kann wie folgt vorgegangen werden:
  - Fehlercode: 2352-1
  - Schnittstelle: Virtuelle Schnittstelle, mit der sich der Fehlercode mit Injektor 6 verbindet
  - Wertebelegung:  $Massenfluss_{aus} \stackrel{qual}{=} \text{niedrig}$
- Fehlercode als Diagnoseregeln: Dieser zeigt direkt auf systemeigene Komponenten oder Komponentengruppen. Die Diagnoseregeln kann wie folgt beschrieben werden:
  - Wenn: Fehlercode 0206-3 - existiert
  - Dann:
    - \* Verdächtige Komponente: Injektor 6
    - \* Anweisung: Bauteil Kraftstoffinjektor 6 prüfen
    - \* Möglicher Fehler: Kurzschluss nach Masse.

Kundenbeanstandungen sind eine der diagnoserelevanten Informationen in den Werkstätten. Sie liegen in Form verbaler Aussagen vor und können verschiedene Ursachen beinhalten. Soweit möglich werden sie auf physikalische Größen des Modells abgebildet. Falls das nicht möglich ist, werden Diagnoseregeln in das Modell integriert. Als Beispiel hierfür wird die Kundenbeanstandung „Motor klingelt oder klopft“ betrachtet (vgl. [122]):

- Kundenbeanstandung als virtueller Sensor. Dieser zeigt auf abgebildete physikalische Modellgrößen:
  - Kundenbeanstandung: „Motor klingelt oder klopft“
  - Schnittstelle: Virtuelle Schnittstelle, mit der sich die Kundenbeanstandung mit Drallklappe verbindet
  - Wertebelegung:  $Massenfluss_{aus} \stackrel{qual}{=} \text{hoch}$

- Kundenbeanstandung als Regel. Direkte Zuweisung der Kundenbeanstandung auf verdächtige Komponenten:
  - Wenn: Kundenbeanstandung „Motor klingelt oder klopft“ existiert
  - Dann:
    - \* Verdächtige Komponente: Brennraum
    - \* Anweisung: Geometrie des Brennraums prüfen
    - \* Möglicher Fehler: Die Änderung beziehungsweise Abweichung der Brennraumgeometrie
  - Optimierungsinformationen:
    - \* Aufwand der Prüfung: Hoch
    - \* Wahrscheinlichkeit: Selten.

Eine weitere Information im Feld sind die Offboard-Messungen. Im Unterschied zu den systemeigenen Sensoren erfolgt die Modellierung des externen Messgeräts ohne die Verhaltensbeschreibung. Analog zu dem systemeigenen Sensor beinhaltet das externe Messgerät folgende Messeigenschaften:

- Physikalische Modellgröße, die gemessen wird
- Wertebereich der Modellgröße.

Die Anbindung von Komponenten- und Stellgliedtests in das Modell ist unkompliziert (vgl. [3]), da sie direkt an eine systemeigene Komponente beziehungsweise Komponentengruppe gebunden sind. Eine automatische Anbindung in Form einer Datenbankabfrage ist möglich, wenn die systemeigenen Komponenten und deren Tests den gleichen Identifizierungscode besitzen. Die Optimierungsinformationen wie Testdauer, Testkosten beziehungsweise Aufwand (soweit vorhanden) werden direkt an die Komponenten- und Stellgliedtests gebunden. Diese Aufteilung des Komponentenmodells ermöglicht die Anwendung des Schichtkonzeptes bei dem Modellerstellungsvorgang. Dabei wird ein Systemmodell in drei Hauptschichten unterteilt:

- Grundsicht: In dieser Schicht werden folgende Aktivitäten durchgeführt:
  - Festlegen der Systemschnittstellen und damit der Systemgrenzen
  - Modellierung der systemeigenen Komponenten und deren Verhalten
  - Abbilden der Verbindungen zwischen Komponenten sowie zwischen Komponenten und Systemschnittstellen
- Mittelschicht: Hier werden die Fehlercodes aus der Onboard-Diagnostik abgebildet
- Oberschicht: Die Kundenbeanstandung sowie externe Mess- und Testmöglichkeiten werden in dieser Schicht modelliert.

Häufig kommt es vor, dass in unterschiedlichen Fahrzeugen das gleiche System eingebaut ist. Die Unterschiede liegen im Fehlercode des einzelnen Fahrzeugs oder bei externen Mess- und Testmöglichkeiten. Zum Beispiel existiert der Fehler-Code P-0016 (Funktionsstörung bei Nockenwellen- und Kurbelwellensignal) beim Corsa D - Z12XEP und beim Agila - Z12XEP



nicht, obwohl in beiden Fahrzeugtypen der gleiche Benzinmotor eingebaut ist. Mit dem Schichtkonzept ist es möglich, die Mittel- und Oberschicht auszutauschen. Damit wird der Modellierungsaufwand bei verändertem Fehlercode beziehungsweise Kundenbeanstandungen sowie externen Mess- und Testmöglichkeiten reduziert, weil die Änderung durch das Austauschen der Schicht oder in unterschiedlichen Schichten vorgenommen werden kann. Der Prozess zur Vervollständigung des Diagnosemodells wird in zwei Hauptphasen untergliedert: Die Modellierungsphase und die Bedatungsphase. Der Modellierer konzentriert sich darauf, welche Komponenten im betrachteten System verbaut wurden, welches Verhalten sie aufweisen und wie sie miteinander zusammengeschaltet sind. In der Bedatungsphase werden die technischen Daten wie Testmöglichkeiten, Soll- und Grenzwerte einer Komponente in das Modell integriert. Die Bedatungsphase kann wiederum in zwei unabhängige Phasen unterteilt werden: Die erste Teilphase umfasst die Integration der Informationen über die Sollwertbereiche der Sensoren und Testprozeduren der Komponenten in das Diagnosemodell. Die Anbindung der Optimierungsdaten (Ausfallwahrscheinlichkeit, entstehende Kosten bei Messung, Prüfung oder Reparatur von systemeigenen Komponenten, etc.) zur Priorisierung der verdächtigen Komponenten im Diagnosevorgang bildet die zweite Teilphase. Das Ergebnis der Trennung von Modellen und komponentenspezifischen Daten führt zu einer enormen Reduzierung der Anzahl benötigter Modelle. Die Wartung und weitere Entwicklung des Modells sowie die Pflege der Parameter können unabhängig voneinander geschehen. Mit den oben beschriebenen Trennungsmöglichkeiten ist das Diagnosemodell skalierbar, weil für die Diagnostik nicht alle diagnoserelevanten Informationen im Diagnosemodell enthalten sein müssen. So funktioniert das System auch, wenn zum Beispiel die Ausfallwahrscheinlichkeit von Komponenten noch unbekannt ist, weil das Wissen über die Ausfallwahrscheinlichkeit erst über Jahre gesammelt wird.

### 1.3.2 Beschreibung des Ansatzes des Inferenzmechanismus

Da der Diagnoseprozess in der Werkstatt ein symptomorientierter Diagnoseprozess ist, besteht der Algorithmus aus folgenden Schritten:

---

**Algorithmus 1.1** : Diagnostik-Algorithmus (D)

---

**Eingabe:** (*Input Algorithmus D*)

- Menge der Kundenbeanstandungen, die sich bei der Fahrzeugannahme beziehungsweise bei der Probefahrt durch das Werkstattpersonal ergaben.
- Menge der vorhandenen Fehlercodes.
- Menge der Modellgrößen mit Wertebelegung, die sich durch systemeigene Sensoren bestimmen lassen.
- Menge der Modellgrößen mit Wertebelegung, die sich durch externe Messungen bestimmen lassen.
- Menge aller Stellglied- und Komponententests, die durchgeführt wurden.

**Ausgabe:** (*Output Algorithmus D*)

- Menge der verdächtigen Komponenten mit Bewertungspunktzahl.

- Eine Menge der Systemsensoren, externen Messungen, Stellgliedtests und Komponententests mit Bewertungspunkten, die zum Ausschließen der Systemkomponenten aus der verdächtigen Menge dienen soll.

**Algorithmus:**

**D1** Aufbereitung der eingegebenen Diagnoseinformationen: Analyse der Werte der Ein- und Ausgangsattribute sowie der internen Variablen und der zu diagnostizierenden Komponenten anhand der Eingabedaten.

**D1.1** Setzen der Attribute im Modell gemäß der Eingabedaten:

Für alle Kundenbeanstandungen, Fehlercodes, Messungen und Tests beziehungsweise Prüfungen werden mögliche Abbildungen in dem Modell ermittelt.

**D1.2** Einschränkung der zu diagnostizierenden Komponenten:

- Durch Fehlercodes, die bestimmte Komponenten verdächtigen.
- Durch Ausschluss offensichtlich korrekt funktionierender Komponenten.
- Durch Testergebnisse von Komponenten- und Stellgliedtests.

**D2** Verdachtsgenerierung:

**D2.1** Propagierung der Attribute im Modell zur Ermittlung verdächtigter Komponenten;

**D2.2** Bewertung der verdächtigten Komponenten mit Bewertungspunktzahlen;

**D3** Koordinierung der Sensoren- und Messeinsätze sowie der Komponententests: Ermittlung von möglichen Messaktionen, Komponententest mit Ranking.

**D4** Erneuter Aufruf des Diagnostik-Algorithmus bis keine Kundenbeanstandungen oder Fehlercodes mehr vorliegen.

Zur Erläuterung dieses Diagnosealgorithmus kann hierfür das Beispielsystem (Abbildung 1.2) herangezogen werden. Dabei beinhaltet das Modell des Rohres eine Verhaltenstabelle 1.2:

<i>Rohr</i>					
<i>In</i>			<i>Out</i>		
$S_0$			$S_1$		<i>Variable</i>
<i>Kraftstoff</i>		<i>Kraftstoff</i>			
$\dot{V}$	...	$\dot{V}^1$	...	<i>Leckage</i>	...
<i>H</i>	...	<i>H</i>	...	<b>N</b>	...
<i>H</i>	...	$\{H, N, L\}$	...	<b>H</b>	...
<i>N</i>	...	<i>N</i>	...	<b>N</b>	...
...	...	...	...	...	...
<i>L</i>	...	<i>L</i>	...	<b>N</b>	...

Tabelle 1.2: Ein Abschnitt der Verhaltenstabelle des Rohres

Neben dem Modell existieren folgende Diagnoseinformationen:

<sup>1</sup> $\dot{V}$  := Volumenstrom; *N* := normal; *L* := niedrig; *H* := hoch

- Fehlercode ist nicht vorhanden.
- Die Kundenbeanstandung lautet „Höchstleistung kann nicht erreicht werden“.
- Es existiert keine Leckage in dem System (Sichtprüfung).
- Das Messsignal des Volumenstromsensors vor dem Druckregelventil ist im Sollbereich.

Mit Hilfe der Methode **D1.1** wird aus den einzelnen Diagnoseinformationen von Kundenbeanstandungen, Fehlercodes aus Steuergeräten, Messwerten der systemeigenen sowie externen Sensoren und Komponententests eine Menge der Diagnoseparameter gebildet:

- Kundenbeanstandung:  $Kraftstoffsystem.P_3.Kraftstoff.\dot{V} = L$ ,
- Fehlercode:  $DTC = \emptyset$ ,
- Messung:  $Volumenstromsensor.S_1.Kraftstoff.\dot{V} = N$ ,
- Test:
  - $Druckregelventil.Leckage = N$ ,
  - $Rohr.Leckage = N$ ,
  - $Ruecklaufleitung.Leckage = N$ ,

Die Ergebnisse der Methode **D1.1** dienen als Parameter der Methode **D1.2**. Unter der Voraussetzung, dass alle systemeigenen Komponenten eine Verhaltenstabelle besitzen, wird die erste Reduktion der Verhaltenstabelle der gefundenen Komponenten durch den Aufruf der Methode **D1.2** durchgeführt. Dabei werden alle nicht relevanten Zeilen der Verhaltenstabelle einer Komponente, die nicht mit der Wertebelegung in den Diagnoseparametern übereinstimmen, entfernt. Zum Beispiel existiert als Ergebnis der Komponentenüberprüfung keine Leckage im Kraftstoffrohr. Dadurch reduziert sich die Verhaltenstabelle des Rohres. Die daraus resultierende Verhaltenstabelle 1.3 besteht aus folgenden Zeilen:

		<i>Rohr</i>			
		<i>In</i>	<i>Out</i>		
		$S_0$	$S_1$	<i>Variable</i>	
<i>Kraftstoff</i>		<i>Kraftstoff</i>			
$\dot{V}$	...	$\dot{V}$	...	<i>Leckage</i>	...
<i>H</i>	...	<i>H</i>	...	<b>N</b>	...
<i>N</i>	...	<i>N</i>	...	<b>N</b>	...
<i>L</i>	...	<i>L</i>	...	<b>N</b>	...

Tabelle 1.3: Ein Abschnitt der reduzierten Verhaltenstabelle mit  $Leckage \stackrel{qual}{=} N$  des Rohres

Analog zu diesem Rohr werden die Verhaltenstabellen des Druckregelventils mit dem Ergebnis der Sichtprüfung reduziert (Tabelle 1.4).

Zur weiteren Eingrenzung der verdächtigen Komponenten im Modell werden die Diagnoseparameter rückwärts und vorwärts propagiert (**D2.1**). Diese zielorientierte Inferenz unterliegt der transitiven Verknüpfungen von Komponenten im System. Weil sich das Modell durch einen gerichteten Graphen darstellen lässt - wobei ein Knoten eine Komponente und eine Kante die Verbindung zwischen zwei Komponenten symbolisiert - sind die transitiven Verknüpfungen die gerichteten Kanten des Modells.

<i>Druckregelventil</i>									
<i>In</i>					<i>Out</i>				
$S_0$		$S_3$		$S_5$		$S_6$		<i>Variable</i>	
<i>Kraftstoff</i>		<i>Steuersignal</i>		<i>Kraftstoff</i>		<i>Kraftstoff</i>			
$\dot{V}$	...	<i>Signal</i>		$\dot{V}$	...	$\dot{V}$	...	<i>Leckage</i>	...
...	...	...		...	...	...	...	...	...
<i>L</i>	...	<i>N</i>		<i>L</i>	...	<i>L</i>	...	<b>N</b>	...
<i>N</i>	...	<i>L</i>		<i>H</i>	...	<i>L</i>	...	<b>N</b>	...
<i>N</i>	...	<i>N</i>		<i>N</i>	...	<i>N</i>	...	<b>N</b>	...
<i>N</i>	...	<i>N</i>		<i>H</i>	...	<i>H</i>	...	<b>N</b>	...
<i>N</i>	...	<i>N</i>		<i>L</i>	...	<i>L</i>	...	<b>N</b>	...
<i>L</i>	...	<i>N</i>		<i>L</i>	...	<i>N</i>	...	<b>N</b>	...
<i>N</i>	...	<i>L</i>		<i>L</i>	...	<i>N</i>	...	<b>N</b>	...
<i>N</i>	...	<i>L</i>		<i>L</i>	...	<i>H</i>	...	<b>N</b>	...
...	...	...		...	...	...	...	...	...

Tabelle 1.4: Ein Abschnitt der reduzierten Verhaltenstabelle mit  $Leckage \stackrel{qual}{=} N$  des Druckregelventils

Im Rückwärtspropagierungsvorgang wird das Verhalten der Komponenten gegen die Richtung der Verbindungen im Modell analysiert. Die Modellgrößen, die in den Diagnoseparametern enthalten sind, befinden sich in den Ausgangsschnittstellen der Komponenten beziehungsweise in den Ausgangsschnittstellen des Systems. Der Algorithmus propagiert diese Diagnoseinformationen in entgegengesetzter Richtung entlang der Verbindungen vom Ende bis zum Anfang der Wirkkette. Hierzu werden alle Zeilen der Verhaltenstabelle der einzelnen Komponenten, die für die Erklärung der vorgegebenen Symptome nicht relevant sind, entfernt. Bei einer Wirkkette, die eine Rückkopplung oder mehrere Rückkopplungen beinhaltet, muss der Propagierungsvorgang wiederholt werden, bis keine Änderungen an den Zeilen der Verhaltenstabelle mehr existieren. Als Ergebnis entsteht für jede Komponente in der Wirkkette eine reduzierte Verhaltenstabelle. Diese reduzierte Tabelle beinhaltet zum einen den möglichen Eingangsfehler, der nicht durch die betrachtete Komponente verursacht ist, und zum anderen das mögliche Fehlverhalten der Komponente, die die Symptome verursacht.

An dem Beispiel des Kraftstoffsystems soll diese rekursive Methode zur Rückwärtspropagierung veranschaulicht werden. Aus der Verbindung zwischen der Ausgangsschnittstelle  $S_1$  des Rohres und der Ausgangsschnittstelle  $P_3$  des Systems (Abbildung 1.2) kann durch die Rückwärtspropagierung folgende Schlussfolgerung gezogen werden:

$$\text{Rohr}.S_1.Kraftstoff.\dot{V} = L$$

Mit dieser Wertebelegung kann die Verhaltenstabelle 1.3 des Rohres weiter reduziert werden:

<i>Rohr</i>					
<i>In</i>			<i>Out</i>		
$S_0$		$S_1$	<i>Variable</i>		
<i>Kraftstoff</i>		<i>Kraftstoff</i>			
$\dot{V}$	...	$\dot{V}$	...	<i>Leckage</i>	...
$L$	...	$\mathbf{L}$	...	$\mathbf{N}$	...

Tabelle 1.5: Weitere Reduzierung der Verhaltenstabelle durch Wertebelegung der Ausgangsschnittstelle des Rohres ( $\text{Rohr}.S_1.Kraftstoff.\dot{V} \stackrel{qual}{=} L$ )

Die Wertebelegungen an der Eingangsschnittstelle  $S_0$  des Rohres in der Tabelle 1.5 wird durch die Rückwärtspropagierung der Ausgangsschnittstelle  $S_5$  des Druckregelventils weitergereicht. Es gilt:

$$\begin{aligned} & ((\text{Rohr}.S_0.Kraftstoff.\dot{V} = L) \wedge \\ & (\text{Rohr}.S_0.Kraftstoff.\dot{V} = \text{Druckregelventil}.S_5.Kraftstoff.\dot{V})) \\ \Rightarrow & \text{Druckregelventil}.S_5.Kraftstoff.\dot{V} = L \end{aligned}$$

Analog zum Rohr reduziert sich die Verhaltenstabelle 1.4 des Ventils zur Verhaltenstabelle 1.6:

<i>Druckregelventil</i>							
<i>In</i>				<i>Out</i>			
$S_0$		$S_3$	$S_5$		$S_6$		<i>Variable</i>
<i>Kraftstoff</i>		<i>Steuersignal</i>	<i>Kraftstoff</i>		<i>Kraftstoff</i>		
$\dot{V}$	...	<i>Signal</i>	$\dot{V}$	...	$\dot{V}$	...	<i>Leckage</i> ...
...	...	...	...	...	...	...	...
$N$	...	$L$	$\mathbf{L}$	...	$N$	...	$\mathbf{N}$ ...
$L$	...	$N$	$\mathbf{L}$	...	$L$	...	$\mathbf{N}$ ...
$N$	...	$N$	$\mathbf{L}$	...	$L$	...	$\mathbf{N}$ ...
$L$	...	$N$	$\mathbf{L}$	...	$N$	...	$\mathbf{N}$ ...
$N$	...	$L$	$\mathbf{L}$	...	$H$	...	$\mathbf{N}$ ...
...	...	...	...	...	...	...	...

Tabelle 1.6: Weitere Reduzierung der Verhaltenstabelle des Druckregelventils durch Rückwärtspropagierung (Diagnoseparameter  $\text{Druckregelventil}.S_5.Kraftstoff.\dot{V} \stackrel{qual}{=} L$ )

Nach dem Beenden der Rückwärtspropagierung wird die Vorwärtspropagierung gestartet. Im Vorwärtspropagierungsvorgang wird die Verhaltenstabelle der Komponenten entlang der Rich-

tung der Verbindungen im Modell reduziert. Dieser Vorgang ist analog zur Rückwärtsverketung, die Unterschiede sind die Propagierungsrichtung und dass sich die Modellgröße in den Eingangsschnittstellen der Komponenten des Systems befindet.

In dem Beispiel des Kraftstoffsystems steht die Diagnoseinformation für die Vorwärtspropagierung aus dem gemessenen Volumenstrom des Volumenstromsensors. Gemäß dem Messergebnis ist der Volumenstrom an der Ausgangsschnittstelle  $S_1$  in Ordnung:

$$\text{Volumenstromsensor.S}_1.\text{Kraftstoff}.\dot{V} = N$$

Die Wertebelegungen an der Ausgangsschnittstelle  $S_1$  des Volumenstromsensors wird durch die Vorwärtspropagierung der Eingangsschnittstelle  $S_0$  des Druckregelventils weitergereicht. Es gilt:

$$\begin{aligned} \text{Volumenstromsensor.S}_1.\text{Kraftstoff}.\dot{V} &= \text{Druckregelventil.S}_0.\text{Kraftstoff}.\dot{V} \\ \Rightarrow \text{Druckregelventil.S}_0.\text{Kraftstoff}.\dot{V} &= N \end{aligned}$$

Mit dieser Information kann die Verhaltenstabelle (Tabelle 1.6) des Druckregelventils weiter reduziert werden. Als Ergebnis besteht die Tabelle des Ventils nach der Vorwärtspropagierung aus folgenden Zeilen:

<i>Druckregelventil</i>								
<i>In</i>			<i>Out</i>					
$S_0$	$S_3$		$S_5$		$S_6$		<i>Variable</i>	
<i>Kraftstoff</i>	<i>Steuersignal</i>		<i>Kraftstoff</i>		<i>Kraftstoff</i>			
$\dot{V}$	...	<i>Signal</i>	$\dot{V}$	...	$\dot{V}$	...	<i>Leckage</i>	...
...	...	...	...	...	...	...	...	...
<b>N</b>	...	<i>N</i>	<b>L</b>	...	<i>L</i>	...	<b>N</b>	...
<b>N</b>	...	<i>L</i>	<b>L</b>	...	<i>N</i>	...	<b>N</b>	...
<b>N</b>	...	<i>L</i>	<b>L</b>	...	<i>H</i>	...	<b>N</b>	...
...	...	...	...	...	...	...	...	...

Tabelle 1.7: Relationen des Volumenstroms ( $\dot{V}$ ) in der reduzierten Verhaltenstabelle des Druckregelventils bei Vorwärtspropagierung (Diagnoseparameter  $\text{Druckregelventil.S}_0.\text{Kraftstoff}.\dot{V} \stackrel{qual}{=} N$ )

Bei der Bewertung der verdächtigen Komponenten mit Bewertungspunktzahlen werden alle Komponenten in der Wirkkette mit einer reduzierten Verhaltenstabelle untersucht (**D2.2**). Dabei bekommen die Komponenten, welche mit den meisten Zeilen die Symptome durch ihre eigenen internen Variablen erklären können, höhere Verdächtigungspunkte als Komponenten, die die Symptome nur teilweise oder nicht erklären können. In dem Beispiel bekommt das Druckregelventil eine höhere Bewertungspunktzahl als das Rohr beziehungsweise die Rücklaufleitung.

Die Koordination der Sensoren- und Messeinsätze sowie der Komponententests (**D3**) gestaltet sich einfach, wenn weitere Systemsensoren existieren, die noch nicht ausgelesen wurden.

Da das Auslesen der Systemsensoren mit geringerem Aufwand als externe Messungen beziehungsweise Tests oder Prüfungen zu bewerkstelligen ist, werden zuerst die restlichen relevanten Systemsensoren ausgelesen. Die Ermittlung von möglichen externen Messaktionen und Komponententests mit Ranking beruht auf einem einfachen Algorithmus, welcher als Eingabe die Defektwahrscheinlichkeit der Komponenten und die dabei entstehenden Testkosten bei Prüfung der jeweiligen Komponente sowie die entstehenden Messkosten bei Durchführung von Messungen auswertet. Dieser Vorgang ist trivial und gehört zu den Standardaufgaben der Optimierung.

### 1.3.3 Modellierungsapplikation

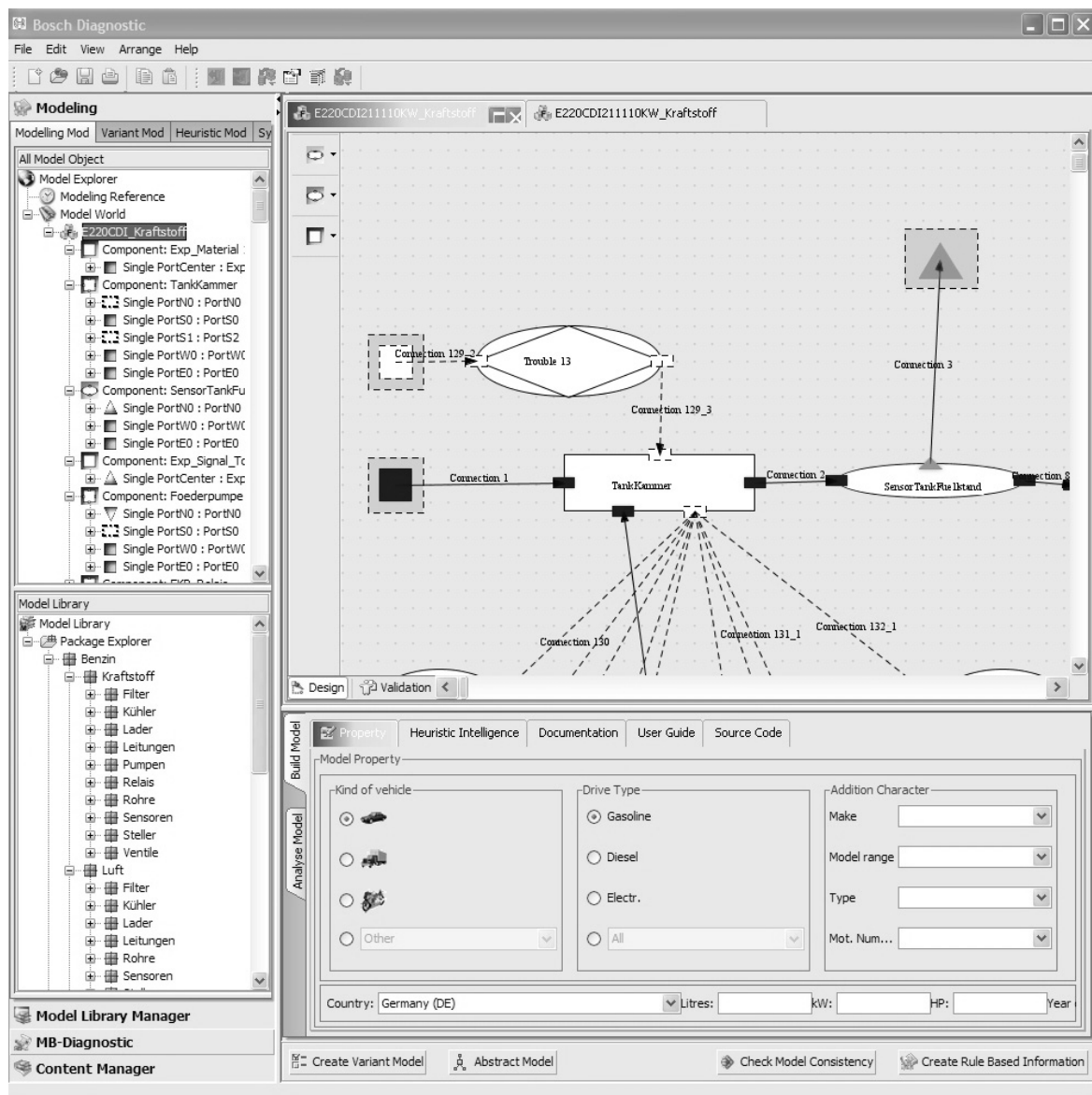


Abbildung 1.4: Entwickelte Modellierungsapplikation (DMB-System)

Der Wissenserhebungsansatz, der in Abschnitt 1.3.1 beschrieben ist, wurde in dieser Arbeit umgesetzt. Die entwickelte Modellierungsapplikation wird im Folgenden als DMB-System (Diagnostic Model Builder System) bezeichnet. Aufgrund des Umfangs sowie der Komplexität

der entwickelten Applikationen wird hierbei nur eine grobe Beschreibung der Funktionen der Applikation gegeben. Aus Benutzersicht ist die Oberfläche der Modellierungsapplikation in vier Bereiche aufgeteilt. Im rechten oberen Bereich in Abbildung 1.4 befindet sich ein Modellbrowser, in welchem der Benutzer unter Verwendung des Navigationsbaums die einzelnen Elemente des erstellten Modells anwählen kann. Die Modell- und Komponentenbibliothek befindet sich im rechten unteren Bereich der Benutzeroberfläche und beinhaltet Komponenten-, Teilsystem- und Systemmodelle, welche per Drag & Drop in den grafischen Editor zur Erstellung eines neuen Modells eingefügt werden können. Die Modellbibliothek verfügt über Suchfunktionen, die neben Namen- und Volltextsuche weitere Suchfunktionen wie Suche über Kriterien, Metadaten beziehungsweise Kategorien beinhaltet. Diese erweiterte Suchfunktion steigert die Effizienz bei der Modellerstellung und reduziert damit den Erstellungsaufwand.

Im linken oberen Bereich der Benutzeroberfläche befindet sich ein grafischer Editor, in welchem die Struktur des realen Systems als ein Objektdiagramm abgebildet wird. Die Systemkomponenten, Systemschnittstellen sowie weitere Elemente des Systemmodells werden durch unterschiedliche Symbole visualisiert. Die Zusammenhänge zwischen den Elementen werden im grafischen Modell durch gerichtete Verbindungslinien zwischen den Schnittstellen der Elemente dargestellt. Neben den Funktionen zur Visualisierung der Systemkomponenten, Beschreibung der Zusammenhänge der Komponenten und Definition der Systemgrenze bietet dieser grafische Editor die Möglichkeit zur Darstellung der Strukturvarianten; dabei wird die Layertechnik umgesetzt. Durch Aus- und Einblenden der einzelnen Layer wird die grafische Darstellung der verschiedenen Varianten des Systemmodells angezeigt.

Für die Editierung von Struktur und Verhalten ist der linke untere Bereich zuständig (Abbildung 1.5). Dabei werden die Anzahl und die Position der Schnittstellen der Komponente über eine eigene Eingabemaske definiert. Außerdem müssen für jede einzelne Schnittstelle ein oder mehrere Transportobjekte festgelegt werden, wobei jedes Transportobjekt ein oder mehrere Attribute hat, welchen physikalische Größen zugeordnet sind. All diese Informationen werden ebenfalls in dieser Maske spezifiziert. Zusätzlich kann eine Komponente eine oder mehrere interne Variablen besitzen, über welche die Eingangsgrößen mit den Ausgangsgrößen verknüpft werden. Die Eigenschaften der internen Variablen werden in einer separaten Eingabemaske in diesem Bereich festgelegt. Zur Beschreibung des Verhaltens einer Komponente werden die Komponentenschnittstellen sowie die internen Variablen über ein qualitatives Gleichungssystem miteinander verknüpft. Die Definition der Gleichungen erfolgt in einer separaten Maske. In dieser Maske wird darüber hinaus auch eine Verhaltenstabelle dargestellt, welche aus den Gleichungen abgeleitet wird.



The screenshot displays the Modeler software interface for configuring a component named 'Kraftstoffhochdruckpumpe'. The interface is divided into several panes:

- Behavior Explorer:** Shows a hierarchical tree of components and variables, including 'Kraftstoffhochdruckpumpe', 'PortW0', 'Kraftstoff', 'Druck', 'Massenfluss', 'Temperatur', 'Verschmutzungsgrad', 'Wassergehalt', 'Zuendwilligkeit', 'PortW1', 'PortW0', 'Rotation', 'Drehmoment', 'Drehwinkel', 'Drehzahl', 'Internal variable', 'Mechanischer Fehler', 'Foerderleistung', 'PortE0', 'Kraftstoff', 'Druck', 'Massenfluss', 'Temperatur', 'Verschmutzungsgrad', 'Wassergehalt', 'Zuendwilligkeit', 'PortE1', 'Kraftstoff', 'Druck', 'Massenfluss', 'Temperatur', 'Verschmutzungsgrad', 'Wassergehalt', 'Zuendwilligkeit'.
- Specification:** Contains the following code:
 

```

1 this.PortW0.Kraftstoff.Druck +
2 this.PortW0.Rotation.Drehzahl*this.Foerderleistung-this.MechanischerFehler=
3 this.PortE0.Kraftstoff.Druck;
4
5 this.PortW0.Kraftstoff.Massenfluss * this.Foerderleistung-this.MechanischerFehler=
6 this.PortE0.Kraftstoff.Massenfluss;
7
8 this.PortW0.Kraftstoff.Temperatur = this.PortE0.Kraftstoff.Temperatur;
9
10 this.PortW0.Kraftstoff.Verschmutzungsgrad =
11 this.PortE0.Kraftstoff.Verschmutzungsgrad;
12
13 this.PortW0.Kraftstoff.Wassergehalt = this.PortE0.Kraftstoff.Wassergehalt;
14
15 this.PortW0.Kraftstoff.Zuendwilligkeit = this.PortE0.Kraftstoff.Zuendwilligkeit;

```
- Internal Variable:** Lists 'Mechanischer Fehler' and 'Foerderleistung'.
- Working State:** Shows states for 'Working Start State' (cold start, warm start, all start) and 'Working Load State' (idle running, partial load, full load, all load).
- Property Table:** A table with columns for 'Drehzahl', 'Druck', 'Massenfluss', 'Temperatur', 'Mechanischer Fehler', 'Foerderleistung', and 'PortE1 Kraftstoff'. The table contains 15 rows of data with values like 'normal', 'low', 'high', and 'all load'.

Abbildung 1.5: Verhaltenseditor von DMB-System

### 1.3.4 Diagnose-Runtime-System

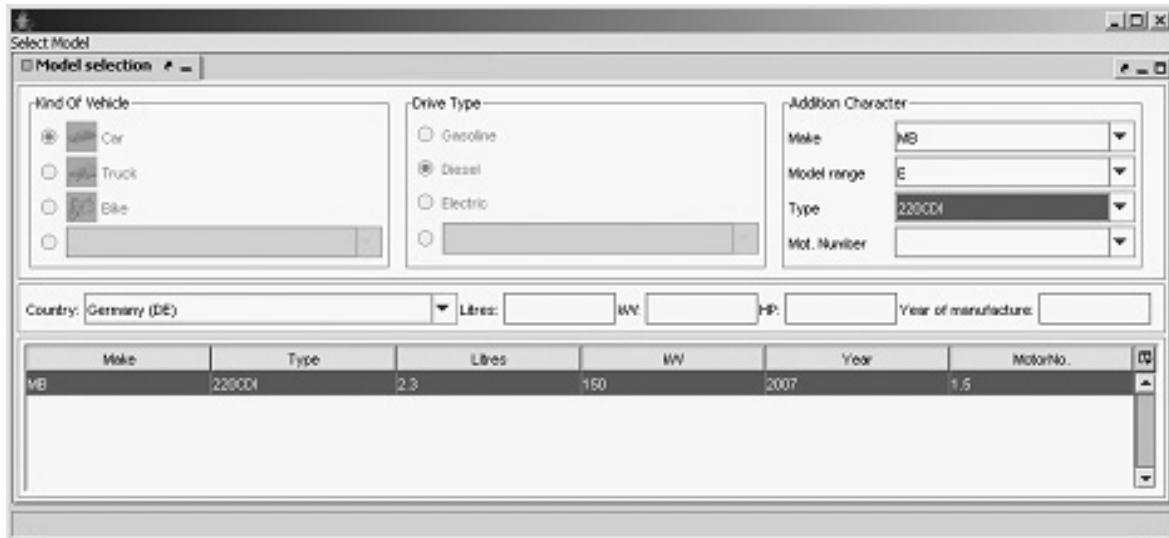


Abbildung 1.6: Auswahl des Modells mit DRT-System

Der Inferenzmechanismus, der in Abschnitt 1.3.2 beschrieben ist, wurde in dieser Arbeit umgesetzt. Das daraus entstehende Diagnoselaufzeitsystem wird im Folgenden als DRT-System (Diagnostic Runtime System) bezeichnet. In diesem Abschnitt werden die Ergebnisse der Realisierung aus Benutzersicht vorgestellt. Nach dem Start von DRT-System sucht dieses in der entsprechenden Datenbank nach abgelegten Modellen. Diese werden, nachdem aus ihnen die wichtigsten Informationen extrahiert wurden, zur Auswahl angezeigt (Abbildung 1.6). In der oberen Hälfte der Oberfläche ist es möglich, über verschiedene Filter das entsprechende Modell auszuwählen. Die Diagnose beginnt mit der Eingabe der Informationen zum Fahrzeug beziehungsweise mit der Systemidentifikation. Dazu werden zunächst der Fahrzeugtyp und die Antriebsart gewählt. Anschließend kann über Marke, Modell, Typ und Motorkennung die Auswahl eingegrenzt werden. Im unteren Bereich werden die Modelle mit Informationen angezeigt, die den gewählten Filterkriterien entsprechen. Über einen Doppelklick auf das entsprechende Modell wird dieses Modell geladen. Nachdem das Modell ausgewählt und geladen wurde, erscheint eine andere Bedienoberfläche im Benutzerinterface. Diese ist in Abbildung 1.7 zu sehen. Die wesentlichen Elemente der Oberfläche von DRT-System sind eine Liste mit sämtlichen verdächtigen Komponenten mit Bewertungspunkten, die zum aktuellen Zeitpunkt vorliegenden Diagnoseinformationen zur Eingrenzung der Fehlerursachen, ein Workflow mit den weiteren Prüfschritten als Handlungsempfehlung für das Werkstattpersonal sowie eine Übersicht mit zusätzlichen Hintergrundinformationen. Zu Beginn des eigentlichen Diagnosevorgangs werden die Diagnoseinformationen durch das Werkstattpersonal im linken unteren Bereich eingegeben. Über den ersten von vier verschiedenen Tabulatoren werden die Symptome oder Kundenbeanstandungen aus einer vordefinierten Liste ausgewählt. In einem separaten Fenster werden nach der Auswahl einer Kundenbeanstandung detaillierte Informationen angezeigt. Diese Informationen dienen zur Symptomerklärung, da die Überführung des Symptoms in verschiedene Abweichungen der Modellgrößen im Modell gespeichert ist. Im zweiten Tabulator werden die automatisch aus den Steuergeräten ausgelesenen Fehlercodes aufgelistet. Analog zur Kundenbeanstandung wird in einem Fenster nach der Auswahl eines Fehlercodes die zugehörige Interpretation in textueller Form angezeigt.

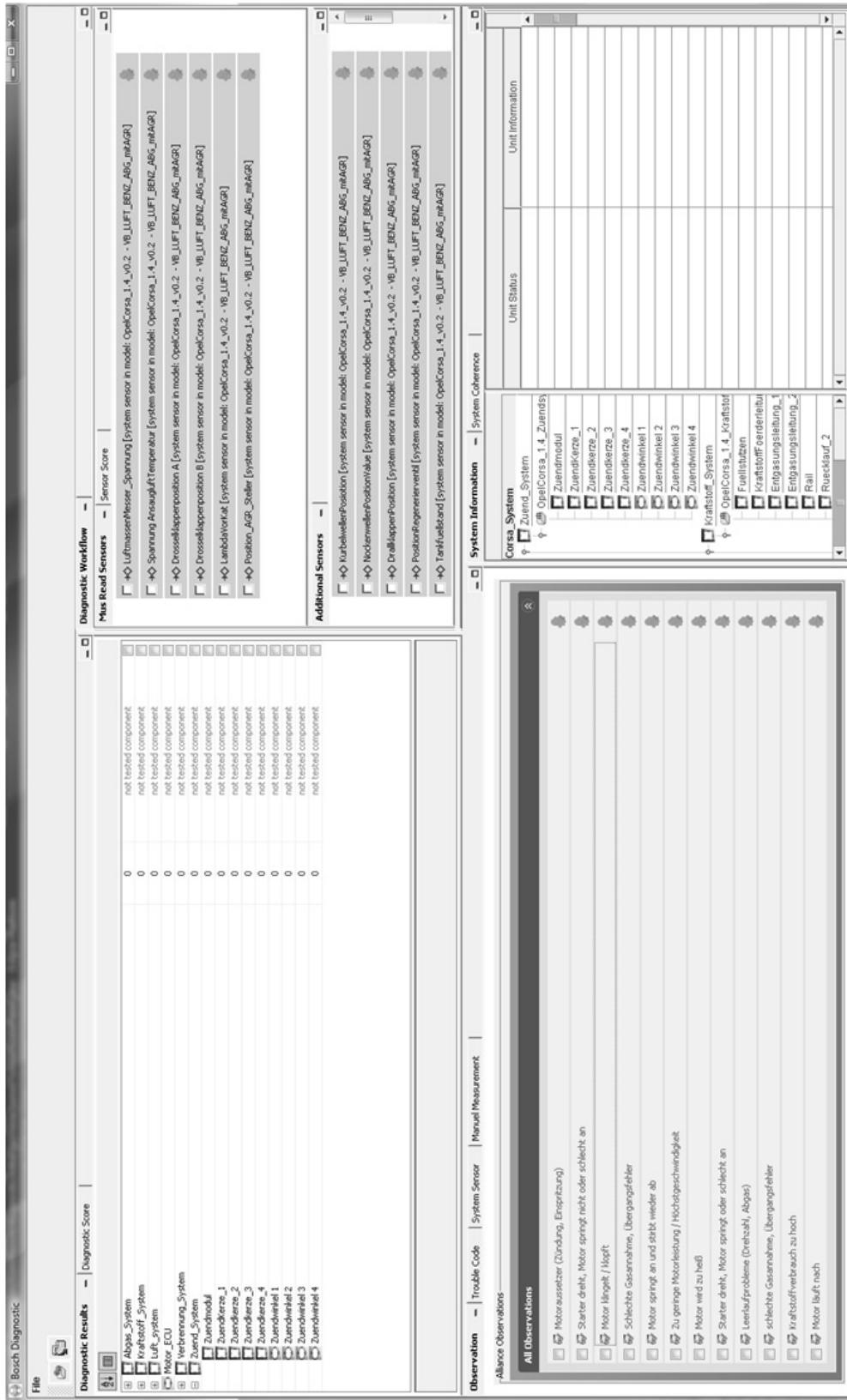


Abbildung 1.7: Entwickeltes DRT-System

Der dritte Tabulator enthält eine Auflistung aller verfügbaren Systemsensoren, von denen einzelne ausgewählt werden können. Nach dem Auswählen eines Sensors wird automatisch eine Messung mittels des angeschlossenen Diagnosegerätes durchgeführt. Das Messergebnis wird anschließend in einem Fenster dargestellt. Eine automatische Auswertung des Messergebnisses erfolgt durch den Abgleich zwischen Istwert und Sollwertbereich. Eine manuelle Auswertung ist trotzdem möglich und obliegt der Entscheidung des Werkstattpersonals. Der vierte Tabulator beinhaltet sämtliche externen Messmöglichkeiten und Stellgliedtests, von denen wiederum nur ein Teil für die Diagnose ausgewählt wird. Die Anzeige für das externe Messergebnis ist analog zu der Anzeige des Systemsensors.

Die Eingrenzung der Fehlerquelle ist ein iterativer Prozess, bei welchem nach den durchgeführten Messungen, Stellgliedtests beziehungsweise Komponententests die am meisten verdächtigen Komponenten von DRT-System vorgeschlagen und in der Regel vom Werkstattpersonal überprüft werden. Das Ergebnis der Überprüfung einer einzelnen Komponente wird in jedem Iterationsschritt DRT-System zur Verfügung gestellt. Das Testergebnis kann mittels eines Dialogs des Benutzerinterface aufgenommen werden. Die Ergebnisse der Stellgliedtests oder der Messungen durch systemeigene Sensoren werden automatisch bewertet und aufgenommen.

Für die Bewertung des Fehlerzustands einer Komponente werden neben den bisher genannten Informationen weitere Angaben über das Alter der Komponente genutzt, welche einen Hinweis auf die Ausfallwahrscheinlichkeit liefern. Die Eingabe dieser zusätzlichen Komponenteneinformationen erfolgt über den rechten unteren Bereich in Abbildung 1.7.



Abbildung 1.8: Maske zur Darstellung der Handlungsempfehlung

In Abbildung 1.8 ist die Maske der Handlungsempfehlung für Werkstattmitarbeiter dargestellt. Dabei sind Messungen und Prüfungen in sortierter Liste abgebildet. Es existiert eine Rückkopplung zwischen der Handlungsempfehlung und allen vier Tabulatoren im linken unteren Bereich der Abbildung 1.7. Diese Kopplung ist wichtig, da ein bereits ausgewählter Sensor oder eine Prüfung in beiden Bereichen maskiert werden sollen. Die getrennte Dar-

stellung der Handlungsempfehlung und der Liste aller Messungen und Prüfungen lässt dem Werkstattpersonal die Handlungsfreiheit und erhört damit die Benutzerakzeptanz.

## 1.4 Überblick über die Evaluierungsergebnisse

In diesem Abschnitt werden die wesentlichen Ergebnisse der Evaluierung beschrieben. Dabei wird auf zwei Aspekte eingegangen:

- Aufwandsabschätzung für die Erstellung und Pflege von Diagnosemodellen mit der entwickelten Modellierungsapplikation,
- Diagnosegüte bei der geführten Fehlersuche unter Verwendung der erstellten Diagnosemodelle und des Runtime-Systems.

### 1.4.1 Aufwand bei der Modellbildung

Mit der Modellierungsapplikation (DMB-System) wurden ein Common Rail CP3-System (Diesel), eine Saugrohreinspritzung ME 7.x-System (Otto) und ein Lichtsystem ALWR 1.0 (automatische Leuchtweiteregelung) modelliert und deren Komponenten parametrisiert beziehungsweise bedatet. Dabei wurde der Aufwand protokolliert und mit den vorhandenen Daten der zurzeit gängigen Redaktionssysteme verglichen. Der Aufwand für nicht modellierte Anteile der Fehlersuchanleitung (wie zum Beispiel Sicherheitshinweise, Bildrecherche, Fahrzeugerprobung, Administration) wurden mit dem gleichem Aufwand wie bei den vorhandenen Redaktionssystemen bewertet. Die Evaluierung hat gezeigt, dass sich mit dem entwickelten DMB-System ein Teil des Aufwands für die Wissenserhebung einsparen lässt. Die Aufwandsreduzierung bei der Wissenserhebung beträgt im Schnitt über 25%. Diese Aussage kann durch einen grundlegenden Vergleich des Erstellprozesses ermittelt und durch die Analyse der während der Modellierung gesammelten Daten für drei Systeme verifiziert werden.

Bei der Modellierung von Fahrzeugsystemen beziehungsweise Fahrzeugfunktionen wird prinzipiell zwischen der Modellierung der Struktur (Vernetzung einzelner Subsysteme und Komponenten untereinander), der Modellierung des Verhaltens (internes Verhalten der Komponenten) und der Parametrisierung (Spezialisierung und Bedatung der Modelle) unterschieden. Dabei ist die Modellierung eines neuen Modells als Leitprojekt zu bezeichnen, wenn:

- Die Vernetzung der Subsysteme und Komponenten neu erstellt werden muss;
- Für mehr als 30%<sup>2</sup> der Komponenten Verhalten neu hinterlegt werden sollen.

Enthält ein Modell nicht alle Subsysteme oder Komponenten des realen Systems, so muss das Modell um diese Komponenten beziehungsweise Subsysteme erweitert werden. Das erweiterte Modell kann dann als neues Basismodell für die Ableitung des vorangegangenen Modells genutzt werden. Wenn die Anzahl der erweiterten Komponenten mehr als 30% der gesamten Komponentenanzahl beträgt, ist die Aktivität zur Erzeugung des Modells als Leitprojekt zu bezeichnen. Sind für die Modellierungen eines Systems weitere Schnittstellen der Subsysteme

---

<sup>2</sup>Der Richtwert 30% ist aus verschiedenen Modellierungsworkshops mit Fehlersuchleitungsautoren, die später DMB-System benutzen sollen, entstanden

oder der Komponenten nicht oder nicht passend vorhanden, so muss eine Überarbeitung der Modellierung des Systems oder der Komponenten erfolgen. Sie lässt sich meistens aus den bestehenden ableiten. Wenn die Anzahl der neuen Schnittstellen mehr als 30% beträgt, wird die Erstellung des Modells ebenfalls als Leitprojekt betrachtet.

Eine Art des Folgeprojekts ist ein Projekt, bei dem die Aktivität auf die Parametrisierung der Modelle beschränkt ist. Die Parametrisierung beinhaltet zum Beispiel die Änderung der Sollwerte oder der Einbaulagen sowie Prüfkosten oder Ausfallwahrscheinlichkeiten. Weitere Arten des Folgeprojekts betreffen die Erzeugung des Modells:

- Nutzen die umgebenden Systeme nicht alle Schnittstellen des betreffende Subsystems oder der Komponenten, so lassen sich durch geeignete Parametrisierung die ungenutzten Schnittstellen für die Diagnose deaktivieren.
- Enthält ein existierendes Systemmodell Subsysteme oder Komponenten, die im zu modellierenden System nicht vorhanden sind, so lassen sich diese Subsysteme oder Komponenten ebenfalls durch geeignete Parametrisierung deaktivieren.
- Wird die Abdeckung von zusätzlichen internen Variablen einer Komponente oder eines Subsystems erforderlich, so ist eine Anpassung oder Erweiterung der Verhaltensmodellierung erforderlich.
- Wird eine interne Variable nicht benötigt, so lässt sich diese Variable durch geeignete Parametrisierung deaktivieren.

Hochgerechnet auf den Gesamtdatenbestand ist die Ersparnis (über 25%) nur für gleiche Verhältnisse von Leit- und Folgeprojekten gültig. Für die Erstellung des Diagnosewissens mit DMB-System sind allerdings wesentlich weniger Leitprojekte erforderlich. Dafür steigt der Anteil der Folgeprojekte. Die Folgeprojekte erfordern einen wesentlich geringeren Aufwand. Daher kann ein erheblich höheres Einsparpotential bei der Erstellung vieler Systeme erwartet werden.

Die Abschätzung des Gesamtaufwands für die Umstellung des Redaktionssystems auf DMB-System erfolgte durch weitere Aktivitäten, zum Beispiel Modellierungsworkshops. Als Ergebnisse existieren folgende Auswertungen von den Autorentams über die Fähigkeiten des entwickelten Ansatzes:

- Machbarkeit: Erstellung der Diagnosemodelle mit vorhandenem Fachwissen und mit der Modellierungsapplikation möglich,
- Effizienz: Zeit- und Kostenersparnis für das Einpflegen der Daten in die Datenbank durch komponentenorientierten Ansatz,
- Durchgängigkeit: Nutzbarkeit der bestehenden Daten sowie Zukaufdaten,
- Überprüfbarkeit: Automatische Test mit vordefinierten Testfällen am Rechner,
- Know-how-Aufbau: Modellierung der Wirkkette ermöglicht weiteren Aufbau des Expertenwissens bei Autoren,
- Robustheit: Sensitivität bezüglich Serienstreuung, Übertragbarkeit auf andere Fahrzeuge, Integrierbarkeit unterschiedlicher Informationsquellen,

- Variantenvielfalt: Wiederverwendbarkeit von Komponenten und Systemen durch Schichtkonzept.

### 1.4.2 Ergebnisse der Feldtests

Ein weiteres Ziel der Evaluierung war die Erprobung des entwickelten, modellbasierten Diagnosesystems (DRT-System) am Realsystem und die Bewertung seiner Leistungsfähigkeit in Bezug auf Diagnosegüte und Diagnosedauer. Als Referenz wurden dafür marktführende Werkstattdiagnosesysteme herangezogen. Um einen objektiven Vergleich bei der Diagnose zu garantieren, wurden die Prüfschritte genau nach Vorgabe gegangen. Damit wurde auf die Erfahrungswerte des Werkstattpersonals verzichtet. Diese Einschränkung war nötig, weil bei der Evaluierung des Diagnosemodells kein heuristisches Wissen enthalten ist. Da die eingebauten Fehler unterschiedliche Situationen in der Werkstatt widerspiegeln sollten, wurden Fehler, die Fehlercodes provozieren, und Fehler, die keinen Fehlercode provozieren, eingebaut. Die Fehlercode provozierenden Fehler sind von Fehlercodes, die direkt auf fehlerhafte Komponenten und auf Systemgrößen zeigen, zu unterscheiden. Zusammenfassend können folgende Aussagen getroffen werden:

- Die Anzahl der Prüfschritte im entwickelten Diagnosetool liegt im Durchschnitt 29% unter der von vorhandenen Tools.
- Bei der Prüfzeit wird im Durchschnitt 45% eingespart.

Neben der Steigerung der Effizienz bei der Fehlersuche existieren weitere Vorteile des entwickelten DRT-Systems gegenüber existierenden Diagnosetools. Folgende Aussagen wurden nach den Feldtests und Live-Demos von Werkstattmitarbeitern, Trainern aus einem Schulungszentrum und Mitarbeitern aus dem Marketing getroffen:

- Effizienz: Schnelles Auffinden gewünschter Informationen durch komponenten- und funktionsorientierten Ansatz,
- Systemverständnis: Die Darstellung der Wirkkette fördert Know-how-Aufbau und damit Verbesserung der Lernkurve der Werkstattmitarbeiter,
- Vermarktbarkeit: Sichtbarkeit der Veränderungen und des Nutzens für den Kunden, schrittweise Vermarktung einzelner Aspekte des Diagnoseansatzes möglich,
- Individualisierung der Diagnose durch: Berücksichtigung regionaler Unterschiede und Reparaturhistorie einzelner Fahrzeuge,
- Verbindung von geführter Fehlersuche und Steuergerätediagnose (Zukunftssicherheit): Automatische Durchführung von vorgeschlagenen Messungen und Stellgliedtest in optimaler Reihenfolge.

## 1.5 Gliederung der Arbeit

Auf die Problemstellung „Randbedingungen, Stand der Forschung und Technik“, die der Gegenstand der Untersuchung des Diagnoseproblems in freien Werkstätten in dieser Arbeit ist,

wird im ersten Abschnitt des Kapitels 2 eingegangen. Hier werden die grundsätzlichen Diagnoseansätze aus der Literatur besprochen und deren Vor- und Nachteile erläutert. Daraus wird das Modellierungskonzept sowie die Entwicklung eines neuen Modellierungs- und Diagnosetools abgeleitet, um der Diagnoseserviceabteilung und den herstellerunabhängigen Werkstätten neue Werkzeuge zur Beherrschung jetziger und zukünftiger Diagnoseprobleme zu geben. In Abschnitt 2.2 werden die Anforderungen und Erwartungen aus Sicht der Domäneexperten in der Serviceabteilung und aus Sicht des Werkstattpersonals besprochen. Daraus folgt im letzten Teil des Kapitels die Entscheidung der Modellform und die Anforderungsanalyse für die entwickelte Software.

Mit den Erkenntnissen aus Kapitel 2 wird in Kapitel 3 der Modellaufbau behandelt. Hierbei wird zuerst über das Komponentenmodell und die Arten der Komponenten diskutiert. Die Schnittstellen, Verbindungen und das Verhalten des Systemmodells bilden den zweiten Teil des Kapitels 3. Im letzten Teil dieses Kapitels wird auf die Anbindung von diagnoseerforderlichem, heuristischem Wissen im Modell eingegangen.

Kapitel 4 beginnt mit der Beschreibung der Modellanalysealgorithmen, die nicht nur für die Verdachtsgenerierung und Fehlerlokalisierung zuständig sind, sondern auch für die Modellüberprüfung eingesetzt werden können. Die Diagnose mittels des Diagnosealgorithmus wird anhand einiger Beispiele beschrieben.

Die Umsetzung des Modellierungsansatzes in Form einer Modellierungsapplikation (MBD-System) wird in Kapitel 5 beschrieben. Abschließend wird in diesem Kapitel auf den Ansatz und die Umsetzung der Modellqualitätssicherung, die in MBD-System eingebettet ist, eingegangen.

Im ersten Abschnitt des Kapitels 6 wird die Systemarchitektur des realisierten Laufzeitsystems (DRT-System) erläutert. Im letzten Teil dieses Kapitels werden verschiedene Möglichkeiten zur Integration des neuen Laufzeitsystems analysiert und bewertet.

Kapitel 7 stellt die Evaluierungsergebnisse des Modellierungstools und des Diagnosetools vor. Dabei soll die Effizienz des Modellierungskonzepts und des Modellierungstools im Vergleich zu einigen, derzeit eingesetzten Redaktionstools ermittelt werden. Ein weiterer Punkt ist der Vergleich zwischen dem Diagnosetool und einigen der besten, gegenwärtig auf dem Markt befindlichen herstellerunabhängigen Diagnosetools. Dies dient zur Untermauerung der Tragfähigkeit des Modellierungskonzepts und der Diagnosealgorithmen.

Mit der Zusammenfassung und einem Ausblick in Kapitel 8 endet die vorliegende Arbeit.



## 2 Werkstattdiagnose im Kraftfahrzeugbereich

Wegen der zunehmenden Fahrzeugkomplexität und der abnehmenden Beherrschbarkeit des Gesamtfahrzeugsystems müssen entlang des gesamten Lebenslaufes eines Fahrzeugs verschiedene Prüf- und Diagnoseanwendungen durchgeführt werden:

- **Entwicklung:** Die Prüfmöglichkeiten in dieser Phase beinhalten die Überprüfungen der Hardware und Software des zu entwickelnden Systems. Die Ansteuerung von Stellgliedern und das Lesen von analogen und digitalen Eingängen werden eingesetzt, um die Funktionsfähigkeit der Hardware zu überprüfen. Bei dem Softwaretest werden die Ansteuerungen und das Austesten von internen Testroutinen, Fehlercodesetz- und Rücksetzbedingungen sowie von Reaktionen auf länderspezifische Einstellungen durchgeführt. Bereits während der Entwicklung eines technischen Systems können Fehlermöglichkeiten, ihre Auswirkungen und mögliche konstruktive Maßnahmen zur Fehlervermeidung untersucht werden. Hierfür werden Methoden wie die Ereignisablaufanalyse, die Fehlerbaumanalyse und die Fehlermöglichkeits- und Einflussanalyse (FMEA) eingesetzt. Die entstehende FMEA kann bei der Fertigung (Massenproduktion) und beim Service für den Funktionstest der einzelnen Komponenten oder Komponentengruppen eingesetzt werden.
- **Fertigung:** Um die Qualität des ausgelieferten Produkts zu gewährleisten, werden in der Fertigung speziell entwickelte Prüftechniken eingesetzt. In der Massenfertigung müssen nicht nur Endkontrollen durchgeführt, sondern Prüfschritte in die Fertigungslinien integriert werden, da Qualitätsprobleme, die erst bei der Endkontrolle festgestellt werden, schnell zu großen Ausschussmengen und damit zu hohen Fertigungskosten führen. Bei der Fertigung der Kraftfahrzeuge ist neben der Verbauprüfung (zum Beispiel Identifizierung des Steuergerätes, Ermittlung der aktuellen Konfiguration, Ansteuern von Stellgliedern und Auslesen von Istwerten) der Funktionstest im verbauten Zustand ein wesentliches Prüfmittel. Dabei wird das Zusammenwirken aller Steuergeräte in der vorliegenden Fahrzeugkonfiguration geprüft. Zusätzlich müssen die beim Verbau entstandenen Fehlercodes gelöscht und die Ermittlung noch bestehender Fehler durchgeführt werden.
- **Service:** Im Service ist der Diagnoseprozess in zwei Hauptteile aufgeteilt: Betrieb und Wartung. Bei der Prozessdiagnose im Fahrbetrieb, auch Onboard-Diagnose genannt, sollen Fehler während des Betriebs möglichst schnell erkannt werden, um unerwünschte, insbesondere sicherheitskritische Zustände verhindern zu können. Die Verarbeitung der gemessenen Signale muss in Echtzeit durchgeführt werden, so dass geeignete Korrekturmaßnahmen noch während des Betriebs ergriffen werden können. Weitere Diagnoseprobleme ergeben sich bei der Wartung und Reparatur eines technischen Systems. Hier wird von der Werkstattdiagnose (Offboard-Diagnose) gesprochen. Im Unterschied

zur Betriebsdiagnose können zusätzliche Messgrößen aufgenommen werden, damit die Messsignale nicht in Echtzeit verarbeitet werden müssen. Außerdem können zusätzliche Informationen aus Systembenutzerbeanstandungen, Ausfallwahrscheinlichkeiten der Komponenten sowie Testkosten und Reparaturkosten herangezogen werden.

Die zu entwickelnde Diagnoseanwendung in dieser Arbeit soll in freien Werkstätten ihren Einsatz finden, daher sind die nachfolgend beschriebenen Randbedingungen, Anforderungen und der Stand der Technik sowie die Forschungen nur auf die Offboard-Diagnose bezogen. Zunächst werden die Anforderungen, Charakteristika und Randbedingungen der Werkstattdiagnose erläutert. Nach der Präzisierung der Problemstellung wird auf existierende Diagnoseverfahren, oft auch als Klassifikationsverfahren bezeichnet, eingegangen und diese werden diskutiert. Abschließend werden die Wissenserhebung und die Diagnosemechanismen der modellbasierten Diagnose vertieft.

## 2.1 Randbedingungen und Anforderungen

Die erfolgreiche Einführung eines neue Lösungsansatzes für die Werkstattdiagnose hängt von verschiedenen Faktoren ab. Die Funktionalitäten der neu entwickelnde Applikationen sollen nicht nur die funktionalen Anforderungen abdecken, sondern auch die nicht funktionalen Anforderungen erfüllen. In diesem Abschnitt werden die Randbedingungen, und damit die abgeleiteten Anforderungen aus Sicht der Autoren des Servicebereichs, die das Diagnosewissen aufbereiten und aus Sicht der Anwender, die in der Werkstatt die Diagnose durchführen, beschrieben.

### 2.1.1 Wissenserhebung für die Werkstattdiagnose

Die Werkstattdiagnoseproblematik im Kraftfahrzeugsektor zeichnet sich durch die hohe Geschwindigkeit technologischer Neuerungen aus. Es ist auf längere Sicht mit der Verwendung neuer Komponententypen oder neuartiger Realisierungen zu rechnen. Außerdem gibt es durch die Variantenvielfalt zum gleichen Zeitpunkt eine Fülle von Modifikationen, hervorgerufen durch unterschiedliche Einbausituationen in unterschiedlichen Fahrzeugtypen. Folglich erfordert es eine hohe Flexibilität und Modularität des Wissenserhebungsystems. Das System sollte einerseits geeignet sein, Wissen aus verschiedenen Anwendungsbereichen darzustellen, andererseits sollten aber auch, unabhängig vom Anwendungsgebiet, die für unterschiedliche Aufgabenstellungen benötigten Informationen effizient darstellbar sein. Die Wissensbasis sollte möglichst einfach veränderbar und erweiterbar sein. Dies ist am einfachsten zu erreichen, wenn die Wissensbasis modular aufgebaut ist. Das bedeutet, spezielle Bereiche des Wissens, wie etwa das Wissen über die Lösung eines bestimmten Teilproblems, sollten möglichst unabhängig vom Rest der Wissensbasis hinzugefügt oder verändert werden können.

Die Informationen, die bei der Wissensakquise einfließen, sind entscheidend für die Diagnosegüte in dem wissensbasierten Diagnosesystem. Ein signifikanter Unterschied zwischen der Wissensakquise der Serviceabteilung der Fahrzeughersteller (OEMs) und der unabhängiger Diagnoseanbieter sind die verfügbaren Informationen über die zu diagnostizierenden Systeme. Trotz klarer gesetzlicher Bestimmungen ist die praktische Durchführung der Kfz-Gruppenfreistellungsverordnung (vgl. [65], [90]) nicht zufriedenstellend für freie Werkstätten. Die hohen Preise der Hersteller-Diagnosetools und die eingeschränkte Verfügbarkeit von

Diagnosedaten für unabhängige Diagnoseanbieter beschränken die Fahrzeugabdeckung. Unabhängige Anbieter von Diagnosetools erhalten nur vereinzelte Fahrzeuginformationen, weil bestimmte OEMs kaum Serviceinformationen, Reparaturinformationen beziehungsweise diagnoserelevante Informationen zur Verfügung stellen und wenig Informationen über aktuell entdeckte Fahrzeugfehler oder Rückrufaktionen von Seiten der OEMs existieren.

Der stetige Anstieg der Auslagerungsaktivitäten bei allen OEMs und eine stärkere Fokussierung auf Kernkompetenzbereiche wirken sich negativ auf den Informationsgehalt der gelieferten Diagnosedaten aus. Für eine systemübergreifende Diagnose und darauffolgende Reparatur reicht dieser Informationsgehalt oft nicht aus, da das Wissen über Funktionen sowie Verhalten der Komponenten, Systeme und Subsysteme nicht immer transparent ist. Die Problematik der gerade erwähnten Fahrzeugabdeckung wird zusätzlich durch die Modell- und Variantenvielfalt verstärkt. Dabei beeinflussen folgende Faktoren die Steigerung der Variantenvielfalt (vgl. [11]):

- Marktsättigung der wichtigsten Märkte in Europa und USA zwingt OEMs zu einer weiteren Marktsegmentierung und stärkt den Wettbewerb zwischen den OEMs
- Kundenwünsche fordern die Fahrzeugindividualisierung und Mikrosegmentierung des Marktes
- Suche der OEMs nach neuen Marktnischen.

Die steigenden Anforderungen an die Modell- und Variantenvielfalt beeinflussen die Modellzyklen sowie die Fahrzeugentwicklungszeiten und damit die Qualitätssicherung des Fahrzeugs. Weil bei kürzeren Entwicklungszeiten mit einer erhöhten Fehleranzahl zu rechnen ist, können viele Fehler erst im Feld entdeckt werden. Damit steigt die Anforderung der Werkstattdiagnose an den Informationsfluss über aktuell häufig entdeckte Fahrzeugfehler einer bestimmten Fahrzeugserie aus dem Feld.

Aus den oben beschriebenen Randbedingungen folgen einige Konsequenzen für den zentralen Servicebereich, welcher für die Einführung und Pflege des Diagnosesystems zuständig ist. Diese Konsequenzen sind gleichzeitig die Anforderung an das zu entwickelnde Autorensystem, das für die Wissenserhebung zuständig ist. Folgende Punkte gelten als hauptfunktionale Anforderungen an das Wissenserhebungstool:

- Integration des bisherigen Wissens: Die existierenden Diagnoseinformationen und deren Fehlersuchlogik für ältere Produktreihen sollen durch die Weiterentwicklungen für neue Produktreihen nicht beeinflusst werden. Bei der Einführung des neuen Lösungsansatzes soll es eine Co-Existenz der älteren Ansätze und des neuen Ansatzes geben, sonst ist der Umstellungsaufwand zu hoch.
- Modularisierung des Diagnosewissens: Weil das Diagnosewissen aus verschiedenen Quellen stammt, mit unterschiedlichen Kompetenzen bearbeitet wird und diese Erhebung zeitversetzt durchgeführt werden kann, soll das Diagnosewissen modularisiert werden. Die Vorteile der Modularisierung sind zum einen die Verkürzung der Akquisezeit und zum anderen eine Lösung der Variantenvielfaltsproblematik.
- Systemübergreifende Diagnose: Die klassischen Vorgehensweisen bei der Erhebung des Diagnosewissens stoßen immer mehr an die Grenzen der Diagnosemöglichkeiten, da die

Komponenten- und Systemsichten nicht immer zu einer erfolgreichen Diagnose führen. Viele Fahrzeugfunktionen sind nur systemübergreifend realisierbar. Zum Beispiel ist bei den meisten Fahrzeugen die ESP-Funktion abhängig von der Motorlast, welche von dem Motorsystem geliefert wird. Ohne die Möglichkeit über die Systemgrenze hinaus zu diagnostizieren ist die Lösung weder zukunftsorientiert noch nachhaltig.

- Beherrschung der Variantenvielfalt: Da ein wesentliches Kriterium für die Güte des herstellerunabhängigen Diagnosesystems die Beherrschung von Produktvarianten ist, sollte idealerweise die Fehlersuchlogik für eine Baureihe nur einmal erstellt werden müssen und damit auch die Varianten abgedeckt sein.
- Automatische Prüfbarkeit des Diagnosewissens: Die eingegebenen Informationen, welche für den Diagnosevorgang in der Werkstatt erforderlich sind, müssen in einer Form bereitgestellt werden, die eine unkomplizierte und effiziente Überprüfung ermöglichen.

Für eine erfolgreiche Einführung des neuen Systems im Servicebereich muss das System neben der Erfüllung der funktionalen Anforderungen folgenden nicht funktionalen Anforderungen entsprechen:

- Einhaltung der Prozessschritte: Der bisher etablierte Wissenserhebungsprozess und dessen Arbeitsschritte sollen durch eventuelle zusätzliche Schritte, die durch den neuen Lösungsansatz erforderlich sind, nicht stark beeinflusst beziehungsweise geändert werden. Es besteht sonst die Gefahr, dass die Fachmitarbeiter im Servicebereich die neue Lösung ablehnen.
- Wissenserhebung mit bestehender Struktur: Der neue Lösungsansatz darf nicht dazu führen, dass das bisherige Erfahrungswissen der Fachmitarbeiter unbrauchbar wird. Daher muss das vorhandene Erfahrungswissen in die neue Diagnosefunktion integriert werden können. Des Weiteren muss die Bearbeitung der erforderlichen neuen Diagnoseinformationen unkompliziert und mit der bestehenden Mannschaft und deren Struktur zu bewältigen sein.
- Aufwand und Nutzen: Das Verhältnis zwischen Aufwand (Wissenserwerb, Modellbildung, Wissensumfang) und Nutzen (Detaillierungsgrad, Vollständigkeit, Kosten) muss berücksichtigt werden und soll wirtschaftlich tragbar sein. Das heißt, dass der Aufwand der Modellerstellung nicht höher als der jetzige Erstellungsaufwand sein darf.

Neben der genannten Problematik im Bereich der Wissensakquise existiert eine Reihe von Anforderungen an das Diagnosesystem, das im Feld zum Einsatz kommen soll. Im nächsten Abschnitt wird auf diese Anforderungen eingegangen. Dabei wird das gesamte Umfeld der Arbeit durchleuchtet und ein klares Bild erschaffen.

### 2.1.2 Das Diagnoselaufzeitsystem in der Werkstatt

Der Druck der Wirtschaftlichkeit zur Kosteneinsparung zeigt sich vor allem in modernen Kraftfahrzeugen an der begrenzten Zahl zusätzlicher Sensoren. Es werden zwar immer mehr Sensoren eingesetzt, diese werden aber im Wesentlichen für die Fahrfunktion entwickelt, so dass für die Diagnosezwecke geeignete Größen oftmals nicht gemessen werden können. Es werden nicht immer alle für die Diagnose gewünschten Größen gemessen.

Aus Kostengründen werden die für die Diagnosefunktionen benötigten Rechenleistungen, der Speicherplatz und die Kommunikationskapazität des Steuergeräts möglichst gering gehalten. Folglich besitzen die Sensoren nur eine beschränkte Auflösung, Messgenauigkeit und die einzelnen für die Diagnose verwendeten Sensorsignale werden nur in festgelegten Abtastzeiten bereitgestellt. Durch temporäre Systemausfälle, zum Beispiel durch einen Wackelkontakt können diese Abtastzeiten jedoch nur bedingt eingehalten werden. Dies führt zu weiteren Messungenauigkeiten. Aufgrund der rauen Umgebungsbedingungen werden die Sensorsignale oftmals durch ein Rauschen überlagert. Zudem variieren zahlreiche Parameter bei Änderung der Umweltbedingungen, bei Massenfertigung und durch alterungsbedingte Bauteiltoleranzen. Dadurch kann das Systemverhalten selbst bei genauer Kenntnis der physikalischen Zusammenhänge nicht immer exakt bestimmt werden. Die Korrelationen zwischen verschiedenen Signalen können vielfach nur grob formuliert werden. Aus solchen inhärenten Unsicherheiten und Unvollständigkeits der Informationen durch Sensorsignale werden unsichere Schlussfolgerungen gezogen. Wegen dieser Randbedingungen reicht die Abdeckung der Onboard-Diagnose nur bis zur systemeigenen Grenze und ist bei der Diagnose von gekoppelten Systemen nicht immer hilfreich. Zudem sind die Diagnoseinformationen aus der Onboard-Diagnose nur bei elektrischen Fehlern zuverlässig; bei Fehlern die nicht durch Kurzschluss, Massenschluss, etc. entstehen, ist die Diagnose oft nicht aussagekräftig.

Aufgrund der beschriebenen Randbedingungen bestehen die funktionalen Anforderungen an das Diagnosesystem aus vier Schwerpunkten:

- **Korrektheit:** Die Diagnose soll korrekt sein. Es sollte nicht passieren, dass Fehlerursachen nicht gefunden werden oder, was noch schlimmer ist, dass falsche Ursachen gefunden werden, die zum Austausch von fehlerfreien Teilen am Fahrzeug führen, ohne dass damit der eigentliche Fehler beseitigt wurde.
- **Vollständigkeit:** Die Diagnose soll vollständig sein. Alle Ursachen von Fehlern sollen gefunden werden, damit eine Instandsetzung des Fahrzeugs innerhalb einer einzigen Diagnosesitzung möglich ist.
- **Prävention:** Nach erfolgreicher Reparatur sollte das Fahrzeug über einen längeren Zeitraum keine Ausfälle zu verzeichnen haben. Idealerweise sollten außer den regelmäßigen Wartungen überhaupt keine Standzeiten zu verzeichnen sein. Es ist daher von Vorteil, wenn während einer Diagnosesitzung zusätzliche Messergebnisse des Fahrzeugs ausgewertet werden können, die Hinweise auf in Kürze zu erwartende Ausfälle liefern. In diesem Fall können präventive Maßnahmen ergriffen werden.
- **Kostengünstigkeit:** Die Diagnose soll kostengünstig sein. Dies fordert im Wesentlichen, dass die Ursache eines Fehlers mit möglichst wenigen Diagnoseschritten gefunden wird. Außerdem sollen während der Diagnosesitzung keine umfangreichen Einstellungen oder Montagen am Fahrzeug durchgeführt werden. Das entscheidende Kriterium ist immer die Zeit, die für eine Diagnosesitzung aufzuwenden ist.

Neben den funktionalen Anforderungen existieren weitere nicht funktionale Anforderungen, zum Beispiel:

- **Lernorientierter Erklärungsansatz:** Neben der Diagnosefähigkeit soll die Erläuterungskomponente des Diagnoselaufzeitsystems die Lernkurve des Anwenders positiv beeinflussen.

Dazu gehören der Wissenstransfer und -aufbau durch genaue Erklärungen der Funktionen und der Wirkketten des zu diagnostizierenden Systems. Die offene Weitergabe diagnoserelevanten Wissens zwischen Experten und Werkstattpersonal schafft einen zusätzlichen Anreiz für die Wissensteilung des Werkstattmitarbeiters über die entwickelte Feedback-Funktion.

- **Optimale Navigation:** Die diagnoserelevanten Informationen, die aus verschiedenen Quellen stammen, sollen in einem kundenorientierten Kontext verknüpft und dargestellt werden. Diese Anforderung ermöglicht eine optimale Benutzbarkeit und Zugänglichkeit des Diagnosewissens und damit eine optimale Nutzerführung.
- **Intuitive Bedienoberfläche:** Für den Werkstattservicetechniker ist eine unkomplizierte Bedienung des Diagnosesystems während des Einsatzes von großer Wichtigkeit. Insbesondere bei rechnergestützten Diagnosesystemen muss die Bedienung so einfach und unmissverständlich sein, dass es zu diesem Zweck keines Computerspezialisten bedarf.
- **Rechenressourcen:** Das Diagnosesystem soll mit geringen Rechenressourcen auskommen, weil nicht davon ausgegangen werden kann, dass die Rechner in den freien Werkstätten über große Leistung und Speicherkapazität verfügen.

## 2.2 Stand der Forschung und Technik

In diesem Abschnitt wird zuerst eine Übersicht über die vorhandenen Diagnoseansätze gegeben. Anschließend werden einige der bisher veröffentlichten Arbeiten auf dem Gebiet der funktionalen modellbasierten Diagnose kurz dargestellt. Danach werden die derzeit vorhandenen Lösungen der Werkstattdiagnose mechatronischer Systeme und Komponenten beschrieben.

### 2.2.1 Bestehende Diagnoseverfahren

Die Einteilung der Diagnoseansätze kann nach Wissensarten erfolgen. Demnach lassen sich die Ansätze in fünf Klassen je nach dem zugrundeliegenden Diagnosewissen unterteilen (vgl. [4], [113], [145], [146], [6]):

- **Sichere Diagnose:** Bei diesem Diagnoseansatz wird das sichere Diagnosewissen in Form von Entscheidungsbäumen oder Entscheidungstabellen repräsentiert. Die Auswertung des Diagnosewissens erfolgt durch Beantwortungen der hierarchischen Fragen zu den beobachteten Symptomen. Hier werden keine komplizierten Problemlösungsansätze benötigt, da die Antworten direkt zu einer Aussage über die Fehlerursache führen können.
- **Heuristische Diagnose:** Der heuristische Ansatz basiert auf der Auswertung von assoziativem Diagnosewissen, das in Form von Heuristiken, also Erfahrungsregeln (*Wirkung*  $\Rightarrow$  *Ursache*), in einer Regelbasis repräsentiert wird. Die Regeln beschreiben direkt und oft ohne Begründung, welche Symptomkombinationen mit welcher Sicherheit auf welche Ursache hindeuten. Diese Regeln können jedoch mit Unsicherheiten behaftet sein und jede Wirkung kann verschiedene Ursachen haben. Daher kommen in einer konkreten Diagnosesituation üblicherweise mehrere Regeln zur Anwendung. Um eine Aussage über die wahrscheinlichste Ursache machen zu können, müssen diese Regeln geeignet verrechnet

werden. Für die Diagnose wird dann die Ursache mit der im Hinblick auf die beobachteten Symptome höchsten Bewertung herangezogen. Eine weitere Möglichkeit mit unsicherem Wissen umzugehen, ist die Anwendung von Fuzzy-Regeln. Bei ihnen wird anstelle der scharfen Zuordnung von numerischen Merkmalswerten zu symbolischen Größen eine unscharfe Zuordnung benutzt. Nach der Fuzzyfizierung der Merkmale und deren Verarbeitung durch Fuzzy-Regeln wird ein entsprechendes Ergebnis mit einer Zuordnung zu gegebenenfalls mehreren Diagnosen ausgegeben. Durch Defuzzyfizierung kann dieses Ergebnis auf eine einzelne, am stärksten zutreffende Diagnose reduziert werden.

- **Statistische Diagnose:** Das Diagnosewissen in diesem Ansatz wird durch die statistische Auswertung von Fallsammlungen erstellt. Demzufolge existieren zunächst keine direkten Beziehungen zwischen Symptomen und Diagnosen. Die Ermittlung dieser Beziehungen erfolgt aus der Berechnung der Wahrscheinlichkeit, wie oft ein Symptom mit einer Diagnose in Verbindung steht. Die Problemlösungsmethode für diesen Ansatz ist das Bayes-Theorem oder die Dempster-Shafer-Theorie. Dieses Theorem lässt sich nur bei speziellen Problemstellungen anwenden. Eine breite Anwendung wird durch restriktive Voraussetzungen, insbesondere die Forderung einer statistischen Unabhängigkeit der Symptome und des gegenseitigen Ausschlusses von Fehlern sowie durch eine exponentielle Laufzeitkomplexität in der Anzahl der möglichen Fehler verhindert. Das Bayes-Netz ist ein probabilistischer Diagnoseansatz, welcher das Diagnosewissen über wahrscheinliche und unwahrscheinliche Fehlerauswirkungen in den Diagnoseregeln voraussetzt. Somit hängt die Diagnosegüte direkt von der ermittelten Wahrscheinlichkeit ab, welche nur garantiert werden kann, wenn die Falldatenbasis eine hinreichend große Auswahl typischer Fehlersituationen enthält, um das Fehlerspektrum geeignet abdecken zu können. Der Aufbau der Falldatenbasis im herstellerunabhängigen Kraftfahrzeugservicebereich hat sich als schwierig erwiesen, da zum einen die freien Werkstätten ihr Wettbewerbswissen ungern untereinander austauschen und zum anderen die erhobenen Daten für andere Zwecke benutzt werden können.
- **Fallbasierte Diagnose:** Die Grundidee des fallbasierten Diagnoseansatzes ist es, zur Lösung eines neuen Problems auf Diagnosewissen zurückzugreifen, das in Form von gelösten Aufgaben des gleichen Problembereichs vorliegt und in einer sogenannten Falldatenbasis gesammelt wird ([5], [17]). Dabei besteht jeder Fall aus einer Problembeschreibung und der zugehörigen Lösung, der Diagnose. Liegt eine Diagnosesituation vor, wird aus der Falldatenbasis ein vergleichbarer, möglichst ähnlicher Fall herausgesucht und dessen Diagnose auf das aktuelle Problem übertragen. Zwei Faktoren sind entscheidend für die Problemlösungsfähigkeit von fallbasierten Diagnosesystemen: Zum einen die Anzahl und die Qualität der Fälle in der Falldatenbasis und zum anderen die Güte des sogenannten Ähnlichkeitsmaßes. Der Grad der Ähnlichkeit beziehungsweise die Übereinstimmung der alten Fälle mit dem aktuellen Fall wird mit dem Ähnlichkeitsmaß berechnet. Die Bestimmung des Ähnlichkeitsmaßes erfordert für komplexe kausale Zusammenhänge in der Regel zusätzliches Wissen. Außerdem muss dem Problembereich eine gewisse „Stetigkeit“ zugrunde liegen, damit die Übertragbarkeit der Lösung des alten Falls auf das aktuelle Problem gewährleistet ist.
- **Modellbasierte Diagnose:** Bei diesem Diagnoseansatz existiert eine Unterteilung der Problemlösungsmethode in überdeckende und funktionale Klassifikation. Im Unterschied zur heuristischen Diagnose steht das assoziative Diagnosewissen der *überdeckenden Klassifikation* in der Form *Ursache*  $\Rightarrow$  *Wirkung*. Die Gewichtung der Ursachen bei existie-

renden Wirkungen (Symptomen beziehungsweise Merkmalen) hängt von der vollständigen Erklärungsfähigkeit gemäß der zuvor definierten kausalen Regeln ab. Die Assoziation zwischen Ursache und Wirkung kann durch Zwischenzustände detailliert beschrieben werden. Diese Möglichkeit verbessert die Gewichtung bei sich überlagernden Ursachen, weil die Systemgrößen (Parameter) mit ihren Wertenbelegungen verknüpft werden können und so detailliertes Diagnosewissen für die Erklärungskomponenten liefern. Der Nachteil jedoch ist, dass, je höher der Detaillierungsgrad, desto ineffizienter die Diagnoselaufzeit ist. Dieses Defizit kann durch ein Verfeinerungskonzept vermindert werden. In diesem Konzept wird das detaillierte Wissen eines Zustands in Form von Zwischenzuständen nur bei Bedarf in dem Diagnosevorgang aufgerufen und bearbeitet. Die Gewichtung der Ursachen kann durch Anreicherung des Zusatzwissens über Wahrscheinlichkeiten verbessert werden. Wegen der graphischen Darstellungsmöglichkeit der Kausalregeln in der überdeckenden Klassifikation und der Berücksichtigung der Zustandwahrscheinlichkeit können die Inferenzalgorithmen mit Hilfe der Methode des Bayes-Netzes realisiert werden.

Bereits während der 80er Jahre wurden einige entscheidende Arbeiten zur *funktionalen Klassifikation* veröffentlicht (vgl. [46], [61], [126], [30]). Dabei wurden viele Forschungen mit Anwendungsdomänen im Bereich der digitalen Schaltungen durchgeführt. In den folgenden Jahren wurden die Anwendungen in mehreren Arbeiten auf verschiedene Domänen ausgedehnt (vgl. [20], [92], [144]). In diesem Diagnoseansatz sind die zur Diagnose notwendigen Informationen in Form eines Modells gespeichert. Dabei enthält das Modell explizite Informationen über Struktur und Verhalten des betrachteten Systems und dessen Komponenten. Die Strukturinformationen eines Modells beinhalten die Angabe über verbaute Komponenten im System und die Verbindungen beziehungsweise Zusammenhänge zwischen den Komponenten. Die Benutzung der Relationsgleichungen von Ein- und Ausgängen sowie der internen Variablen zur Formulierung des Verhaltens des technischen Systems und dessen Komponenten ist hinlänglich bekannt. In verschiedenen Arbeiten und Veröffentlichungen [78], [87], [66], [92], [6] werden relationale Modelle technischer Systeme vorgestellt und zur Lösung der Diagnoseaufgaben verwendet. Bei den Ansätzen in oben genannten Veröffentlichungen besteht das Verhalten einer Komponente nicht nur aus korrektem sondern auch aus fehlerhaftem Verhalten. Diese Art der komponentenorientierten Modellierung (vgl. [71], [102], [103], [104], [105], [47]) ähnelt sehr stark der objektorientierten Modellierung bei der Softwareentwicklung. Daher können die Mechanismen zur Komplexitäts- und Variantenbeherrschung durch Vererbung, Generalisierung und Spezialisierung in diesem Diagnoseansatz an die allgemeine objektorientierte Modellierung angelehnt werden. Beispielsweise wird das Verhalten der Modellkomponenten durch Relationsgleichungen beschrieben und damit die Möglichkeit geschaffen, einen Abstraktionsmechanismus zu entwickeln, um eine Allgemeingültigkeit des Wissens über Komponenten mit gleichen Funktionen zu gewinnen, die sich damit zur Wiederverwendung eignet. Das grundlegende Prinzip der Fehlerlokalisierung in der funktionalen modellbasierten Diagnose ist prinzipiell in verschiedenen Veröffentlichungen immer ähnlich (vgl. [46], [61], [126], [126], [93], [67], [130], [132], [133], [134], [135], [136], [27], [29], [30], [31], [32], [125], [100]). Ausgehend von einer Menge der aktuellen Symptome des realen Systems können mit Hilfe des Modells Vorhersagen über das Verhalten dieses Systems gemacht werden. Wenn das Modell nur das Nominalverhalten des Systems beschreibt, liegt offenbar mindestens ein Fehler im System vor, da eine Abweichung zwischen den Vorhersagen des Nominalverhaltens von



den tatsächlichen Beobachtungen beziehungsweise Symptomen existiert. Eine Diagnose wird unter Zuhilfenahme von Diagnosewissen, das im Modell enthalten ist, erstellt. Dabei handelt es sich um Hypothesenmengen von Systemkomponenten, die fehlerhaft sein können. Die Kardinalität der Verdächtigenmenge ist abhängig von der Nützlichkeit der aktuellen Symptome beziehungsweise Merkmale. Je weniger hilfreich die verfügbaren Merkmale zum Ausschluss verdächtiger Komponenten sind, desto größer ist die Kardinalität der Hypothesenmengen. Die Reduzierung der Hypothesenmengen beziehungsweise Eingrenzung der fehlerhaften Komponenten kann durch das Anfordern weiterer Merkmale realisiert werden. Die zusätzlichen Merkmale entstehen durch weitere Messungen beziehungsweise zusätzliche Überprüfungen am realen System. Wenn das Modell Fehlerzustände beinhaltet, kann sich die erste Ermittlung der potenziell verdächtigen Komponenten an der abduktiven Vorgehensweise orientieren (vgl. [78], [127], [66]). Wenn keine vormodellierten Komponentenfehler die Symptome erklären können beziehungsweise bei der Komponentenprüfung die verdächtigen Komponenten sich als fehlerfrei erweisen, wird bei der weiteren Diagnosevorgehensweise weitgehend die oben beschriebene Strategie der Hypothesenerstellung und Hypothesenprüfung verfolgt.

In Laufe der Forschung im Diagnosebereich entstanden unterschiedliche Strategien, die sich als Standardtechniken zur Verarbeitung des Diagnosewissens herauskristallisiert haben. Dabei kann die Strategie „Hypothesenerstellung und Hypothesenprüfung“ als allgemein gültige Vorgehensweise gesehen werden. Weitere Strategien wie „Vorwärtsverkettung“, „Rückwärtsverkettung“ oder „Etablierung der hierarchischen Diagnoseverfeinerung“ sind Spezialfälle von „Hypothesenerstellung und Hypothesenprüfung“ (Abbildung 2.1). Die Vorgehensweise dieser Strategie kann wie folgt beschrieben werden:

---



---

**Algorithmus 2.1** : Hypothesenerstellung und Hypothesenprüfung
 

---



---

**Eingabe:**

$SYMPTOM^{input}$  - Menge der Symptome, die am Anfang des Diagnosevorgangs existieren

**Ausgabe:**

$DIAGNOSE$  - Menge der Diagnosen, die existierende Symptome erklären

**Algorithmus:**

$istDiagnoseBestaetigt = false;$

**while**( $\neg istDiagnoseBestaetigt$ ) **do**

$DIAGNOSE = hypotheseAufstellung(SYMPTOM^{input});$

$isApproved = hypotheseBestaetigung(DIAGNOSE);$

**if** ( $\neg istDiagnoseBestaetigt$ ) **then**

$SYMPTOM^{add} := neueSymptomeAufforderung$   
( $SYMPTOM^{input}, DIAGNOSE$ );

$SYMPTOM^{input} := symptomFusionierung$   
( $SYMPTOM^{input}, SYMPTOM^{add}$ );

**end**

**end**

$return\ DIAGNOSE;$

Die Fehlerdiagnose mit der Strategie „Hypothesenerstellung und Hypothesenprüfung“ ist ein iterativer Vorgang. Hierbei ist der Eingangspunkt eine Menge der existierenden Symptome

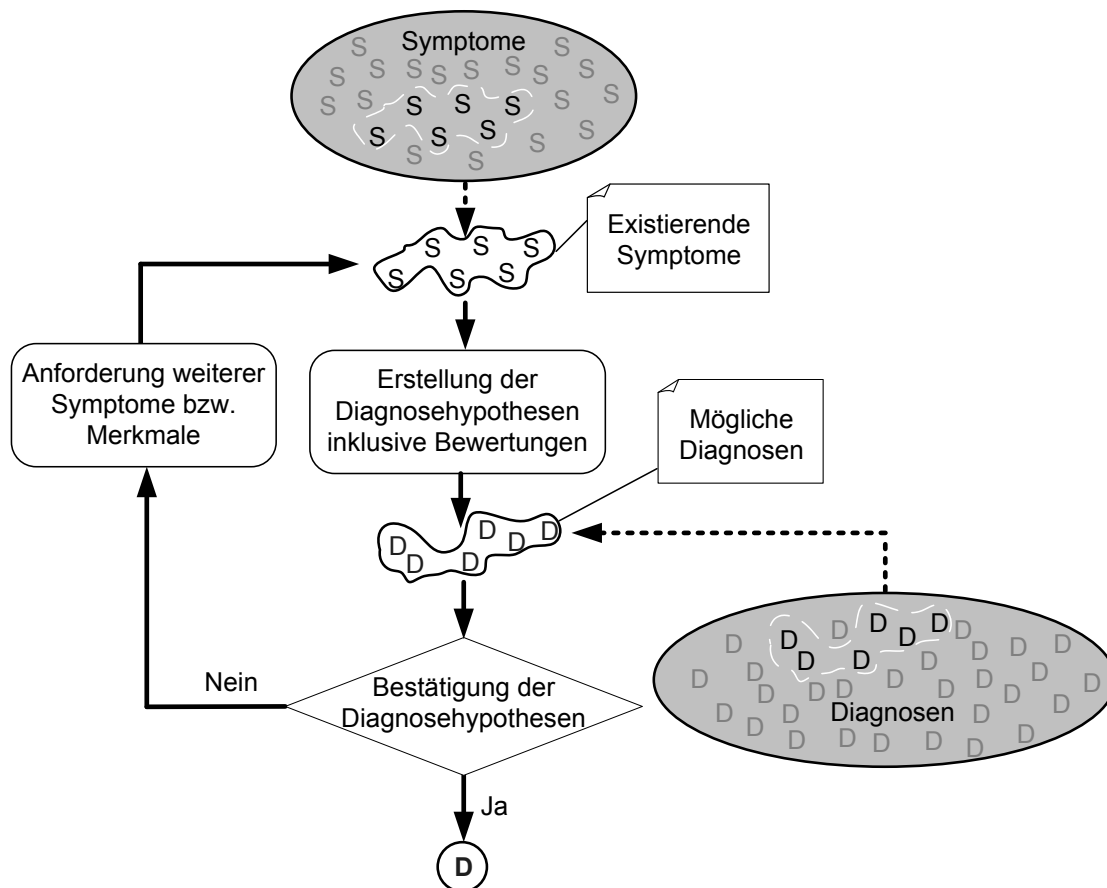


Abbildung 2.1: Prozess der Strategie „Hypothesenerstellung und Hypothesenprüfung“

und das Diagnosewissen, das im Basiswissen zur Verfügung steht. Die Erkennung und Bestätigung der Ursache und deren Behebung gilt als Ausgangspunkt der Diagnose. In der Methode *hypotheseAufstellung* werden die vorhandenen Symptome beziehungsweise Merkmale zunächst untersucht, danach werden Hypothesen basierend auf dem Diagnosewissen über Ursachen aufgestellt. Die Methode *hypotheseBestaetigung* liefert das Ergebnis, ob die hypothetische Ursache bestätigt beziehungsweise nicht bestätigt ist. Bezogen auf Kraftfahrzeugwerkstattendiagnose beinhaltet die Überprüfung das Auslesen weiterer systemeigener Sensorwerte, die Durchführung weiterer externer Messungen oder Komponententests. Wenn die Ursachen bestätigt sind, wird die Fehlerbehebung durchgeführt. Anderenfalls werden die Ergebnisse der durchgeführten Überprüfungen in der Methode *neueSymptomeAufforderung* untersucht und weitere Messungen beziehungsweise Prüfungen vorgeschlagen. Die neu gewonnenen Informationen und die existierenden Symptome werden in der Methode *symptomFusionierung* zusammengefasst und dienen zur neuen Hypothesegenerierung.

### 2.2.2 Existierende Lösungen der Werkstattdiagnose

In Bereich der freien Kfz-Werkstattendiagnose basieren die Diagnoseverfahren auf dem heuristischen, auf dem fallbasierten Ansatz oder auf einer Kombination der beiden Ansätze. Eine Berücksichtigung des statistischen Wissens ist teilweise gegeben, wenn der Servicebereich über ein Servicefax und/ oder eine Hotline verfügt. Eine Internetlösung für Fallsammlungen in freien Werkstätten ist bis jetzt nicht gegeben. Die allgemeine Architektur all dieser Expertensys-

teme aus funktionaler Sicht ist in [111] beschrieben. In Abbildung 2.2 sind die wesentlichen Komponenten sowie Benutzerrollen des Systems dargestellt.

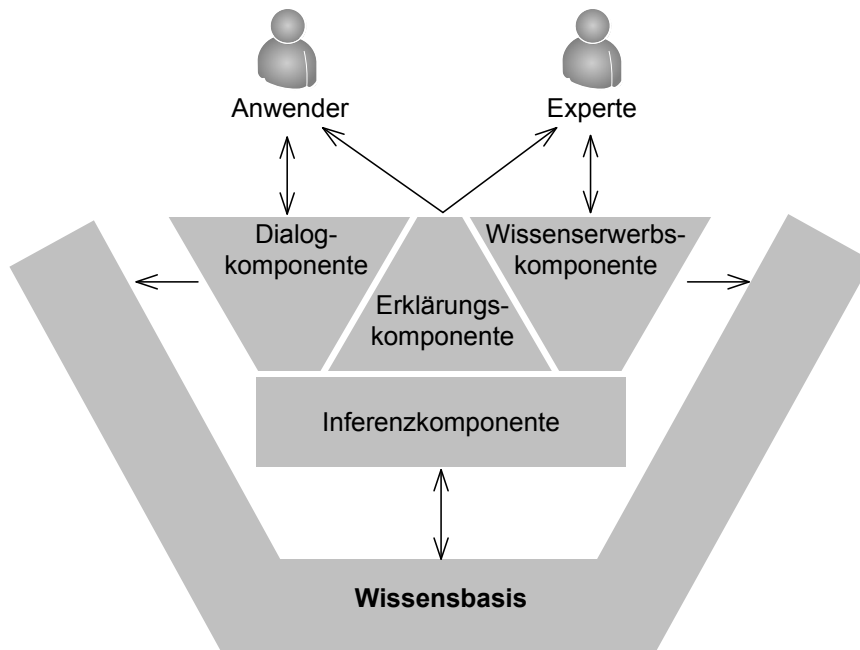


Abbildung 2.2: Architektur des Expertensystems aus funktionaler Sicht

Es existiert eine Rollenaufteilung der Benutzer:

- Experte, der das Wissen in die Wissensbasis mittels einer Wissenserwerbskomponente einpflegt. Mit der Erklärungs- und Inferenzkomponente überprüft der Experte das Wissen in der Wissensbasis.
- Anwender, der mit dem Expertensystem über die Dialogkomponente interagiert. Das System liefert die Lösung an den Anwender durch die Erklärungskomponente. Hierbei analysiert die Inferenzkomponente die Eingabeinformationen und schlussfolgert die Lösung anhand der Wissensbasis.

In diesem Abschnitt wurden zunächst die Arbeitsschritte der Experten in der Wissenserhebungsphase beschrieben. Danach wird auf den generellen Diagnoseablauf in den freien Kfz-Werkstätten eingegangen.

In Abbildung 2.3 sind alle wesentlichen Schritte in der Wissenserhebung dargestellt. Um die Vermittlung des Diagnosewissens zwischen Fachexperten und Anwender zu ermöglichen, ist der erste Schritt die Erschließung des Herstellerwissens. Die Beschaffung dieser Informationen erfolgt hauptsächlich durch Datenzukauf. Neben dieser Aktivität verwendet der Experte im Servicebereich weitere Medien wie Internet oder Literatur, um sich in das System einzuarbeiten. Da das System nicht nur in einer Sprache verfügbar ist und die existierende Terminologie nicht ausreicht, um das neue System beziehungsweise deren Komponenten zu beschreiben, muss gegebenenfalls eine neue Terminologie definiert und abgestimmt werden.

Weil die meisten zu reparierenden Fahrzeuge in freien Werkstätten mindestens drei Jahre auf dem Markt sind, ist es möglich, verschiedene Diagnosefälle zu sammeln. Eine Voraussetzung

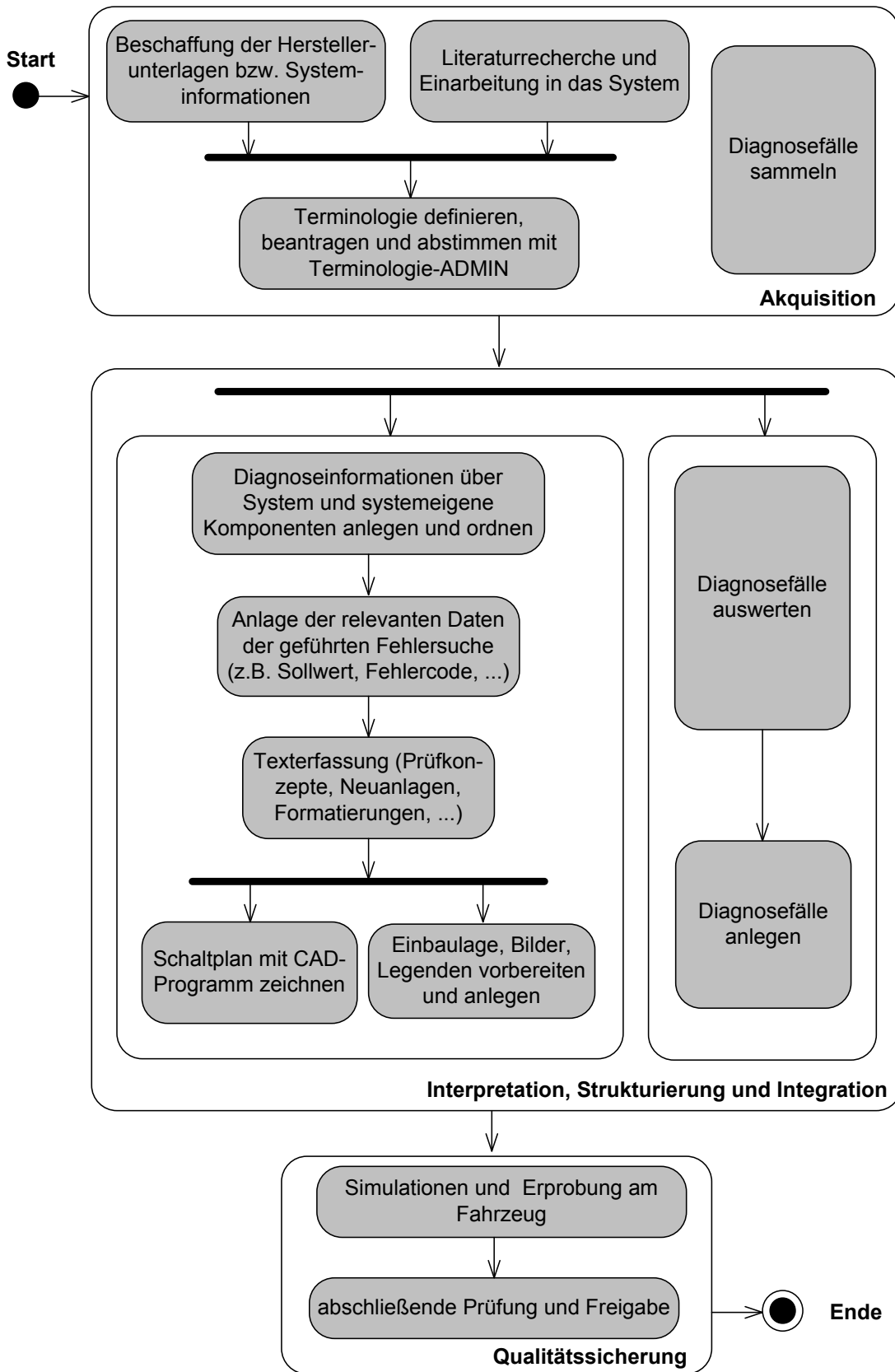


Abbildung 2.3: Wissenserhebung für die Kfz-Werkstattdiagnose

dafür sind gute Kontakte zwischen den Mitarbeitern des Servicebereichs und den Mitarbeitern in den Vertragswerkstätten.

Die zweite große Phase innerhalb der Wissenserhebungsphase ist die Interpretation und Strukturierung des zuvor gesammelten Diagnosewissens sowie dessen Integration in die Wissensbasis. Diese Aktivitäten werden mittels eines existierenden Autorensystems durchgeführt. Um eine Fehlersuchanleitung der geführten Fehlersuche zu erstellen, werden zuerst die Informationen über das System und deren Komponenten im Autorensystem angelegt. Wenn die Informationen in Datenbank existieren, werden diese ausgesucht und neu geordnet. Weitere Daten wie Sollwerte der Systemsensoren, externe Messungen, Fehlercodes und Stellgliedtests werden angelegt und auf zugehörige Komponenten verlinkt. Die Texterfassung wird für einzelne Kundenbeanstandungen erstellt, dabei werden die Prüfkonzepte in Form von festgelegten Mess- und Testreihenfolgen realisiert. Die Erfassung von Bildern und Einbaulagen der Komponenten erleichtert den Diagnosevorgang und die Reparatur in der Werkstatt und soll neben der Schaltplanzeichnung die letzte Aktivität in dieser Phase sein. Wenn fallbasierte Diagnosedaten existieren, werden diese durch den Autor ausgewertet. Relevante Fälle werden als fertige Diagnoselösungen in der Suchanleitung gespeichert.

Die dritte Phase im Wissenserhebungsprozess ist die Qualitätssicherung. Dazu wird ein Fahrzeug bestellt und es werden Fehler eingebaut. Durch eingebaute Fehler werden zum einen der Fehlercode und zum anderem die geführten Fehlersuchen geprüft. Die Abfrage der Istwerte von Systemsensoren und Überprüfungen der Sollwertbereiche werden ebenfalls in dieser Simulations- und Erprobungsaktivität durchgeführt. Die Freigabe der Fehlersuchanleitung erfolgt nach der abschließenden Überprüfung am Rechner durch das Autorenteam.

In Abbildung 2.4 ist der Prozess zur Diagnose und Reparatur in der freien Kfz-Werkstatt grob dargestellt. Dabei beginnt dieser Diagnoseprozess damit, dass ein Kunde bei der Serviceannahme eintrifft und dem Werkstattmitarbeiter seine Beanstandungen beziehungsweise Symptome am Fahrzeug darstellt. Während der Annahme werden die Umgebungsbedingungen der Symptome, Fahrzeughistorie, allgemeine Sichtprüfungen, etc. dokumentiert. Wenn ein Symptom einer bemängelten Funktion entspricht, wird das Symptom durch Kontrolle - zum Beispiel bei einer Probefahrt - reproduziert. Anschließend verbindet der Werkstattmitarbeiter den Diagnosetester an das Fahrzeug über die Diagnose-Schnittstelle und liest die Fehlerspeicher aus. Danach erfolgt die Abfrage, ob ein bekannter Fall in der Datenbank vorhanden ist. Wenn ein bekannter Fall vorliegt, wird die Reparatur nach der Vorgabe zu diesem Fall vom Werkstattmitarbeiter durchgeführt. Die Reparatur wird auf Erfolg geprüft. Sofern kein Fehler vorliegt, wird das Fahrzeug an den Kunden zurückgegeben. Falls die Reparatur nicht erfolgreich war, wird der Prozessrückschritt zum Fehlercode-Auslesen eingeleitet und dort erneut begonnen. Wenn kein bekannter Fall existiert, startet der Werkstattmitarbeiter die geführte Fehlersuche mit dem vorliegenden Fehlercode und den Symptomen. Dabei gibt der Diagnosetester die durchzuführenden Prüfschritte vor und der Werkstattmitarbeiter arbeitet diese Vorgabe ab. Wird während dieser Prüfarbeiten ein Fehler gefunden, das heißt, eine defekte Komponente festgestellt, erfolgt die Reparatur durch Umtausch beziehungsweise Instandsetzen der fehlerhaften Komponente. Anschließend werden die Fehlerspeicher gelöscht. Falls kein Fehler gefunden wurde, beginnt der nächste Prozessschritt ohne Reparatur. Um eine erfolgreiche Reparatur zu bestätigen, können zuerst die Fehlerspeicher ausgelesen und als nächstes die bemängelte Funktion kontrolliert werden. Zeigt die Reparatur Erfolg, dokumentiert der Werkstattmitarbeiter alle ausgeführten Arbeitsschritte auf dem Auftrag und bereitet das Fahrzeug für die Rückgabe vor.

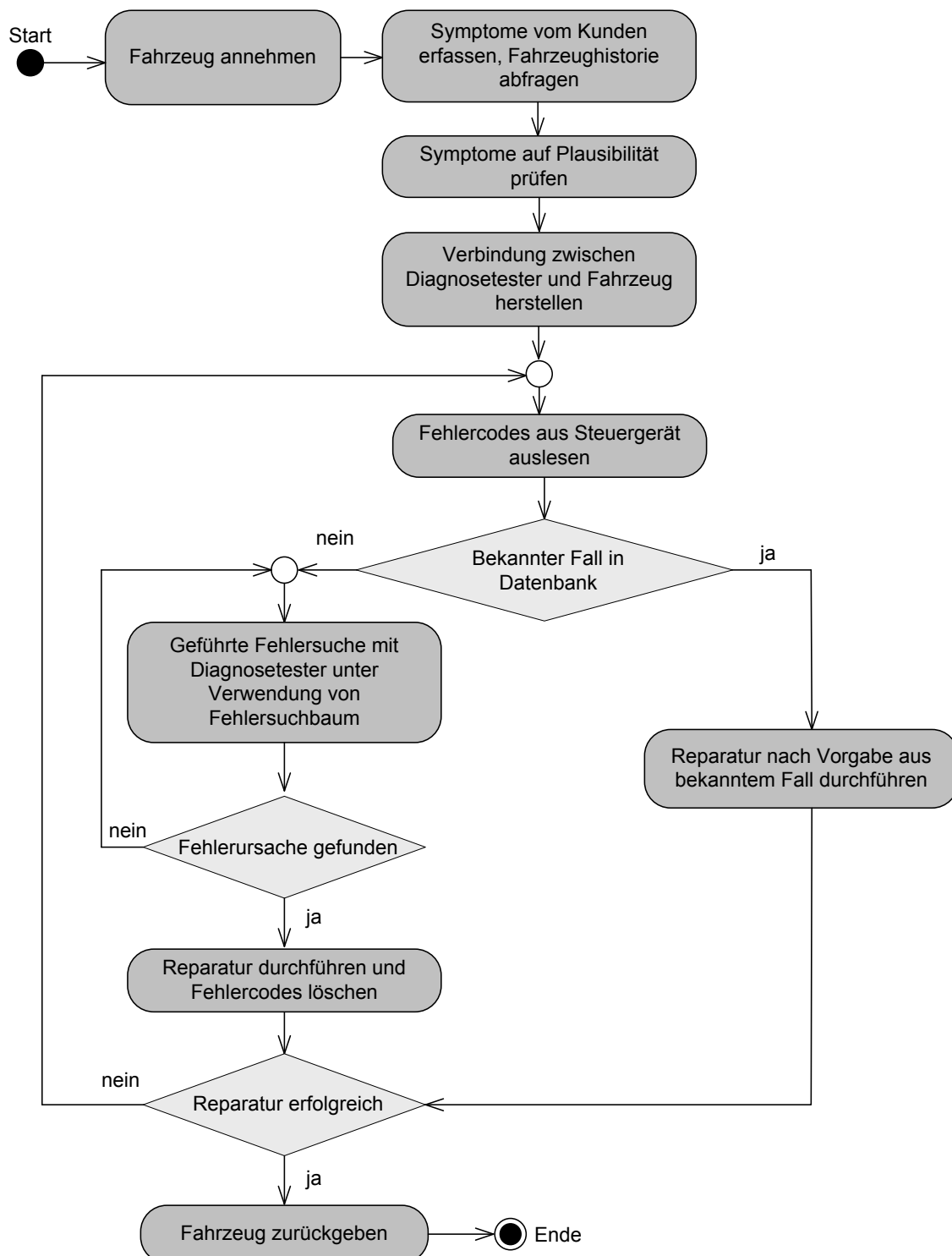


Abbildung 2.4: Diagnose- und Reparaturprozess in der Kfz-Werkstatt

## 2.3 Lösungsweg

Die bestehenden Lösungsansätze können die sich steigernde Problematik der Wissenserhebung für die Diagnoseanwendung in der freien Werkstatt nicht mehr beherrschen. Aus den Anforderungen an systemübergreifende Diagnose, Variantenvielfalt, automatischer Wissensprüfung und Integration des vorhandenen Wissens folgt, dass die Lösung nur ein hybrider skalierbarer Ansatz sein kann. Diese hybride Lösung besteht aus funktionaler und heuristischer Klassifikation. Dabei löst die funktionale Klassifikation die Probleme der systemübergreifenden Diagnose und der Variantenvielfalt. Des Weiteren ermöglicht sie einen strukturierten Wissensaufbau der neuen Mitarbeiter im Autorenteam und erhöht das Systemverständnis (Struktur- und Funktionszusammenhang) der Autoren. Mit der heuristischen Klassifikation kann das bisherige und zukünftig heuristische Diagnosewissen weiter verwendet und gepflegt werden. Die Eigenschaft der Skalierbarkeit ermöglicht den schrittweisen Aufbau des Diagnosewissens und die Erstellung einer Diagnose mit unterschiedlicher Tiefe abhängig vom verfügbaren Wissen.

Der erste Unterschied bei der Vorgehensweise in der modellbasierten Kraftfahrzeugwerkstatt-diagnose zu allen gängigen, modellbasierten Diagnoseverfahren ist domänenspezifisch. Da das mechatronische System über ein Eigendiagnose-Informationssystem verfügt, muss das Vorgehen an diese Randbedingung angepasst werden. Wenn ein Fehlercode im Fehlerspeicher gesetzt ist und dieser Fehlercode auf bestimmte Komponenten zeigt, gilt es immer, zuerst diese Komponente zu überprüfen. Sollte ein Fehlercode auf mehrere Komponenten im System zeigen und ein Stellgliedtest existieren, gilt es immer, diese Tests zuerst durchzuführen. Wenn die beiden genannten Situationen nicht auftreten, kann das Diagnoseverfahren symptomorientiert erfolgen. Dabei kann sich die Ermittlung der potenziell verdächtigen Komponenten an der Strategie der „Hypothesenerstellung und Hypothesenprüfung“ orientieren. Aufgrund des oben erwähnten vorhandenen Fehlercodes beziehungsweise der Eigendiagnosemöglichkeiten des Steuergeräts und dadurch, dass die Diagnoseinformationen des Fehlercodes häufig in der Wenn-Dann-Form interpretierbar sind, müssen die Diagnoseparadigmen modellbasiert und mit der heuristischen Diagnose kombiniert werden. Dabei kann das heuristische Wissen in unterschiedlichen Modellebenen eingebunden werden. Diese Kombination ermöglicht die Ausnutzung des Funktionswissens des Modells zur Erklärung der Diagnoseschlussfolgerung durch heuristisches Wissen. Die Anreicherung des Diagnosemodells durch heuristisches Wissen erhöht nicht nur die Diagnosegüte und die Diagnosegeschwindigkeit, sondern auch die Ausnutzung des vorhandenen Diagnosewissens, das zum einen durch langjährige Erfahrung gesammelt wurde und zum anderen durch Gesetzgebung vorgeschrieben ist. Im Gegensatz zu den Diagnoseinformationen aus Steuergeräten, Sensor- und Messwerten kann die Kundenbeanstandung beziehungsweise Beobachtung durch das Werkstattpersonal mehrdeutig sein, zum Beispiel kann das Symptom „Klingeln und Klopfen bei Beschleunigung“ mehrere Deutungen wie Zündzeitpunkte, Drehzahl, Kraftstoff, Verdichtungsverhältnis, Luftverhältnis, Ansaugdruck, Ansaugtemperatur, Restgas, Brennraumform annehmen. Es kann nicht erwartet werden, dass die Beanstandungen vollständig sind; in manchen Fällen könnte es sein, dass die Beanstandungen sogar falsch sind. Um die Robustheit der Diagnoseergebnisse zu garantieren, sollen heuristische Diagnostiktabellen (vgl. [112]) als Erweiterung bei der Priorisierung der verdächtigen Komponenten verwendet werden. Die Hauptschwierigkeit, die Festlegung der vielen Einzelbewertungen bei der heuristischen Diagnostiktafel, kann unter Zuhilfenahme von weiteren Diagnoseinformationen wie Ausfallwahrscheinlichkeiten, Zeit und Kosten gelöst werden.

## 3 Wissenserhebung

In diesem Kapitel wird der zu entwickelnde Modellierungsansatz beschrieben. In Abschnitt 3.1 werden die Anforderungen an die zu erstellenden Diagnosemodelle zusammengestellt. Sie dienen als Grundlage und gleichzeitig als Maßstab für die Bewertung der zu entwickelnden Techniken. Auf den Inhalt eines Diagnosemodells wird in Abschnitt 3.2 eingegangen. Dabei werden die Zusammenhänge zwischen den verschiedenen Bestandteilen des Modells dargestellt. Im Anschluss daran werden in Abschnitt 3.3 die theoretischen Grundlagen, Details und Überlegungen im Rahmen des zu entwickelnden Modellierungsansatzes, die für die Abbildung der Komponenten im Diagnosemodell eines Systems notwendig sind, erklärt. Für ausgewählte Beispiele verschiedener Komponentenarten wird die Modellierungstechnik angewandt und erläutert. Nach der Bereitstellung der Grundlagen der physikalischen qualitativen Modellbildung der systemeigenen Komponenten wird in Abschnitt 3.4 auf die Abbildung der Onboard- und Offboard-Messungen eingegangen. Die Modellierung der Fehlercodes wird in Abschnitt 3.5 beschrieben. Die Abbildung der Beobachtungen beziehungsweise Beanstandungen wird in Abschnitt 3.6 erläutert. Die Vorgehensweise zur Modellierung des Diagnosewissens in Form von Stellglied- und Komponententests wird in Abschnitt 3.7 dargestellt. Demgegenüber bieten die Abschnitte 3.8.1 und 3.8.2 eine einführende Modellierungsübersicht der Systemschnittstellen sowie der Verbindungen im Modell. Für einen konkreten Diagnosefall muss die Grenze des Wertebereichs der Systemsensoren und der Messungen festgelegt werden. Ohne diese Festlegung können in dem Diagnosevorgang keine Komponenten ausgeschlossen und verdächtigt werden. In Abschnitt 3.9 wird auf die Möglichkeit eingegangen, die Modellgrößen zu parametrisieren.

### 3.1 Anforderungsanalyse des Diagnosemodells

Die Modelle sind ein zentrales Element der modellbasierten Diagnose, da sie das für die Problemlösungskomponenten im Diagnosesystem nötige Diagnosewissen beinhalten. Aus diesem Grund soll das Modell eines realen Systems alle Eigenschaften des Systems, die für die Diagnose notwendig sind, abbilden. Da bei der Werkstattdiagnose zusätzlich weitere Messgrößen aufgenommen und Informationen aus Kundenbeanstandungen herangezogen werden können, besteht das Diagnosemodell des zu entwickelnden Ansatzes nicht allein aus einem physikalischen Modell des realen Systems. Nachfolgend werden die detaillierten Anforderungen an das Diagnosemodell und die möglichen Lösungen vorgeschlagen.

#### 3.1.1 Funktionale Modellierung auf Basis der objektorientierten Methode

Da ein mechatronisches System im Kraftfahrzeug sich durch seine Konstruktion in mehrere Komponenten aufteilen lässt und jede dieser Komponenten bestimmte Funktionen des Fahrzeugs übernimmt ([111], [112]), bietet sich die Möglichkeit, das Diagnosewissen entsprechend



den Komponenten in eigenständigen Wissensbasen abzubilden. Wenn im Fehlerfall die zu behandelnde Komponente möglicherweise eine relativ hohe Komplexität aufweist, kann diese in Subkomponenten durch Modularisierungsvorgänge weiter aufgeteilt werden. Dieses Zerlegungsprinzip trägt nicht nur zu Komplexitätsreduzierung, sondern auch zur Lösung des Problems der Variantenvielfalt bei, weil eine Modularisierung des Diagnosewissens vorliegt und sich so der Wiederverwendungsgrad des Diagnosewissens steigern lässt.

Die Funktion und das Verhalten der Komponenten im mechatronischen System können durch das Fließen und Umwandeln von Transportobjekten (Material, Energie oder Signal) beschrieben werden ([46], [111]). Hierbei können die Ein- und Ausgänge als Schnittstellen der Komponente betrachtet werden. Diese Schnittstellen ermöglichen die Interaktion zwischen der modellierten Komponente und ihrer Umgebung und stimmen mit dem Schnittstellenparadigma der objektorientierten Methode überein. Die Beschreibungsform des Verhaltens einer Komponente kann durch Verhaltenstabellen beziehungsweise Relationsgleichungen dargestellt werden. Relationsgleichungen beschreiben die Beziehungen und Zusammenhänge zwischen dem Transportobjekt, dessen Attributen in Schnittstellen der Komponente sowie internen Variablen. Aus dem Verhalten seiner Komponenten in verschiedenen Betriebszuständen und den Verbindungen zwischen den Komponenten ergibt sich das Verhalten des gesamten Systems beziehungsweise Subsystems. Dieses Kompositionsprinzip des Modells wird in verschiedenen Veröffentlichungen, welche sich mit dem Diagnosethema beschäftigen, wie zum Beispiel [87], [22], [92], propagiert. Analog zur Modularisierung dient das Kompositionsprinzip zur Beherrschung der Modellkomplexität.

Wegen der Möglichkeit, die Komplexität durch das Zerlegungs- und Kompositionsprinzip zu reduzieren sowie das Verhalten durch die Eingangs- und Ausgangsbetrachtungsweise des mechatronischen Systems zu beschreiben, ist die funktionale Modellierung auf Basis der objektorientierten Methode der richtige Lösungsansatz für die Diagnosewissensrepräsentation der Diagnoseproblematik in freien Werkstätten.

### 3.1.2 Qualitative Modellierung

Der Aufwand der Wissenserhebung ist ein großes Hindernis bei der Einführung des Experten-diagnosesystems im praktischen Einsatz. Bei einem modellbasierten Diagnosesystem hängt der Aufwand der Wissenserhebung von der Wahl der Form des Modells ab. Hier stehen zwei Möglichkeiten zur Auswahl: Die qualitative und die quantitative Modellform.

Quantitative Modelle benötigen relativ exakte Vorstellungen über systemare Mechanismen. Die zum Bau solcher Modelle notwendigen Informationen lassen sich erst durch aufwändige und langwierige Messungen und Beobachtungen gewinnen. Die Berücksichtigung von Unsicherheiten erfolgt in solchen Modellen häufig durch Variation der Modellparameter innerhalb vertretbarer Grenzen, wobei anschließend die Menge der Lösungen betrachtet wird.

In der Fahrzeugdiagnose ist aufgrund der Vielschichtigkeit der zu verbindenden Informationen nicht zu erwarten, dass das zur Konstruktion solcher Modelle notwendige Prozesswissen jemals ausreichend ist. Oft existieren weder quantitative Variablen noch exakte Vorstellungen über detaillierte Wirkungsmechanismen und exakte Daten über das betrachtete System. Häufig stützen sich Entscheider in solchen Situationen auf Expertenmeinungen, die nach subjektiver Kenntnis der Lage eine Einschätzung über mögliche systemare Zusammenhänge liefern. Allerdings besteht hier die Möglichkeit der Mehrdeutigkeit, eventuell sogar völlig gegenteiliger

Aussagen, was zu einem Abwägungskonflikt führt. Das Expertenwissen ist zu einem großen Teil durch die Kenntnis prinzipieller Zusammenhänge und Einflüsse charakterisiert, ohne dass genaue und funktionale Abhängigkeiten bekannt sein müssen. Hier finden sich also sowohl parametrische Unsicherheiten als auch unvollständig bekannte Abhängigkeiten. Oft werden daher generelle Tendenzen und Grenzfälle zur Schlussfolgerung über reale Systeme herangezogen. Auch ist das Wissen über den aktuellen Systemzustand durch die eingeschränkte Beobachtbarkeit zwangsläufig unvollständig. In kaum einem realistischen Szenario ist eine vollständige räumliche und zeitliche Abdeckung durch Messungen denkbar. Auch sind viele relevante Stellgrößen, Messgrößen und Beobachtungen inhärent qualitativer Natur. Beispielsweise werden anstelle von genauen Werten wie  $U = 15,3V$  qualitative Werte wie „normale Spannung“ oder  $[U] = [14V - 18V]$  benutzt, was bedeutet, dass für die Spannung  $U$  lediglich bekannt ist, dass sie einen Wert aus dem Intervall von 14V bis 18V annimmt.

Die Methode der qualitativen Modellierung bietet geeignete Abstraktionsebenen, Modellierungsparadigmen und Schlussverfahren, um dieses vorhandene Wissen verarbeiten zu können ([22]). Die Betrachtung des qualitativen Verhaltens eines Systems im Diagnoseverfahren ermöglicht außerdem Aussagen größerer Robustheit und Stabilität gegenüber Parameterabweichungen [92], [149]. Aus diesen Gründen wird in dieser Arbeit die qualitative Modellform für die Modellbeschreibung gewählt.

### 3.1.3 Stationäre Betrachtung

Stationäre und transiente Phasen durchläuft ein mechantronisches System immer abwechselnd. Dabei verhält sich das System in der stationären Phase im Gegensatz zur transienten Phase stabil, das heißt, die Zustandswahrscheinlichkeit während der stationären Phase ist relativ unabhängig von der Zeit, während der transienten Phase hingegen zeitabhängig. Die Vorgehensweise in der Werkstatt ist eher stationärer Natur; in vielen Fällen wird das zu diagnostizierende System in eine stationäre Phase gebracht, bevor die manuellen Messungen beziehungsweise das Auslesen von systemeigenen Sensoren durchgeführt werden. Kommt zum Beispiel ein Kunde mit dem Symptom „Starter dreht, Motor springt nicht oder schlecht an“ in die Werkstatt, ist eine mögliche Diagnose zur Erklärung dieser Symptome der anormale Istwert des Kraftstoffdrucks in der Vorlaufleitung des Kraftstoffsystems. Die Vorgehensweise des Werkstattpersonals zur Überprüfung des Kraftstoffdrucks läuft folgendermaßen ab:

- Das Fahrzeug in Leerlauf bringen
- Bei abgezogenem Unterdruckschlauch Istwert mit Sollwert vergleichen
- Bei aufgestecktem Unterdruckschlauch Istwert mit Sollwert vergleichen
- Istwert mit Sollwert des Haltedrucks nach 10 Minuten vergleichen.

Bei dieser Vorgehensweise ist deutlich zuerkennen, dass die Druckmessungen im zu diagnostizierenden System innerhalb der stationären Phase (in diesem Fall der Leerlaufphase) durchgeführt werden.

Bei Symptomen, die in der transienten Phase auftreten, wird die zeitliche Abhängigkeit des Systemverhaltens in der Werkstatt durch Komponenten- und Stellgliedtests bei dem Diagnosevorgang abgedeckt. Zur Veranschaulichung wird hierfür das Symptom „Schlechte Gasannahme, Übergangsfehler“ betrachtet. Eine Ursache des Symptoms ist das Fehlverhalten der Elektrokraftstoffpumpe. Zur Überprüfung der Pumpe wird das Werkstattpersonal die Kennlinie

des Systemdrucks im Verhältnis zur Fördermenge (Abbildung 3.1) bei stehendem Motor mit dem tatsächlichen Verlauf abgleichen. Die Feststellung der Abweichung, soweit diese existiert, erfolgt manuell durch das Werkstattpersonal oder automatisiert durch Abgleich per Datenanalyse. Bei diesem Vergleichsvorgang ist die zeitliche Abhängigkeit berücksichtigt worden. Bei dem Beispiel der Elektrokraftstoffpumpe verhält sich im normalen Fall die Fördermenge der Pumpe über die Zeit proportional zu dem Systemdruck.

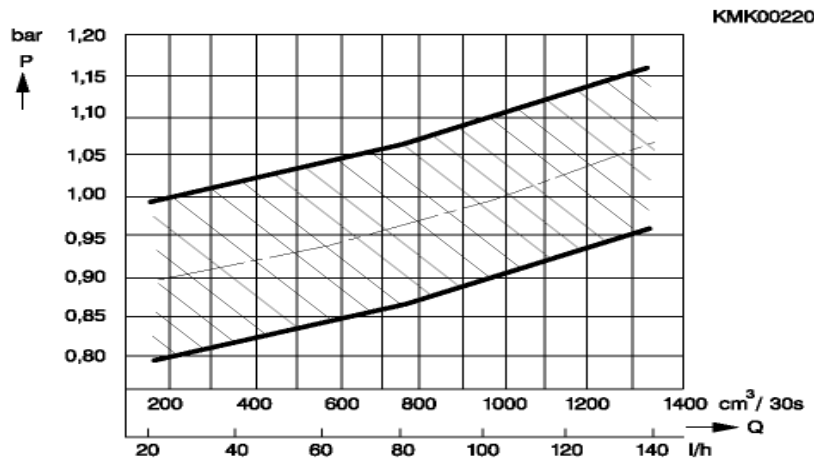


Abbildung 3.1: Kennlinien der Elektrokraftstoffpumpe beim Golf III 1.8 (Quelle: BOSCH Group)

Das oben beschriebene Beispiel verdeutlicht, dass der zeitliche Aspekt in das Modell eingebunden werden muss, um das Verhalten in der transienten Phase zu modellieren. Diese Erweiterung erlaubt eine Simulationsmöglichkeit und damit die Vorhersagemöglichkeit des Modells, bedarf aber erheblicher Erweiterungen in verschiedenen Modellierungsebenen: Komponentenmodell, Subsystem- sowie Systemmodell. So erhöht sich der Modellierungsaufwand auf ein Vielfaches und das Verhältnis zwischen Aufwand und Nutzung ist wirtschaftlich nicht vertretbar.

Da in den meisten Fällen die Symptome bereits bei dem Werkstattbesuch vorliegen und die Reproduzierbarkeit der Symptome in bestimmten Betriebszuständen, bei denen die Symptome aufgetreten sind, in freien Werkstätten möglich ist, wird in dieser Arbeit nur die symptomorientierte Diagnose und keine präventive Diagnose beziehungsweise Überwachung mit Risikoanalyse berücksichtigt. Diese symptomorientierte Diagnose reduziert damit nicht nur das Modellwissen hinsichtlich des zeitlichen Verhaltens (System im quasi stationären Zustand) sondern auch den Umfang der Diagnosealgorithmen, weil keine Algorithmen für die Erkennung von Trends, die in Zukunft zu Fehler führen können, erforderlich sind. Weiterhin werden in dem Diagnosevorgang Fragen zur Auftretshäufigkeit des Symptoms gestellt (reproduzierbare Symptome). Mit solchen Fragen wird der zeitliche Aspekt zum Teil berücksichtigt. Diese Berücksichtigung äußert sich in der Auswertung der verdächtigen Komponenten zur Symptomerklärung.

### 3.1.4 Berücksichtigung des Fehlermodus bei der Modellierung

Da in dieser Arbeit die Verhaltensbeschreibung über die Komponenten im System und das System in qualitativer Form unter stationärer Betrachtung erfolgen soll, leiden die Diagno-

seergebnisse bei der Schlussfolgerung an Effizienz und bei der Fehlerlokalisierung an Ungenauigkeit. Die Integration von konkreten Fehlerzuständen im Modell kann oft irrelevante verdächtige Komponenten aussortieren und von der weiteren Berechnung ausschließen (vgl. [30], [31], [78], [134]). Sie verringert die Komplexität der Diagnoseberechnung, der konkrete Fehler einer Komponente kann bei der Erklärungskomponente des Diagnosesystems angegeben werden. Deshalb ist diese Integration bei dem zu entwickelnden Modellierungstool notwendig und als Muss-Kriterium zu betrachten.

Nicht garantiert werden kann jedoch, dass die verwendeten Modelle hinsichtlich ihres Fehlverhaltens vollständig sind. Vollständigkeit bedeutet in jedem Fall für eine Komponente oder ein System, dass alle Fehlermodelle im Diagnosemodell vorhanden sind. Die Verwendung von konkreten Fehlerzuständen kann deshalb nur einen Teil der Probleme hinsichtlich Effizienz und Genauigkeit des modellbasierten Ansatzes lösen; zusätzliche Erweiterungen sind notwendig.

Eine mögliche Ergänzung ist die Verwendung eines unbekanntes Fehlerzustands. Diese Erweiterung garantiert die Vollständigkeit der Modelle in Bezug auf Fehlverhalten, ist aber keine Abhilfe bei der Fehlerlokalisierung. Weiterhin darf der unbekanntes Fehlerzustand erst verwendet werden, wenn kein anderes Fehlermodell die Verhaltensabweichung erklären kann, sonst wirkt er sich negativ auf die Effizienz aus.

### 3.1.5 Verwendung zusätzlichen heuristischen Wissens

Zur Verbesserung der Fehlerlokalisierung ist neben der Erweiterung von konkreten Fehlerzuständen die Anbindung von heuristischem Wissen über Komponenten und Systeme nötig. Diese Anbindung ermöglicht eine Priorisierung der verdächtigen Komponenten zum einen durch die Kosten der Komponentenprüfung und zum anderen durch die Ausfallwahrscheinlichkeiten der Komponenten im System. Zudem kann durch die Angabe der Häufigkeit der konkreten Fehlerzustände einer Komponente der Diagnosealgorithmus die Komponenten, die für die Erklärung der Symptome wegen der nicht vorhandenen beziehungsweise geringen Häufigkeit der konkreten Fehlerzustände selten in Frage kommen, vorübergehend im ersten Durchlauf ausschließen. Auf die Möglichkeit zur Anbindung solcher heuristischen Diagnoseinformationen wird im Folgenden eingegangen.

Das heuristische Diagnosewissen in freien Werkstätten stammt aus unterschiedlichen Quellen - zum Beispiel aus Onboard-Diagnose-Informationen oder aus Erfahrungswissen beziehungsweise direkt aus Werkstätten. Dieses heuristische Teilwissen gibt das unterschiedliche Wissen über Komponenten (zum Beispiel: Ausfallwahrscheinlichkeit, Test- beziehungsweise Prüfkosten, Prüfzeit) und den Kausalzusammenhang in Form von Wenn-Dann-Regeln zwischen Symptom und Ursache (insbesondere für Fehlercodes) wieder. Das benötigte Diagnosewissen, das in der Werkstatt bei dem Diagnosevorgang verwendet wird, soll aus all diesen diagnoserelevanten Informationen zusammengesetzt werden.

Aufgrund des vorhandenen Fehlercodes beziehungsweise der Eigendiagnosemöglichkeiten des Steuergeräts und weil die Diagnoseinformationen des Fehlercodes häufig in der Wenn-Dann-Form interpretierbar sind, erfolgt in dieser Arbeit die Kombination zweier Diagnoseparadigmen: Modell- und regelbasierter Ansatz. Diese Kombination (im Folgenden als hybrider Ansatz bezeichnet) ermöglicht die Ausnutzung des vorhandenen Diagnosewissens, das durch langjährige Erfahrung in den Werkstätten gesammelt wurde, des physikalischen Verhaltens

und der Strukturen, die im Modell abgebildet sind. Bei seltener Fehlerursache ist eine Diagnose mit Unterstützung des Modellwissens wegen der Erhöhung der Modellkomplexität kaum zu erwarten; zum Beispiel kann eine der Ursachen des Symptoms „Klingeln und Klopfen bei Beschleunigung“ die Änderung der Brennraumgeometrie sein. Da diese Diagnose sehr selten vorkommt, ist es empfehlenswert, dieses Diagnosewissen in Form von Wenn-Dann-Regeln zu formulieren und in geeignete Modellierungsebenen einzubinden. Damit ist eine Reduzierung der Modellkomplexität möglich, weil bei seltener Fehlerursache, welche durch Modellwissen nicht abgedeckt ist, das Modell nicht erweitert werden muss.

Bei der Diagnose in der freien Werkstatt steht das Werkstattpersonal oft vor folgender Situation: Es gibt eine Menge verdächtiger Komponenten und die Messmöglichkeiten sind ausgeschöpft. Die Entscheidung, welche Komponente zuerst geprüft werden muss, obliegt dem Werkstattpersonal. Mit Hilfe der Integration von Ausfallwahrscheinlichkeit, allgemeiner Wirkung der Komponente auf bestimmte Systemgrößen und Verschleißzustand der Komponenten im System kann die Lokalisierung der fehlerhaften Komponenten im Schlussfolgerungsprozess automatisiert und gegebenenfalls verbessert werden. Weitere Informationen wie Test- beziehungsweise Prüfkosten, Prüfzeit, etc. können die Effizienz des Diagnosevorgangs erheblich steigern. Die Zusammenführung dieser Informationen wird durch einen Identifizierungscode der Komponenten beziehungsweise des Systems durch Datenbankabfrage realisiert und bei der Parametrisierung des Modells durchgeführt.

## 3.2 Inhalt des Diagnosemodells

Auf Grund der angewendeten Dekompositions- und Kompositionsansätze bei der Konstruktion des *Fahrzeugs*<sup>1</sup> besteht das Diagnosemodell eines Fahrzeugs aus mehreren Modellen der fahrzeugeigenen mechatronischen Systeme. Ein mechatronisches System wird in dieser Arbeit durch ein Diagnosemodell dargestellt. Dieses Modell ist in Abbildung 3.2 mittels der Klasse *VollständigesSystem* repräsentiert.

Nachfolgend werden die Assoziationen, die Kardinalitäten und die Generalisierung beziehungsweise Spezialisierung der Klassen in dem Klassendiagramm (Abbildung 3.2) kurz erläutert. Anschließend wird auf die Inhalte der Klassen eingegangen. Da ein Teil des Diagnosewissens fahrzeugabhängig ist, existieren mehrere 1:n Beziehungen zwischen dem Modell eines Fahrzeugs und verschiedenen fahrzeugspezifischen Diagnosewissen:

- *Testwissen*,
- *Messungswissen*,
- *Heuristisches Komponentenwissen*,
- *Ersatzteilwissen*,
- *Komponentenzusatzwissen*.

Heuristisches Komponentenwissen, Ersatzteilwissen und Komponentenzusatzwissen sind komponentenbezogene Diagnosewissen. Daher gilt eine 1:1 Beziehung mit der zugehörigen *Komponente*.

---

<sup>1</sup>Die Klassen zur Abbildung des Diagnosewissens werden in diesem Abschnitt bei Ersterwähnung kursiv dargestellt

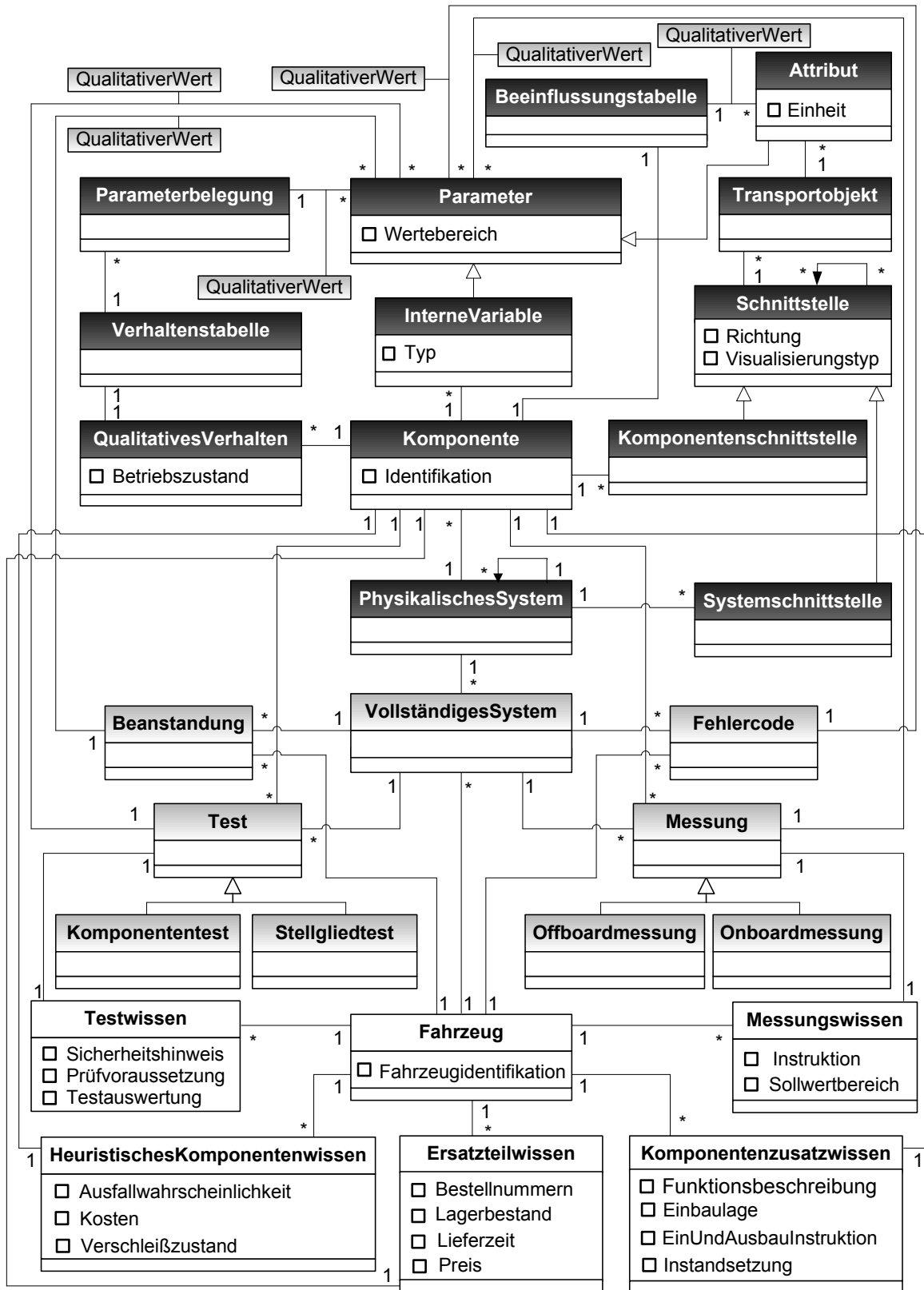


Abbildung 3.2: Bestandteile des Basis- und des vollständigen Diagnosemodells

Das Testwissen wird einem bestimmten Test zugeordnet, wobei ein Modell des vollständigen Systems mehrere Tests enthält. Ein Test ist eine Generalisierung des Komponenten- und

Stellgliedtests. Analog dazu besitzt eine Messung das Messungswissen und ist die Generalisierung von *Offboard-* und *Onboardmessung*. Ein vollständiges System enthält natürlich auch mehrere Messungen. Wenn eine Komponente in dem Diagnosevorgang verdächtigt wird, muss die Möglichkeit gegeben sein, dass diese Komponente als fehlerhaft beziehungsweise fehlerfrei identifiziert werden kann. Aus diesem Grund existiert eine 1:n Beziehung zwischen Komponente und Test sowie zwischen Komponente und Messung. Neben Tests und Messungen beinhaltet ein Modell des vollständigen Systems mehrere *Beanstandungen* und *Fehlercodes*. Es gibt 1:n Beziehungen zwischen Fahrzeug und Fehlercode und zwischen Fahrzeug und Beanstandungen, weil Fehlercode beziehungsweise Beanstandung Fahrzeug abhängig sind.

Dass ein Modell des *physikalischen Systems* sich in verschiedenen vollständigen Systemen befinden kann, wird durch die 1:n Beziehung dargestellt. Diese Beziehung beschreibt die Wiederverwendbarkeit des Modells. Es beinhaltet mehrere Komponenten, *Systemschnittstellen* und gegebenenfalls Subsysteme. Analog zum System besitzt eine Komponente mehrere *Komponentenschnittstellen*. Die Klasse *Schnittstelle* ist die Generalisierung der Komponenten- und Systemschnittstelle. Die Verbindung zwischen den Schnittstellen wird durch die n:n Beziehung beschrieben. Eine Schnittstelle kann ein beziehungsweise mehrere *Transportobjekte* enthalten und ein Transportobjekt enthält mehrere *Attribute*. Die internen Eigenschaften einer Komponente werden durch mehrere *interne Variablen* definiert. Interne Variablen und Attribute bilden *Parameter*, die für die Diagnose notwendig sind. Die Wertebelegungen dieser Diagnoseparameter werden durch 1:n Beziehungen zwischen Beanstandung, Fehlercode, Test beziehungsweise Messung und Parameter beschrieben.

Das qualitative Verhalten einer Komponente ist betriebszustandabhängig. Daher besitzt eine Komponente mehrere *qualitative Verhalten* und jedes Verhalten eine 1:1 Beziehung zu der *Verhaltenstabelle*. Eine Verhaltenstabelle hat mehrere Zeilen und eine Zeile wird durch die *Parameterbelegungen* repräsentiert. Die Inhalte einer Parameterbelegung bestehen aus mehreren Parameter und den zugehörigen qualitativen Werten.

Eine Komponente beeinflusst durch ihre vordefinierten Funktionen bestimmte Ausgangsattribute. Dieses Wissen ist einfach zu erheben und eine Möglichkeit zur Priorisierung der verdächtigen Komponenten in einem Diagnoseschritt. Die Klasse *Beeinflussungstabelle* ist für die Erhebung dieses Wissens zuständig.

Ein Fahrzeug wird durch die Fahrzeugidentifikation eindeutig identifiziert. Diese Eigenschaft legt fest, welche Fahrzeugtypen dieses Modell abdeckt. Dabei beinhaltet die Identifikation mehrere charakteristische Merkmale des Fahrzeugs, zum Beispiel Fahrzeughersteller, Fahrzeugart (PKW, LKW, Motorad, etc.), Antriebsart (Benzin, Diesel, Elektro, Gas, Hybrid), Motorkenndaten (Motorhersteller, Motorkennbuchstabe, Hubraum, Motorleistung), Antriebsart, Länderausführung, Baujahr, Modelljahr, Fertigungsdatum, etc. Neben der Fahrzeugidentifikation besitzt ein Fahrzeugmodell verschiedenes fahrzeugbezogenes Diagnosewissen. Dabei beinhaltet das heuristische Komponentenwissen folgende Informationen über die eingebauten Komponenten:

- Ausfallwahrscheinlichkeit der systemeigenen Komponenten: Diese Information kann aus verschiedenen Quellen erhoben werden. Zum einen können sie aus der Informationsrückführung aus dem Feld gewonnen, zum anderen aus dem Verkauf der Ersatzteile abgeleitet werden.
- Kosten: Diese Optimierungsgröße besteht zum einen aus Test-, Prüf- sowie Messkosten,

zum anderen aus Test- beziehungsweise Prüfzeit, da die Information über den benötigten Aufwand in Form von Zeit in Kosten überführt werden kann.

- Verschleißzustand der Komponenten: Wenn eine oder mehrere Komponenten des Systems Verschleißteile sind, können die Angaben über den Verschleißzustand für die Priorisierung der Prüfung in dem Diagnosevorgang genutzt werden.

Die Effizienz der Diagnose kann gesteigert werden, wenn dem Werkstattpersonal weitere hilfreiche Hinweise zur Verfügung gestellt werden. In dieser Arbeit wird dieses Wissen als Komponentenzusatzwissen bezeichnet und besteht aus folgenden Informationen:

- Einbaulage: Die Effizienz der Komponentenprüfungen sowie der Durchführung von manuellen Messungen im Diagnosevorgang ist abhängig von der Beschreibung der Einbaulage und von der Einbaulage selbst. Die Einbaulage kann entweder textuell und/oder mittels Bildern erfolgen.
- Ein- und Ausbauinstruktion: Diese Informationen beschreibt, wie eine Fahrzeugkomponente aus- beziehungsweise eingebaut werden kann. Dieses Wissen reduziert die möglichen entstehenden Zusatzkosten durch Verringerung des Aufwands und Vermeidung von Beschädigungen beim Ein- und Ausbau der Komponenten.
- Instandsetzung: Die erforderlichen Tätigkeiten, um eine Komponente wieder in den funktionsfähigen Zustand zu versetzen, werden als Hinweistexte beschrieben.

Das Ersatzteilwissen ist bei dem Reparaturvorgang unabdingbar und bei vielen existierenden Diagnosesystemen ein Bestandteil. Im Folgenden werden nur die wesentlichen Inhalte des Ersatzteilwissens erläutert:

- Bestellnummern: Mit dieser Information kann das Werkstattpersonal das Ersatzteil bestellen, wenn das Teil nicht vorrätig ist.
- Lagerbestand und Lieferzeit: Die Standzeit bei der Reparatur ist abhängig von der Verfügbarkeit der Ersatzteile im Lager.
- Preis: Die Höhe der Reparaturkosten ist zum einen von den Stundenkosten, die während der Diagnose und Reparatur anfallen, zum anderen vom Preis des Ersatzteils abhängig. Eine Auskunft darüber ist für den Kunden, der in die Werkstatt kommt, wichtig und muss immer gegeben werden.

Weiteres fahrzeugspezifisches Diagnosewissen, das die Durchführung und Auswertung von Tests und Messungen ermöglicht, ist direkt an das Fahrzeug gebunden:

- Messungswissen:
  - Instruktion: Diese Information beinhaltet den Anschlussplan der Komponenten und den Anschluss von Testgeräten und Werkzeugen. Je nach Art der Komponenten kann es ein elektrischer, hydraulischer oder pneumatischer Plan sein. Die Darstellung erfolgt vornehmlich anhand von Bildern. Zum Beispiel werden bei einem elektrischen System beziehungsweise einer elektrischen Komponente die Klemmenbelegungen dargestellt. Dabei werden sowohl die Einbaulage der Klemmen und damit die Messpunkte innerhalb der Steckanschlüsse als auch die funktionale Belegung der elektrischen Klemmen beschrieben.



- Sollwertbereich: Soll- beziehungsweise Richtwerte sind erforderlich, um Messergebnisse zu bewerten.
- Testwissen:
  - Sicherheitshinweis: Dieser beschreibt mögliche Gefahren für Personen, Sachwerte und Umwelt. Für sicherheitskritische Systeme wie zum Beispiel AirBag, ABS, ESP sind Hinweise über Eingriffsmöglichkeiten in das System zwingend erforderlich.
  - Prüfvoraussetzung: Die Einhaltung der Prüfvoraussetzung ist notwendig, um eine korrekte Schlussfolgerung zu ziehen. Es existiert zum Beispiel bei vielen Komponenten und Systemen eine Grundeinstellung und nur wenn die Grundeinstellung richtig eingestellt ist, kann die Prüfung beziehungsweise der Test durchgeführt werden.
  - Testauswertung: Die Beschreibung, wie ein Test oder eine Prüfung durchzuführen und wie die Ergebnisse zu bewerten sind, ist in der Testauswertung enthalten. Hierbei ist zu beachten, dass ein Test beziehungsweise eine Prüfung mehrere Test- oder Prüfschritte beinhaltet.

Da das systemeigene Komponentenmodell der Hauptbestandteil des Systemmodells ist, soll auf die Modellierung der systemeigenen Komponenten vor der Modellierung des Systemmodells eingegangen werden.

Im Folgenden werden die Inhalte der Klassen in einem Klassendiagramm kurz zusammengefasst. Um ein Modell des Systems aufzustellen, muss das System analysiert werden. Die Analyse erfolgt über zwei Betrachtungsweisen - von innen und von außen. Bei der Betrachtung von innen bietet die Darstellung des Systems durch seine Subsysteme, Komponenten und Verbindungen zwischen den Komponenten Erklärungen in unterschiedlicher Detailtiefe, wie die Eingangsgrößen in die gewünschten Ausgangsgrößen umgewandelt oder transformiert werden oder wie umgekehrt die Ausgangsgrößen aus unterschiedlichen Konstellationen der Eingangsgrößen entstehen. Demzufolge besteht ein System aus folgenden Elementen:

- Subsysteme: Um ein komplexes physikalisches System abzubilden hat der Modellierer die Möglichkeit, dieses in mehrere Subsysteme zu unterteilen. Dies verringert nicht nur die Komplexität des Systems, sondern erhöht auch die Wartbarkeit des Modells. Die Subsysteme beinhalten wiederum einzelne Komponenten und deren Verbindungen.
- Komponenten: Das Modell des realen Systems wird aus einzelnen Komponentenmodellen aufgebaut. Dabei ist eine Komponente die kleinste, in der Werkstatt austauschbare Einheit. Bei diesen Modellen muss neben den Schnittstellen auch das Verhalten der einzelnen Komponente spezifiziert werden. So stellen Kraftstoffleitung, Kraftstofffilter, Überströmventil, Hochdruckpumpe, Injektor und Rücklaufbehälter zum Beispiel Komponenten des Dieselsystems dar.
- Verbindungen: Die Verbindungen beschreiben die Interaktion der einzelnen Komponenten untereinander. Zu beachten gilt es, dass die Verbindungen nicht Verbindungsstücke (Schläuche etc.) oder Leitungen des realen Systems darstellen. Vielmehr sind Verbindungsstücke oder Leitungen des realen Systems als Komponenten zu interpretieren.

Analog zum System gibt es bei einer Komponente auch eine innere und eine äußere Betrachtungsweise. In dieser Arbeit enthält das Modell einer systemeigenen Komponente folgende Inhalte:

- Schnittstellen, mit denen eine Komponente mit seinen benachbarten Komponenten oder seiner Umgebung interagiert. Neben der Flussrichtung beinhaltet eine Schnittstelle ein oder mehrere Transportobjekte. Das Transportobjekt hat die Form von Materialien beziehungsweise Signalen, die hinein beziehungsweise aus der Komponente heraus fließen. Daher besitzt das Transportobjekt ein oder mehrere Attribute, die die physikalischen Eigenschaften des Objekts beschreiben. Zum Beispiel besteht das Transportobjekt „Luft“ aus mehreren Attributen wie Druck, Temperatur, Massenfluss, etc.
- Interne Funktions-, Zustands- und Fehlervariablen der Komponente, die die internen Komponenteneigenschaften oder Fehlerzustände beschreiben.
- Verhaltenstabelle, die das Nominal- und Fehlverhalten der Komponente in unterschiedlichen Betriebszuständen beschreibt. Eine Komponente kann in unterschiedlichen Betriebszuständen, zum Beispiel Leerlauf, Teillast, Vollast unterschiedliche Parameterbelegungen zeigen. Eine Parameterbelegung beschreibt eine Zeile der Verhaltenstabelle und beinhaltet die Zuweisungen der Parameter und deren zugehörigen qualitativen Werte. Dabei ist ein Parameter Ein- oder Ausgangsattribut beziehungsweise eine interne Variable der Komponente.

Für den systemübergreifenden Diagnosezweck liegt der Fokus der Betrachtung eines Systems auf dessen Übertragungsverhalten. Es stehen nicht die systemeigenen Komponenten und deren Interaktionen im Vordergrund der Betrachtung, sondern die Ein- und Ausgänge des Systems. Deswegen besteht ein Basismodell aus dem Modell eines System, dessen Schnittstellen und den Verlinkungen zwischen systemeigenen Komponenten und Systemschnittstellen. Diese Modellverlinkungen beschreiben die Zugehörigkeit der explizit modellierten Systemschnittstellen und der realen systemeigenen Komponentenschnittstellen. Systemschnittstellen sind mit den Komponentenschnittstellen verlinkt, die nicht mit anderen Komponentenschnittstellen durch Verbindungen in das Modell des Systems eingebunden sind. Im realen System sind sie die Systemgrenze und bilden damit die realen Systemschnittstellen. Mit der expliziten Modellierung dieser Art von Schnittstellen legt der Modellierer nicht nur fest, wie das System mit seiner Umgebung beziehungsweise benachbarten Systemen interagieren kann, sondern auch die Systemgrenze.

Durch die Anforderung der Abbildung verschiedener diagnoserelevanter Informationen aus Onboard- und Offboard-Quellen muss das Basismodell weitere Diagnoseinformationen beinhalten. Die Onboard- und Offboard-Messungen sollen durch den Modellierer im Modell abgebildet werden:

- Onboard-Messung: Die Messergebnisse dieser Messungen werden von Sensoren, die im System verbaut sind, geliefert.
- Offboard-Messung: Ein Beispiel dafür ist die Förderdruckprüfung im Kraftstoff-Niederdruck-Kreislauf des Dieselsystems. Als externes Messgerät kommt ein Manometer zum Einsatz, das mit zusätzlichen Prüflleitungen zwischen dem Kraftstofffilter und der Hochdruckpumpe angeschlossen wird. Mit Hilfe der Messwerte, die wahlweise zu hoch,

zu niedrig oder normal sein können, werden geeignete Fehlerhypothesen mittels Diagnosealgorithmen gestellt.

Die Lokalisierung der verdächtigen Komponenten in der Werkstattdiagnose stützt sich nicht nur auf Messergebnisse, sondern auch auf weitere Diagnoseinformationen, die speziell in der Offboard-Diagnose verfügbar sind. Diese Informationen stammen aus zwei unterschiedlichen Quellen:

- Fehlercodes: Dies sind Informationen, die aus den Fehlercodetabellen der Steuergerätdiagnose ausgelesen werden. Hier bedarf es einer Unterscheidung zwischen Fehlercodes, die unmittelbar auf Komponenten zeigen und Fehlercodes, die auf physikalische Größen zeigen. Ein Beispiel für einen Fehlercode, der auf eine Komponente deutet, stellt der Fehlercode „P2643 - Raildrucküberwachung Tank leer“ dar. Diese Information kann als Regel für die Diagnose erfasst werden. Fehlercodes, die auf physikalische Größen zeigen, treten zum Beispiel bei der Überwachung des Ladedrucks des Luftsystems auf: „P1470 - Ladedruckregelung Druck zu hoch“ oder „P2632 - Ladedruckregelung Maximalwert überschritten“. Diese Art des Fehlercodes bedarf einer Zuordnung des qualitativen Werts des Diagnoseparameters. Die Parameter- und Wertezuordnung obliegt dem Modellierer.
- Beanstandung: Subjektive Betrachtungen des Kunden beziehungsweise des Werkstattpersonals sind in Form von verbalen Aussagen wie „Unruhiger Leerlauf“, „Ungenügende Motorleistung“, „Fehlzündung bei konstanter Drehzahl“, „Klopfen beim Beschleunigen“, „Kraftstoffverbrauch zu hoch“ oder „Beschleunigungsloch“ vorhanden. Analog zum Fehlercode soll die Zuordnung der Beanstandung zu Diagnoseparametern mit qualitativen Werten erfolgen und als Diagnoseregeln aufgenommen werden.

Der Test einer oder mehrerer Komponenten in einem Diagnosevorgang ist unverzichtbar bei der Verdachtsprüfung beziehungsweise Verdachtsbestätigung. In der Werkstattdiagnose existieren zwei Arten von Tests:

- Stellgliedtest: Dies ist ein Test, bei dem eine gezielte Anregung einer Regelstrecke, die im System vorkommt, über einen oder mehrere systemeigene Aktoren vorgenommen wird. Als Ergebnis liefert der Test die Abweichung zwischen dem tatsächlichen Verhalten und dem erwarteten Verhalten der Regelstrecke. Im Fall einer fehlerfreien Regelstrecke ist die Abweichung gleich null. Dann können die Komponenten, die Teil dieser Regelstrecke sind, als fehlerfrei angenommen werden. Die Voraussetzung für diese Folgerung ist die Anregung mit Signalen, unter denen sich alle Fehler im Verhalten äußern.
- Komponententest: Beim Komponententest wird das Testobjekt ohne Wechselwirkungen mit benachbarten Komponenten geprüft. Somit können auftretende Symptome eindeutig auf die getestete Komponente zurückgeführt werden. Ziel dieses Tests ist es, zu prüfen, ob die verdächtige Komponente die spezifizierten funktionalen Anforderungen erfüllt. Eine Leckage-Prüfung einer Kraftstoffleitung ist beispielsweise ein Komponententest.

Durch die Modellierung der Parameter- und Komponentenzuordnung sowie die Abbildung der Diagnoseregeln bei Fehlercodes beziehungsweise Kundenbeanstandungen existieren mehrere Verlinkungen in dem Basismodell:

- Verlinkungen zwischen systemeigenen Komponenten und Onboard-Diagnose-Informationen: Diese Art der Verbindungen beschreiben die Zusammenhänge zwischen Systemkomponenten und Fehlercodes oder Stellgliedtests.
- Verlinkungen zwischen systemeigenen Komponenten und Offboard-Diagnose-Informationen: Die Berücksichtigung der weiteren Messmöglichkeiten, der Komponententests sowie der subjektiven Betrachtungen im Diagnosevorgang wird mit dieser Art der Verbindungen im Modell abgebildet.

### 3.3 Komponentenmodellierung

Da systemeigene Sensoren alle Eigenschaften systemeigener Komponenten aufweisen und im Vergleich zu anderen Systemkomponenten weitere Rollen bei dem Diagnosevorgang übernehmen, erfolgt als erster Schritt bei der Modellierung der Systemkomponenten die Festlegung verschiedener systemeigener Komponentenarten (Tabelle 3.1).

Systemkomponenten	Bezeichnung
Sensor	Tankfüllstand, Kraftstofftemperatursensor, Drucksensor
Aktor	Kraftstoffüberströmventil, Hochdruckpumpe, Injektor
Passiv	Tank, Kraftstoffleitung, Kraftstofffilter, Rücklaufbehälter

Tabelle 3.1: Verschiedene reale Komponentenarten im Kraftstoffsystem

Neben Sensoren, die verschiedene Messgrößen erfassen, existieren im System Aktoren und passive Komponenten. Ein Aktor wandelt die Eingangsgrößen in andersartige Ausgangsgrößen um, womit bestimmte Funktionen des Systems realisiert werden. Passive Komponenten sind Systemkomponenten, die keine Messsignale liefern und nicht aktiv gesteuert werden können, zum Beispiel Behälter, Verbindungen jeglicher Art (elektrisch, mechanisch, hydraulisch, etc.). Es seien mit:

*COM* - Menge aller systemeigenen Komponenten

*SEN* - Menge aller systemeigenen Sensoren

*ACT* - Menge aller systemeigenen Aktoren

*PAS* - Menge aller systemeigenen passiven Komponenten

in einem System, das zu modellieren ist, dann gilt:  $COM = SEN \cup ACT \cup PAS$ . Wenn  $com_k$  das k-te systemeigene Komponentenmodell ist, das im Systemmodell abgebildet wird, dann gilt:  $com_k \in COM$ .

Aus der Diagnoseanforderung und den Reparaturmöglichkeiten folgt, dass eine Komponente eine kleinste austauschbare Einheit im System ist. In Abbildung 3.3 ist das Komponentenklassendiagramm dargestellt. Dabei sind alle wesentlichen Klassen, die zur Beschreibung des qualitativen Verhaltens einer systemeigenen Komponente notwendig sind, aufgezeigt. Nachfolgend werden all diese Klassen detailliert vorgestellt. Abschließend wird im letzten Abschnitt dieses Kapitels die Darstellung eines beispielhaften Objektdiagramm zur Erläuterung des Modellierungskonzepts beschrieben.

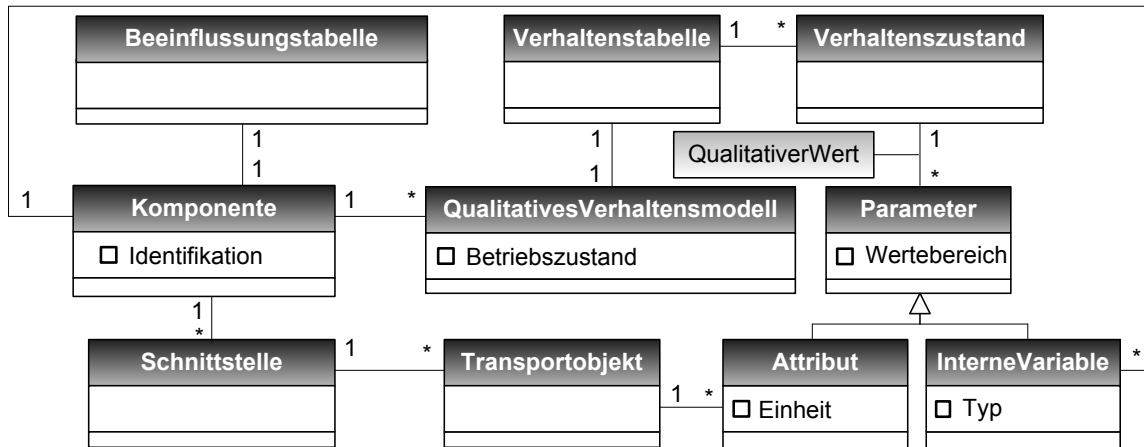


Abbildung 3.3: Klassendiagramm einer systemeigenen Komponente

### 3.3.1 Komponentenschnittstelle

Nach der Festlegung von verschiedenen systemeigenen Komponentenarten erfolgt die Festlegung der realen und virtuellen Schnittstellen der einzelnen Komponenten. Dabei ermöglichen die realen Schnittstellen die Interaktion und den Informationsaustausch zwischen den systemeigenen Komponenten untereinander sowie zwischen den Systemschnittstellen und Komponenten. Durch virtuelle Schnittstellen kann die Verbindung zwischen systemeigenen Komponenten und virtuellen beziehungsweise externen Komponenten realisiert werden. Es seien mit:

$PORT_k$  - Menge aller Schnittstellen einer Komponente  $com_k$

$PORT_k^{real}$  - Menge aller realen Schnittstellen der Komponente  $com_k$

$PORT_k^{vir}$  - Menge aller virtuellen Schnittstellen der Komponente  $com_k$

$port_l$  - Die  $l$ -te Schnittstelle einer Komponente  $com_k$ .

Dann gilt:

$$port_l \in PORT_k$$

$$PORT_k = PORT_k^{real} \cup PORT_k^{vir}.$$

Die Spezifikation der Schnittstelle erfolgt durch die Festlegung der Flussrichtung sowie die Transportobjekte, die diese Schnittstelle beinhaltet. Es gibt nur zwei mögliche Flussrichtungen: „In“ und „Aus“ (Tabelle 3.2). Es seien mit:

$PORT_k^{real,in}$  - Menge aller realen Eingangsschnittstellen der Komponente  $com_k$

$PORT_k^{real,out}$  - Menge aller realen Ausgangsschnittstellen der Komponente  $com_k$

$PORT_k^{vir,in}$  - Menge aller virtuellen Eingangsschnittstellen der Komponente  $com_k$

$PORT_k^{vir,out}$  - Menge aller virtuellen Ausgangsschnittstellen der Komponente  $com_k$ .

Dann gilt:

$$PORT_k^{real} = PORT_k^{real,in} \cup PORT_k^{real,out}$$

$$PORT_k^{vir} = PORT_k^{vir,in} \cup PORT_k^{vir,out}.$$

Für eine Schnittstelle  $port_l$  existiert eine Menge der Transportobjekte  $TRAN_l$ , die sich in dieser Schnittstelle befinden. Die Zuweisung der Transportobjekte auf eine Schnittstelle erfolgt durch den Modellierer. In den meisten Fällen beinhaltet eine Schnittstelle ein Transportobjekt. Dabei beschreibt ein Transportobjekt das Material, das durch die Schnittstelle in die Komponente einfließt oder aus der Komponente herausströmt. Bei dem Luftsystem wäre dies zum Beispiel Frischluft, im Kraftstoffsystem Diesel oder Benzin und im Abgassystem Abgas.

Transportobjekte in Schnittstellen sind nicht nur Materie, sondern auch Mess- und Steuersi-

Systemkomponenten	Bezeichnung	Schnittstellen		
		Anzahl	Richtung	Transportobjekt
Sensor	Temperatursensor	1	in	Temperatursignal
		1	out	Temperatursignal
Aktor	Druckregelventil	1	in	Ventilsteuerungssignal
		1	out	Kraftstoff
Passiv	Kraftstoffleitung	1	in	Kraftstoff
		1	out	Kraftstoff

Tabelle 3.2: Systemeigene Komponenten und deren Schnittstellen

gnale, welche für die Modellierung von Sensoren und Aktoren nötig sind. In Tabelle 3.2 werden einige Schnittstellen und deren zugehörige Transportobjekte von stellvertretenden Sensoren, Aktoren und passiven Komponenten des CP3-Kraftstoffsystems beschrieben. Es seien mit:

$TRAN_l$  - Menge aller Transportobjekte einer Schnittstelle  $port_l$

$tran_m$  - Das  $m$ -te Transportobjekt einer Schnittstelle  $port_l$ .

Dann gilt:  $tran_m \in TRAN_l$ .

Attribut	Abk.	Einheit	Transportobjekt		
			Abgas	Kraftstoff	Luft
CO	CO	$mg/m^3$	X		
Druck	P	Pa	X	X	X
HC	HC	%	X		
Massenfluss	m	g/s	X	X	X
NOx	NOx	$mg/m^3$	X		
O2	O2	%	X		X
SOx	SOx	$mg/m^3$	X		
Staubgehalt	s	%	X		X
Temperatur	T	°C	X	X	X
Verschmutzungsgrad	V	%		X	
Wassergehalt	W	%	X	X	
Zündwilligkeit	Z	Dez.		X	

Tabelle 3.3: Einige Transportobjekte und einige ihrer Attribute

Ein Transportobjekt wird durch dessen zugehörige Attribute spezifiziert. Die Attribute eines Transportobjekts sind die physikalischen Größen, die sich durch Systemsensoren oder durch externe Messgeräte bestimmen lassen. Durch Attribute werden die Eigenschaften der Transportobjekte beschrieben; Luft zum Beispiel durch verschiedene Größen wie Druck, Masse, Temperatur, Sauerstoffgehalt. Es seien mit:

$ATT_m$  - Menge aller Attribute eines Transportobjekts  $tran_m$

$att_n$  - Das n-te Attribut eines Transportobjekts  $tran_m$ .

Dann gilt:  $att_n \in ATT_m$ .

In Tabelle 3.3 sind einige Attribute von verschiedenen Transportobjekten dargestellt. Es gilt die Regel: Ein Attribut soll nur definiert werden, wenn sich durch Systemsensoren oder Messungen seine Größe ermitteln lässt. Ableitbare Attribute, die sich nicht durch Sensoren oder Messungen bestimmen lassen, sollen nicht definiert werden. Da die Werte eines Attributes in qualitativer Form  $\{hoch, normal, niedrig\}$  vorliegen, soll im Zuge der Spezifikation des Attributes die Zuweisung des Wertebereiches erfolgen. Zum Beispiel kann der Umgebungsdruck alle drei Werte  $\{hoch, normal, niedrig\}$  annehmen, der Staubgehalt der Umgebungsluft kann dagegen nur  $\{hoch, normal\}$  sein. Es seien mit:

$VAL_n^{att}$  - Menge aller Wertebelegungen eines Attributes  $att_n$

$val_m^{att}$  - Die m-te Wertebelegung eines Attributes  $att_n$ .

Dann gilt:  $val_m^{att} \in VAL_n^{att}$ .

### 3.3.2 Interne Variablen einer Komponente

Neben der Spezifikation der Komponentenschnittstellen müssen weitere Eigenschaften der Komponenten definiert werden, denn die Ausgangsgröße einer Komponente ist nicht nur von der Eingangsgröße abhängig. Im Beispiel des Kraftstofffilters ist der Verschmutzungsgrad im Kraftstoff vom *Abscheidegrad* des Kraftstofffilters abhängig. Eine weitere Aufgabe des Kraftstofffilters ist die Senkung des Wassergehalts. Der Wassergehalt des Kraftstoffs im Ausgang ist von dem *Wasserstand* des Kraftstofffilters abhängig. Solche komponenteneigenen Attribute sind in dieser Arbeit als interne Funktionsvariablen definiert. Als formale Formulierung kann hierfür folgendes definiert werden:

$VAR_k^{func}$  - Menge aller internen Funktionsvariablen der Komponente  $com_k$

$var_i^{func}$  - Die i-te interne Funktionsvariable der Komponente  $com_k$ .

Es gilt:  $var_i^{func} \in VAR_k^{func}$ .

Weil der *Abscheidegrad* bestenfalls der definierten Vorgabe entsprechen kann und bei aufgetretenem Fehler durch Kraftstoffverschmutzung der *Abscheidegrad* immer niedrig ist, kann der Wertebereich der internen Funktionsvariable *Abscheidegrad* nur zwei Werte annehmen  $\{normal, niedrig\}$ . Mit einer analogen Überlegung beträgt der Wertebereich der internen Funktionsvariable *Wasserstand*  $\{normal, niedrig\}$ . Es seien mit:

$VAL_i^{func}$  - Menge aller Wertebelegungen einer internen Funktionsvariable  $var_i^{func}$

$val_n^{func}$  - Die n-te Wertebelegung einer internen Funktionsvariable  $var_i^{func}$ .

Dann gilt:  $val_n^{func} \in VAL_i^{func}$ .

Viele Aktoren, wie zum Beispiel Drosselklappe, Injektor, AGR-Ventil, Druckregelventil, besitzen wegen ihrer Funktion definierte interne Zustände. Diese internen Zustände der Komponenten werden durch Steuersignale oder mechanische Steuerung geregelt, wobei die Komponente in Abhängigkeit des Steuerbefehls in verschiedene Zustände gesetzt wird. Dadurch verhält sich die Komponente je nach Zustand unterschiedlich. Diese Zustände müssen ebenfalls im Komponentenmodell abgebildet werden und werden in dieser Arbeit als interne Zustandsvariablen bezeichnet. Im Beispiel des Druckregelventils hat das Druckregelventil bei dem Druckregelungsvorgang im Normalfall zwei stationäre Zustände (*Offen* und *Geschlossen*) und zwei Übergangszustände (*Öffnen* und *Schließen*). Außer bei dem stationären Zustand *Geschlossen*

variiert bei allen anderen drei Zuständen der Ausgangsdruck in Abhängigkeit von den verschiedenen Eingangsgrößen und dem internen Verhalten des Ventils. Wegen der stationären Betrachtungsweise in der Werkstatt sollen die zwei stationären Zustände *Offen* und *Geschlossen* als Zustandsvariablen im Komponentenmodell des Druckregelventils definiert werden. Es seien mit:

$VAR_k^{sta}$  - Menge aller internen Zustandsvariablen der Komponente  $com_k$

$var_i^{sta}$  - Die i-te interne Zustandsvariable der Komponente  $com_k$ .

Dann gilt:  $var_i^{sta} \in VAR_k^{sta}$ .

Da eine Komponente sich nicht gleichzeitig in verschiedenen stationären Zuständen befinden kann, soll bei der Modellierung beachtet werden, dass die Zustandsvariablen sich gegenseitig ausschließen können. Im Unterschied zu internen Funktionsvariablen wie Abscheidegrad und Wasserstand hat die Wertebelegung von internen Zustandsvariablen, die die Komponentenzustände beschreiben, den Charakter eines logischen Terms  $\{true, false\}$ . Es seien mit:

$VAL_i^{sta}$  - Menge aller Wertebelegungen einer internen Zustandsvariable  $var_i^{sta}$

$val_n^{sta}$  - Die n-te Wertebelegung einer internen Zustandsvariable  $var_i^{sta}$ .

Dann gilt:  $val_n^{sta} \in VAL_i^{sta}$ .

Um das Nominalverhalten einer Komponente zu beschreiben, reicht die Spezifikation der Komponentenschnittstellen, der internen Funktionsvariablen und der internen Zustandsvariablen. Um das Fehlverhalten einer Komponente zu beschreiben, wird eine weitere interne Variablenart benötigt. Nachfolgend wird diese Art als interne Fehlervariable bezeichnet. Als Exempel kann hierfür weiterhin das Druckregelventil dienen. Zu beobachten ist, dass der Ausgangsdruck des Druckregelventils von mindestens drei weiteren Größen abhängig ist:

- Federkraft der Druckfeder
- Anziehungskraft des Elektromagnetfelds
- Verstopfung.

Ein mechanischer Fehler kann die Federkraft des Ventils beeinflussen. Ein elektrischer Fehler wirkt auf die Anziehungskraft des Ventil-Elektromagnetfelds. Eine weitere Abhängigkeit ist die mögliche Verstopfung des Ventils. Die drei Größen Federkraft, Anziehungskraft und Verstopfung sollen als interne Fehlervariablen im Modell definiert werden. Es seien mit:

$VAR_k^{faul}$  - Menge aller internen Fehlervariablen der Komponente  $com_k$

$var_i^{faul}$  - Die i-te interne Fehlervariable der Komponente  $com_k$ .

Dann gilt:  $var_i^{faul} \in VAR_k^{faul}$ .

Die Wertebelegungen der internen Fehlervariablen sollen analog zu internen Zustandsvariablen und internen Funktionsvariablen durch den Modellierer festgelegt werden. Beispielsweise ist bei der internen Fehlervariablen *Verstopfung* die Wertebelegung  $\{hoch, normal\}$ . Das bedeutet:

- $Verstopfung = hoch \Leftrightarrow$  Verstopfung vorhanden
- $Verstopfung = normal \Leftrightarrow$  Verstopfung nicht vorhanden.

Es seien mit:

$VAL_i^{faul}$  - Menge aller Wertebelegungen einer internen Fehlervariable  $var_i^{faul}$



$val_n^{faul}$  - Die n-te Wertebelegung einer internen Fehlervariable  $var_i^{faul}$ .  
 Dann gilt:  $val_n^{faul} \in VAL_i^{faul}$ .

Im Unterschied zu Funktionsvariablen und Zustandsvariablen können Fehlervariablen des Komponentenmodells mit Wahrscheinlichkeiten belegt werden. Die Belegung der Wahrscheinlichkeiten von aufgetretenen Fehlern erfolgt in dieser Arbeit in qualitativer Form. Hierfür werden Worte beziehungsweise Wortgruppen vordefiniert, zum Beispiel: „häufig“, „nicht häufig“, „selten“ und „sehr selten“. Mit Wissen über die Wahrscheinlichkeit des Fehlverhaltens der einzelnen Komponenten in einem System kann die Priorisierung der Komponentenprüfungen im Diagnosevorgang verbessert werden. Weiterhin benutzt die Erklärungskomponente des Diagnosetools die Fehlervariablen zur Beschreibung der Eigenschaften der Komponenten im System in Fehlerfällen. In Verbindung mit Beschreibungstexten können Fehlervariablen die Ursache des Fehlverhaltens erklären. Zum Beispiel kann eine Verstopfung des Ventils auf Ablagerung der Schmutzpartikel im Kraftstoff zurückzuführen oder eine niedrige Federkraft des Druckregelventils durch Materialermüdung beziehungsweise durch mechanische Fehler bedingt sein.

### 3.3.3 Qualitatives Verhaltensmodell

Zur Beschreibung des Verhaltens einer Systemkomponente bei einem bestimmten Betriebspunkt im nominalen Fall und im Fehlerfall werden die Transportobjekte und die zugehörigen Attribute der Schnittstellen sowie die Funktions- und Fehlervariablen genutzt. Es existiert für jedes Ausgangsattribut  $att_x^{out}$  einer Systemkomponente  $com_k$  eine qualitative relationale Übergangsfunktion  $f_{k,x}^{out}$ , welche den relationalen Zusammenhang zwischen diesem Ausgangsattribut und einer Menge der Eingangsattribute  $ATT_x^{in}$ , einer Menge der Ausgangsattribute  $ATT_{\neg x}^{out}$ , einer Menge der internen Funktionsvariablen  $VAR_x^{func}$ , einer Menge der internen Zustandsvariablen  $VAR_x^{sta}$  und einer Menge der internen Fehlervariablen  $VAR_x^{faul}$  beschreibt. Es seien mit:

- $ATT_k^{in}$  - Menge aller Eingangsattribute der Systemkomponente  $com_k$
- $ATT_k^{out}$  - Menge aller Ausgangsattribute der Systemkomponente  $com_k$
- $VAR_k^{func}$  - Menge aller internen Funktionsvariablen der Systemkomponente  $com_k$
- $VAR_k^{sta}$  - Menge aller internen Zustandsvariablen der Systemkomponente  $com_k$
- $VAR_k^{faul}$  - Menge aller internen Fehlervariablen der Systemkomponente  $com_k$ .

Dann gelten folgende Bedingungen:

$$\begin{aligned} ATT_x^{in} &\subseteq ATT_k^{in} \\ att_x^{out} \in ATT_k^{out} \wedge ATT_{\neg x}^{out} &\subseteq ATT_k^{out} \wedge att_x^{out} \notin ATT_{\neg x}^{out} \\ VAR_x^{func} &\subseteq VAR_k^{func} \\ VAR_x^{sta} &\subseteq VAR_k^{sta} \\ VAR_x^{faul} &\subseteq VAR_k^{faul} \end{aligned}$$

Die qualitative relationale Übergangsfunktion  $f_{k,x}^{out}$  wird wie folgt definiert:

$$\begin{aligned} f_{k,x}^{out} : (ATT_x^{in}, ATT_{\neg x}^{out}, VAR_x^{func}, VAR_x^{sta}, VAR_x^{faul}) &\xrightarrow{qual} att_x^{out} | \\ f_{k,x}^{out} (VAL_x^{in}, VAL_{\neg x}^{out}, VAL_x^{func}, VAL_x^{sta}, VAL_x^{faul}) &\stackrel{qual}{=} VAL_x^{out} \end{aligned}$$

Es seien mit:

- $F_k^{out}$  - Menge aller qualitativen relationalen Übergangsfunktionen einer Komponente.

Dann gilt:  $f_{k,x}^{out} \in F_k^{out}$ .

Um den relationalen Zusammenhang zwischen Eingangs- und Ausgangsattributen sowie in-

ternen Funktions-, Zustands- und Fehlervariablen in der Übergangsfunktion zu beschreiben, müssen die qualitativen Gleichheitszeichen und die qualitativen Operationen wie Addition, Subtraktion, etc. definiert und zur Verfügung gestellt werden. Im Rahmen von verschiedenen Arbeiten [28], [131], [143] wurde das Konzept der Berechnung mit qualitativen Operatoren beschrieben. Nachfolgend erfolgt eine kurze Ausführung zur Erklärung der qualitativen Berechnungen anhand einer abstrakten Komponente mit zwei Eingangsattributen  $att_x^{in}$ ,  $att_y^{in}$  und einem Ausgangsattribut  $att_z^{out}$  (Abbildung 3.4).

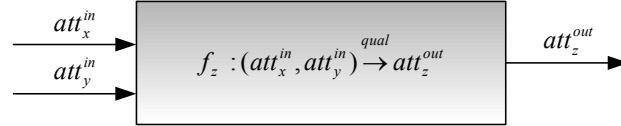


Abbildung 3.4: Beispiel einer qualitativen Übergangsfunktion

Das Verhalten dieser Komponente wird bestimmt durch die Übergangsfunktion  $f_z : (att_x^{in}, att_y^{in}) \xrightarrow{qual} att_z^{out}$ , welche die Ausgangsattribute  $att_z^{out}$  durch eine qualitative Operation der Eingangsattribute erzeugt.

$att_x$	$att_y$	$att_x \overset{qual}{+} att_y \overset{qual}{=} att_z$	$att_x \overset{qual}{-} att_y \overset{qual}{=} att_z$
hoch	hoch	hoch	hoch, normal, niedrig
hoch	normal	hoch	hoch
hoch	niedrig	hoch, normal, niedrig	hoch
normal	hoch	hoch	niedrig
normal	normal	normal	normal
normal	niedrig	niedrig	hoch
niedrig	hoch	hoch, normal, niedrig	niedrig
niedrig	normal	niedrig	niedrig
niedrig	niedrig	niedrig	hoch, normal, niedrig

Tabelle 3.4: Rechenbeispiel für qualitative Addition und Subtraktion

Bei dem Diagnosevorgang wird anhand der Abweichung der ermittelten beziehungsweise gemessenen Werte der Ausgangsattribute auf die möglichen Ursachen, das bedeutet auf die Werte der Eingangsattribute beziehungsweise den möglichen Fehlerzustand der Komponente geschlossen. Eine qualitative Abweichung in  $att_z^{out}$  liegt dann vor, wenn der zugehörige Zahlenwert der betrachteten physikalischen Größe entweder zu niedrig oder zu hoch ist. Es gilt:

$$\begin{aligned}
 & att_z^{out, gemessen} < att_z^{out, erwartet} \mid att_z^{out, gemessen} > att_z^{out, erwartet} \\
 & \Rightarrow att_z^{out, gemessen} - att_z^{out, erwartet} < 0 \mid att_z^{out, gemessen} - att_z^{out, erwartet} > 0 \\
 & \Rightarrow \Delta att_z^{out} = (-) \mid \Delta att_z^{out} = (+).
 \end{aligned}$$

Unter Zuhilfenahme des totalen Differentials der Überföhrungsfunktion  $f_z$  kann die Änderung  $\Delta att_z^{out}$  des Attributwertes  $att_z^{out}$  in Abhängigkeit der Änderungen  $\Delta att_x^{in}$  und  $\Delta att_y^{in}$  der Eingangsattributen  $att_x^{in}$  und  $att_y^{in}$  wiedergeben werden:

$$\Delta att_z^{out} = \frac{\delta f_z}{\delta att_x^{in}} \Delta att_x^{in} + \frac{\delta f_z}{\delta att_y^{in}} \Delta att_y^{in}$$

Mit  $f_z(att_x^{in}, att_y^{in}) \stackrel{qual}{=} att_x^{in} + att_y^{in}$  gilt für die qualitative Addition:

$$\Delta att_z^{out} \stackrel{qual}{=} \Delta att_x^{in} + \Delta att_y^{in}$$

Mit  $f_z(att_x^{in}, att_y^{in}) \stackrel{qual}{=} att_x^{in} - att_y^{in}$  gilt für die qualitative Subtraktion:

$$\Delta att_z^{out} \stackrel{qual}{=} \Delta att_x^{in} - \Delta att_y^{in}$$

Aus diesen Überlegungen gilt für die Berechnung mit qualitativer Addition und Subtraktion die Relationstabelle 3.4. In der qualitativen relationalen Übergangsfunktion wird die physikalische Einheit der Eingangs- und Ausgangsattribute nicht in Betracht gezogen. Diese Vernachlässigung ist notwendig, weil in den meisten Fällen die internen Funktions-, Zustands- und Fehlervariablen mit den Eingangsattributen in Beziehungen zu Ausgangsvariablen stehen und dabei selbst oft keine Einheit besitzen. Zudem ist diese Vernachlässigung berechtigt, da bei dieser Art der qualitativen Verhaltensbeschreibung die Vernachlässigung der Glieder höherer Ordnung möglich ist.

Zur Veranschaulichung der Verhaltensmodellierung kann als Beispiel die Verhaltensmodellierung einer Kraftstoffleitung (Abbildung 3.5) betrachtet werden.



Abbildung 3.5: Modelldarstellung einer Kraftstoffleitung

Die Beschreibung der relationalen Zusammenhänge zwischen Ein- und Ausgangsattributen der Kraftstoffleitung im nominalen Fall unter Verwendung der qualitativen Operationen kann wie folgt definiert werden:

$$\begin{aligned} & \text{Kraftstoffleitung.Eingang.Kraftstoff.Massenfluss} \\ & \stackrel{qual}{=} \text{Kraftstoffleitung.Ausgang.Kraftstoff.Massenfluss} \\ & \text{Kraftstoffleitung.Eingang.Kraftstoff.Druck} \\ & \stackrel{qual}{=} \text{Kraftstoffleitung.Ausgang.Kraftstoff.Druck} \\ & \stackrel{qual}{=} \text{Kraftstoffleitung.Ausgang.Kraftstoff.Massenfluss} \end{aligned}$$

Die erste Übergangsfunktion, die den Zusammenhang zwischen Eingangs- und Ausgangsmassenfluss im nominalen Fall beschreibt, kann durch die Kontinuitätsgleichung mit der Vernachlässigung der Glieder höherer Ordnung erklärt werden. Die zweite Übergangsfunktion kann durch die Bernoulli-Gleichung erklärt werden. Die Anwendung der Bernoulli-Gleichung ist begründet durch die stationäre Betrachtung in der Werkstatt und die Vernachlässigung der Glieder höherer Ordnung beziehungsweise den Abbruch nach dem ersten Glied. Unter Berücksichtigung des Fehlverhaltens bei *Leckage* in der Kraftstoffleitung wird das Verhalten der Kraftstoffleitung im Fehlerfall wie folgt erweitert:

$$\begin{aligned} & \text{Kraftstoffleitung.Eingang.Kraftstoff.Massenfluss} \\ & \stackrel{qual}{=} \text{Kraftstoffleitung.Leckage} \\ & \stackrel{qual}{=} \text{Kraftstoffleitung.Ausgang.Kraftstoff.Massenfluss} \end{aligned}$$

Für die passive Komponente Kraftstofffilter, die so konstruiert ist, dass der *Verschmutzungsgrad* im Kraftstoff am Ausgang direkt von dem Abscheidungsgrad der Komponente abhängt, kann die nominale Verhaltensbeschreibung für *Verschmutzungsgrad* mit Hilfe der internen Funktionsvariablen *Abscheidegrad* wie folgt beschrieben werden:

*Kraftstofffilter.Eingang.Kraftstoff.Verschmutzungsgrad*

$\stackrel{qual}{=}$  *Kraftstofffilter.Abscheidegrad*

$\stackrel{qual}{=}$  *Kraftstofffilter.Ausgang.Kraftstoff.Verschmutzungsgrad*

Im Beispiel des Kraftstoffventils sind weitere interne Zustandsvariablen wie *Offen* beziehungsweise *Geschlossen*, die die Komponentenzustände beschreiben, definiert. Hierbei ist die Verhaltensbeschreibung abhängig von dem Wert, den die internen Zustandsvariablen annehmen:

```

IF (Kraftstoffventil.Offen  $\stackrel{qual}{=}$  true){
  Kraftstoffventil.Eingang.Kraftstoff.Massenfluss
   $\stackrel{qual}{=}$  Kraftstoffventil.Ausgang.Kraftstoff.Massenfluss
  Kraftstoffventil.Eingang.Kraftstoff.Druck
   $\stackrel{qual}{=}$  Kraftstoffventil.Ausgang.Kraftstoff.Druck
   $\stackrel{qual}{=}$  Kraftstoffventil.Ausgang.Kraftstoff.Massenfluss
} ELSE {
  Kraftstoffventil.Ausgang.Kraftstoff.Massenfluss  $\stackrel{qual}{=}$  null
  ...
}

```

### 3.3.4 Verhaltenstabelle einer Komponente

<i>com<sub>k</sub></i>													
<i>port<sub>0</sub><sup>in</sup></i>										<i>port<sub>0</sub><sup>out</sup></i>			
<i>tran<sub>x</sub></i>	...	<i>tran<sub>y</sub></i>	<i>var<sub>i</sub><sup>func</sup></i>	...	<i>var<sub>j</sub><sup>sta</sup></i>	...	<i>var<sub>k</sub><sup>faul</sup></i>	...	<i>tran<sub>z</sub></i>				
<i>att<sub>0</sub></i>	...	<i>att<sub>m</sub></i>	...	<i>att<sub>0</sub></i>						<i>att<sub>0</sub></i>	...	<i>att<sub>n</sub></i>	
H	...	L	...	N	N	...	L	...	H	...	L	...	L
H	...	N	...	H	N	...	L	...	L	...	H	...	L
...													
H	...	L	...	N	N	...	N	...	N	...	H	..	L

Tabelle 3.5: Aufbau einer Verhaltenstabelle<sup>2</sup>

In dieser Arbeit wird die Verhaltensrepräsentation, die für den Diagnosevorgang verwendet wird, nicht in Form von qualitativen relationalen Übergangsgleichungen, sondern in Form von Verhaltenstabellen dargestellt. Dabei wird das Verhalten einer Komponente durch eine zweidimensionale Tabelle repräsentiert. Eine Verhaltenstabelle besteht aus zwei Teilen, dem Tabellenkopf und dem Tabellenrumpf, in welchem die Wertebelegungen stehen, die die Verhaltensrelation beschreiben (Tabelle 3.5). Der Tabellenkopf kann als Baumstruktur betrachtet

<sup>2</sup>Jede Zeile der Tabelle 3.5 entspricht einer Abbildung des Tupels:  $\{ATT_k^{in}, ATT_k^{out}, ATT_k^{func}, ATT_k^{sta}, ATT_k^{faul}\} \xrightarrow{qual} \{VAL_k^{in}, VAL_k^{out}, VAL_k^{func}, VAL_k^{sta}, VAL_k^{faul}\}$ . Zum Beispiel kann die erste Zeile wie folgt gelesen werden:

$(com_k.port_0^{in}.tran_x.att_0 = H) \wedge (\dots) \wedge (com_k.port_0^{in}.tran_x.att_m = L) \wedge (\dots) \wedge (com_k.port_i^{in}.tran_y.att_0 = N) \wedge (com_k.var_i^{func} = N) \wedge (\dots) \wedge (com_k.var_j^{sta} = L) \wedge (\dots) \wedge (com_k.var_k^{faul} = H) \wedge (\dots) \wedge (com_k.port_0^{out}.tran_z.att_0 = L) \wedge (\dots) \wedge (com_k.port_0^{out}.tran_z.att_n = L)$

werden: Die „Wurzel“ ist die Komponentenbezeichnung. Die erste Verzweigung beschreibt die Eingangs- und Ausgangsschnittstellen, die internen Funktions-, Zustands- sowie die internen Fehlervariablen. In der dritten Ebene stehen die einzelnen Schnittstellen und in der vierten Ebene folgt die Darstellung der Transportobjekte, die sich in den Schnittstellen befinden. Für „Blätter“ stehen die zugehörigen Attribute der jeweiligen Transportobjekte und die einzelnen internen Funktions-, Zustand- und Fehlervariablen. Jede Zeile im Tabellenrumpf entspricht einer Verhaltensrelation, die die Abhängigkeit zwischen Eingangs- und Ausgangsattributen unter Berücksichtigung von internen Fehler-, Zustands- sowie Funktionsvariablen beschreibt und zur Aufklärung der Symptome im Diagnosevorgang dient. Es seien mit:

$tab_k^{beh}$  - Verhaltenstabelle einer Komponente  $com_k$

$ROW_k^{beh}$  - Menge aller Zeilen der Verhaltenstabelle  $tab_k^{beh}$

$row_n^{beh}$  - Die n-te Zeile der Verhaltenstabelle  $tab_k^{beh}$ .

Dann gilt:

$row_n^{beh} \in ROW_k^{beh}$

$tab_k^{beh} := TUPEL(VAL_k^{in}, VAL_k^{func}, VAL_k^{sta}, VAL_k^{faul}, VAL_k^{out})$ .

Dieses Tupel ist eine Abbildung aus:

$\{ATT_k^{in}, VAR_k^{func}, VAR_k^{sta}, VAR_k^{faul}, ATT_k^{out}\}$

$\xrightarrow{qual} \{VAL_k^{in}, VAL_k^{func}, VAL_k^{sta}, VAL_k^{faul}, VAL_k^{out}\}$ .

Die Verwendung der Verhaltenstabelle ist sinnvoll, weil die Wertebelegung in den Zeilen der Verhaltenstabelle durch die automatische Auswertung der Übergangsfunktionen generiert wird. Die Wertebelegung kann auch durch den Modellierer manuell festgelegt werden, lässt sich doch die Verhaltensbeschreibung der Komponente im Modell nicht immer durch relationale Gleichungssysteme einfach beschreiben. Ein Beispiel dafür ist die Verhaltensbeschreibung des Brennraums: Mit einer Vielzahl von Eingangs- und Ausgangsattributen sowie internen Variablen ist bei dem Brennraum der Aufwand für die relationale Verhaltensbeschreibung im Vergleich zur manuellen Füllung der Tabelle relativ hoch. Ein Vorteil der Verhaltenstabelle bei der Erstellung von Diagnosehypothesen ist, dass nur die Suchmethode und nicht die Analyse der qualitativen relationalen Übergangsfunktionen aufgerufen werden muss. Diese Möglichkeit reduziert den Rechenaufwand und die Rechenzeit während der Diagnose erheblich.

### 3.3.5 Beeinflussungstabelle einer Komponente

Neben der Möglichkeit zur Verhaltensbeschreibung durch qualitative relationale Übergangsfunktionen und Repräsentation dieses Verhaltens in Tabellenform erlaubt die Modellierung in dieser Arbeit weitere Möglichkeiten zur Abbildung diagnoserelevanter Informationen in Form einer Auswertung des Beeinflussungsgrads aller Ausgangsattribute einer Systemkomponente durch seine erwarteten Funktionen. Aufgrund der Eingangs- und Ausgangsbetrachtungsweise einer Komponente bei der Modellierung und der Tatsache, dass systemeigene Komponenten durch ihre eigenen Funktionen die Ausgangsattribute beeinflussen, ist die Abbildung dieser Informationen im Gegensatz zur Verhaltensmodellierung relativ einfach.

Als Beispiel soll die Funktion der Komponente *Ladeluftkuehlung* dienen: Die Funktion der *Ladeluftkuehlung* ist die Begrenzung der Ladelufttemperatur. Sie ermöglicht die Reduzierung der thermischen Belastung des Motors, die Senkung der Abgastemperatur und damit der NOx-Emission, die Verringerung des Kraftstoffverbrauchs und die Erhöhung der Klopfestigkeit beim Ottomotor. Es ist offensichtlich, dass das Attribut *Temperatur* des Transportobjekts *Luft* in der Ausgangsschnittstelle der Komponente *Ladeluftkuehlung* dessen Funktion

im Vergleich zu anderen Attributen wie *Druck*, *Massenfluss* des Transportobjekts *Luft* am stärksten beeinflusst. Wenn ein Symptom im Diagnosevorgang auftaucht, das in Verbindung zur Abweichung der Ladelufttemperatur gebracht werden kann und sich in der verdächtigen Liste der Komponente *Ladeluftkuehlung* befindet, dann sollte in den meisten Fällen zuerst die *Ladeluftkuehlung* und deren Funktion überprüft werden. Mit dieser Überlegung und der Anforderung, dass eine Komponente unabhängig von ihrer Umgebung modelliert werden soll, ergibt sich die Modellierung der Beeinflussung der Ausgangsattribute durch die erwartete Komponentenfunktion. Es seien mit:

$INF_k^{out}$  - Menge aller Bewertungsbelegungen des Beeinflussungsgrads aller Ausgangsattribute einer Komponente  $com_k$  durch seine erwarteten Komponentenfunktionen

$inf_n^{out}$  - die n-te Bewertungsbelegung des Beeinflussungsgrads durch die Komponente  $com_k$  auf ein Ausgangsattribut  $att_n^{out}$ .

Dann gilt:  $inf_n^{out} \in INF_k^{out}$ .

			$com_k$									
			$port_0^{out}$			$port_1^{out}$			$\dots$			$port_n^{out}$
$var_0^{sta}$	$\dots$	$var_i^{sta}$	$tran_0$	$\dots$	$tran_x$	$tran_z$	$\dots$	$tran_v$				
			$att_0$	$\dots$	$att_n$	$\dots$	$att_m$	$att_i$	$\dots$	$att_k$	$\dots$	$att_j$
T	$\dots$	T	+	$\dots$	0	$\dots$	+	-	$\dots$	-	$\dots$	+
$\dots$												
F	$\dots$	T	-	$\dots$	+	$\dots$	-	0	$\dots$	-	$\dots$	-

Tabelle 3.6: Aufbau einer Beeinflussungstabelle<sup>3</sup>

Der Aufbau einer Beeinflussungstabelle ist in Tabelle 3.6 dargestellt. Im Unterschied zur Verhaltenstabelle werden nur die interne Zustandsvariablen und die Ausgangsschnittstellen der Systemkomponente, deren Transportobjekte und die zugehörigen Attribute für den Aufbau des Tabellenkopfs benötigt. Jede Zeile im Tabellenrumpf beschreibt den Beeinflussungsgrad der Komponente auf einzelne Ausgangsattribute in Abhängigkeit von den Werten der internen Zustandsvariablen dieser Komponente. Die Festlegung des Beeinflussungsgrads erfolgt durch den Modellierer. Es stehen drei mögliche Belegungen  $\{+, 0, -\}$  zur Verfügung; wobei  $\{+\}$  für stark beeinflusst,  $\{0\}$  für kaum beeinflusst und  $\{-\}$  für nicht beeinflusst stehen. Es seien mit:

$tab_k^{inf}$  - Beeinflussungstabelle einer Komponente  $com_k$

$ROW_k^{inf}$  - Menge aller Zeilen der Beeinflussungstabelle  $tab_k^{inf}$

$row_n^{inf}$  - Die n-te Zeile der Beeinflussungstabelle  $tab_k^{inf}$ .

Dann gilt:

$row_n^{inf} \in ROW_k^{inf}$

$tab_k^{inf} := TUPLEL(VAR_k^{sta}, ATT_k^{out})$ .

Dieses Tupel ist eine Abbildung aus:  $\{VAR_k^{sta}, ATT_k^{out}\} \xrightarrow{qual} \{VAL_k^{sta}, INF_k^{out}\}$

<sup>3</sup>Jede Zeile der Tabelle 3.6 entspricht einer Abbildung des Tupels:  $\{VAR_k^{sta}, ATT_k^{out}\} \xrightarrow{qual} \{VAL_k^{sta}, INF_k^{out}\}$ . Zum Beispiel kann die erste Zeile wie folgt gelesen werden:

$(com_k. var_0^{sta} = T) \wedge (\dots) \wedge (com_k. var_i^{sta} = T) \wedge (com_k. port_0^{out}. tran_0. att_0 = +) \wedge (\dots) \wedge (com_k. port_0^{out}. tran_0. att_n = 0) \wedge (\dots) \wedge (com_k. port_0^{out}. tran_x. att_m = +) \wedge (com_k. port_1^{out}. tran_z. att_i = -) \wedge (\dots) \wedge (com_k. port_1^{out}. tran_z. att_k = -) \wedge (\dots) \wedge (port_n^{out}. tran_v. att_j = +)$

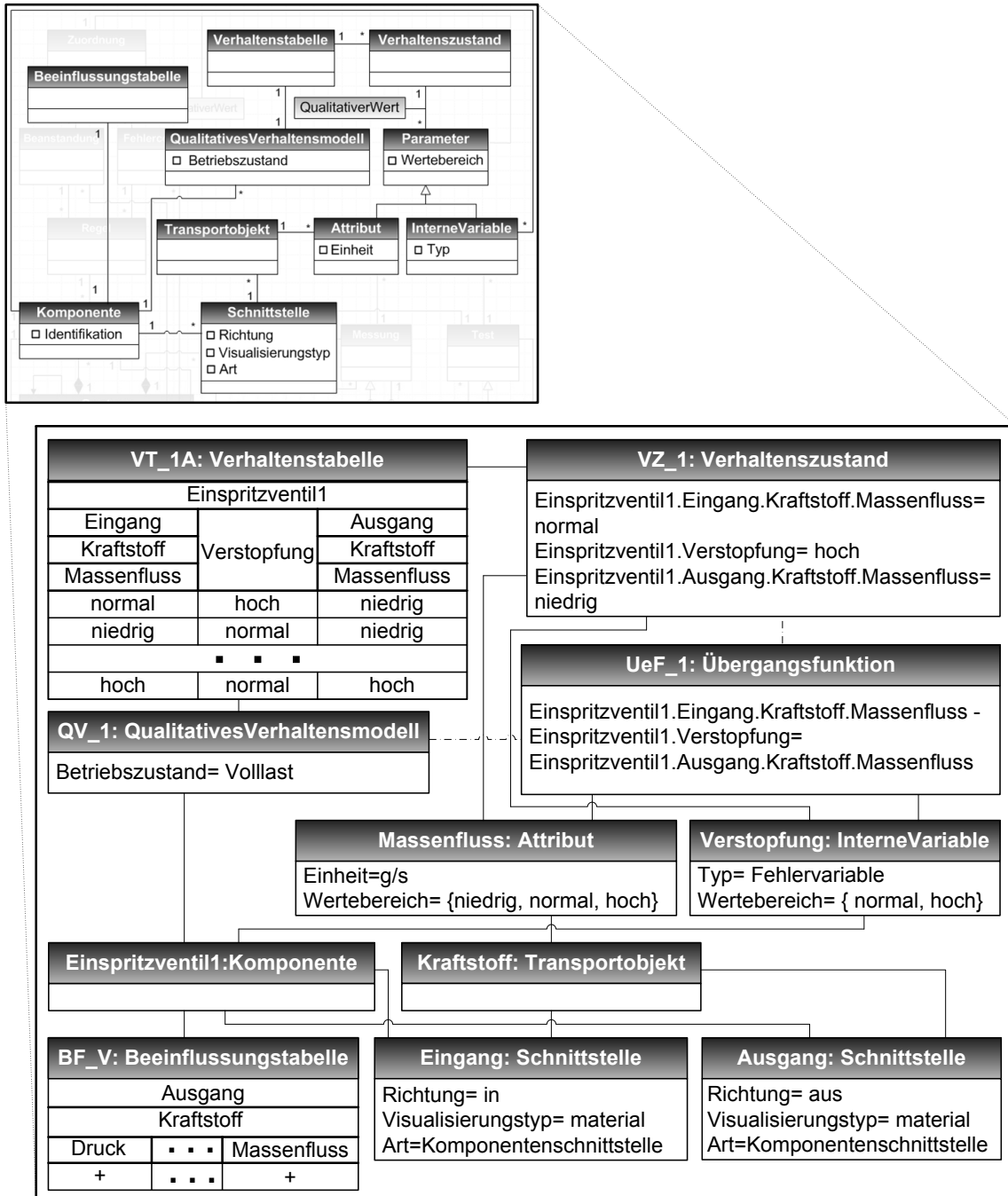


Abbildung 3.6: Objektdiagramm eines Einspritzventils

Die Verwendung der Bewertungsbelegung der Beeinflussung durch die Komponente auf ein Ausgangsattribut ist eine Möglichkeit zur Priorisierung der Liste der verdächtigen Komponenten. Die Priorisierung ist trival: Wenn sich zwei Komponenten mit gleicher Bewertung in der Verdächtigungsliste befinden und eine Komponente eine höhere Bewertungsbelegung des relevanten Attributs als die andere Komponente aufweist, dann soll die erste Komponente eine höhere Priorität als die zweite Komponente haben. Diese Information ist besonders nützlich, da nicht immer das Wissen über die Ausfallwahrscheinlichkeit der systemeigenen Komponenten verfügbar ist.

### 3.3.6 Ein beispielhaftes Komponentenobjektdiagramm

In Abbildung 3.6 sind einige Instanzen zur qualitativen Verhaltensbeschreibung der Komponente *Einspritzventil* dargestellt. In der Abbildung beinhaltet das Ventil eine Eingangs- und eine Ausgangsschnittstelle, durch welche das Transportobjekt *Kraftstoff* fließt. Ein Attribut dieses Transportobjekts ist der *Massenfluss*, welcher die Masse des Kraftstoff, die sich in einer Zeiteinheit durch einen Querschnitt bewegt beschreibt, und in Gramm je Sekunde angegeben wird ([80]). Die Verstopfung wird hierbei als interne Fehlervariable definiert und beeinflusst direkt das Verhalten des Massenflusses in der Ausgangsschnittstelle. Die Übergangsfunktion *UeF\_1* beschreibt die Abhängigkeit zwischen dem Ausgangs- und Eingangsmassenfluss sowie der Variable *Verstopfung*. Aus dieser Übergangsfunktion können mehrere Verhaltenszustände generiert werden. Die Generierungsoption ist in dem Objektdiagramm als gestrichelte Linie dargestellt. Das Objekt *VZ\_1* ist einer dieser Zustände und beschreibt eine Zeile der Verhaltenstabelle *VT\_1A*. Das qualitative Verhaltensmodell *QV\_1* beschreibt das Verhalten des Einspritzventils in dem Betriebszustand Volllast und besitzt die Verhaltenstabelle *VT\_1A*. Die gestrichelte Linie zwischen dem Verhaltensmodell *QV\_1* und der Übergangsfunktion *UeF\_1* deutet daraufhin, dass diese Übergangsfunktion für den Betriebszustand Volllast gilt.

## 3.4 Messungsmodellierung

Die Modellierung der externen Messung und der Messung mittels systemeigenen Sensors ist gleich. Hierbei ist nur die Angabe der Messpunkte sowie der Messgröße wichtig. Nur bei dem Diagnosevorgang wird zwischen den beiden Messarten unterschieden, weil oft die Kosten der Messungsdurchführung der externen Messung viel höher sind als bei der Messung mit Systemsensoren.

### 3.4.1 Messung mittels Systemsensor

Im Vergleich zu Aktoren und passiven Komponenten enthalten Systemsensoren zusätzliche Messeigenschaften. Ein Systemsensor ermittelt eine physikalische Modellgröße, die sich durch das Transportobjekt und das zugehörige Attribut beschreiben lässt. Der Sensor liefert den aktuellen Wert der Modellgröße und damit die Information zur Verdichtung des Diagnoseergebnisses auf wenige, verdächtige Komponenten. Für die Modellierungsaufgabe ist die Ankopplung zwischen Sensor und physikalischer Modellgröße notwendig. Bei der Formalisierung der Messgröße im Systemsensor (Abbildung 3.7) wird wie folgt vorgegangen:



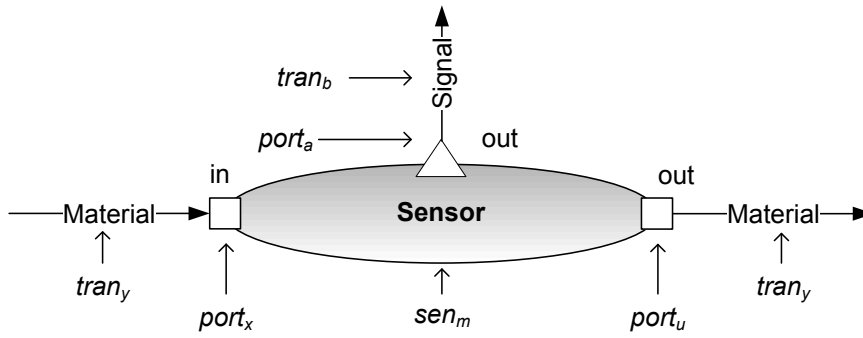


Abbildung 3.7: Beispielhafte Visualisierung eines Systemsensors

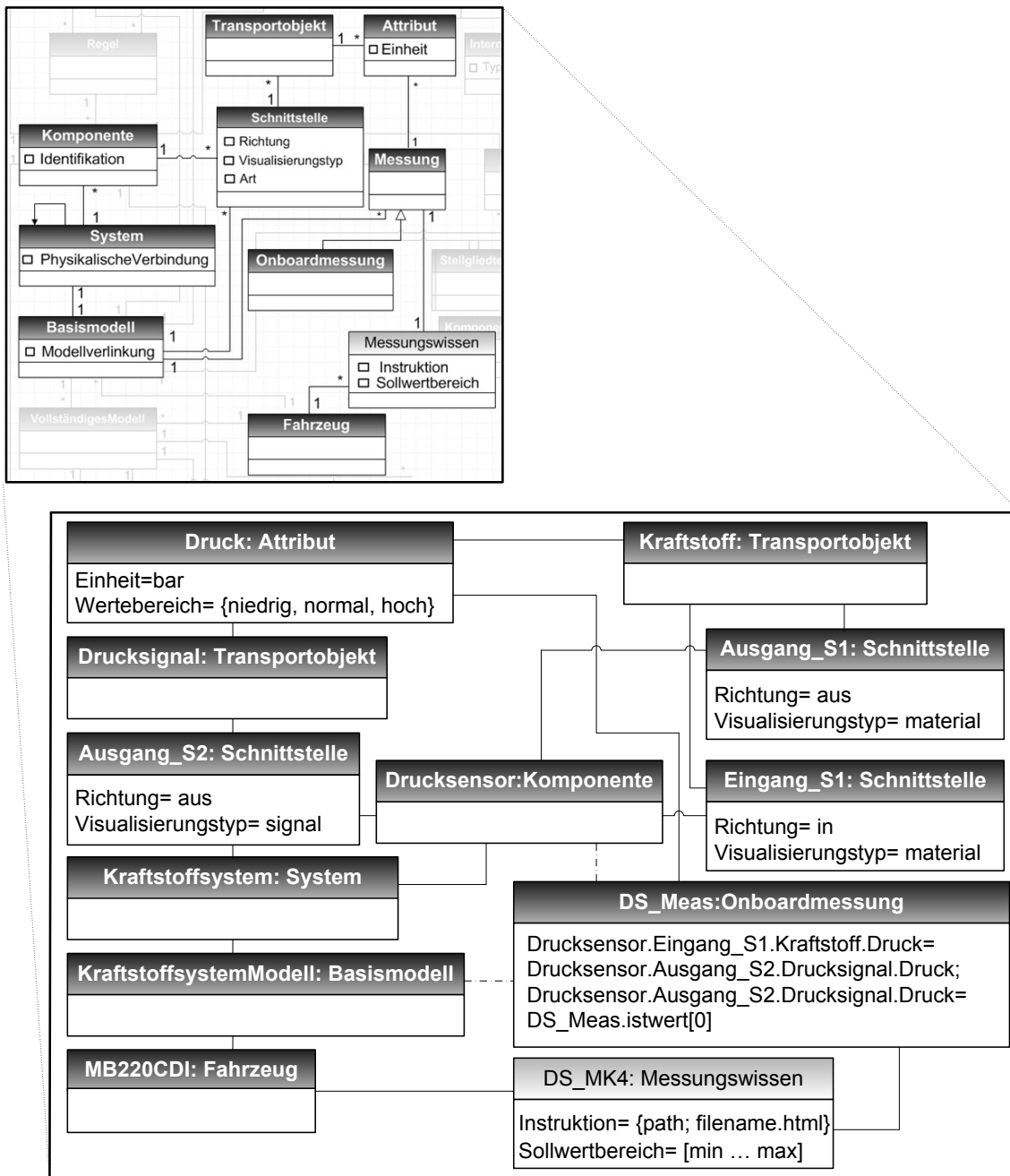


Abbildung 3.8: Beispielhaftes Objektdiagramm eines Drucksensors

$$\begin{aligned} sen_m.port_x^{in}.tran_y^{material}.att_z &:= sen_m.port_a^{out}.tran_b^{signal}.att_c \\ sen_m.port_a^{out}.tran_b^{signal}.att_c &:= sen_m.istwert_i. \end{aligned}$$

Die Instruktion zur Abfrage der Istwerte und die Angabe der Sollwertbereiche in verschiedenen Betriebszuständen sind abhängig vom Fahrzeugmodell und sollen mit der Datenbankabfrage in das Sensormodell bei der Diagnose eingebunden werden. In einem Diagnosevorgang wird der gemessene Wert  $sen.istwert[a]$  eines Systemsensors durch Anfrage des zuständigen Steuergeräts ermittelt. Nach der Wertermittlung erfolgt die Wertezuweisung des zugehörigen Attributes  $sen_m.port_x^{in}.tran_y^{material}.att_z$  und der Vergleich zwischen den Ist- und Sollwerten. Mit diesem neuen Diagnosewissen kann die Einschränkung der Liste der verdächtigen Komponenten fortgesetzt werden.

In Abbildung 3.8 ist das Objektdiagramm eines Drucksensors im Kraftstoffsystem des Fahrzeugs MB220CDI beispielhaft dargestellt. Dabei ist das Objekt  $DS\_Meas$  eine Instanz der Klasse  $Onboardmessung$  und beinhaltet die Messinformationen, die soeben beschrieben wurden. Das  $Messungswissen$  dieses Sensors, das für das Fahrzeug MB220CDI spezifisch ist, wird durch die Instanz  $DS\_MK4$  repräsentiert. Dabei enthält diese Instanz zum einen den relativen Pfad der Messinstruktionsdatei und zum anderen die Sollwertbereiche des Messergebnisses im fehlerfreien Fall.

### 3.4.2 Abbildung der externen Messung

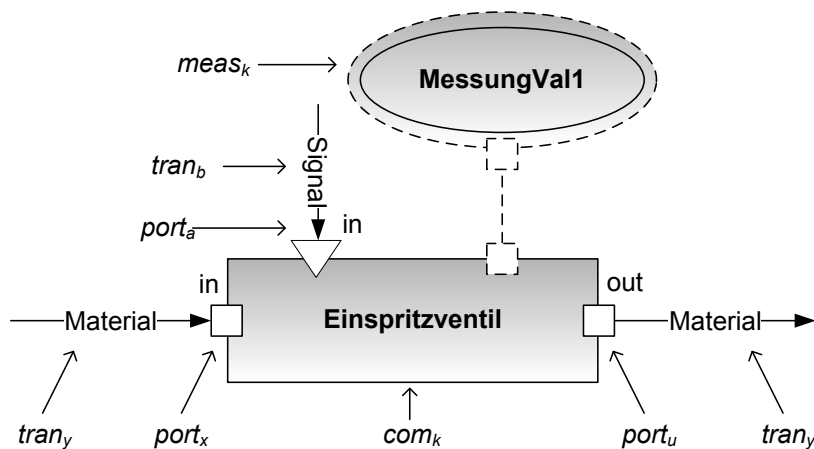


Abbildung 3.9: Beispielhafte Visualisierung einer Messung

Im Unterschied zu den systemeigenen Sensoren erfolgt die Modellierung der externen Messungen ohne eine Verhaltensbeschreibung. Die Schnittstelle des externen Messmodells ist eine virtuelle Schnittstelle, weil der Einbau des Messgeräts in das zu diagnostizierende System das Verhalten des Systems nicht beeinflussen soll. Es seien mit:

$MEAS$  - Die Menge aller externen Messungen, die im Modell abgebildet sind

$meas_k$  - Die  $k$ -te externe Messung im Modell.

Dann gilt:  $meas_k \in MEAS$ .

Analog zu dem systemeigenen Sensor beinhaltet das externe Messgerät die Messeigenschaften:

- Instruktion zur Abfrage des Istwerts

- Angabe der Sollwertbereiche in verschiedenen Betriebszuständen.

Eine externe Messung (Abbildung 3.9) ermittelt eine physikalische Modellgröße, die sich durch das Transportobjekt und das zugehörige Attribut beschreiben lässt. Mit dem gemessenen Wert der physikalischen Modellgröße kann die Fehlerhypothese verfeinert werden. Die Instruktion zum Auslesen des Messgeräts und die Angabe der Sollwertbereiche sind abhängig vom Fahrzeugmodell und werden bei der Parametrisierungsphase in das Diagnosemodell integriert. Für die Modellierung der externen Messung ist nur die Ankopplung zwischen externem Messgerät und der physikalischen Modellgröße wichtig. Die Formalisierung der Messgröße im externen Messgerät erfolgt analog zum Systemsensor:

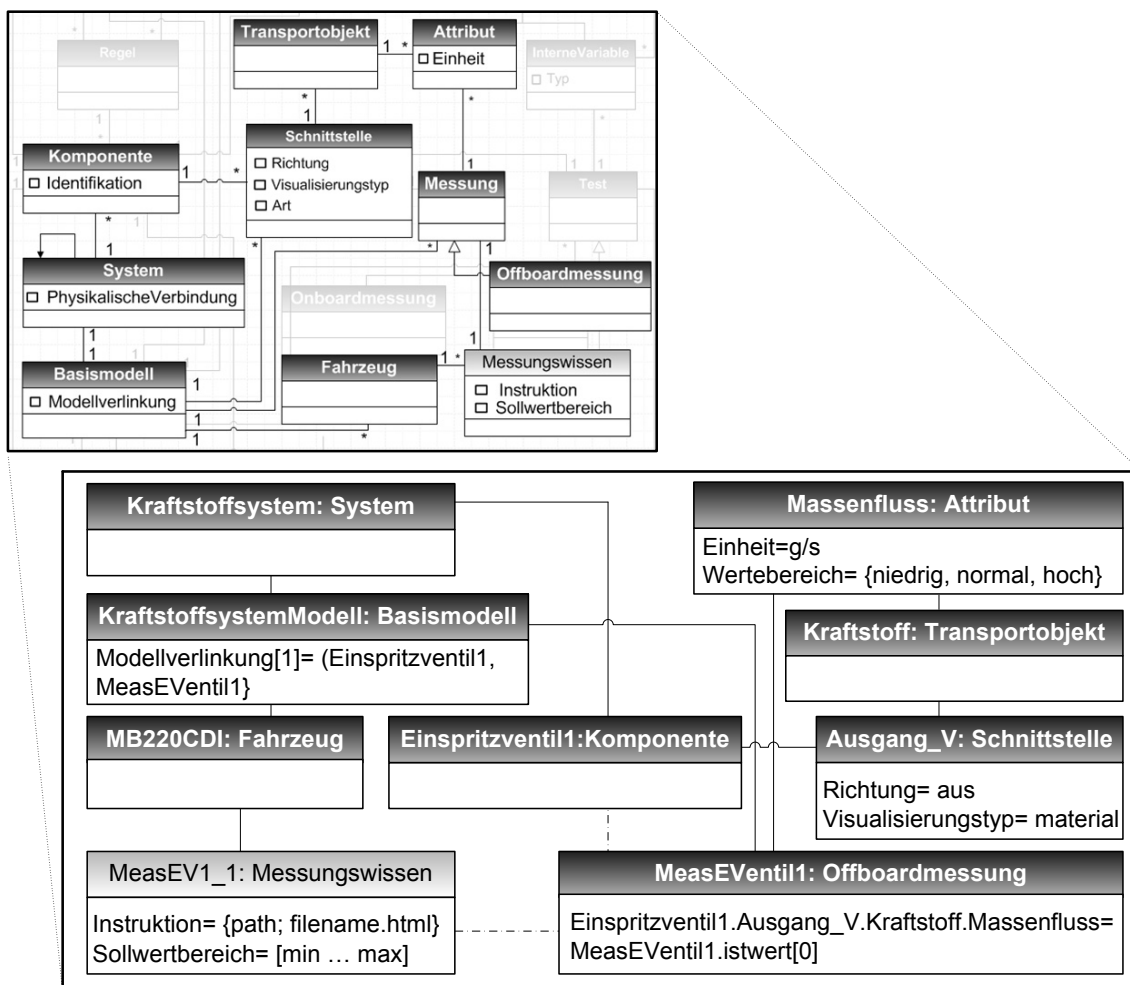
$$com_k.port_u.tran_y.att_z := meas_k.istwert_i.$$


Abbildung 3.10: Beispielhaftes Objektdiagramm einer manuellen Messung

Analog zur Onboardmessung beinhaltet die Instanz *MeasEV1\_1* der Klasse *Messungswissen* einen relativen Pfad zur Messinstruktionsdatei und zu den Sollwertbereichen des Messergebnisses. Da diese Informationen fahrzeugabhängig sind, sind diese mit der Instanz *MB220CDI* der Klasse *Fahrzeug* zugeordnet (Abbildung 3.10). Die physikalische Modellierung der manuellen Messung wird durch die Instanz *MeasEVentil1* repräsentiert und entspricht der oben beschriebenen Formalisierung.

### 3.5 Abbildung des Fehlercodes

Zum Grundumfang des Motorsteuergeräts in modernen Kraftfahrzeugen gehört die Onboard-Diagnose (OBD). Im Unterschied zur Offboard-Diagnose erfolgt die Onboard-Diagnose durch den permanenten Abgleich der Aktionen und der Reaktionen des Systems im Betrieb mit den Befehlen des Steuergerätes und der Signale der verschiedenen systemeigenen Sensoren untereinander auf ihre Plausibilität. Wenn die Plausibilität nicht gegeben ist, erfolgt ein Eintrag in den Fehlerspeicher (Fehlercode) des Steuergeräts. Dabei ist das Hauptziel der Onboard-Diagnose des Motorsteuergeräts und seine Funktion die Überwachung relevanter Komponenten, die bei Fehlverhalten oder Ausfall zur Erhöhung der schädlichen Emissionen führen können. Dieses Ziel ist bedingt durch die Gesetzgebung zur Minderung der Umweltverschmutzung. Andere kundendienstrelevante Diagnoseinformationen können ebenfalls im Fehlerspeicher abgespeichert werden, sind aber von Fahrzeug zu Fahrzeug hinsichtlich des Umfangs und des Detaillierungsgrades verschieden. Der Fehlercode enthält folgende, diagnoserelevante Informationen:

- Fehlerpfad: Systembereiche (Komponente beziehungsweise Komponentengruppe), die vermutlich das Fehlverhalten verursachen
- Fehlerart: Benennung der Art der Abweichung von der physikalischen Systemgröße in qualitativer Form (zu hoch, zu niedrig, unplausibel); Benennung der Fehlerarten im Stromkreis: Kurzschluss, Masseschluss, Plusschluss, Unterbrechung
- Fehlerbeschreibung: Kurzer Text zur Erklärung des Fehlverhaltens und mögliche Tests beziehungsweise Messungen zur Bestätigung der Verdachtshypothese; kann zusätzliche Informationen zu den Randbedingungen enthalten, bei denen der Fehler aufgetreten ist, wie zum Beispiel: Drehzahl, Umgebungsdruck, Umgebungstemperatur, Motortemperatur.

Es seien mit:

$DTC$  - Menge aller Fehlercodes, die im Modell abgebildet sind

$dtc_k$  - Das  $k$ -te Fehlercode im Systemmodell.

Dann gilt:  $dtc_k \in DTC$ .

Zur Veranschaulichung der Diagnoseinformationen des Fehlercodes werden nachfolgend einige beispielhafte Fehlercodes im Motorsystem beziehungsweise in dessen Subsystemen vorgestellt:

- P 2352[1] - Fehlercode:
  - Fehlerpfad: Kraftstoffsystem - Injektoren
  - Fehlerart: Zu geringe Einspritzmenge
  - Fehlerbeschreibung: Anzahl Einspritzung - begrenzte Anzahl von Einspritzungen durch zu geringe Einspritzmenge
- P 2017 - Fehlercode:
  - Fehlerpfad: Kraftstoffsystem - Mengenregelventil
  - Fehlerart: Minimalwert unterschritten

- Fehlerbeschreibung: Raildruck - Überwachung über Mengenregelventil; der Raildruck ist zu klein
- P 0204[1]- Fehlercode:
  - Fehlerpfad: Kraftstoffsystem - Injektor Zylinder 4
  - Fehlerart: Plusschluss
  - Fehlerbeschreibung: Bauteil Injektor 4. Zylinder prüfen; Kurzschluss nach Plus
- P 0204[2]- Fehlercode:
  - Fehlerpfad: Kraftstoffsystem - Injektor Zylinder 4
  - Fehlerart: Kurzschluss
  - Fehlerbeschreibung: Bauteil Injektor 4. Zylinder prüfen; Kurzschluss untereinander.

Durch die unterschiedlichen Diagnoseinformationen, die ein Fehlercode beinhaltet, ist es notwendig, eine Abbildung des Fehlercodes im Modell in zwei Arten zu unterscheiden:

- Fehlercode als weitere Messungen (virtueller Sensor)
- Fehlercode als Diagnoseregel.

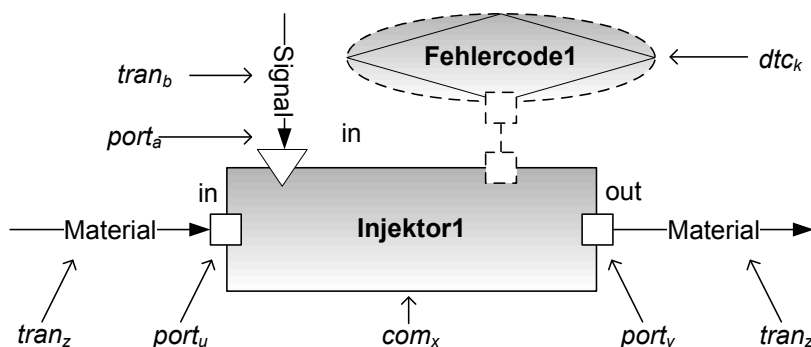


Abbildung 3.11: Beispielhafte Notation eines Fehlercodes als virtueller Sensor

Die Unterscheidung ist notwendig, weil die modellbasierten Algorithmen nur mit Fehlercodes als virtuellen Sensoren arbeiten können. Der Fehlercode als Diagnoseregel beträgt im Regelfall 50% des gesamten Fehlercodes und muss bei dem Diagnosevorgang berücksichtigt werden. Die Verarbeitung und Bewertung eines solchen Fehlercodes erfordert erweiterte Diagnosealgorithmen und eine Änderung des Diagnoseablaufs im Tester. Diese Erweiterung der Algorithmen kombiniert regel- und modellbasierte Diagnoseansätze. Ein Fehlercode als Sensor ist analog einem Systemsensor, der eine physikalische Modellgröße ermittelt, die sich durch das Transportobjekt und das zugehörige Attribut beschreiben lässt. Im Unterschied zu einem Systemsensor liefert der Fehlercode nicht den aktuellen Wert der Modellgröße zurück, sondern die Vergleichsergebnisse zwischen Soll- und Ist-Werten, wobei die qualitative Wertebelegung der Modellgrößen der Fehlerart im Fehlercode entnommen werden kann und der Fehlerort im Fehlerpfad gespeichert ist.

Die Messgrößen im Fehlercode können ähnlich wie die Messungen des Systemsensors formalisiert werden (Abbildung 3.11). Der Unterschied ist, dass die Ergebnisse der Messungen bereits

existieren:

$$com_x.port_y.tran_z.att_t := dtc_k.qualitativwert[i].$$

Bei der Modellierung von Fehlercode 2352[1] und 2017 soll wie folgt vorgegangen werden:

- P 2352[1] - Fehlercode:
  - Ort: Ausgangsschnittstelle der Injektoren
  - Abweichung:  $Injektor_1.Ausgang.Kraftstoff.Massenfluss^{qual} = niedrig$
- P 2017 - Fehlercode:
  - Ort: Ausgangsschnittstelle des Mengenregelventils
  - Abweichung:  $Mengenregelventil.Ausgang.Signal.Raildruck^{qual} = niedrig$

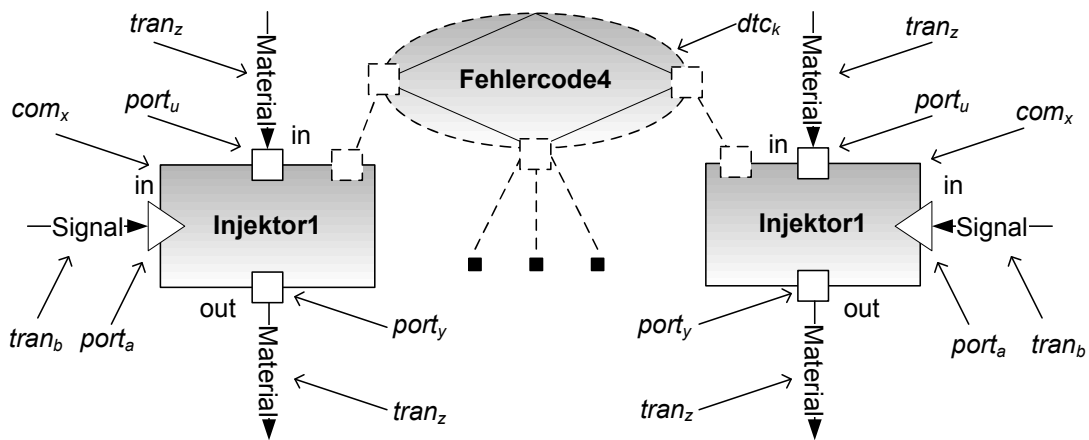


Abbildung 3.12: Beispielhafte Notation eines Fehlercodes als Diagnoseregeln

Ein Fehlercode als Diagnoseregeln zeigt direkt auf systemeigene Komponenten oder eine Komponentengruppe (Abbildung 3.12). Dabei kann die verdächtige Komponente anhand des Fehlerpfades entnommen werden; die Überprüfungsöglichkeiten werden in der detaillierten Fehlerbeschreibung des Fehlercodes angegeben. In dieser Arbeit hat Fehlercode, der als Diagnoseregeln abgebildet werden soll, folgende Form:

- Wenn: Fehlercode  $dtc_k$  existiert
- Dann:
  - Verdächtige Komponenten:  $com_i, \dots, com_{i+n}$
  - Prüfanweisungen: Folgende Prüfschritte  $X_1, \dots, X_n$  sollen durchgeführt werden
  - Erklärungstext: Fehlerhypothesen  $Y_1, \dots, Y_n$  beschreiben, die die Verdächtigungen der Komponenten  $com_i, \dots, com_{i+n}$  erklären

Die Diagnoseregeln der Fehlercodes 0204[1] und P 0204[2] lautet wie folgt:

- Wenn: 0204[1] - Fehlercode existiert
- Dann:

- Verdächtige Komponenten: Komponente: Kraftstoffsystem - Injektor Zylinder 4
  - Prüfanweisungen: Bauteil Injektor 4. Zylinder prüfen
  - Erklärungstext: Plusschluss; Kurzschluss nach Plus.
- Wenn: P 0204[2] - Fehlercode existiert
  - Dann:
    - Verdächtige Komponenten: Komponente: Kraftstoffsystem - Injektor Zylinder 4
    - Prüfanweisungen: Bauteil Injektor 4. Zylinder prüfen
    - Erklärungstext: Kurzschluss untereinander.

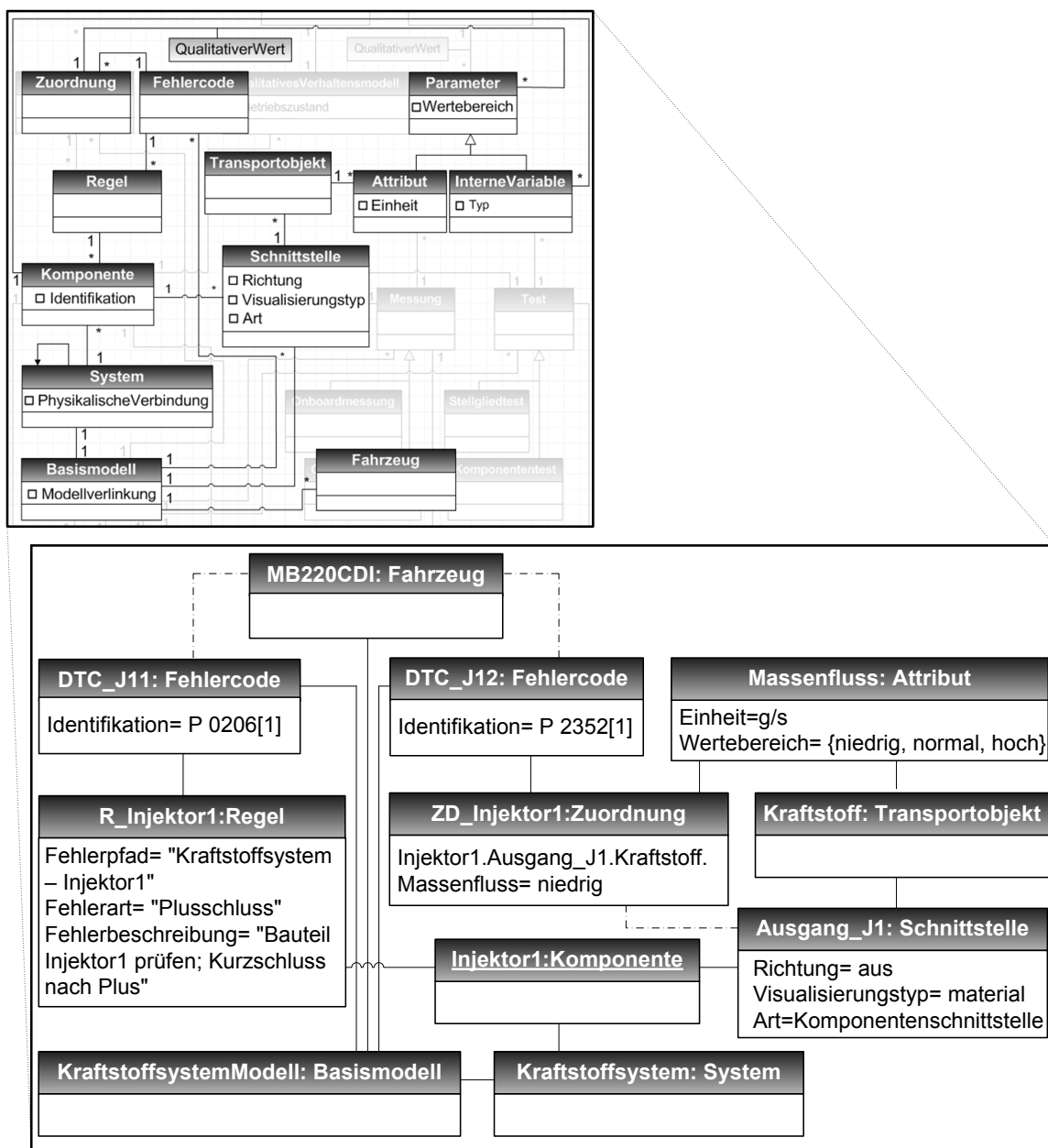


Abbildung 3.13: Beispielhaftes Objektdiagramm einiger Fehlercodes

In Abbildung 3.13 steht ein mögliches Objektdiagramm der Fehlercodes, die oben beschrieben sind. Dabei stellt dieses Diagramm neben den Instanzen von verschiedenen Klassen wie zum Beispiel *Basismodell*, *System*, *Komponente*, *Schnittstelle*, *Transportobjekt*, *Attribut*, auch die Objekte *ZD\_Injektor1* (Instanz der Zuordnung), *R\_Injektor1* (Instanz der Regel), *DTC\_J11* und *DTC\_J12* (Instanzen der Fehlercodes) dar.

Mit dem Objekt *MB220CDI* sind die beiden Fehlercodeinstanzen jeweils über einen Link verbunden, da die Identifikation der Fehlercodes Fahrzeug abhängig ist. Die Instanz der Regel *R\_Injektor1* ist direkt an das Objekt *Injektor1* und das Objekt *ZD\_Injektor1* an das Objekt *Massenfluss* gebunden. Die Verlinkung zwischen den beiden Instanzen *Schnittstelle* und *Zuordnung* beschreibt den Ort beziehungsweise die Messpunkte und damit eine indirekte Zuordnung zum Objekt *Injektor1*.

### 3.6 Abbildung der Beanstandung

Die Anforderungen an Diagnosetools sind vielfältig, eine davon ist die Verarbeitung der Kundenbeanstandungen. Für das Diagnosesystem, das das Diagnosewissen aus den Modellen abrufen, bedeutet dies zum einen die Überführung der subjektiven Beobachtungen beziehungsweise Kundenbeanstandungen in physikalische Größen, die in dem Modell abgebildet sind. Zum anderen sollen die Eigenschaften der Beanstandung, die sehr aufwändig sind beziehungsweise sich nicht in physikalische Größen überführen lassen, in geeigneter Form im Modell abgebildet werden. Als Beispiel wird hierfür eine Kundenbeanstandung, die der Kunde oder das Werkstattpersonal über das Motorsystem äußert, betrachtet. Die Beanstandungen entstehen in der Regel aus Abweichungen der verschiedenen physikalischen Größen oder Parameter, die für die Funktion des zu diagnostizierenden Systems verantwortlich sind. Diese Abweichungen sind Resultate des Fehlverhaltens der Komponenten im System. Die Komponente, die das Symptom verursacht, soll mit Hilfe des Diagnosesystems lokalisiert werden. Die entscheidenden Diagnoseinformationen, die für die Lokalisierung der fehlerhaften Komponenten benötigt werden, sind die Antworten auf folgende Fragen:

- Wo im System ist die Abweichung zu beobachten?
- Welche Abweichung der physikalischen Größe kann das Symptom erklären?

Aufgrund des Systemzusammenhangs und der Interaktion zwischen den Komponenten im System, welche durch die Strukturmodellierung in Form von Graphen abgebildet sind, und da die Fehlerlokalisierung durch Rückwärts- und Vorwärtsverkettung anhand der Struktur des Systems erfolgt, ist die Bestimmung des Ortes der beobachtbaren physikalischen Ursachen bei der Modellierung der Beobachtungen sehr wichtig. Der Ort beeinflusst nicht nur den Rechenaufwand der Diagnosealgorithmen bei der Generierung der Liste der verdächtigen Komponenten, sondern auch die Fehlerlokalisierung. Bei falscher Ortsbestimmung können bei bestimmten Konstellationen wichtige Sensoren, die für die Fehlerlokalisierung relevante Daten liefern, erst zu einem späteren Zeitpunkt durch die Diagnosealgorithmen berücksichtigt werden, obwohl sie zur Verkürzung der Berechnungszeit bereits vorher in Betracht gezogen werden sollten. Die Abbildung der Kundenbeanstandungen soll daher in zwei Schritten erfolgen:

- Ort: Bei welchen Komponenten beziehungsweise Systemschnittstellen sind die Abweichungen zu beobachten



- Abweichungen in physikalische Ursachen überführen und als qualitative relationale Gleichungen abbilden.

Eine Formalisierung der Eigenschaften von Kundenbeanstandungen hinsichtlich des Beobachtungsortes und der Abweichungsmöglichkeiten der Ausgangs- beziehungsweise Eingangsattribute ist in dieser Arbeit wie folgt definiert:

$OBS$  - Die Menge aller Kundenbeanstandungen, die im Modell abgebildet sind

$obs_k$  - Die  $k$ -te Kundenbeanstandung im Systemmodell ( $obs_k \in OBS$ )

$E_k^{obs}$  - Menge aller qualitativen relationalen Gleichungen, die die Kundenbeanstandungen beziehungsweise Beobachtungen  $obs_k$  beschreibt

$e_{k,n}^{obs}$  - Die  $n$ -te qualitative relationale Gleichung ( $e_{k,n}^{obs} \in E_k^{obs}$ )

Die qualitative Relation  $e_{k,n}^{obs}$  wird folgendermaßen definiert:

$$e_{k,n}^{obs} : (ATT_{k,n}^{in}, ATT_{k,n}^{out}, VAR_{k,n}^{func}, VAR_{k,n}^{sta})^{qual} = (VAL_{k,n}^{in}, VAL_{k,n}^{out}, VAL_{k,n}^{func}, VAL_{k,n}^{sta}).$$

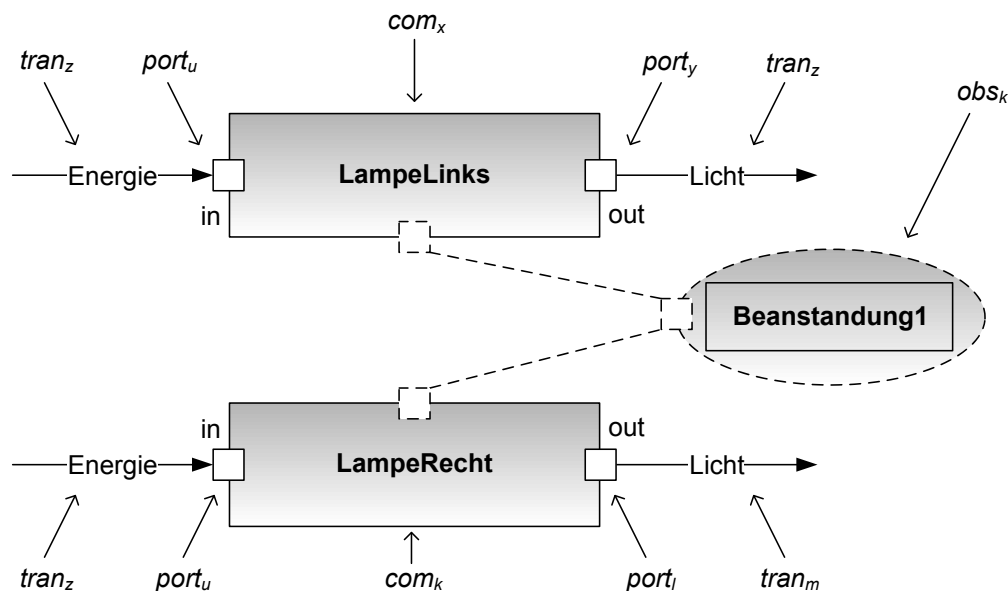


Abbildung 3.14: Beispielnotation einer Kundenbeanstandung im Lichtsystem

Zur Verdeutlichung wird dieses Vorgehen anhand der Kundenbeanstandung „Starter dreht, Motor springt nicht oder schlecht an“ erläutert. Hier gibt es in einige Interpretationsmöglichkeiten, wie zum Beispiel:

- 1. Möglichkeit
  - Ort: Ausgangsschnittstelle des Abgassystems - Ausgangsschnittstelle des Abgasrückführungsventils
  - Abweichung:  $AGRVentils.Ausgangsschnittstelle.Abgas.Massenfluss^{qual} = hoch$ ;
- 2. Möglichkeit
  - Ort: Ausgangsschnittstelle des Luftsystems - Ausgangsschnittstelle der Drallklappe
  - Abweichung:  $Drallklappe.Ausgangsschnittstelle.Luft.Massenfluss^{qual} = niedrig$ .

Im oben genannten Beispiel ist die Abweichung der Modellgröße nur an eine Komponente gebunden. Es existieren Kundenbeanstandungen, die an mehrere Modellgrößen und an verschiedene Komponenten gebunden sind. Ein Beispiel dafür ist die Beanstandung „Leuchtweite unterschiedlich“ (Abbildung 3.14). Hier ist eine Abweichung im Lichtsystem bei den Komponenten „linke Lampe“ und „rechte Lampe“ beobachtet worden und die physikalischen Ursachen sollen in qualitativer Relation stehen:

$$\begin{aligned} & \text{LampeLinks.Ausgang.Licht.Neigungswinkel} \\ & \stackrel{qual}{=} \text{LampeRechts.Ausgang.Licht.Neigungswinkel}. \end{aligned}$$

Da diese Abweichung in Form einer qualitativen relationalen Funktion aufgefasst wird, wird der Algorithmus zur Erzeugung der Verhaltenstabelle verwendet, um alle möglichen Wertebeglegungen der Attribute zu berechnen. Es seien mit:

$$\begin{aligned} & \text{tab}_k^{obs} \text{ - Beobachtungstabelle einer Kundenbeanstandung } \text{obs}_k \\ & \text{ROW}_k^{obs} \text{ - Menge aller Zeilen der Beobachtungstabelle } \text{tab}_k^{obs} \\ & \text{row}_{k,n}^{obs} \text{ - Die n-te Zeile der Beobachtungstabelle .} \end{aligned}$$

Dann gilt:

$$\begin{aligned} & \text{row}_{k,n}^{obs} \in \text{ROW}_k^{obs} \\ & \text{tab}_k^{obs} := \text{TUPEL}(E_k^{obs}, \text{VAL}_k^{in}, \text{VAL}_k^{out}, \text{VAL}_k^{func}, \text{VAL}_k^{sta}). \end{aligned}$$

Dieses Tupel ist eine Abbildung aus:

$$(\text{ATT}_k^{in}, \text{ATT}_k^{out}, \text{VAR}_k^{func}, \text{VAR}_k^{sta}) \stackrel{qual}{\rightarrow} (\text{VAL}_k^{in}, \text{VAL}_k^{out}, \text{VAL}_k^{func}, \text{VAL}_k^{sta}).$$

Für das Beispiel „Leuchtweite unterschiedlich“ wird folgende Tabelle erzeugt:

<i>LampeLinks.Ausgang.Licht. Neigungswinkel</i>	<i>LampeRechts.Ausgang.Licht. Neigungswinkel</i>
H	H
H	N
...	
L	L

Tabelle 3.7: Beispiel Inhalts einer Beobachtungstabelle

Da sich nicht alle Diagnoseeigenschaften der Beobachtungen mit relativ geringem Aufwand in physikalische Modellgrößen überführen lassen oder im Modell verortet werden können, sollen weitere Modellierungsmöglichkeiten für solche Eigenschaften angeboten werden. Im Beispiel der Kundenbeanstandung „Motor klingelt/ klopft“ existiert unter anderem als Fehlerursache, dass die Geometrie des Brennraums verändert worden ist. Die Abweichung der Brennraumgeometrie kann nur mit höherem Aufwand in Modellgrößen abgebildet werden. Die Existenz dieser Modellgrößen dient nur zur Abbildung einer speziellen Diagnoseinformation der Kundenbeanstandung „Motor klingelt/ klopft“, ansonsten ist die Änderung beziehungsweise Abweichung der Brennraumgeometrie in der Regel selten der Fall. Da das Verhältnis zwischen Aufwand und Nutzen bei der Modellierung und der Diagnose nicht vertretbar ist, soll die Abbildung solcher Diagnoseinformationen nicht über Modellgrößen und den dazugehörigen Ort erfolgen (Abbildung 3.15), sondern durch eine direkte Zuweisung der Kundenbeanstandung auf verdächtige Komponenten (Diagnoseregeln), in diesem Fall den Brennraum. Dabei soll ein Erklärungstext mit angegeben werden. Für die Erklärungskomponente des Diagnosetools ist der Erklärungstext notwendig, weil in diesem Fall das Diagnosemodell über kein „tieferes Wissen“ verfügt. Die Diagnoseregeln für Kundenbeanstandungen in dieser Arbeit hat folgende

Form:

- Wenn: Kundenbeanstandung  $obs_k$  existiert
- Dann:
  - Verdächtige Komponenten:  $com_i, \dots, com_{i+n}$
  - Erklärungstext: Fehlerhypothesen  $Y_1, \dots, Y_n$  beschreiben, die die Verdächtigungen der Komponenten  $com_i, \dots, com_{i+n}$  erklären

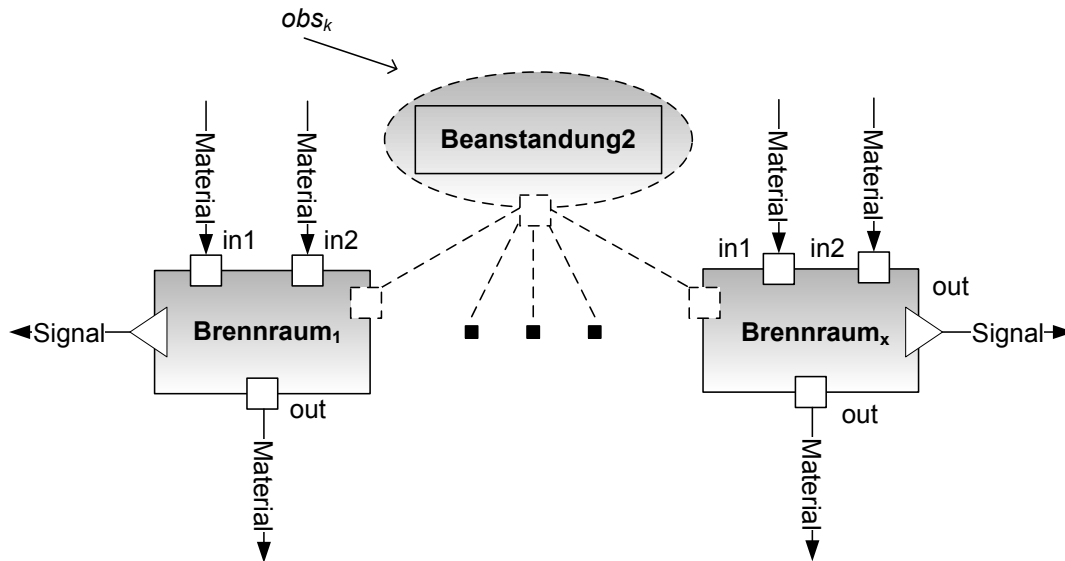


Abbildung 3.15: Kundenbeanstandung als Diagnoseregel

Die Formalisierung der Kundenbeanstandung „Motor klingelt/ klopft“ mit der Fehlerhypothese „Änderung der Brennraumgeometrie“ kann wie folgt abgebildet werden:

- Wenn: Kundenbeanstandung „Motor klingelt/ klopft“ existiert
- Dann:
  - Verdächtige Komponenten:  $Brennraum_1, \dots, Brennraum_x$
  - Erklärungstext: Änderung beziehungsweise Abweichung der Brennraumgeometrie

Eine Zuordnung zwischen den Beanstandungen und den verursachenden physikalischen Größen des Modells erfolgt durch eine virtuelle Schnittstelle des Kundenbeanstandungsmodells. Damit können sich die physikalischen Größen für die Abbildung einer bestimmten Systemanregung sowohl eingangsseitig befinden, als auch zur Beobachtung des Antwortverhaltens ausgangsseitig angeordnet sein. Des Weiteren seien:

$COM_k^{\exists obs}$  - Menge aller systemeigenen Komponenten, die durch eine Kundenbeanstandung  $obs_k$  mit Expertenwissen verdächtigt wurde

$COM_k^{-obs}$  - Menge aller systemeigenen Komponenten, die durch eine Kundenbeanstandung  $obs_k$  mit Expertenwissen nicht verdächtigt wurde.

Dann gilt:  $COM_k^{\exists obs} \cap COM_k^{-obs} = \emptyset$ .

Die Möglichkeiten der oben beschriebenen Modellierung der Kundenbeanstandungen sind als Objektdiagramm in Abbildung 3.16 dargestellt. Das Objekt *OBS\_K1* ist die Instanz der Kundenbeanstandung „Motor klingelt/ klopft“. Die Instanz *AttObs\_1* der Klasse *Zuordnung* repräsentiert die Zuordnung der Beanstandung zur physikalischen Größe im Modell. Dabei wurde eine der Ursachen dieser Beanstandung an die Ausgangsschnittstelle (*Ausgang\_SV\_1* der Instanz der Systemschnittstelle) des Kraftstoffsystems gebunden. Die Möglichkeit, eine Beanstandung als Diagnoseregeln aufzustellen, wurde mit der Instanz *DR\_K1* der Klasse *Regel* dargestellt.

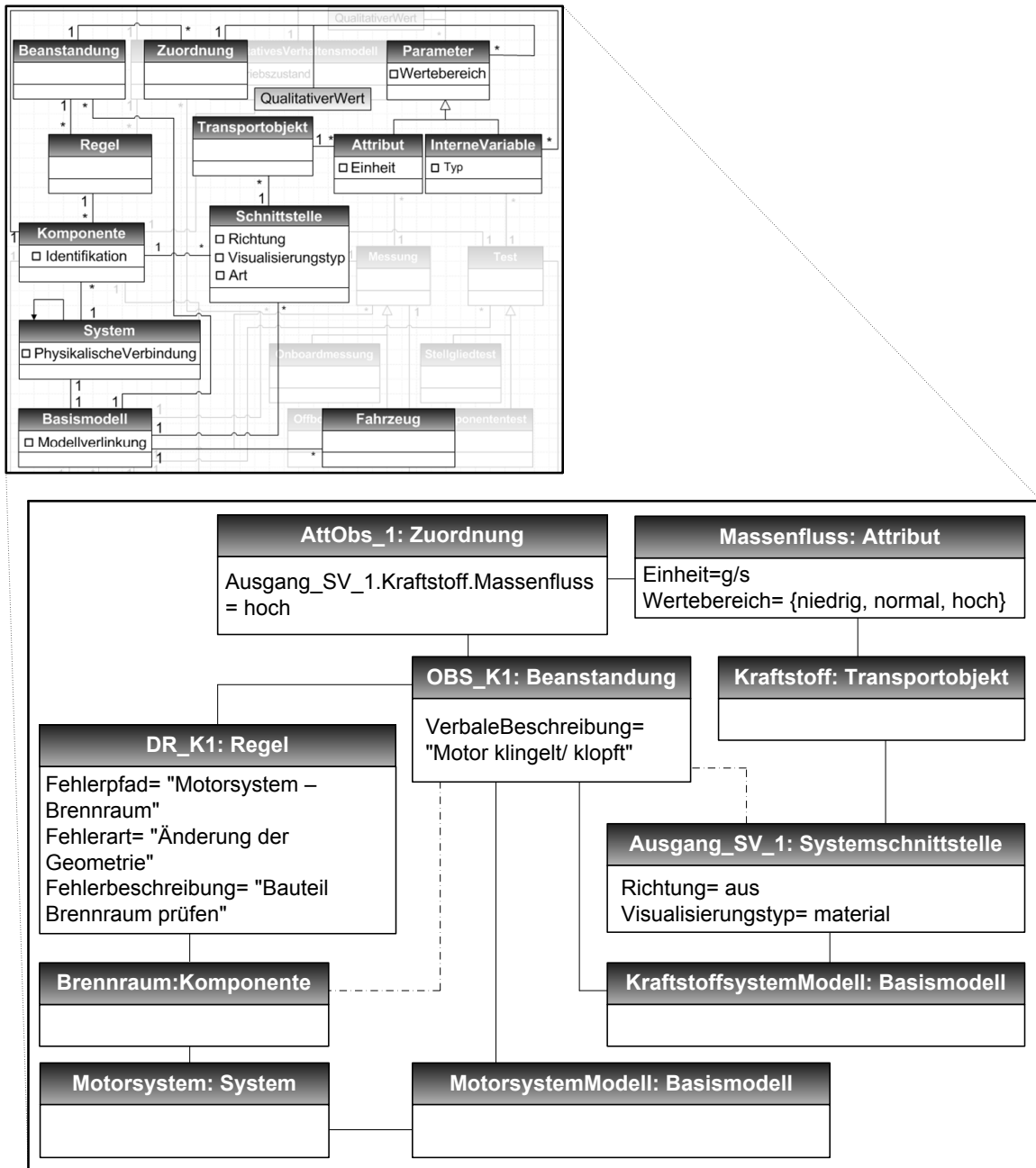


Abbildung 3.16: Beispielhaftes Objektdiagramm einer Kundenbeanstandung

### 3.7 Abbildung der Tests

Auf die Modellierung der verschiedenen Testarten in der Werkstattdiagnose wird in diesem Abschnitt eingegangen. Die Onboard-Diagnose kann eine gezielte Anregung beziehungsweise Ansteuerung von Stellgliedern (Aktoren) der Regelstrecke innerhalb des Systems durchführen. Diese Funktionsfähigkeit der Stellglieder zu überprüfen, ist unverzichtbar in einem Diagnosevorgang und wird im ersten Teil dieses Abschnitts beschrieben. Im zweiten Teil wird die Abbildung des Komponententests erläutert.

#### 3.7.1 Modellierung des Stellgliedertests

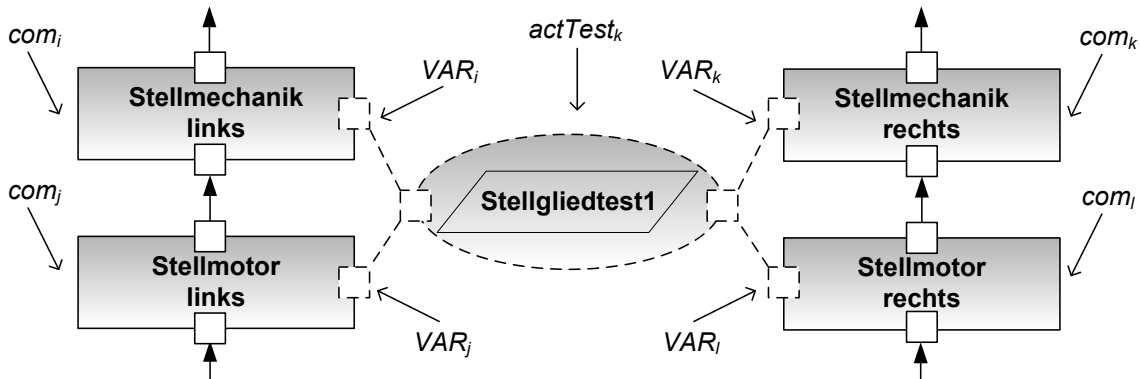


Abbildung 3.17: Beispielnotation eines Stellgliedertests im Modell des Lichtsystems

Der Stellgliedertest ist ein Test, bei dem die Regelstrecke in einem System gegen die funktionalen Anforderungen geprüft wird. Die Modellierung des Stellgliedertests ist unkompliziert, da dieser direkt an eine systemeigene Komponentengruppe gebunden ist. Da das Ergebnis des Tests der Nachweis von vorhandenen beziehungsweise nicht vorhandenen Komponentenfehlern sowie die Korrektheit der Komponentenfunktionalität ist, beinhalten die Diagnoseinformationen die Wertebelegungen der internen Funktions-, Zustands- und Fehlervariablen der zugehörigen Komponenten in der Regelstrecke (Abbildung 3.17). Es seien mit:

$ActTEST$  - Die Menge aller Stellgliedertests, die im Modell abgebildet sind

$actTest_k$  - Der  $k$ -te Stellgliedertest im Modell.

Dann gilt:  $actTest_k \in ActTEST$ .

Eine Formalisierung der Ergebnisse von Stellgliedertests hinsichtlich der Funktions-, Zustands- und Fehlervariablen ist in dieser Arbeit wie folgt definiert:

$E_k^{actTest}$  - Menge aller qualitativen Relationen, die die Stellgliedertests  $actTest_k$  beschreiben

$e_{k,n}^{actTest}$  - Die  $n$ -te qualitative Relation des Stellgliedertests  $actTest_k$ .

Dann gilt:  $e_{k,n}^{actTest} \in E_k^{actTest}$ .

Die qualitative relationale Gleichung  $e_{k,n}^{actTest}$  wird wie folgt definiert:

$$e_{k,n}^{actTest} : (VAR_{k,n}^{func}, VAR_{k,n}^{sta}, VAR_{k,n}^{faul})^{qual} = (VAL_{k,n}^{func}, VAL_{k,n}^{sta}, VAL_{k,n}^{faul}).$$

In Abbildung 3.18 sind drei Instanzen der Klassen *InterneVariable*, *Stellgliedtest* und *Messungswissen* dargestellt. Da die Ergebnisse des Stellgliedertests  $STest\_1$  eine Aussage über die interne Variable *MechanischeFehler* der Komponente *StellmechanikLinks* treffen, existiert eine direkte Verlinkung zwischen den beiden Objekten  $STest\_1$  und *MechanischeFehler*. Die Instanz  $MeasEV1_1$  der Klasse *Testwissen* beinhaltet Informationen über Sicherheitshinweise, Prüfvoraussetzung und Testauswertung. Diese Informationen sind fahrzeugabhängig,

daher existiert eine direkte Verlinkung zum Objekt *GoldV* und ein Link zum Objekt *STest<sub>1</sub>*.

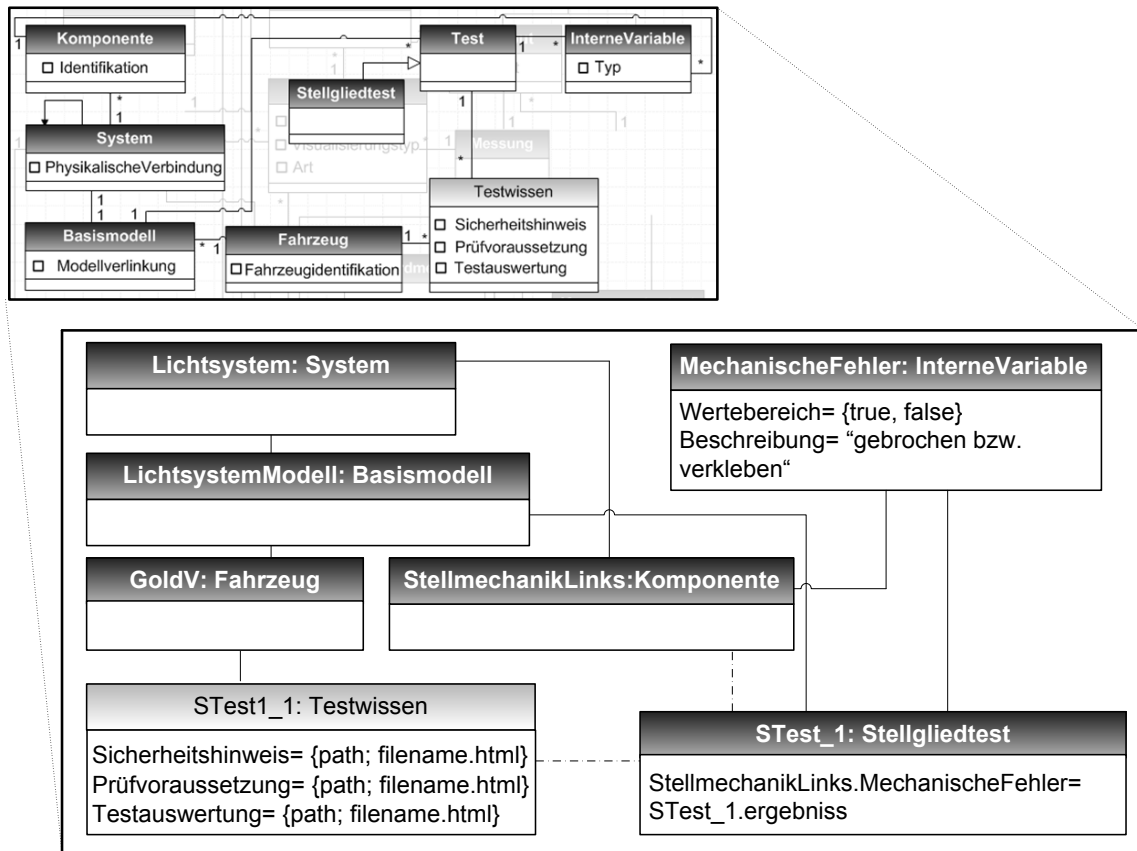


Abbildung 3.18: Beispielhaftes Objektdiagramm eines Stellgliedtests im Modell des Lichtsystems

### 3.7.2 Modellierung des Komponententests

Eine weitere Diagnoseinformation, die nur bei der Offboard-Diagnose verfügbar ist, sind die Komponentenprüfungen. Die Anbindung solcher Prüfungen beziehungsweise Tests ist unkompliziert, da auch sie direkt an eine systemeigene Komponente gebunden sind (Abbildung 3.19). Eine automatische Anbindung zwischen der Komponente und der zugehörigen Prüfung in Form einer Datenbankabfrage ist möglich, wenn die systemeigene Komponente und deren Prüfung den gleichen Identifizierungscode besitzen. Es seien mit:

$ComTEST$  - Die Menge aller Komponentenprüfungen, die im Modell abgebildet sind

$comTest_k$  - Die k-te Komponentenprüfung im Modell.

Dann gilt:  $comTest_k \in ComTEST$ .

Eine Formalisierung der Aussagen von Komponentenprüfungen hinsichtlich der Funktions-, Zustands- und Fehlervariablen ist in dieser Arbeit wie folgt definiert:

$E_k^{comTest}$  - Menge aller qualitativen Relationen, die die Komponentenprüfungen  $comTest_k$  beschreiben

$e_{k,n}^{comTest}$  - Die n-te qualitative Relation der Komponentenprüfung  $comTest_k$ .

Dann gilt:  $e_{k,n}^{comTest} \in E_k^{comTest}$ .

Die qualitative relationale Gleichung  $e_{k,n}^{comTest}$  wird wie folgt definiert:

$$e_{k,n}^{comTest} : (VAR_{k,n}^{func}, VAR_{k,n}^{sta}, VAR_{k,n}^{faul})^{qual} = (VAL_{k,n}^{func}, VAL_{k,n}^{sta}, VAL_{k,n}^{faul}).$$

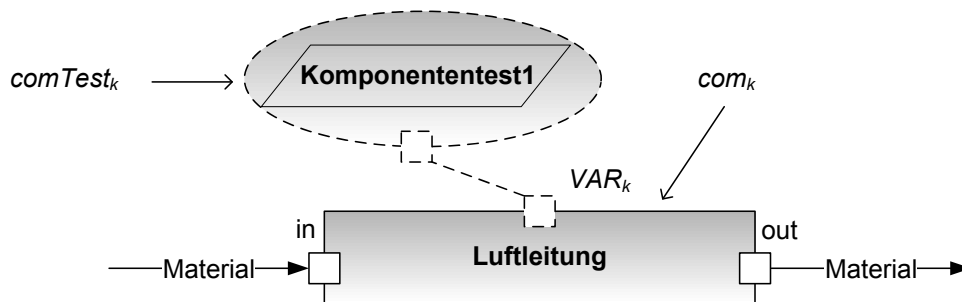


Abbildung 3.19: Beispielnotation eines Komponententests im Modell des Kraftstoffsystems

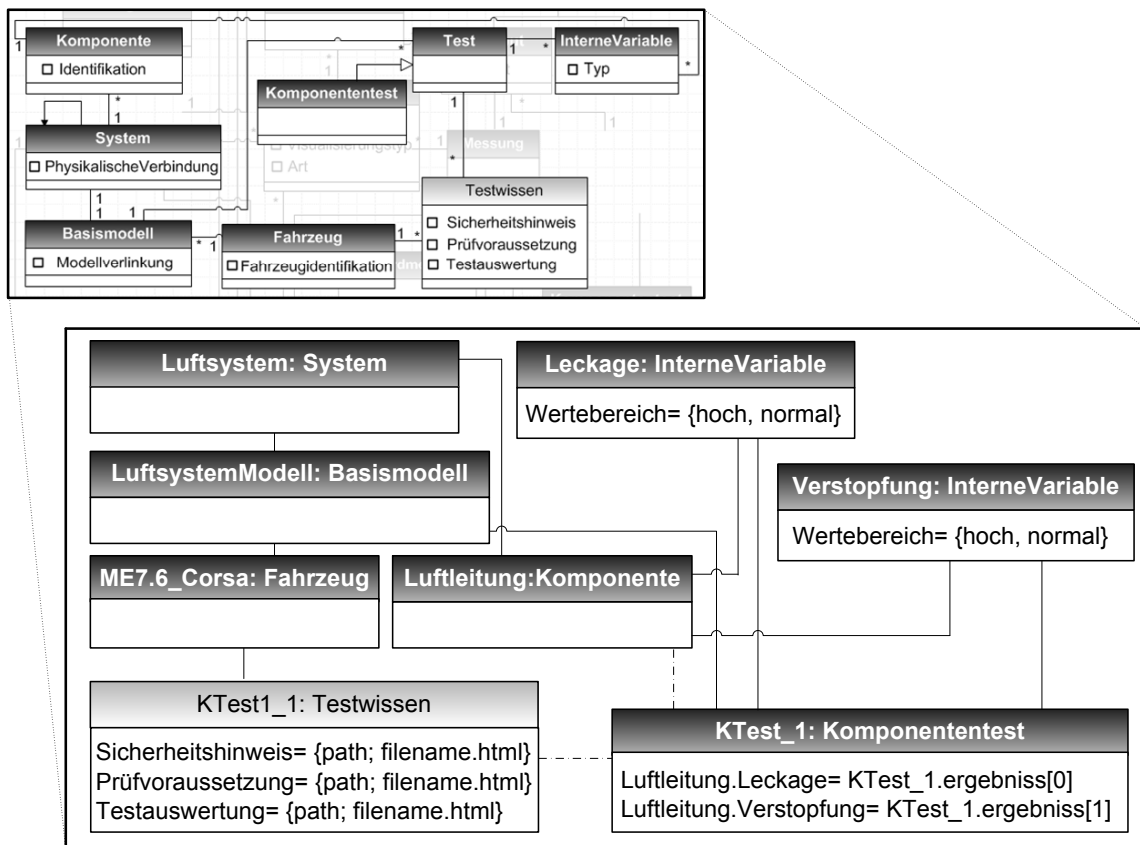


Abbildung 3.20: Beispielhaftes Objektdiagramm eines Komponententests im Modell des Lichtsystems

Um den beschriebenen theoretischen Modellierungsansatz des Komponententests zu erläutern, ist in Abbildung 3.19 die Instanz  $KTest_1$  eines Komponententests dargestellt. Die Komponente *Luftleitung* im *Luftsystem* hat die zwei internen Fehlervariablen *Leckage* und *Verstopfung*. Die Ergebnisse der Prüfung der Luftleitung auf Leckage und Verstopfung sind in dem Objekt  $KTest_1$  abgebildet. Das fahrzeugabhängige *Testwissen* wird durch das Objekt  $KTest_1$  repräsentiert.

## 3.8 Systemmodellierung

Nachdem die modelltechnische Umsetzung zur Modellierung der Systemkomponente sowie die interne Messung (systemeigener Sensor) beschrieben wurden, wird in diesem Abschnitt auf die Modellierung der Systemschnittstelle und auf die Verbindungen zwischen den Komponenten eingegangen, die notwendig sind, damit das System vollständig modelliert ist.

### 3.8.1 Systemschnittstelle

In der realen Welt sind die Schnittstellen eines Systems eine Teilmenge der Schnittstellen seiner Komponenten. Daher beinhaltet eine Systemschnittstelle analog zur Komponentenschnittstelle drei Eigenschaften: Orientierung, Transportobjekt und Attribut. Obwohl eine Komponente- und eine Systemschnittstelle ein und dasselbe reale Objekt repräsentieren, existiert in dem entwickelten Ansatz eine klare Unterscheidung zwischen Komponente- und Systemschnittstellen. Komponentenschnittstellen sind mit dem Komponentenmodell gebunden, Systemschnittstellen hingegen eigenständige Objekte.

Die Unterscheidung ist nötig, da sie zur Verbesserung der Wiederverwendbarkeit des Komponentenmodells und zur Kapselung des Systemmodells dient. Einige Schnittstellen der systemeigenen Komponenten besitzen eine zusätzliche Eigenschaft im Vergleich zu den übrigen Schnittstellen, da sie gleichzeitig Systemschnittstellen sind. Ohne eine Trennung zwischen Komponente- und Systemschnittstellen wäre die Wiederverwendung des Komponentenmodells mittels Modellbibliothek erschwert, da die genannte Eigenschaft bei der Wiederverwendbarkeit entweder bei der Schnittstelle hinzugefügt oder entfernt werden müsste. Bei einem verhältnismäßig kleinen Systemmodell, das nur eine geringe Anzahl an Systemschnittstellen beinhaltet, ist eine manuelle Modifizierung durch den Modellierer vertretbar. Bei einem komplexen Systemmodell mit einer großen Anzahl an Schnittstellen ist eine manuelle Modifizierung nicht sinnvoll. Eine Lösung dieses Problems ist die Entwicklung eines zusätzlichen Algorithmus, der diesen Unterschied erkennt und das verwendete Komponentenmodell der Modellbibliothek automatisch ändert. Der Aufwand zur Implementierung dieses Algorithmus ist gering, es sinkt damit jedoch die Wartbarkeit des Modells und eine wichtige Eigenschaft der Schnittstelle würde vor dem Modellierer versteckt werden. Mit der Definition der Systemschnittstellen ist sich der Modellierer bewusst, dass er in diesem Schritt die Systemgrenze festlegt (Abbildung 3.21).

In der Modellierungsebene, in der die Komponentenmodelle abgebildet sind, ist die Systemschnittstelle mit einem eigenständigen Symbol dargestellt. In der nächst höheren Modellierungsebene werden die Systemschnittstellen in Komponentenschnittstellen umgewandelt, weil in dieser Ebene die modellierten Systeme als Komponenten zu betrachten sind. Die Transformation ist einfach, da Systemschnittstellen alle Eigenschaften einer Komponentenschnittstelle beinhalten. Indem die Systemgrenze nur durch Systemschnittstellen definiert ist, eröffnen sich für den Diagnosevorgang mehrere Möglichkeiten zur Einsparung des Speicherbedarfs und der Rechnerzeit, wird doch nicht das komplette Modell zur Berechnung herangezogen, sondern nur ein Teil des Modells. Erst bei Bedarf werden weitere Teilmodelle geladen, bis die Diagnose abgeschlossen ist. Es seien mit:

$EXTP$  - Menge aller Systemschnittstellen

$extp_n$  - die n-te Schnittstelle eines Systems.

Dann gilt:  $extp_n \in EXTP$



Die Spezifikation der Systemschnittstelle erfolgt analog zur Festlegung der Komponentenschnittstellen. Auch hier existieren nur zwei mögliche Flussrichtungen: „In“ und „Aus“. Es seien mit:

- $EXTP^{in}$  - Menge aller Eingangsschnittstellen
- $EXTP^{out}$  - Menge aller Ausgangsschnittstellen

Dann gilt:  $EXTP = EXTP^{in} \cup EXTP^{out}$ .

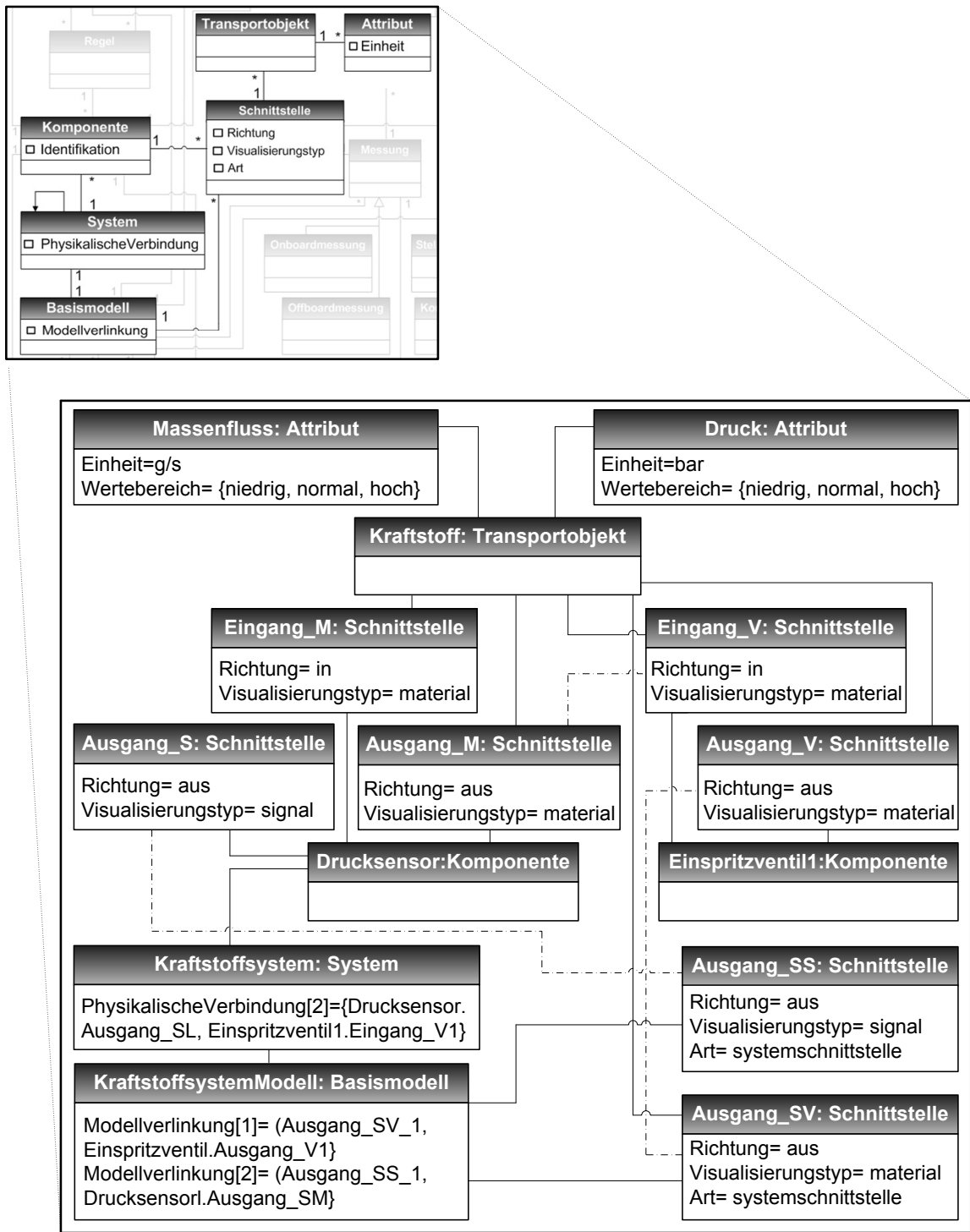


Abbildung 3.21: Beispielhaftes Objektdiagramm der Systemschnittstellen und Verbindungen

Die systemübergreifende Diagnose kann einfach über Systemschnittstellen erfolgen. Ist zum Beispiel das Symptom „Höchstleistung kann nicht erreicht werden“ im Verbrennungssystem beobachtet worden, liegt nach der Analyse des Verbrennungssystems eine der Ursachen an der Kraftstoffmenge, die in den Brennraum eingeflossen ist. Um die Ursache des Kraftstoffmangels zu bestimmen, wird das darüber liegende Systemmodell „Motorsystem“ analysiert. Anhand der Schnittstellen des Verbrennungssystems können die Analysealgorithmen in das Kraftstoffsystem gelangen und dort die Ursache lokalisieren. Diese Einfachheit der systemübergreifenden Diagnose ist mit Systemschnittstellen realisierbar; der Wechsel zwischen verschiedenen Systemen und Subsystemen bei der Analyse ist Voraussetzung für eine erfolgreiche Diagnose in großen Systemen mit komplexer Steuerung, Querswirkungen und einer großen Komponentenzahl.

### 3.8.2 Verbindung

Die Strukturmodellierung eines Systems repräsentiert folgende Eigenschaften: Zum einen die Wiedergabe des kompositionalen Charakters der modellierten Komponenten beziehungsweise der dekompositionalen Eigenschaften des modellierten Systems, zum anderen die Darstellung der Zusammenhänge zwischen den abgebildeten Komponenten und Systemschnittstellen. Die innere Struktur eines Systems, die durch die Dekompositionsbetrachtungsweise entsteht, wird in dieser Arbeit mit Hilfe von Graphen dargestellt. Dabei sind die Knoten eines Graphen die systemeigenen, externen oder virtuellen Komponenten und Systemschnittstellen. Die Zusammenhänge zwischen den Komponentenmodellen untereinander sowie zwischen Systemschnittstellen und Komponentenmodellen werden im Systemmodell durch Verbindungslinien abgebildet und symbolisieren die Interaktion der Komponenten und Systemschnittstellen. Die Verbindungen werden durch Kanten des Graphen dargestellt. Für Verbindungen, die die reale beziehungsweise physikalische Interaktion abbilden, werden Linien ohne Unterbrechung zur Visualisierung verwendet.

Dabei kann eine Verbindung ( $link_x$ ) nur existieren, wenn folgende Bedingungen gelten:

- Nur eine Ausgangsschnittstelle  $com_k.port_i^{out}$  einer Komponente  $com_k$  kann über eine Verbindung mit einer Eingangsschnittstelle  $com_{k'}.port_j^{in}$  einer anderen Komponente  $com_{k'}$  verbunden sein (Abbildung 3.21)
- In verbundenen Schnittstellen liegen dieselben transportierten Objekte mit denselben charakterisierenden Attributen vor (Abbildung 3.21). Es gelten:
 
$$\begin{aligned}
 &com_k.port_i^{out}.TRAN \subseteq com_{k'}.port_j^{in}.TRAN \\
 &\quad \wedge com_{k'}.port_j^{in}.TRAN \subseteq com_k.port_i^{out}.TRAN \\
 &com_k.port_i^{out}.tran_n.ATT \subseteq com_{k'}.port_j^{in}.tran_n.ATT \\
 &\quad \wedge com_{k'}.port_j^{in}.tran_n.ATT \subseteq com_k.port_i^{out}.tran_n.ATT
 \end{aligned}$$
- Seien  $att_y$  ein Attribut eines transportierten Objekts  $tran_n$  einer Ausgangsschnittstelle  $port_i^{out}$  einer Komponente  $com_k$  und das entsprechende Attribut  $att_z$  des transportierten Objekts  $tran_n$  an der verbundenen Eingangsschnittstelle  $port_j^{in}$  der nachfolgenden Komponente  $com_{k'}$ , dann gilt  $val_y = val_z$ , das heißt, der Wert eines Attributs an der Eingangsschnittstelle einer verbundenen Komponente ist gleich dem Wert des gleichen Attributs an der Ausgangsschnittstelle der verbundenen Komponente.

### 3.9 Parametrisierung und Bedatung des Modells

Das Einbinden weiterer diagnoserelevanter Informationen wie Soll- und Grenzwerte der Sensorensignale, Ausfallwahrscheinlichkeit, entstehende Kosten bei Messung, Prüfung oder Reparatur von systemeigenen Komponenten, etc. erfolgt bei der Parametrisierung beziehungsweise Bedatung des Modells.

Die individuellen Komponentendaten beziehungsweise Informationen sind von Fahrzeugtyp zu Fahrzeugtyp unterschiedlich und erschweren die Wiederverwendbarkeit, Pflege und Wartung des erstellten Modells erheblich. Eine Trennung der individuellen Daten von den Komponentenmodellen ist notwendig, weil dies den generischen Charakter der Modelle unterstützt und damit die Variantenvielfalt beherrschbar macht. Die Informationen über Soll- und Grenzwerte sowie Prüfungsmöglichkeiten der Komponenten sind notwendig, da ohne Sollwerte die Verwendung der qualitativen Werte  $\{niedrig, normal, hoch\}$  nicht realisierbar und ohne Prüfungsmöglichkeiten die Ausschließung der verdächtigen Komponenten nicht möglich ist. Dagegen sind Informationen über Ausfallwahrscheinlichkeiten und Kosten optional; diese Informationen dienen zur Optimierung des Diagnosevorgangs aber nicht zur Minimierung der Menge der verdächtigen Komponenten.

Der Vorteil dieser Lösung ist die Kompletttrennung zwischen Modellierungs- und Parametrisierungsphase. Zusätzlich erfolgt in der Parametrisierungsphase eine Trennung zwischen Basis- und Optimierungsmodellparametern. Dank dieser Aufteilung beschäftigt sich der Modellierer nicht mit technischen Daten wie Testmöglichkeiten, Soll- und Grenzwerten einer Komponente, sondern konzentriert sich darauf, wie das System aufgebaut ist, welche Komponente im betrachteten System verbaut wurde und welches Verhalten sie aufweist.

Technische Daten einzelner Komponenten, die in dieser Arbeit als Basismodellparameter bezeichnet werden, können in einer separaten Datenbank eingepflegt werden. Wenn Daten der entsprechenden Hersteller vorliegen, können solche Daten mit geringem Aufwand in die Datenbank importiert werden. Eine Automatisierung des Datenimports ist möglich. Die Existenz einer derartigen externen Datenbank mit den Basismodellparametern der Komponenten ist unbedingte Voraussetzung für das korrekte Arbeiten des Diagnosesystems. Der größte Teil der darin gespeicherten Daten wird bereits in vorhandenen Systemen verwendet, denn diese Informationen sind unverzichtbar, gleichgültig welcher Diagnoseansatz verfolgt wird. Daher ist der erforderliche Aufwand zur Erstellung des neuen Diagnosesystems, das mit dem vorgestellten Modellierungskonzept arbeitet und einer zugehörigen Basismodellparameterdatenbank, relativ gering und der Aufbau ist mit vergleichsweise niedrigem Aufwand verbunden.

Das Ergebnis der Trennung von Modellen und komponentenspezifischen Daten führt zu einer enormen Reduzierung der Anzahl benötigter Modelle und gleichzeitig zur Erzeugung getrennter Aufgabenbereiche, welche parallel abgearbeitet werden können. Die Wartung und weitere Entwicklung des Modells sowie der Basis- und Optimierungsmodellparameter können unabhängig voneinander geschehen. Die einzige Voraussetzung dieser Lösung ist die Konsistenz der Fahrzeug- und Komponentenidentifizierungsnummer in allen Datenbanken.

### 3.10 Behandlung der Variantenvielfalt

Die zunehmende Komplexität und Variantenvielfalt der Systeme im Fahrzeug ergeben sich aus der steigenden Kundenheterogenität und der Kostensenkung durch das Baukastenprin-

zip in der Entwicklung sowie Fertigung ([141], [106], [140], [10]). In der Vergangenheit wurden viele Arbeiten zum Thema Variantenmanagement am mechatronischen System veröffentlicht, in denen jedoch ausschließlich die Beherrschung der Varianten bei der Produktion und im Vertrieb behandelt wurde. Die Beherrschung der Komplexität und Variantenvielfalt der Fahrzeugsysteme/-funktionen in der freien Werkstattdiagnose bezieht sich auf die letzte Phase des Produktlebenszyklus - die Wartung und wird kaum betrachtet. Im Bereich der Softwareentwicklung existieren einige Lösungen zum Variantenmanagement, die speziell für die Wiederverwendung der Modelle geeignet sind ([60], [96], [58], [59], [97], [101], [117], [76]). Des Weiteren wurde in der modellgetriebenen Softwareentwicklung (*MDA*) der Modelltransformationsansatz vorgestellt. Dieser Ansatz ist zum Teil für die System-/Funktionsmodellierung sowie Wiederverwendung der Submodelle geeignet ([91], [8], [2], [79], [153], [45], [139]). Da die Diagnosemodelle sich durch einen Graphen repräsentiert lassen, sind solche Ansätze auf die Wiederverwendung der Struktur von Diagnosemodellen anwendbar ([123], [98]). Aufgrund der Gegebenheiten bei der Konstruktion und Produktion der Fahrzeugsysteme/-funktionen basiert das Lösungskonzept für die Problematik der Komplexität und Variantenvielfalt in dieser Arbeit auf verschiedenen Lösungsansätzen in der Softwareentwicklung. Neben der Wiederverwendung des Modellwissens in Form von Komponenten- und Systemmodellbibliotheken werden weitere Lösungen entwickelt.

Eine Differenz zwischen unterschiedlichen Diagnosemodellvarianten liegt bei der Parametrisierung (siehe Abschnitt 3.9). Hier existiert eine Parallelität zwischen objektorientierter Programmierung und der Erzeugung der Diagnosemodelle (Abbildung 3.22).

$$(Klasse \xrightarrow{\text{Instanziierung}} \text{Objekt}) \Leftrightarrow (PhysikalischesSystemModell \xrightarrow{\text{Instanziierung}} \text{VollstaendigesSystemModell})$$

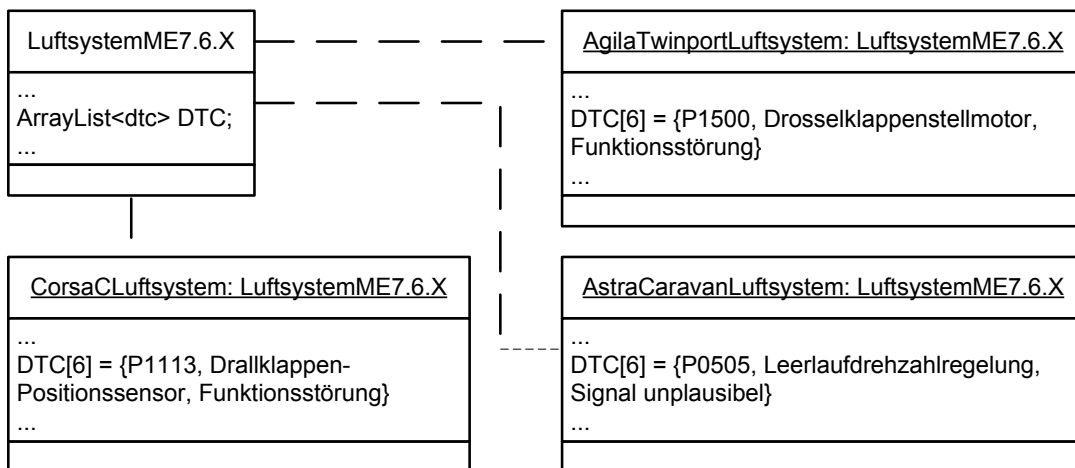


Abbildung 3.22: Vollständige Modelle als Instanzen des physikalischen Modells am Beispiel des Luftsystems-ME7.6.x

Wegen der unterschiedlichen Konstruktions- und Produktionsanforderungen existieren viele Komponenten in mechatronischen Systemen, die die gleichen Funktionen haben, doch unterschiedlich ausgelegt sind. Zum Beispiel existieren für die Verdichtung der Frischluft im Ansaugtrakt unterschiedliche Aufladekomponenten: Mechanische Kreisel-/Verdrängerlader, Druckwellenlader. Dabei wird die Verdichtungsleistung aus einer mechanischen beziehungsweise strömungstechnischen Koppelung oder einer Kombination beider Ansätze gewonnen.

Für die Modellbildung können all diese Auflader zu einer Komponente *Kompressor* generalisiert werden (Abbildung 3.23). Der Umgang mit Spezialisierung und Generalisierung in [58] für existierende Source-Codes bei der Softwareentwicklung wird in dieser Arbeit für die Modellentwicklung adaptiert und im Wissenserhebungstool implementiert.

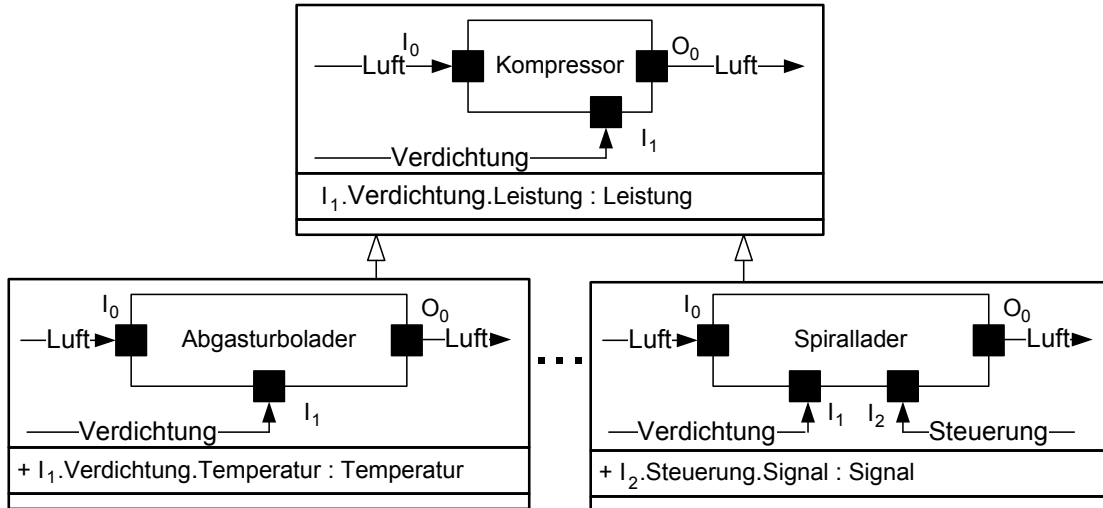


Abbildung 3.23: Generalisierung der Varianten am Beispiel der Luftlader

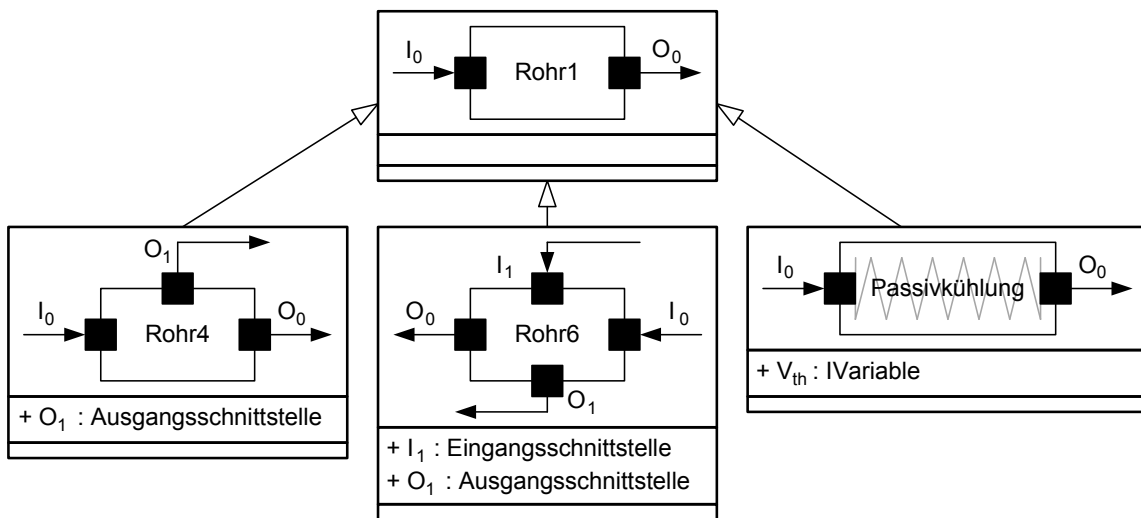


Abbildung 3.24: Vererbung der Schnittstelle sowie der internen Variable am Beispiel des Kraftstoffrohrs

Ein weiteres Basiskonzept der Objektorientierung, das sich auch in dieser Arbeit wiederfindet, ist die Vererbung. Bei dem Komponentenmodell äußert sich dieses Konzept durch die Vererbung der Komponentenschnittstellen und der internen Variablen der Komponente (Abbildung 3.24). Die Erweiterung der Ausgangsschnittstelle  $O_1$  der Komponente *Rohr4* oder der internen Variable  $V_{th}$  der Komponenten *Passivkühlung* bewirkt eine Änderung des vorhandenen Verhaltens. Analog zur Klasse, die eine andere Klasse implementiert, ist die Änderung der Verhaltensbeschreibung gleich der Änderung der Methode in der implementierten Klasse. Zum Beispiel wird das Verhalten des *Rohr1* wie folgt durch *Rohr4* beziehungsweise *Passivkühlung* erweitert:

1. Rohr1  $\xrightarrow{\text{Vererbung}}$  Rohr4:

$$\text{Rohr1.I}_0.\text{Kraftstoff.volumenstrom} \stackrel{\text{qual}}{=} \text{Rohr1.O}_0.\text{Kraftstoff.volumenstrom} \xrightarrow{\text{Vererbung}}$$

$$\text{Rohr4.I}_0.\text{Kraftstoff.volumenstrom} \stackrel{\text{qual}}{=} \text{Rohr4.O}_0.\text{Kraftstoff.volumenstrom} \stackrel{\text{qual}}{+} \text{Rohr4.O}_1.\text{Kraftstoff.volumenstrom}$$

2. Rohr1  $\xrightarrow{\text{Vererbung}}$  Passivkuehlung:

$$\text{Rohr1.I}_0.\text{Kraftstoff.temperatur} \stackrel{\text{qual}}{=} \text{Rohr1.O}_0.\text{Kraftstoff.temperatur} \xrightarrow{\text{Vererbung}}$$

$$\text{Passivkuehlung.I}_0.\text{Kraftstoff.temperatur} \stackrel{\text{qual}}{-} \text{Passivkuehlung.V}_{th} \stackrel{\text{qual}}{=} \text{Passivkuehlung.O}_0.\text{Kraftstoff.temperatur}$$

Die Wiederverwendung des Diagnosewissens durch Generalisierung und Spezialisierung in der Komponentenmodellebene unterstützt das Modelltransaktionskonzept zur Variantenbehandlung in der Systemmodellebene.

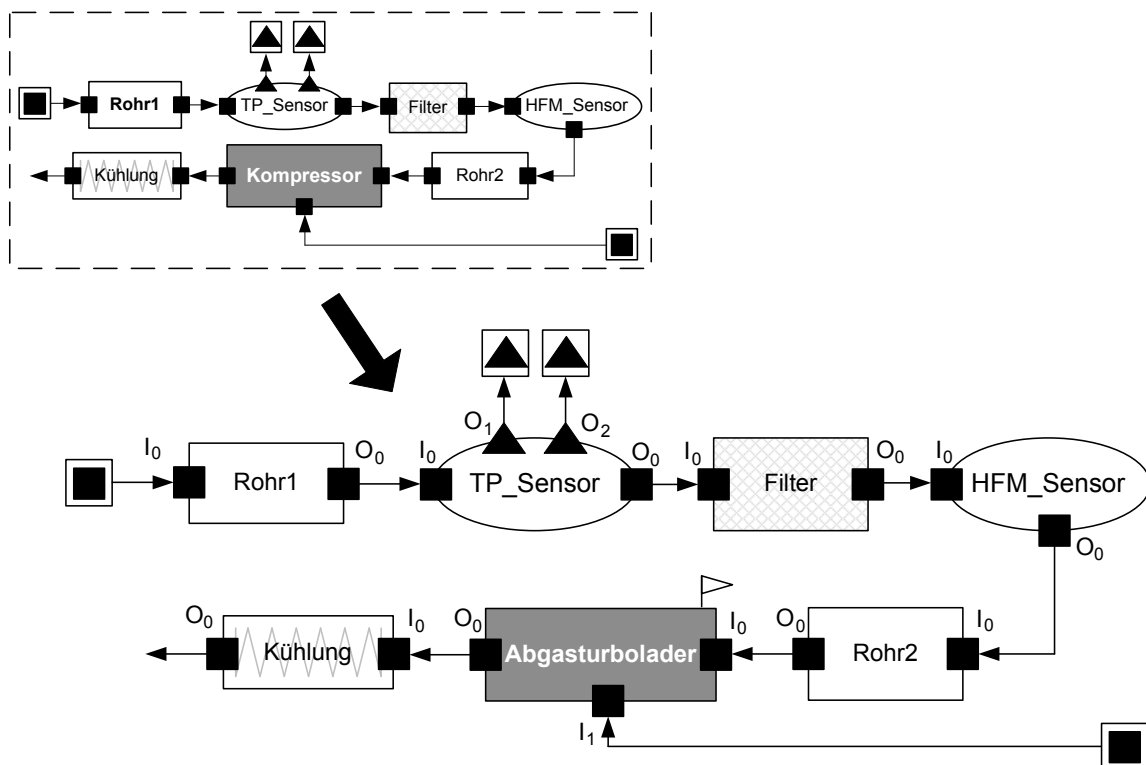


Abbildung 3.25: Modelltransformation am Beispiel eines Teils des Luftsystems mit Lader

Eine Art der Modelltransformation in MDA ist die Modellmodifikation (*Inplace Transformation*). Dabei wird kein neues Modell erzeugt; die Transformation begrenzt sich auf die Modifizierung des Quellmodells ([115]). In Abbildung 3.25 wurde ein Teil des Luftsystems als Beispiel für die Anwendung dieses Ansatzes dargestellt. Das Quellmodell beinhaltet die Komponente *Kompressor*, die in dem konkreten Diagnosemodell für Mercedes Benz 220CDI (CP3-DS) durch die Komponente *Abgasturbolader* ersetzt wird.

Ein weiterer Ansatz zur Behandlung von Modellvarianten in MDA ist die Modelltransformation. Diesem Lösungsansatz ähnelt die Wiederverwendung eines Teilmodells bei der Modellzeugung in System- beziehungsweise Subsystemebene in dieser Arbeit. Diese Transformati-

onsart ermöglicht eine komplexere Änderung des Quellmodells. Hierbei werden neue Komponenten hinzugefügt und bestehende Komponenten verändert beziehungsweise entfernt ([139]). Am Beispiel des Lichtsystems mit Kurvenlichtfunktion (Abbildung 3.26) wird das Quellmodell des ALWR-Systems (automatisches Leuchtweitenregulierungssystem) um fünf Komponenten, die für die Kurvenlichtfunktion zuständig sind, erweitert. Weitere Änderungen betreffen die existierenden Komponenten. So ist die Komponente *Leistungsmodul rechts* um eine Eingangs- und eine Ausgangsstelle erweitert worden.

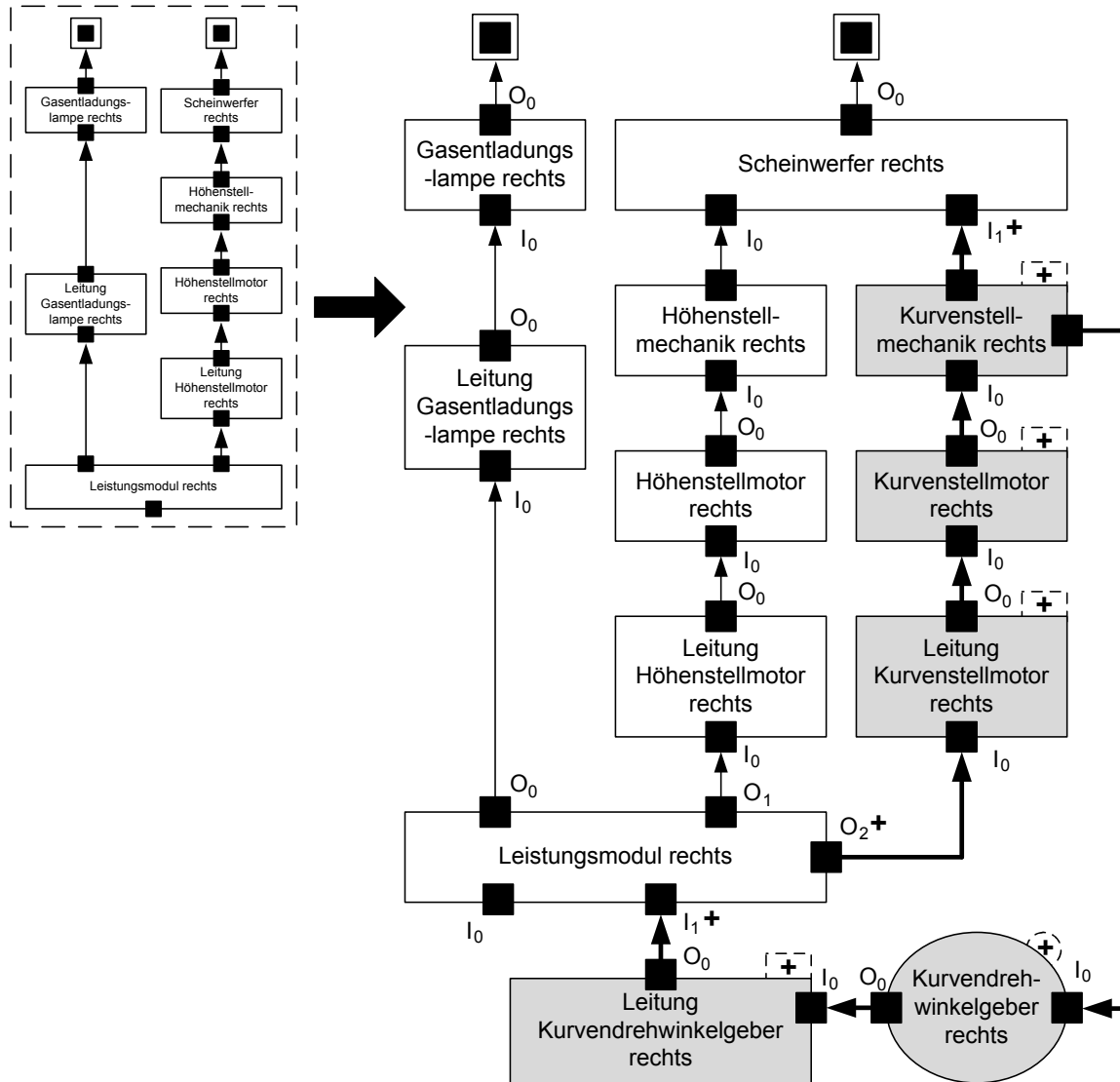


Abbildung 3.26: Modellerweiterung am Beispiel eines Teils des Lichtsystems

Neben der Spezifikation des Verhaltens in Form von Relationsgleichungen beinhaltet die Wissenserhebung die Modellierung der System-/Subsystemstruktur. Die Präsentation des Strukturwissens in dieser Arbeit ist durch Graphen formalisiert. Es existiert eine Reihe von Veröffentlichungen über Graphentransformation mit dem Ziel der Wiederverwendung des Modellwissens ([1], [18], [13], [48], [49], [75], [116], [119], [120], [142]). Die Modelle, die in verschiedenen Artikeln als Beispiele verwendet werden, sind häufig State-Diagramme beziehungsweise Klassen-Diagramme. Daher sind die Lösungsansätze nicht direkt in dieser Arbeit anwendbar. Ein nützlicher Ansatz für die Wiederverwendung des Diagnosemodells ist die

Markierung der Knoten in Graphen. Im Unterschied zum Variantenmanagement in DMA ist eine Art der Markierung die Verwendung von eindeutigen Funktionsbezeichnern der Knoten (Komponenten beziehungsweise Subsysteme). Im Gegensatz zu [8] werden hierbei nicht nur die eindeutigen Funktionsbezeichnungen für den Vergleich zwischen den Modellen (Graphen), sondern auch die internen Variablen, Komponentenschnittstellen und Zeilen der Verhaltenstabelle verwendet. Die Differenz zwischen den Modellen der Komponenten wird durch die Bildung der Schnittmenge berechnet. Der Abstand zwischen den Komponenten  $com_i$  und  $com_j$ , die gleiche Funktion besitzen ( $Funktion(com_i) \equiv Funktion(com_j)$ ), wird wie folgt berechnet:

1.  $PORT_{j/i}^{real} = PORT_j^{real} / PORT_i^{real}$  (Differenz der Komponentenschnittstellen)
2.  $VAR_{j/i} = VAR_j / VAR_i$  (Differenz der internen Variablen)
3.  $ROW_{j/i}^{beh} = ROW_i^{beh} / ROW_j^{beh}$  (Differenz der Zeilen der Verhaltenstabelle)
4. Der Abstand zwischen den Komponenten  $com_i$  und  $com_j$  ( $distance(com_i, com_j)$ ) ergibt sich aus:

$$distance(com_i, com_j) = p * |PORT_{j/i}^{real}| + v * |VAR_{j/i}| + r * |ROW_{j/i}^{beh}|$$

wobei  $p$ ,  $v$  und  $r$  die Faktoren für Schnittstellen, interne Variablen und die Zeilen der Verhaltenstabelle sind.

Die automatisierte Zusammenführung des Modellwissens auf Komponentenebene erfolgt durch die Bildung der Vereinigungsmengen bezüglich Komponentenschnittstellen, internen Variablen sowie Verhaltenstabellen. Analog der Berechnung des Abstands zwischen zwei Komponenten erfolgt die Vereinigung der beiden Komponenten. Es seien mit:

$$com_k = com_i \cup com_j | Funktion(com_i) \equiv Funktion(com_j)$$

Dann werden die Komponentenschnittstellen, internen Variablen und die Zeilen der Verhaltenstabelle der Komponente  $com_k$  wie folgt berechnet:

1.  $PORT_k^{real} = PORT_j^{real} \cup PORT_i^{real}$  (Schnittmenge der Komponentenschnittstellen)
2.  $VAR_k = VAR_j \cup VAR_i$  (Schnittmenge der internen Variablen)
3.  $ROW_k^{beh} = ROW_i^{beh} \cup ROW_j^{beh}$  (Schnittmenge der Zeilen der Verhaltenstabelle)

Weitere Markierungen kennzeichnen Komponenten, die in einem System mit bestimmten Funktionen zwingend erforderlich sind ([44]). Zum Beispiel beinhalten alle Luftsysteme des Dieselmotors einen Luftfilter, mindestens einen Luftlader, eine Drosselklappe und einen Drucksensor. Diese Markierung erleichtert die Wiederverwendung des Strukturwissens bei der Modellierung eines neuen Systems, das die gleiche Funktion wie das vorhandene System besitzt. Zu dem unterstützt diese Markierung die Ermittlung der gemeinsamen Teilstrukturen im Modell. Die Differenz zwischen zwei Modellen, die die gleiche Funktion erfüllen ( $Funktion(MOD_i) \equiv Funktion(MOD_j)$ ), wird wie folgt berechnet:

1.  $COM_{j/i} = COM_j / COM_i$  (Differenz der Komponentenmenge)
2.  $EXTP_{j/i} = EXTP_j / EXTP_i$  (Differenz der Systemschnittstellenmenge)
3.  $LINK_{j/i} = LINK_j / LINK_i$  (Differenz der Menge der Verbindungen)
4. Der Abstand zwischen zwei Modellen  $MOD_i$  und  $MOD_j$  ( $distance(MOD_i, MOD_j)$ ) ergibt sich aus:



$$\text{distance}(\text{MOD}_i, \text{MOD}_j) = c * |\text{COM}_{j/i}| + e * |\text{EXTP}_{j/i}| + l * |\text{LINK}_{j/i}|$$

wobei  $c$ ,  $e$  und  $l$  die Faktoren für Komponenten-, Systemschnittstellen- und die Verbindungenmenge sind.

Bei der Zusammenführung der Systemmodelle werden die erforderlichen Komponenten, die maskiert sind, automatisch mitgenommen. Die Zusammenführung der restlichen Komponenten, Verbindungen und Systemschnittstellen obliegt dem Modellierer. Die letzte Markierungsart in dieser Arbeit markiert Komponenten, die mehrere Spezialisierungen besitzen. Die Komponente *Luftlader* (Abbildung 3.23) ist ein gutes Beispiel dafür. Diese Art erlaubt die Nutzung von abstrakten Modellen bei der Diagnose. Die Konkretisierung der Komponente für einen bestimmten Fahrzeugtyp beziehungsweise Systemart kann zu einem späteren Zeitpunkt erfolgen.

### 3.11 Modellprüfung

Die Vollständigkeit und Korrektheit eines Modells kann durch formale Beweise beziehungsweise durch Modellchecking sichergestellt werden. Diese Verfahren sind in der Regel nur auf einfache Modelle mit vertretbarem Aufwand anwendbar. Da das erstellte Diagnosemodell nicht nur physikalische Verhalten, sondern auch heuristisches Wissen beinhaltet, ist das einzige Mittel neben Review, um eine Aussage über die Qualität des erstellten Diagnosemodells mit vertretbarem Aufwand zu treffen, der Modelltest. Der Modelltest ist eine Aktivität, bei der das zu überprüfende Modell oder Teile des zu überprüfenden Modells mit Eingaben stimuliert und ihre Reaktionen beobachtet und bewertet werden ([9], [12], [16], [50], [89], [94], [63], [64], [108], [118]). Aufgrund des kontextfreien, lokalen Modellierungsansatzes der Funktionen und des Verhaltens der Komponenten, Teilsysteme und Systeme können Tests auf verschiedenen hierarchischen Detailebenen beziehungsweise Teststufen stattfinden (Abbildung 3.27):

- Der Komponententest testet die kleinsten testbaren Einheiten eines Systemsmodells. Hierfür werden die Komponentenschnittstellen und das Verhalten der Komponenten im Normal- und Fehlerfall geprüft.
- Der Integrationstest testet die Interaktionen der Komponenten durch sukzessives Aufbauen und überprüft das Verhalten immer größerer Teilsysteme.
- Der Systemtest testet das erstellte Modell als Ganzes. Dabei wird das gesamte Systemmodell im Sinne der Diagnosefähigkeit getestet.

Analog zur Methode des Softwaretests kann der Komponententest mit dem White-Box-Test, der Integrationstest mit dem Grey-Box-Test und der Systemtest mit dem Black-Box-Test verglichen werden ([14], [77], [84], [95], [147]). Entsprechend dem White-Box-Test kann der Modellersteller beziehungsweise Modellierer den Testfällen in Bezug auf interne Variablen und Komponentenschnittstellen des Komponentenmodells die Werte zuweisen. Testfälle, die das Komponentenmodell erfüllen muss, beinhalten folgende Überdeckungskriterien:

- Schnittstellenüberdeckung:
  - Überprüfung der Anzahl der Eingangs- und Ausgangsschnittstellen

- Betrachtung aller Schnittstellen mit ihren Transportobjekten und der beinhalteten Attribute mit dem dazugehörigen abstrakten Wertebereich
- Interne Variablenüberdeckung: Untersuchung aller modellierten Variablen und deren qualitativen Auftretenswahrscheinlichkeiten.

Testfälle, die die Überdeckungskriterien überprüfen, werden in dieser Arbeit Strukturtestfälle genannt. Diese Testfälle können aus der Funktionsbeschreibung beziehungsweise Spezifikation der Komponente gewonnen werden.

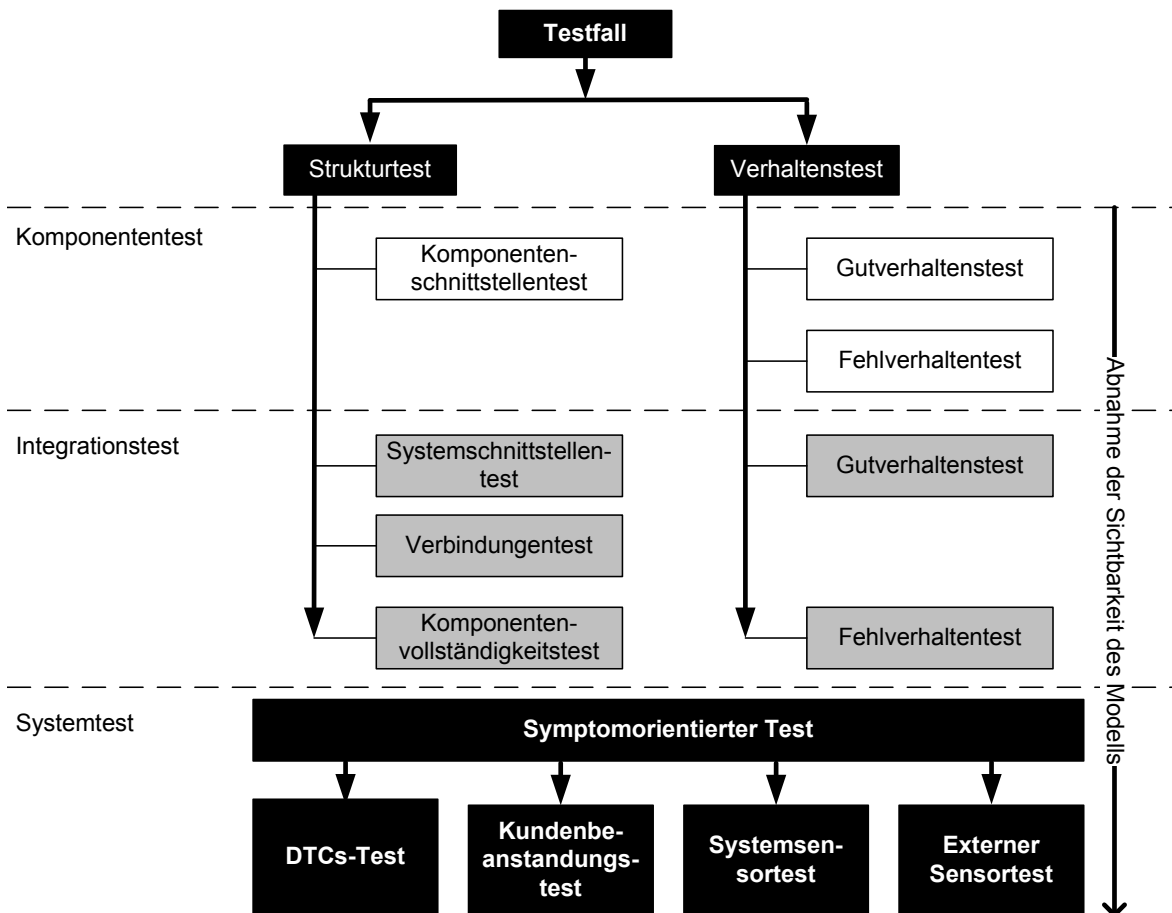


Abbildung 3.27: Verschiedene Testarten des erstellten Diagnosemodells

Analog zum White-Box-Test kann der Modellierer die Erfüllung der Verhaltensspezifikation der modellierten Komponente prüfen (Verhaltenstest). Hierbei kann der Modellierer den Eingangs- und Ausgangsattributen sowie den internen Variablen Werte zuweisen. Anhand der vordefinierten Werte kann der Komponententest angestoßen werden. Die dafür benötigte Methode reduziert die Verhaltenstabelle der Komponente und splittet die Ergebnisse in mehrere Teiltabellen, zum Beispiel Tabelle mit Eingangsfehlern und internen Fehlern, Tabelle mit ausschließlich internen Fehlern, etc. Der Algorithmus, der diese Vorhaben unterstützt, wird in 4.2 beschrieben. Der Test beziehungsweise die Prüfung des modellierten Komponentenverhaltens kann durch Vergleich des erwarteten Komponentenverhaltens mit dem tatsächlich modellierten Verhalten durchgeführt werden. Die Testfälle für das „normale“ Verhalten einer Komponente werden automatisch von der Testplattform generiert. Dabei werden alle

Eingangsattribute sowie interne Variablen auf „normal“ gesetzt, die erwarteten Ausgangsattribute müssen logischerweise auch den Wert „normal“ annehmen. Für eine Komponente  $com_i$  können die Testfälle für das „normale“ Verhalten wie folgt formal definiert werden:

1. Vorbedingungen:

Es seien mit:

- $\forall port_j^{real,in} \in PORT_i^{real,in}$ , wobei  $PORT_i^{real,in}$  die Menge der Eingangsschnittstellen der Komponente  $com_i$  ist,
- $\forall tran_k^{in} \in TRAN_j^{in}$ , wobei  $TRAN_j^{in}$  die Menge der Transportobjekte ist, die sich in der Eingangsschnittstelle  $port_j^{real,in}$  befinden,
- $\forall att_l^{in} \in ATT_k^{in}$ , wobei  $ATT_k^{in}$  die Menge der Attribute ist, die sich in dem Transportobjekt  $tran_k^{in}$  befinden,
- $\forall var^j \in VAR_i$ , wobei  $VAR_i$  die Menge der internen Variablen der Komponente  $com_i$  ist,

dann:

- $val_l^{att,in} = normal$ , wobei  $val_l^{att,in}$  die Wertebelegung des Attributes  $att_l^{in}$  ist,
- $val_j^{var} = normal$ , wobei  $val_j^{var}$  die Wertebelegung der internen Variable  $var^j$  ist.

2. Nachbedingungen:

Es seien mit:

- $\forall port_j^{real,out} \in PORT_i^{real,out}$ , wobei  $PORT_i^{real,out}$  die Menge der Ausgangsschnittstellen der Komponente  $com_i$  ist,
- $\forall tran_k^{out} \in TRAN_j^{out}$ , wobei  $TRAN_j^{out}$  die Menge der Transportobjekte ist, die sich in der Eingangsschnittstelle  $port_j^{real,out}$  befinden,
- $\forall att_l^{out} \in ATT_k^{out}$ , wobei  $ATT_k^{out}$  die Menge der Attribute ist, die sich in dem Transportobjekt  $tran_k^{out}$  befinden,

dann:

- $val_l^{att,out} = normal$ , wobei  $val_l^{att,out}$  die Wertebelegung des Attributes  $att_l^{out}$  ist.

Wenn eine Abweichung bei den Nachbedingungen existiert, dann liegt bei der Modellierung des Komponentenverhaltens mindestens ein Fehler vor.

Ähnlich wie der Komponententest können die Testfälle bei dem Integrationstest in Strukturtestfälle und Verhaltenstestfälle unterschieden werden. Die Strukturtestfälle in dem Integrationstest beinhalten folgende Testarten:

- Systemschnittstellentest: Analog zum Komponententest beinhaltet diese Testart die gleichen Überdeckungskriterien wie bei dem Schnittstellenüberdeckungstest
- Verbindungentest:
  - Überprüfung der Anzahl und der Richtung der Verbindungen zwischen den Komponenten
  - Überprüfung der Kompatibilität der Schnittstellen, die durch eine Verbindung gebunden sind

- Komponentenvollständigkeitstest: Überprüfung der Anzahl der systemeigenen, virtuellen und externen Komponenten, die das Modell beinhaltet.

In dem Integrationstest existieren zwei Möglichkeiten, um den Verhaltenstest durchzuführen:

- Erste Variante: Der Tester kann den Eingangs- und Ausgangsattributen sowie den internen Variablen die Werte, die im Testfall vordefiniert sind, zuweisen. Nach den Wertezuweisungen wird eine Simulation mit diesen Werten durchgeführt. Als Ergebnis erhält er für einzelne Komponenten eine reduzierte Verhaltenstabelle, die er mit dem erwarteten Verhalten vergleichen kann.
- Zweite Variante: Im Unterschied zur ersten Variante bestehen die erwarteten Ergebnisse nicht aus reduzierten Verhaltenstabellen, sondern aus einer Menge der verdächtigen Komponenten und einer Menge der nicht verdächtigen Komponenten. Ein Testfall in dieser Variante besteht aus drei Teilen:
  - Vorbedingungen sind aus vordefinierten Beobachtungen, zugewiesenen Sensorwerten oder Fehlercodes zusammengesetzt
  - Invarianten können aus den im Systembetriebsmodus zugewiesenen Werten der internen Variablen der Systemkomponenten bestehen
  - Nachbedingungen definieren die erwarteten Ergebnisse in Form von Mengen der verdächtigen Komponenten und nicht verdächtigen Komponenten.

Die Formalisierung der zweiten Variante des Integrationstests kann wie folgt beschrieben werden:

1. Vorbedingungen (*Precondition<sub>j</sub>*):

- $MOD_i$  - das erstellte Modell, das sich im Integrationstest befindet,
- $OBS_j^{testcase} \subseteq OBS_i$ , wobei  $OBS_i$  die Menge der Beobachtungen in dem Modell  $MOD_i$  ist,
- $DTC_j^{testcase} \subseteq DTC_i$ , wobei  $DTC_i$  die Menge der Fehlercodes in dem Modell  $MOD_i$  ist,
- $(SEN_j^{testcase} \xrightarrow{qual} ISTWERT_j^{testcase, sen}) | SEN_j^{testcase} \subseteq SEN_i$ , wobei  $SEN_i$  die Menge der Systemsensoren ist, die sich in dem Modell  $MOD_i$  befinden;  $ISTWERT_j^{testcase, sen}$  ist die Menge der Systemsensornistwerte, die für den Integrationstest zur Verfügung stehen,
- $(MEAS_j^{testcase} \xrightarrow{qual} ISTWERT_j^{testcase, meas}) | MEAS_j^{testcase} \subseteq MEAS_i$ , wobei  $MEAS_i$  die Menge der externen Messungen ist, die sich in dem Modell  $MOD_i$  befinden.  $ISTWERT_j^{testcase, meas}$  ist die Menge der Messungenistwerte, die für den Integrationstest zur Verfügung stehen,
- $(ActTEST_j^{testcase} \xrightarrow{qual} VAL_j^{testcase, acttest}) | ActTEST_j^{testcase} \subseteq ActTEST_i$ , wobei  $ActTEST_i$  die Menge der Stellgliedtests ist, die sich in dem Modell  $MOD_i$  befinden.  $VAL_j^{testcase, acttest}$  ist die Menge der qualitativen Wertebelegungen, die für den Integrationstest zur Verfügung stehen,
- $(ComTEST_j^{testcase} \xrightarrow{qual} VAL_j^{testcase, comtest}) | ComTEST_j^{testcase} \subseteq ComTEST_i$ , wobei  $ComTEST_i$  die Menge der Komponententests ist, die sich in dem Modell  $MOD_i$  befinden.  $VAL_j^{testcase, comtest}$  ist die Menge der qualitativen Wertebelegungen, die für den Integrationstest zur Verfügung stehen.

2. Invarianten (*Invariant<sub>j</sub>*):

- $(VAR_j^{testcase,sta} \xrightarrow{qual} VAL_j^{testcase,sta}) | VAR_j^{testcase,sta} \subseteq VAR_i^{sta}$ , wobei  $VAR_i^{sta}$  die Menge der internen Zustandvariablen der Komponenten ist, die in dem Modell  $MOD_i$  abgebildet sind.  $VAL_j^{testcase,sta}$  ist die Menge der qualitativen Wertebelegungen, die für den Integrationstest immer gelten müssen.

3. Nachbedingungen (*Postcondition<sub>j</sub>*):

- $COM_j^{-OK,post} | COM_j^{-OK,post} \subseteq COM_i$ , wobei  $COM_i$  die Menge der Komponenten ist, die in dem Modell  $MOD_i$  abgebildet sind.  $COM_j^{-OK,post}$  ist die Menge der verdächtigen Komponenten,
- $COM_j^{OK,post} | COM_j^{OK,post} \subseteq COM_i$ , wobei  $COM_i$  die Menge der Komponenten ist, die in dem Modell  $MOD_i$  abgebildet sind.  $COM_j^{OK,post}$  ist die Menge der nicht verdächtigen Komponenten.

Nach der Spezifikation der Testfälle kann eine automatische Testdurchführung und Testauswertung mittels der Testplattform gestartet werden. Die Testauswertung basiert auf dem Vergleich zwischen den erwarteten Komponentemengen, die in den Nachbedingungen stehen, und den berechneten Komponentemengen, die durch den Aufruf des Diagnosealgorithmus mit den Vorbedingungen, Invarianten und dem unter Test stehenden Modell als Eingaben entstanden sind. Diese Testauswertung ist analog zu dem Testorakel in der Softwareentwicklung ([121], [82], [129]). Es gelten:

- $(COM_j^{OK,soll} \subseteq COM_j^{OK,post}) \wedge (COM_j^{OK,post} \subseteq COM_j^{OK,soll})$
- $(COM_j^{-OK,soll} \subseteq COM_j^{-OK,post}) \wedge (COM_j^{-OK,post} \subseteq COM_j^{-OK,soll})$
- $COM_j^{OK,soll} \cap COM_j^{-OK,soll} = \emptyset$

Analog dazu ist die Menge der verdächtigen Komponenten, die aus der Analyse stammt ( $COM_j^{-OK,ist}$ ), und die Menge der nicht verdächtigen Komponenten ( $COM_j^{OK,ist}$ ) disjunkt:

- $COM_j^{OK,ist} \cap COM_j^{-OK,ist} = \emptyset$

Das Ergebniss der Auswertung des Integrationstests ist abhängig von zwei Schnittmengen. Eine dieser Mengen ist die Schnittmenge zwischen Soll- und Ist-Menge der nicht verdächtigen Komponenten:

- $COM_j^{OK} = COM_j^{OK,soll} \cap COM_j^{OK,ist}$

Die zweite Menge, die für die Auswertung zur Verfügung steht, ist die Schnittmenge zwischen Soll- und Ist-Menge der verdächtigen Komponenten:

- $COM_j^{-OK} = COM_j^{-OK,soll} \cap COM_j^{-OK,ist}$

Analog zur Testnotationssprache TTCN-3 ([57]) besteht das Testurteil in dieser Arbeit aus verschiedenen Stufen: *Bestanden*, *nicht entscheidbar positiv*, *nicht entscheidbar*, *nicht entscheidbar negativ* und *fehlgeschlagen*.

Mengenrelation	Testurteil	Bewertung $B(COM_j^{OK})$
$COM_j^{OK,soll} = COM_j^{OK,ist}$	<i>bestanden</i>	++
$COM_j^{OK,ist} \subset COM_j^{OK,soll}$	<i>nicht entscheidbar positiv</i>	+

$COM_j^{OK,soll} \subset COM_j^{OK,ist}$	<i>nicht entscheidbar</i>	0
$(COM_j^{OK,ist} / COM_j^{OK} \neq \emptyset) \wedge$ $(COM_j^{OK,soll} / COM_j^{OK} \neq \emptyset) \wedge$ $(COM_j^{OK} \neq \emptyset)$	<i>nicht entscheidbar negativ</i>	-
$COM_j^{OK,soll} \cap COM_j^{OK,ist} = \emptyset$	<i>fehlgeschlagen</i>	--

Tabelle 3.8: Testurteil für die nicht verdächtige Menge  $COM_j^{OK} = COM_j^{OK,soll} \cap COM_j^{OK,ist}$ 

Die Unterteilung in verschiedene Stufen ermöglicht die Priorisierung der Bugfixe des Modells. Im Folgenden werden die Testurteile in Abhängigkeit zur Kardinalität der Menge  $COM_j^{OK}$  und deren Relationen zu den Mengen  $COM_j^{OK,soll}$ ,  $COM_j^{OK,ist}$  in Form von Tabelle 3.8 dargestellt.

Analog zur Auswertung der entstehenden Menge  $COM_j^{OK}$  wird die Bewertungstabelle 3.9 für  $COM_j^{-OK}$  aufgebaut. Der Unterschied zwischen den beiden Auswertungstabellen besteht in den Zeilen 3 und 4, hier werden Zeilen wegen der Negation ausgetauscht.

Mengenrelation	Testurteil	Bewertung $B(COM_j^{-OK})$
$COM_j^{-OK,soll} = COM_j^{-OK,ist}$	<i>bestanden</i>	++
$COM_j^{-OK,soll} \subset COM_j^{-OK,ist}$	<i>nicht entscheidbar positiv</i>	+
$COM_j^{-OK,ist} \subset COM_j^{-OK,soll}$	<i>nicht entscheidbar</i>	0
$(COM_j^{-OK,ist} / COM_j^{-OK} \neq \emptyset) \wedge$ $(COM_j^{-OK,soll} / COM_j^{-OK} \neq \emptyset) \wedge$ $(COM_j^{-OK} \neq \emptyset)$	<i>nicht entscheidbar negativ</i>	-
$COM_j^{-OK,soll} \cap COM_j^{-OK,ist} = \emptyset$	<i>fehlgeschlagen</i>	--

Tabelle 3.9: Testurteil für die verdächtige Menge  $COM_j^{-OK} = COM_j^{-OK,soll} \cap COM_j^{-OK,ist}$ 

Da die Menge  $COM_j^{OK}$  und die Menge  $COM_j^{-OK}$  disjunkt sind, erfolgt das gesamte Testurteil  $B_j^{MOD_i}$  durch die Addition der Bewertungszahl des einzelnen Testurteils:

$$- B_j^{MOD_i} = B(COM_j^{OK}) + B(COM_j^{-OK})$$

Die letzte Testphase des Modells in dieser Arbeit ist der Systemtest. Ziel dieses Tests ist die Überprüfung des Diagnosemodells gegen seine Anforderungen. Ausgehend von Diagnosefunktionsanforderungen werden Testfälle abgeleitet. Da die Diagnose in der Werkstatt symptomorientiert ist, sollen die Testfälle auch symptomorientiert spezifiziert werden. Es existieren drei Vorgehensweisen, wie der Systemtest durchgeführt werden kann:

- Testfall im Systemtest besteht aus einer Testsequenz. Ein Test in dieser Sequenz ist analog zur zweiten Variante des Gray-Box-Tests aufgebaut (Test mit Vorbedingungen, Invarianten und Nachbedingungen). Bei der Spezifizierung des einzelnen Tests in der Sequenz müssen folgende Bedingungen beachtet werden. Es seien mit:
  - $Test_k$  - der k.te Test in der Sequenz,
  - $Test_{k+1}$  - der (k+1).te Test in der Sequenz, dieser Test muss nach dem k.ten Test durchgeführt werden.

Dann gilt:

- $Precondition_k \subseteq Precondition_{k+1}$
- $Invariant_k \subseteq Invariant_{k+1}$
- $COM_{k+1}^{OK} \supseteq COM_k^{OK}$  - wobei  $(COM_{k+1}^{OK} \subseteq Postcondition_{k+1}) \wedge (COM_k^{OK} \subseteq Postcondition_k)$
- $COM_{k+1}^{-OK} \subseteq COM_k^{-OK}$  - wobei  $(COM_{k+1}^{-OK} \subseteq Postcondition_{k+1}) \wedge (COM_k^{-OK} \subseteq Postcondition_k)$ .

Das Testurteil eines Testfalls dieser Art ergibt sich aus der gesamten Auswertung der einzelnen Testfälle.

- Die zweite Möglichkeit wird in dieser Arbeit Rekursionstest genannt. Es wird nur ein Test definiert, in welchem neben Vor- und Nachbedingungen, die analog zum Gray-Box-Test aufgebaut sind, die Invarianten des Tests erweitert werden, indem in den Invarianten die Abbruchbedingung des Rekursionsaufrufs spezifiziert wird. In dieser Arbeit ist die Abbruchbedingung die Anzahl des rekursiven Aufrufs. Die Testausführung wird wie folgt beschrieben. Es seien mit:

- $Test_k$  - der k.te Testaufruf; mit  $n$  als Anzahl des rekursiven Aufrufs gilt:  $0 < k \leq n$
- $Test_{k+1}$  - der (k+1).te Testaufruf ( $(k+1) \leq n$ ).

Dann gilt:

- $(Precondition_{k+1} = Precondition_k \cup Postcondition_k) | (k = 1)$ .

Die Reduzierung der verdächtigen Komponenten kann nicht nur durch die Vereinigung der Vor- und Nachbedingungen des k.ten Testaufrufs erfolgen. Weitere Diagnoseinformationen müssen hinzugefügt werden. Da die Durchführung des nächsten Tests automatisiert laufen soll, wird die benötigte neue Diagnoseinformation mittels hypothetischer Wertebelegung der Modellgrößen angenommen. Hierbei wird der Diagnosealgorithmus  $K$ , der in Kapitel 4 Abschnitt 3 beschrieben ist, gestartet. Die Ausgabe dieses Algorithmus ist eine priorisierte Liste der Messungen und einfachen Tests, die im nächsten Schritt im Diagnosevorgang vom Werkstattpersonal durchgeführt werden soll. Für die hypothetische Wertebelegung wird das erste Element ( $Element_k$ ) der Liste ausgewählt und es werden alle möglichen Wertebelegungen ( $Val = val_1, \dots, val_m$ ) angenommen. Die Weiterführung des Tests erfolgt mit einzelnen Werten der möglichen Wertebelegungen. Dabei existiert eine Verzweigung ab dem zweiten Rekursivaufruf.

- $(Precondition_{k+1} = Precondition_k(Element_{k-1} = val_j) \cup Postcondition_k) \cup (Element_k = val_i) | (1 < k) \wedge (1 \leq i \leq m)$ .

Wegen der hypothetischen Wertebelegung bestehen die Ergebnisse des Testdurchlaufs ( $TestResult$ ) aus einer Menge des Ergebniselements ( $result_i | result_i \in TestResult$ ) in Abhängigkeit zur Hypothese. Das Testurteil entsteht durch den Vergleich zwischen  $result_i$  und der vordefinierten Nachbedingung  $Postcondition$ .

- Eine dritte Möglichkeit ist die Überprüfung direkt am Versuchsträger. Dazu werden zuerst die Fehler am Fahrzeug eingebaut. Unter Werkstattbedingungen werden anhand des erstellten Modells und mittels des Diagnosetools die zuvor eingebauten Fehler gesucht. Für die Auswertung des Testes steht das Diagnoseprotokoll, das alle Aktivitäten des

Tester protokolliert, zur Verfügung. Diese Vorgehensweise kann auch als Abnahmetest betrachtet werden.



# 4 Inferenzmechanismen zur Diagnoseerstellung

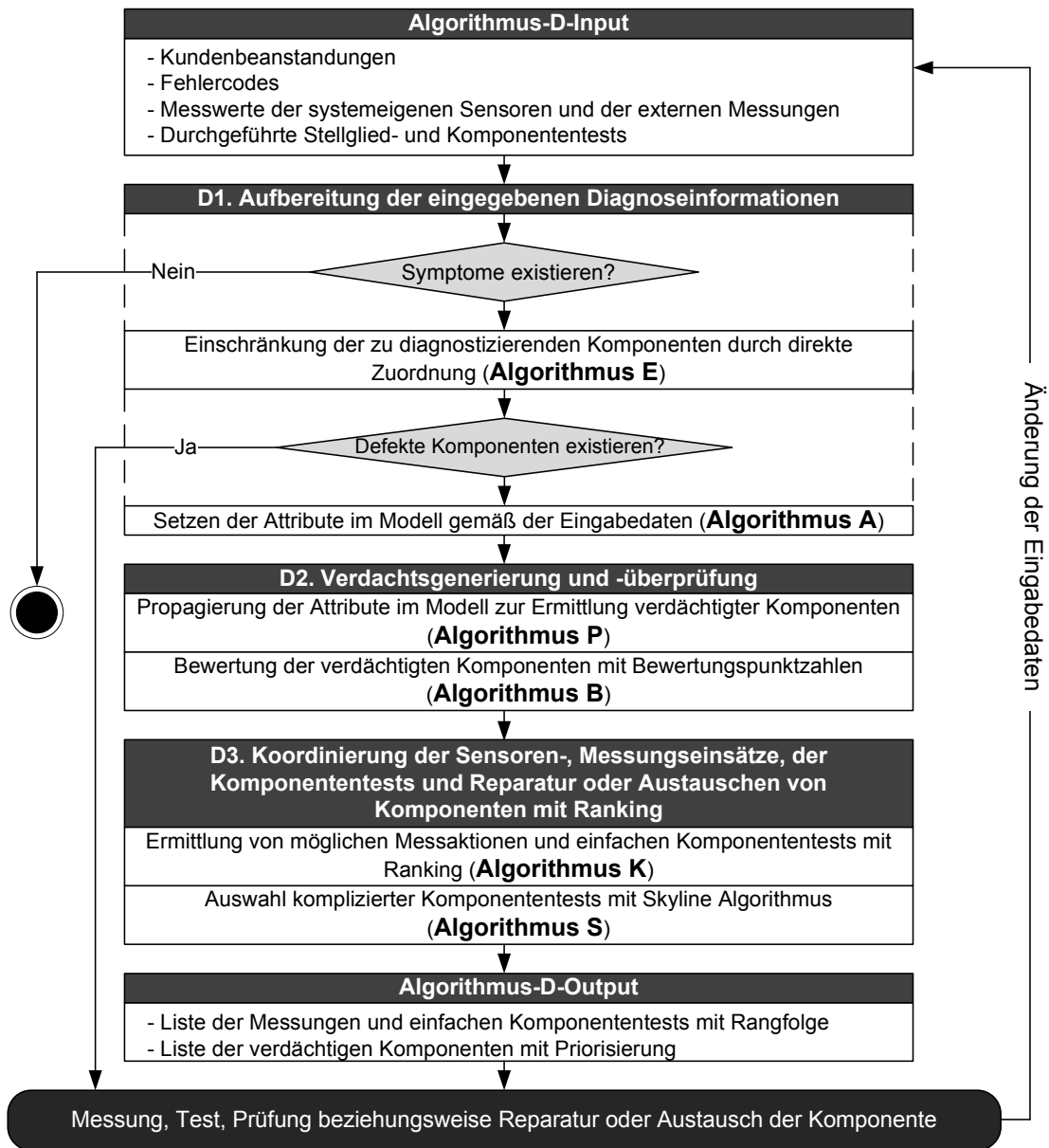


Abbildung 4.1: Diagnostik-Algorithmus *D*

Neben dem angewendeten Diagnoseansatz und dem System- und Erfahrungswissen des Werkstattmitarbeiters hat auch der Diagnoseprozess große Auswirkung auf die Fehlerlokalisierung und Fehlerbehebung im Bereich der Werkstattdiagnose. Da der Diagnoseprozess direkten Einfluss auf die Entwicklung der Inferenzkomponente in einem Diagnoselaufzeitsystem hat, wird

in diesem Kapitel der zu entwickelnde Diagnosealgorithmus unter Berücksichtigung des Diagnoseprozesses in der freien Werkstatt anhand der wesentlichen Schritte erläutert. Da der Diagnosealgorithmus  $D$  im Kapitel 1 bereits beschrieben ist (Algorithmus 1.1), werden hier die Schritte dieses Algorithmus in Form eines Ablaufdiagramms dargestellt (Abbildung 4.1). Dabei besteht der Input des Diagnosealgorithmus aus folgenden Mengen:

- Menge der Kundenbeanstandungen  $OBS^{input}$ , die sich bei der Fahrzeugannahme beziehungsweise bei Probefahrt durch das Werkstattpersonal ergaben ( $OBS^{input} \subset OBS$ ).
- Menge der vorhandenen Fehlercodes ( $DTC^{input} \subset DTC$ ).
- Menge der Modellgrößen mit Wertebelegung, die sich durch systemeigene Sensoren bestimmen lassen ( $SEN^{input} \subset SEN$ ).
- Menge der Modellgrößen mit Wertebelegung, die sich durch externe Messungen bestimmen lassen ( $MEAS^{input} \subset MEAS$ ).
- Menge aller Stellglied- und Komponententests, die durchgeführt wurden ( $ActTEST^{input} \subset ActTEST \wedge ComTEST^{input} \subset ComTEST$ ).

Als Ergebnisse liefert der Algorithmus zwei Listen. Die erste Liste beinhaltet Systemsensoren, externe Messungen und einfache Tests beziehungsweise Prüfungen mit Rangfolge (zum Beispiel: Stellgliedtests oder Sichtprüfung). Diese Liste ist die Handlungsempfehlung an das Werkstattpersonal und soll abgearbeitet werden, weil das Laufzeitsystem anhand der vorhandenen Diagnoseinformationen den Benutzer bei der Fehlersuche führen soll. Die Führung muss aber weiterhin die Möglichkeit geben, dass der Nutzer frei entscheiden kann. Die zweite Liste enthält die verdächtigen Komponenten und gibt dem Werkstattpersonal einen Überblick über verschiedene Optionen, die für ihn interessant sind, zum Beispiel: Ausfallwahrscheinlichkeit, Prüfzeit, Prüfkosten, Verfügbarkeit der Ersatzteile.

Da der Diagnoseprozess in der Werkstatt ein symptomorientierter Diagnoseprozess ist, sind die Abbruchbedingungen dieses Algorithmus die nicht vorhandenen Symptome. Demzufolge besteht die Abbruchbedingung aus folgenden Bedingungen:

$$(OBS^{input} = \emptyset) \wedge (DTC^{input} = \emptyset) \wedge (param_i = (att_i, OK) | param_i \in (SEN^{input} \cup MEAS^{input})) \wedge (param_j = (var_j, OK) | param_j \in (ActTEST^{input} \cup ComTEST^{input}))$$

Der Algorithmus wird immer neu aufgerufen, wenn die Ergebnisse der gerade durchgeführten Messungen beziehungsweise Prüfungen oder Tests vorliegen. Mit diesen neuen Diagnoseinformationen kann der Algorithmus die Liste der verdächtigen Komponenten reduzieren und die Komponenten neu bewerten. Außerdem wird mit diesen zusätzlichen Informationen die Rangfolge der Messungen und einfachen Tests beziehungsweise Prüfungen neu berechnet.

## 4.1 Aufbereitung der Eingabedaten (Algorithmen E und A)

Zu Beginn werden die verfügbaren Diagnoseinformationen in Form von Kundenbeanstandungen, Fehlercodes, Messungen und Komponentenzuständen (*Algorithmus-D-Input*) analysiert. Dieser Schritt wird in diesem Abschnitt mit den beiden Algorithmen  $E$  und  $A$  beschrieben. Es ist erforderlich, dass der Algorithmus  $E$  vor dem Algorithmus  $A$  geschaltet ist, da mechatronische Systeme häufig über eine Onboard-Diagnose-Funktion verfügen, welche die elektrischen Fehler der Sensoren und Aktoren sofort identifiziert und in Fehlerspeichern als

Fehlercodes ablegt. Diese Diagnoseinformationen verkürzen den Diagnosevorgang, weil damit defekte Komponenten direkt aufgezeigt werden und zuerst bearbeitet werden sollen. Mit den Ergebnissen des Algorithmus E werden die Einschränkungen der zu diagnostizierenden Komponenten berechnet. Dagnach ermittelt der Algorithmus A die möglichen modellgrößenbezogenen Abbildungen der Kundenbeanstandungen, Fehlercodes, Messungen und Tests beziehungsweise Prüfungen.

Die Eingabe, die für Algorithmus D zur Verfügung steht, ist auch die Eingabe des Algorithmus E. Dabei wird die Diagnoseinformation zur Einschränkung der verdächtigen Komponenten ausschließlich aus existierenden Fehlercodes ( $0 < |DTC^{input}|$ ) ermittelt.

---

**Algorithmus 4.1** : Diagnostik-Algorithmus (E)
 

---

**Eingabe:** (*Input Algorithmus E*)

- Menge der existierenden Fehlercodes, die aus dem Input des Algorithmus D stammen

**Ausgabe:** (*Output Algorithmus E*)

- Menge der systemeigenen Komponenten  $COM^{-OK}$ , die durch Fehlercodes verdächtig sind.

**Algorithmus:**

**Wissen für Algorithmus E:** Wenn Fehlercodes, die durch die Eigendiagnosefähigkeit des Steuergeräts bei Fehlverhalten des Systems auf bestimmte Komponenten zeigen, existieren, dann werden die entsprechenden Fehlercodes im Diagnosemodell und die zugehörigen Komponenten in einer Tabelle erfasst (Tabelle 4.1).

$DTC^{input}$	Komponentenbezogene Diagnoseparameter
$\{DTC_i\}$	$\{com_3, \dots, com_{12}\}$
$\{DTC_j\}$	$\{com_8\}$
	...
$\{DTC_k, \dots, DTC_m\}$	$\{com_9, \dots, com_{13}\}$

Tabelle 4.1: Darstellung der Menge  $COM^{DTC,-OK}$  in Tabellenform

**E** Die Zuordnung der Fehlercodes zu defekten Komponenten gemäß Wissen erfolgt mittels eines einfache Suchalgorithmus.

---

**Algorithmus 4.2** : Diagnostik-Algorithmus (A)
 

---

**Eingabe:** (*Input Algorithmus A*)

- Diagnoseinformationen, die aus dem Input des Algorithmus D stammen
  - Menge der Kundenbeanstandungen
  - Menge der Fehlercodes, die auf Modellgrößen zeigen
  - Menge der Modellgrößen mit Wertebelegung, die sich durch Messungen mittels systemeigene Sensoren und externe Messgeräte entstanden sind

- Menge der entstehenden Diagnoseinformationen aus den Komponententests und Stellgliedtests

**Ausgabe:** (*Output Algorithmus A*)

- Menge der attributbezogenen Diagnoseparameter  $PARAM$ . Dabei ist ein Element  $param_i \in PARAM$  ein Tupel  $((ATT_i^{input}, VAL_i^{input}), (VAR_i^{func}, VAL_i^{func}), (VAR_i^{sta}, VAL_i^{sta}), (VAR_i^{faul}, VAL_i^{faul}), (ATT_i^{out}, VAL_i^{out}))$

**Algorithmus:**

### Wissen für Algorithmus A

Die Abbildungen der Kundenbeanstandungen werden im Modell gesucht. Die Attribute und deren Wertebelegungen werden gefiltert und bilden eine Menge der modellgrößenbezogenen Diagnoseparameter ( $PARAM^{OBS}$ ), die aus Kundenbeanstandungen stammen. Diese Menge der Diagnoseparameter kann in Tabellenform dargestellt werden (Tabelle 4.2).

Diagnoseparameter	$OBS^{input}$					
	$\{obs_1\}$	$\{obs_1, obs_2\}$	$\{obs_2\}$	...	$\{obs_2, obs_3, \dots, obs_6\}$	$\{obs_6\}$
$com_1.port_2.tran_1.att_4$	$\{H\}$	$\{L\}$	–	...	$\{H\}$	–
$com_4.port_3.tran_3.att_2$	–	–	–	...	$\{H\}$	$\{L\}$
			...			
$com_8.port_3.tran_4.att_1$	–	$\{L\}$	$\{H\}$	...	–	–

Tabelle 4.2: Darstellung der  $PARAM^{OBS}$  in Tabellenform<sup>1</sup>

Eine Spalte der Tabelle 4.2 entspricht einem Parametersatz  $param_i^{OBS}$ , der propagiert werden muss, da die Modellgrößen und deren zugehörige Wertebelegungen eines Parametersatzes, die sich in  $param_i^{OBS}$  befinden, eine beziehungsweise mehrere Kundenbeanstandungen erklären. Es gilt:  $param_i^{OBS} \in PARAM^{OBS}$

Die Fehlercodes, die in Steuergeräten gespeichert sind, werden analysiert. Analog zur Kundenbeanstandung soll für Fehlercodes, die auf Modellgrößen zeigen, ein Tupel der Diagnoseparameter ermittelt werden (Tabelle 4.3).

Diagnoseparameter	$DTC^{input}$				
	$dtc_2$	$dtc_4$	...	$dtc_7$	$dtc_8$
$com_3.port_1.tran_2.att_1$	$\{L, H\}$	–	...	–	–
$com_3.port_3.tran_1.att_2$	–	$\{H\}$	...	$\{H\}$	–
			...		
$com_9.port_1.tran_2.att_1$	–	–	...	–	$\{L, H\}$

Tabelle 4.3: Darstellung der  $PARAM^{DTC}$  in Tabellenform

Die Analyse der Diagnoseinformationen von ausgelesenen systemeigenen Sensoren und

<sup>1</sup> $com_i.port_j.tran_k.att_l := i$ -te Komponente.  $j$ -te Schnittstelle.  $k$ -tes Transportobjekt.  $l$ -tes Attribut  
 $\{H\} :=$  hoch;  $\{L\} :=$  niedrig;  $\{N\} :=$  normal

den durchgeführten externen Messungen ist gleich. Die Tabelle 4.4 stellt die Menge der Diagnoseparameter ( $PARAM^{SEN}$ ) dar, die aus der Analyse von Sensoren entstanden sind. Die Tabelle 4.5 stellt die Menge der Messungen mit den zugehörigen Messwerten ( $PARAM^{MEAS}$ ) dar.

Diagnoseparameter	$SEN^{input}$			
	$sensor_1$	$sensor_3$	...	$sensor_5$
$com_1.port_2.tran_3.att_1$	$\{H\}$	–	...	–
...				
$com_6.port_2.tran_1.att_3$	–	–	...	$\{L\}$

Tabelle 4.4: Darstellung der  $PARAM^{SEN}$  in Tabellenform

Diagnoseparameter	$MEAS^{input}$			
	$messung_1$	...	$messung_2$	$messung_6$
$com_2.port_2.tran_3.att_1$	$\{L\}$	...	–	–
$com_3.port_2.tran_4.att_1$	–	...	–	$\{H\}$
...				
$com_5.port_5.tran_3.att_1$	–	...	$\{L\}$	–

Tabelle 4.5: Darstellung der  $PARAM^{MEAS}$  in Tabellenform

Aus den Komponenten- beziehungsweise den Stellgliedtests entstehen zwei Komponentenmengen. Dabei ergibt sich eine Menge der systemeigenen Komponenten, die durch einen Test als in Ordnung identifiziert wurden. Analog resultiert eine Menge der Komponenten, die als fehlerhaft befunden wurden. Die Aussagen über die Zustände der Komponenten können aus den Testergebnissen gezogen werden. Diese Informationen müssen bei der Diagnose auch berücksichtigt werden. Die Menge der entstehenden Informationen aus den Komponententests wird als  $PARAM^{ComTEST}$  bezeichnet und die Menge der Informationen aus den Stellgliedtests als  $PARAM^{ActTEST}$  (Tabelle 4.6, Tabelle 4.7).

Diagnoseparameter	$ComTEST^{input}$			
	$comtest_2$	$comtest_3$	...	$comtest_5$
$com_3.var_1^{func}$	$\{L\}$	–	...	–
$com_3.var_3^{faul}$	–	$\{N\}$	...	–
...				
$com_7.var_1^{faul}$	–	–	...	$\{H\}$

Tabelle 4.6: Darstellung der  $PARAM^{ComTEST}$  in Tabellenform

Aus den einzelnen Analyseergebnissen von Kundenbeanstandungen, Fehlercodes aus Steuergeräten, Messwerten der systemeigenen sowie externen Messungen, Komponenten- und Stellgliedtests werden in diesem Schritt die modellgrößenbezogenen Diagnoseparameter

zusammengeführt. Es gilt:

$$\begin{aligned}
& (|PARAM| = |PARAM^{OBS}) \wedge (\forall param_i \in PARAM \mid \\
& param_i = param_i^{OBS} \cup PARAM^{DTC} \cup PARAM^{SEN} \cup PARAM^{MEAS} \cup \\
& PARAM^{ComTEST} \cup PARAM^{ActTEST}) \\
& \Leftrightarrow param_i = ((ATT_i^{input}, VAR_i^{func}, VAR_i^{sta}, VAR_i^{faul}, ATT_i^{out}) \rightarrow \\
& (VAL_i^{input}, VAL_i^{func}, VAL_i^{sta}, VAL_i^{faul}, VAL_i^{out}))
\end{aligned}$$

**A** Die Zuordnung der Modellgröße und deren Wertenbelegungen gemäß Wissen erfolgt mittels eines einfache Suchalgorithmus.

Diagnoseparameter	<i>ActTEST</i> <sup>input</sup>			
	<i>acttest</i> <sub>1</sub>	<i>acttest</i> <sub>2</sub>	...	<i>acttest</i> <sub>6</sub>
<i>com</i> <sub>1</sub> . <i>var</i> <sub>1</sub> <sup>func</sup>	{N}	–	...	{N}
<i>com</i> <sub>2</sub> . <i>var</i> <sub>1</sub> <sup>func</sup>	–	{L}	...	–
	...			
<i>com</i> <sub>3</sub> . <i>var</i> <sub>2</sub> <sup>faul</sup>	–	–	...	{N}

Tabelle 4.7: Darstellung der  $PARAM^{ActTEST}$  in Tabellenform

## 4.2 Verdachtsgenerierung

Aus den Diagnoseinformationen, die der Algorithmus A liefert, wird der Algorithmus P zur Verdachtsgenerierung angestoßen. Nach der Verdachtsgenerierung werden die verdächtigen Komponenten mittels Algorithmus B mit Punktzahlen bewertet. In diesem Abschnitt werden die Algorithmen zur Verdachtsgenerierung und -überprüfung beschrieben.

### 4.2.1 Ermittlung der verdächtigen Komponenten (Algorithmus P)

Die Ermittlung der verdächtigen Komponenten basiert auf der Modellanalyse der Verhaltenstabellen der Komponentenmodelle und deren Verbindungen unter Berücksichtigung der Diagnoseparameter, die aus den Analyseergebnissen von Kundenbeanstandungen, Fehlercodes aus Steuergeräten, Messwerten der systemeigenen Sensoren sowie externen Messungen und Komponenten-/ Stellgliedtests entstehen. Nachfolgend wird die Propagierung der belegten Attribute im Modell zur Ermittlung verdächtiger Komponenten beschrieben.

---

**Algorithmus 4.3** : Diagnostik-Algorithmus (P)

---

**Eingabe:** (*Input Algorithmus P*)

- Modell mit teilweise belegten Parameter, wobei die Menge der belegten Diagnoseparameter  $PARAM$  die Ausgabe des Algorithmus A ist. Das Diagnosemodell  $MOD$  stammt aus der Identifizierung des zu diagnostizierenden Systems beziehungsweise der zu diagnostizierenden Funktion.

**Ausgabe:** (Output Algorithmus P)

- Menge der verdächtigen Komponenten ( $COM^{sus}$ ) mit deren reduzierten Verhaltenstabellen

**Algorithmus:**

**P1.** Für jedes Attribut beziehungsweise jede interne Variable, deren Werte belegt sind, werden die Verhaltenstabellen aller Komponenten, die mit dem Attribut oder der Variable direkt oder indirekt zu tun haben, reduziert. Die Reduzierung der Verhaltenstabelle erfolgt durch Löschung aller Zeilen der Tabelle, die den Wertebelegungen nicht entsprechen. Für eine bestimmte Komponente  $com_x$  des Diagnosemodells  $MOD$  reduziert diese Methode die vollständige Verhaltenstabelle  $tab_x$  zu  $tab_x^{param_{i,x}}$ , wenn  $|param_{i,x}| > 0$  ist. Die Reduzierung erfolgt durch die sequentielle Suche der Menge aller Zeilen  $ROW_x$  der Verhaltenstabelle  $tab_x$  unter der Vergleichsbedingung, dass einzelne Zeilen mit den Wertebelegungen in  $param_{i,x}$  übereinstimmen. Analog wird die Beeinflussungsgradtabelle reduziert; dabei wird nur die Übereinstimmung der Ausgangsattribute berücksichtigt.

$com_x$														
$port_0^{in}$ ... $port_i^{in}$										$port_0^{out}$				
$tran_a$ ... $tran_b$					$var_i^{func}$	...	$var_j^{sta}$	...	$var_k^{faul}$	...	$tran_n$			
$att_0$	...	$att_m$	...	$att_0$							$att_0$	...	$att_n$	
H	...	L	...	N	N	...	L	...	H	...	L	...	L	
H	...	N	...	H	N	...	L	...	L	...	L	...	H	
H	...	N	...	N	N	...	N	...	N	...	H	...	H	
N	...	N	...	N	N	...	N	...	N	...	N	...	N	
L	...	N	...	H	N	...	H	...	N	...	L	...	L	
N	...	N	...	N	N	...	L	...	N	...	L	...	L	
...														
H	...	L	...	N	N	...	N	...	N	...	H	...	H	

Tabelle 4.8: Beispiel eines Abschnitts der vollständigen Verhaltenstabelle  $tab_x$  einer Komponente  $com_x$

Als Beispiel besteht  $param_{i,x} = ((ATT_{i,x}^{input}, VAR_{i,x}^{func}, VAR_{i,x}^{sta}, VAR_{i,x}^{faul}, ATT_{i,x}^{out}) \rightarrow (VAL_{i,x}^{input}, VAL_{i,x}^{func}, VAL_{i,x}^{sta}, VAL_{i,x}^{faul}, VAL_{i,x}^{out}))$  aus folgenden Einträgen:

- $(ATT_{i,x}^{input} = \{ com_x \cdot port_0^{in} \cdot tran_a \cdot att_0 \}) \wedge (VAL_{i,x}^{input} = \{ H \})$
- $(VAR_{i,x}^{func} = \{ com_x \cdot var_i^{func} \}) \wedge (VAL_{i,x}^{func} = \{ N \})$
- $(ATT_{i,x}^{out} = \{ com_x \cdot port_0^{out} \cdot tran_n \cdot att_n \}) \wedge (VAL_{i,x}^{out} = \{ H \})$

Demzufolge gilt:  $param_{i,x} = (\{ com_x \cdot port_0^{in} \cdot tran_a \cdot att_0 \}, \{ com_x \cdot var_i^{func} \}, \{ com_x \cdot port_0^{out} \cdot tran_n \cdot att_n \}) \rightarrow (\{ H \}, \{ N \}, \{ H \})$ . Damit wird die Verhaltenstabelle  $tab_x$  (Tabelle 4.8) unter Berücksichtigung der Parameter  $param_{i,x}$  zu  $tab_x^{param_{i,x}}$  (Tabelle 4.9) reduziert.

$com_x$													
$port_0^{in}$ ... $port_i^{in}$										$port_0^{out}$			
$tran_a$ ... $tran_b$					$var_i^{func}$	...	$var_j^{sta}$	...	$var_k^{faul}$	...	$tran_n$		
$att_0$	...	$att_m$	...	$att_0$						$att_0$	...	$att_n$	
H	...	N	...	H	N	...	L	...	L	...	L	...	H
H	...	N	...	N	N	...	N	...	N	...	H	...	H
...													
H	...	L	...	N	N	...	N	...	N	...	H	..	H

Tabelle 4.9: Beispiel eines Abschnitts der reduzierten Verhaltenstabelle  $tab_x^{param_{i,x}}$  einer Komponente  $com_x$  mit dem Diagnoseparameter  $param_{i,x}$

**P2.** Die Teilmenge der Diagnoseparameter, die sich in den Ausgangsschnittstellen der Komponenten beziehungsweise in den Ausgangsschnittstellen des Systems befinden, wird rückwärts gegen die Richtung der gerichteten Verbindungen im Modell propagiert. Dabei werden alle Zeilen der Verhaltenstabelle der einzelnen Komponenten, die mit der Wertebelugung der propagierten Diagnoseparameter nicht übereinstimmen, entfernt. Wenn das Modell eine Rückkopplung oder mehrere Rückkopplungen beinhaltet, muss der Propagierungsvorgang wiederholt werden, bis keine Änderungen an den Zeilen der Verhaltenstabelle mehr existieren.

$com_x^{pre}$													
$port_0^{in}$ ... $port_i^{in}$										$port_0^{out}$			
$tran_x$ ... $tran_y$					$var_i^{func}$	...	$var_j^{sta}$	...	$var_k^{faul}$	...	$tran_a$		
$att_l$	...	$att_m$	...	$att_n$						$att_0$	...	$att_m$	
N	...	L	...	L	N	...	N	...	H	...	N	...	L
H	...	N	...	H	N	...	L	...	N	...	H	...	N
N	...	L	...	N	N	...	H	...	N	...	H	...	L
H	...	H	...	N	N	...	N	...	L	...	N	...	L
N	...	N	...	H	N	...	H	...	N	...	L	...	L
L	...	N	...	N	N	...	L	...	N	...	N	...	L
...													
H	...	L	...	N	N	...	N	...	N	...	H	..	N

Tabelle 4.10: Beispiel eines Abschnitts der vollständigen Verhaltenstabelle  $tab_x^{pre}$  einer Komponente  $com_x^{pre}$

Ein Schrittes in dieser Rückwärtspropagierung wird zur Veranschaulichung in diesem Abschnitt anhand der Komponente  $com_x^{pre}$  beschrieben.  $com_x^{pre}$  ist eine Komponente in der Wirkkette, die direkt vor der Komponente  $com_x$  steht. Die Komponente  $com_x^{pre}$  besitzt ei-



ne Ausgangsschnittstelle  $port_0^{out}$ , mit der die Eingangsschnittstelle  $port_0^{in}$  der Komponente  $com_x$  verbunden ist. Tabelle 4.10 zeigt das Beispiel eines Abschnitts der vollständigen Verhaltenstabelle  $tab_x^{pre}$  der Komponente  $com_x^{pre}$  vor dem Rückwärtspropagierungsvorgang.

In Tabelle 4.9 existieren folgende Wertebelegungen:

$$(com_x \cdot port_0^{in} \cdot tran_a \cdot att_0, com_x \cdot port_0^{in} \cdot tran_a \cdot att_m) \rightarrow (\{H\}, \{N\})$$

$$(com_x \cdot port_0^{in} \cdot tran_a \cdot att_0, com_x \cdot port_0^{in} \cdot tran_a \cdot att_m) \rightarrow (\{H\}, \{L\})$$

Aus diesen Diagnoseinformationen gelten folgende Wertebelegungen für die Ausgangsschnittstellen  $port_0^{out}$  der Komponente  $com_x^{pre}$ :

$$param_{i,x,j}^{pre} = (com_x^{pre} \cdot port_0^{out} \cdot tran_a \cdot att_0, com_x^{pre} \cdot port_0^{out} \cdot tran_a \cdot att_m) \rightarrow (\{H\}, \{N\})$$

$$param_{i,x,j+k}^{pre} = (com_x^{pre} \cdot port_0^{out} \cdot tran_a \cdot att_0, com_x^{pre} \cdot port_0^{out} \cdot tran_a \cdot att_m) \rightarrow (\{H\}, \{L\})$$

Mit den Diagnoseparametern  $param_{i,x,j}^{pre}$  und  $param_{i,x,j+k}^{pre}$  wird die Verhaltenstabelle 4.10 zur Tabelle 4.11 reduziert.

$com_x^{pre}$														
$port_0^{in}$ ... $port_i^{in}$										$port_0^{out}$				
$tran_x$ ... $tran_y$					$var_i^{func}$	...	$var_j^{sta}$	...	$var_k^{faul}$	...	$tran_a$			
$att_l$	...	$att_m$	...	$att_n$						$att_0$	...	$att_m$		
H	...	N	...	H	N	...	L	...	N	...	H	...	N	
N	...	L	...	N	N	...	H	...	N	...	H	...	L	
...														
H	...	L	...	N	N	...	N	...	N	...	H	..	N	

Tabelle 4.11: Beispiel eines Abschnitts der reduzierten Verhaltenstabelle mit den Diagnoseparametern  $param_{i,x,j}^{pre}$  und  $param_{i,x,j+k}^{pre}$  der Komponente  $com_x^{pre}$

**P3.** Nach dem Beenden der Rückwärtspropagierung wird die Vorwärtspropagierung gestartet. Hierzu wird die Verhaltenstabelle der Komponenten entlang der Richtung der gerichteten Verbindungen im Modell mittels Diagnoseparametern, die sich in den Eingangsschnittstellen der Komponenten des Systems befinden, reduziert. Dieser Vorgang ist analog zur Rückwärtsverkettung, Unterschiede sind die Propagierungsrichtung und dass sich die Modellgröße in den Eingangsschnittstellen der Komponenten des Systems befindet.

$com_x^{post}$														
$port_0^{in}$ ... $port_i^{in}$										$port_0^{out}$				
$tran_n$ ... $tran_b$					$var_i^{func}$	...	$var_j^{sta}$	...	$var_k^{faul}$	...	$tran_l$			
$att_0$	...	$att_n$	...	$att_0$						$att_0$	...	$att_q$		
L	...	H	...	N	N	...	L	...	H	...	L	...	H	
H	...	L	...	H	N	...	L	...	L	...	L	...	H	
N	...	H	...	N	N	...	N	...	N	...	N	...	H	

N	...	N	...	N	N	...	N	...	N	...	N	...	N
H	...	H	...	H	N	...	H	...	N	...	H	...	H
N	...	H	...	N	N	...	L	...	N	...	H	...	N
...													
N	...	L	...	N	N	...	N	...	N	...	L	..	L

Tabelle 4.12: Beispiel eines Abschnitts der vollständigen Verhaltenstabelle  $tab_x^{post}$  einer Komponente  $com_x^{post}$

Es sei mit  $com_x^{post}$  eine Komponente in der Wirkkette, die direkt hinter Komponente  $com_x$  steht. Die Komponente  $com_x^{post}$  besitzt eine Eingangsschnittstelle  $port_0^{in}$ , welche mit der Ausgangsschnittstelle  $port_0^{out}$  der Komponente  $com_x$  verbunden ist. Tabelle 4.12 ist ein Beispiel eines Abschnitts der vollständigen Verhaltenstabelle  $tab_x^{post}$  der Komponente  $com_x^{post}$  vor dem Vorwärtspropagierungsvorgang.

In der Verhaltenstabelle 4.9 existieren zwei Wertebelegungen der Ausgangsattribute der Komponente  $com_x$ :

$$(com_x. port_0^{out}. tran_n. att_0, com_x. port_0^{out}. tran_n. att_n) \rightarrow (\{L\}, \{H\})$$

$$(com_x. port_0^{out}. tran_n. att_0, com_x. port_0^{out}. tran_n. att_n) \rightarrow (\{H\}, \{H\})$$

Demzufolge gilt folgende Wertebelegung für die Ausgangsschnittstellen  $port_0^{in}$  der Komponente  $com_x^{post}$ :

$$param_{i,x,j}^{post} = (com_x^{post}. port_0^{in}. tran_n. att_0, com_x^{post}. port_0^{in}. tran_n. att_n) \rightarrow (\{L\}, \{H\})$$

$$param_{i,x,j+k}^{post} = (com_x^{post}. port_0^{in}. tran_n. att_0, com_x^{post}. port_0^{in}. tran_n. att_n) \rightarrow (\{H\}, \{H\})$$

Die Projektion der Parameter  $param_{i,x,j}^{post}$  und  $param_{i,x,j+k}^{post}$  auf die Verhaltenstabelle 4.12  $tab_x^{post}$  reduziert diese Tabelle zu Tabelle 4.13.

$com_x^{post}$													
$port_0^{in}$ ... $port_i^{in}$									$port_0^{out}$				
$tran_n$ ... $tran_b$				$var_i^{func}$	...	$var_j^{sta}$	...	$var_k^{faul}$	...	$tran_l$			
$att_0$	...	$att_n$	...	$att_0$						$att_0$	...	$att_q$	
L	...	H	...	N	N	...	L	...	H	...	L	...	H
...													
H	...	H	...	H	N	...	H	...	N	...	H	...	H

Tabelle 4.13: Beispiel eines Abschnitts der reduzierten Verhaltenstabelle mit den Diagnoseparametern  $param_{i,x,j}^{post}$  und  $param_{i,x,j+k}^{post}$  der Komponente  $com_x^{post}$

**P4.** Für alle Komponentenzustände aus Schritt P2 wird geprüft, ob ein Fehlerzustand alle Attribute erklären kann. Alle diese Komponenten werden in die Menge der verdächtigen Komponenten ( $COM^{susp}$ ) aufgenommen. Falls kein Fehlerzustand gefunden wird, werden die Komponenten, die sich in der Wirkkette befinden, verdächtig und müssen in die Menge

der verdächtigen Komponenten ( $COM^{susp}$ ) aufgenommen werden. Die Wirkkette ist ein Teil des zudiagnostizierenden Systems beziehungsweise des Subsystems und alle Komponenten in diese Wirkkette wird durch die Rückwärtsverkettung markiert. Desweiteren wird in diesem Fall die Systemeingangsschnittstellen durch die Rückwärtsverkettung mit Wertenebelegung *hoch* oder *niedrig* belegt, mit diesem Wert wird die Propagierung in benachbarten Systemen, wenn Diagnosemodelle der benachbarten Systeme existiert, fortgeführt.

#### 4.2.2 Bewertung der verdächtigen Komponenten (Algorithmus B)

Der Algorithmus B dient zur Bewertung der verdächtigen Komponenten mit Bewertungspunktzahlen.

---

##### Algorithmus 4.4 : Diagnostik-Algorithmus (B)

---

**Eingabe:** (*Input Algorithmus B*)

Output Algorithmus P :

- Menge der verdächtigen Komponenten ( $COM^{susp}$ ) mit ihren reduzierten Verhaltenstabellen.

**Ausgabe:** (*Output Algorithmus B*)

- Bewertungspunktzahl für verdächtige Komponenten.

**Algorithmus:**

**B1.** Jede Zeile  $row_i^{beh}$  der reduzierten Verhaltenstabelle  $tab_k^{beh}$  der Komponente  $com_k \in COM^{susp}$ , die sich in der Verdächtigenliste befindet, wird mit einem Faktor  $f_i^{beh}$  (s.u.) in Abhängigkeit von der Anzahl der belegten internen Variablen und der belegten Eingangsattribute berechnet und gewichtet. Der Grad der Verdächtigung einer Komponente, der auf der reduzierten Verhaltenstabelle basiert, ist die gesamte Summe des Faktors  $f_i^{beh}$  der einzelnen Zeilen:

$$f_k^{beh} = \sum_{i=1}^{|ROW_k^{beh}|} f_i^{beh}$$

Daraus folgt, dass eine Komponente umso stärker gewichtet wird, je mehr Fehlerzustände sie hat, falls keine Wahrscheinlichkeiten der Fehlerzustände bekannt sind.

**B2.** Jede Zeile  $row_i^{inf}$  der reduzierten Beeinflussungstabelle  $tab_k^{inf}$  wird mit einem Faktor  $f_i^{inf}$  (s.u.) in Abhängigkeit von der Anzahl der belegten Ausgangsattribute berechnet und gewichtet. Analog zur Verhaltenstabelle wird die Summe des Faktors  $f_k^{inf}$  der einzelnen Zeilen gebildet:

$$f_k^{inf} = \sum_{i=1}^{|ROW_k^{inf}|} f_i^{inf}$$

**B3.** Der Verdächtigungsgrad der Komponente  $com_k$  ergibt sich aus der Summe von  $f_k^{beh}$  und  $f_k^{inf}$ :

$$f_k = f_k^{beh} + f_k^{inf}$$

Dieser Wert drückt aus, wie wahrscheinlich die Komponente für die abweichenden Outputvariablen verantwortlich ist.

Der Bewertungsfaktor  $f_i^{\tilde{b}eh}$  der Zeile  $row_i^{\tilde{b}eh}$  der reduzierten Verhaltenstabelle  $tab_k^{\tilde{b}eh}$  einer Komponente  $com_k \in COM^{susp}$  ist abhängig von dem Inhalt dieser Zeile. Nach 3.3.4 ist eine Zeile  $row_i^{\tilde{b}eh}$  der Verhaltenstabelle  $row_i^{\tilde{b}eh}$  ein Tupel, welches eine Abbildung ist aus:

$$\{ATT_k^{in}, VAR_k^{func}, VAR_k^{sta}, VAR_k^{faul}, ATT_k^{out}\} \\ \xrightarrow{qual} \{VAL_k^{in}, VAL_k^{func}, VAL_k^{sta}, VAL_k^{faul}, VAL_k^{out}\}.$$

Daher ist eine Zeile der reduzierten Verhaltenstabelle auch ein Tupel und kann mit folgenden Einschränkungen definiert werden:

$$\forall att_{k,i}^{out} \in ATT_k^{out} | (\exists att_{k,i}^{out} = val_{k,i}^{out}) \wedge (val_{k,i}^{out} \in VAL_{k,i}^{out}) \wedge (normal \notin VAL_{k,i}^{out})$$

Da mindestens eine Abweichung eines Ausgangsattributes in der Zeile  $row_i^{\tilde{b}eh}$  existieren muss, wird der Bewertungsfaktor  $f_i^{\tilde{b}eh}$  wie folgt berechnet:

- Zeile  $row_i^{\tilde{b}eh}$  wird mit  $f_i^{\tilde{b}eh} = f^{var}$  bewertet, wenn:

$$\forall att_{k,i}^{in} \in ATT_k^{in} | (att_{k,i}^{in} = val_{k,i}^{in}) \wedge (val_{k,i}^{in} \in VAL_{k,i}^{in}) \wedge (VAL_{k,i}^{in} \subseteq \{normal\})$$

Dieser Ausdruck sagt aus, dass nur die Abweichung der Wertebelegungen der internen Variablen für die Abweichungen der Ausgangsattribute verantwortlich ist.

- Zeile  $row_i^{\tilde{b}eh}$  wird mit  $f_i^{\tilde{b}eh} = f^{in,var}$  bewertet, wenn:

$$(\forall att_{k,i}^{in} \in ATT_k^{in} | (\exists (att_{k,i}^{in} = val_{k,i}^{in}) \wedge (val_{k,i}^{in} \in VAL_{k,i}^{in}) \wedge (normal \notin VAL_{k,i}^{in}) \wedge (1 \leq i))) \wedge \\ (\forall var_{k,j}^{prop} \in VAR_k^{prop} \wedge prop \in \{func, sta, faul\} | (\exists (var_{k,j}^{prop} = val_{k,j}^{prop}) \wedge (val_{k,j}^{prop} \in \\ VAL_{k,j}^{prop}) \wedge (normal \notin VAL_{k,j}^{prop}) \wedge (1 \leq j)))$$

Dieser Ausdruck sagt aus, dass durch die Abweichung der Wertebelegungen der Eingangsattribute und der internen Variablen die Abweichungen der Ausgangsattribute hervorgerufen werden.

- Zeile  $row_i^{\tilde{b}eh}$  wird mit  $f_i^{\tilde{b}eh} = f^{in}$  bewertet, wenn:

$$\forall var_{k,j}^{prop} \in VAR_k^{prop} \wedge prop \in \{func, sta, faul\} | (\exists (var_{k,j}^{prop} = val_{k,j}^{prop}) \wedge (val_{k,j}^{prop} \in \\ VAL_{k,j}^{prop}) \wedge (VAL_{k,j}^{prop} \subseteq \{normal\}))$$

Dieser Ausdruck sagt aus, dass nur die Abweichung der Wertebelegungen der Eingangsattribute für die Abweichungen der Ausgangsattribute verantwortlich ist.

Demnach liefert die Zeile mit dem Bewertungsfaktor  $f^{var}$  die Erklärung der vorhandenen Symptome durch Einfachfehler. Die Mehrfachfehler werden durch Zeilen mit dem Bewertungsfaktor  $f^{in,var}$  erklärt. Fehler, die ausschließlich weitergereicht werden, sind durch Zeilen mit dem Bewertungsfaktor  $f^{in}$  berücksichtigt worden. Da die bisherigen Erfahrungen bei der Werkstattdiagose gezeigt haben, dass Einfachfehler häufiger auftreten als Mehrfachfehler und Komponenten, die die Fehler ausschließlich weiterreichen, nicht verdächtigt werden sollen, gelten folgende Relationen zwischen den Bewertungsfaktoren:

$$f^{var} > f^{in,var} > f^{in} .$$

### 4.3 Koordination der Messungen und Tests

Der Teilalgorithmus *D3* (Abbildung 4.1) zur Koordinierung der Sensoren- und Messungseinsätze, der Komponententests und der Reparatur oder dem Austausch von Komponenten mit

Ranking besteht aus zwei Algorithmen. Zum einen ist dies der Algorithmus  $K$ , welcher die Messungen und einfachen Tests wie zum Beispiel den Stellgliedtest priorisiert. Zum anderen gruppiert der Algorithmus  $S$  die verdächtigen Komponenten mit verschiedenen Kriterien, die für das Werkstattpersonal bei der Diagnose entscheidend sind. Die Liste der priorisierten Messungen und einfachen Tests dient als Handlungsempfehlung und ist für den Benutzer, der die geführte Fehlersuche in Anspruch nehmen will, ausschlaggebend. Wegen der begrenzten Mess- und Testmöglichkeiten im Fahrzeug und wegen des Assistenzaspekts des Diagnosetools liefert die zweite Liste keine direkte Handlungsempfehlung. Mit dieser Liste kann der Nutzer eigenständig entscheiden, welche verdächtigen Komponenten er zunächst prüfen oder austauschen möchte. Die beiden Listen werden nach jedem Diagnoseschritt, wie zum Beispiel der Durchführung von Messungen, Tests oder Prüfungen neu berechnet.

### 4.3.1 Koordinierung der Sensoren-, Messungseinsätze und Tests (Algorithmus K)

Die Messungsmöglichkeiten am mechatronischen System, das im Fahrzeug eingebaut ist, sind begrenzt. Die Ermittlung neuer Messpunkte zur Reduzierung der Anzahl der verdächtigen Komponenten im Sinne der Einbringung neuer Sensoren beziehungsweise Messgeräte ist auf Grund des begrenzten Bauraums und der notwendigen Umbaumaßnahmen in freien Werkstätten schwer durchführbar und erhöht damit die Diagnosekosten. Aus diesem Grund werden in dieser Arbeit keine Verfahren, die neue Messpunkte ermitteln, entwickelt. Es werden ausschließlich die vorhandenen Sensoren, Messungen, Tests und Prüfungen berücksichtigt und priorisiert.

Da bei dem Diagnosevorgang nach jeder neuen Messung, dem Auslesen des Istwerts des Systemsensors oder der Durchführung von Tests neue Diagnoseinformationen zur Verfügung stehen, beinhaltet der Algorithmus  $K$  folgende Schritte:

---

#### Algorithmus 4.5 : Diagnostik-Algorithmus (K)

---

**Eingabe:** (*Input Algorithmus K*)

- Teil des Input Algorithmus D:
  - Menge der Modellgrößen mit Wertebelegung, die sich durch systemeigene Sensoren bestimmen lassen.
  - Menge der Modellgrößen mit Wertebelegung, die sich durch externe Messungen bestimmen lassen.
  - Menge aller Stellglied- und Komponententests, die durchgeführt wurden.
- Output Algorithmus B:
  - Menge der verdächtigen Komponenten mit Bewertungspunktzahl

**Ausgabe:** (*Output Algorithmus K*)

- Priorisierte Liste der Messungen und einfachen Tests

**Algorithmus:**

**K1.** Berechnung der mittleren Reduktion der verdächtigen Komponentenmenge durch zukünftige Ausführung einer Messung oder eines Tests. Diese Berechnung erfolgt über die Mittelwertbildung unter der Annahme, dass Messungen beziehungsweise Tests bestimmte Werte annehmen, da in dem nächsten Diagnoseschritt die Wertebelegung der Modellgrößen, die sich durch Messung bestimmen lassen, sowie die Ergebnisse der durchgeführten Tests nicht vorhersagbar sind.

**K2.** Ermittlung der mittleren Reduktion unter Berücksichtigung der Bewertungspunktzahl der verdächtigen Komponenten, da die reine mittlere Reduktion der verdächtigen Komponentenmenge durch Messungen oder Tests nicht ausreichend ist.

Die Menge der verfügbaren Systemsensoren, Messungen und einfachen Tests ergibt sich aus der Differenz zwischen den Mengen, die im Diagnosemodell abgebildet sind, und den Mengen der durchgeführten externen Messungen, ausgelesenen Systemsensor-Istwerten sowie der Menge der durchgeführten einfachen Tests [35]. Es gilt:

$$\begin{aligned} SEN^{todo} &= SEN / SEN^{todo} \\ MEAS^{todo} &= MEAS / MEAS^{todo} \\ ActTEST^{todo} &= ActTEST / ActTEST^{todo} \\ ComTEST^{todo} &= ComTEST / ComTEST^{todo} \end{aligned}$$

Da die Wertebelegungen der zu messenden Parameter eines Sensors, der zur Diagnoselaufzeit noch nicht abgefragt wurde, nicht bekannt sind, muss mit allen möglichen auftretenden Werten gerechnet werden. Folglich wird für jede Wertebelegung die Reduktion anhand der Menge der verdächtigen Komponenten ermittelt und der Mittelwert der Reduktion mit Hilfe der Anzahl der Möglichkeiten gebildet. Es seien mit:

- $sen_i$  - der  $i$ -te Sensor, der sich in der Menge der noch nicht ausgelesenen Sensoren befindet ( $sen_i \in SEN^{todo}$ )
- $istwert_k$  - der  $k$ -te Istwert, der von dem  $i$ -ten Sensor  $sen_i$  gemessen wird
- $VAL_k$  - die qualitative Wertemenge, in der der  $k$ -te Istwert  $istwert_k$  einen Wert annehmen kann.

Für jede mögliche Wertebelegung  $val_j \in VAL_k | 0 \leq j < |VAL_k|$  wird der Algorithmus  $D2$  mit der hypothetischen Wertebelegung des Istwerts  $istwert_k = val_j$  und die  $D2$ -Inputinformation aufgerufen. Die Reduktionsgröße der Menge der verdächtigen Komponenten auf Grund der Hypothese ist die Differenz zwischen der Kardinalität der vorhandenen Verdächtigungsmenge und der Kardinalität der hypothetischen Verdächtigungsmenge [36]:

$$R(istwert_k, val_j) = |COM^{susp}| - |COM^{suspHYP}|$$

Die mittlere Reduktion der Menge der verdächtigen Komponenten durch Auslesen eines Istwerts  $istwert_k$  des systemeigenen Sensors  $sen_i$  mit  $sen_i \in SEN^{todo}$  ergibt sich aus dem arithmetischen Mittelwert der einzelnen, oben berechneten Reduktionsgrößen:

$$\bar{R}(istwert_k) = \frac{\sum_{j=1}^{|VAL_k|} R(istwert_k, val_j)}{|VAL_k|}$$

Die Berechnung der mittleren Reduktion der verdächtigen Komponentenmenge durch externe Messungen erfolgt analog zur Berechnung der Systemsensoren. Die mittlere Reduktion des einfachen Tests erfolgt identisch wie oben. Der Unterschied besteht darin, dass das Argument nicht der Istwert ist, sondern die internen Variablen der zu testenden Komponenten [34]:

$$\bar{R}(var_l) = \frac{\sum_{j=1}^{|VAL_l|} R(var_l, val_j)}{|VAL_l|}$$

Da die Bewertungspunktzahl der Komponenten in der verdächtigen Menge möglicherweise unterschiedlich ist, ist die Betrachtung der mittleren Reduktionsgröße des Sensoreinsatzes, des Messungseinsatzes beziehungsweise des einfachen Tests im Zusammenhang mit der Bewertungspunktzahl zur Erstellung der Liste der Handlungsempfehlung sinnvoll. Es seien mit:

-  $com_i$  - die  $i$ -te Komponente, die sich in der Menge der verdächtigen Komponenten befindet ( $com_i \in COM^{sup_s}$ )

-  $w_i$  - die Bewertungspunktzahl der Komponenten  $com_i$ , die aus dem Algorithmus B bezogen auf vorhandene Beanstandungen und durchgeführte Messungen sowie Tests ermittelt wurden.

Für jede verdächtige Komponente  $com_i$  kann die auftretende Wahrscheinlichkeit  $p_i$  zur Erklärung der vorhandenen Symptome wie folgt berechnet werden:

$$p_i = \frac{w_i}{\sum_{j=1}^{|COM^{sup_s}|} w_j}$$

Für die Berücksichtigung der Bewertungspunktzahl bei der Erstellung der Handlungsempfehlung kann wie folgt vorgegangen werden [34], [36]:

$$\overline{R^p}(istwert_k) = \frac{R(istwert_k, val_{j=1}) \times \sum_{i=g}^h p_i + \dots + R(istwert_k, val_{j=|VAL_k|}) \times \sum_{i=n}^m p_i}{|VAL_k|}$$

$$\overline{R^p}(var_l) = \frac{R(var_l, val_{j=1}) \times \sum_{i=g}^h p_i + \dots + R(var_l, val_{j=|VAL_l|}) \times \sum_{i=n}^m p_i}{|VAL_l|}$$

Anhand der berechneten Werte ( $\overline{R^p}$ ) der durchzuführenden Messungen und einfachen Tests werden die Elemente der Handlungsempfehlungsliste sortiert.

### 4.3.2 Gruppierung der verdächtigen Komponenten (Algorithmus S)

---

**Algorithmus 4.6** : Diagnostik-Algorithmus (S)

---

**Eingabe:** (*Input Algorithmus S*)

- Ausgabe des Output Algorithmus B:
- Menge der verdächtigen Komponenten

**Ausgabe:** (*Output Algorithmus S*)

- Liste der Komponenten, die in Gruppen zusammengefasst sind

**Algorithmus:**

**S1.** Gruppierung der verdächtigen Komponenten nach Branch-And-Bound-Skyline-Algorithmus<sup>3</sup>.

Die Gruppierung der verdächtigen Komponenten erfolgt in dieser Arbeit nach dem Skyline-Konzept. Dieses Konzept löst das Problem der ungenauen Benutzeranfragen bei der Datenbankanfrage. Es seien mit:

$obj$  - Objekt, das im Vergleich steht. Dieses Objekt wird von  $n$ -Attributen repräsentiert:

$$- obj = \{att_1, att_2, \dots, att_n\}$$

$R^n$  -  $n$ -dimensionaler Raum, wobei jede Dimension ein Attribut  $att_i$  des Objekts  $obj$  repräsentiert

$OBJ$  - endliche Menge der Objekte, die im Vergleich zur Verfügung stehen:

$$- obj_i \in OBJ | obj_i \in R^n$$

$OBJ^s$  - endliche Menge der Objekte, die als Skyline bezeichnet werden. Dann muss

$OBJ^s$  folgende Eigenschaften erfüllen:

$$- OBJ^s \in OBJ$$

$$- \forall obj^s \in OBJ^s | obj^s = \{att_1^s, att_2^s, \dots, att_n^s\} \wedge \forall obj \in OBJ/OBJ^s | obj = \{att_1, att_2, \dots, att_n\} \text{ gilt:}$$

$$\forall obj_i^s, obj_j^s | att_i^s \geq att_j^s \wedge \exists obj_j^s, obj_j^s | att_j^s > att_i^s \text{ wobei } i, j = \{1, 2, \dots, n\}.$$

Die verdächtigen Komponenten werden mit einem Skyline-Algorithmus zu Gruppen zusammengefasst. Dabei werden die Komponenten entsprechend ihrer Punktzahl beziehungsweise ihres Verdächtigungsgrads, ihrer Test-/ Prüfkosten und -zeit, Alterszustand sowie der Verfügbarkeit von zum Austausch notwendigen Ersatzteilen gruppiert. Die vergleichenden Kriterien bei der Diagnose entsprechen einem fünfdimensionalen Raum. Die Entscheidung für den Skyline-Algorithmus beruht auf folgenden Randbedingungen des verfügbaren Diagnosewissens und der Werkstattanforderungen, die mit den Anforderungen an Online-Algorithmen übereinstimmen [73]:

- Einfache Anwendung des Algorithmus auf bestehende Diagnosedaten ohne Aufforderung zur Änderung der Datenstruktur
- Möglichkeit zur Verarbeitung aller verfügbaren und gegebenenfalls fehlenden Diagnosedatensätze;
- Keine Einschränkung der Veränderung der Benutzeranfrage während des Diagnosevorgangs, so dass keine manuelle Anpassung des Algorithmus nötig ist. Der Algorithmus nimmt neue Komponenten in Skyline auf und entfernt Komponenten, die nicht mehr dazu gehören, im Laufe des Diagnosevorgangs unter veränderter Anfrage;
- Ein Teil der Gruppierungsergebnisse liegt in kurzer Zeit vor und der Algorithmus liefert die kompletten Ergebnisse in endlicher Zeit. Damit ist die Terminierung des Algorithmus immer garantiert und die Wartezeit des Benutzers wird verkürzt;
- Der Algorithmus berücksichtigt alle Vergleichsdimensionen der Datensätze bei der Gruppierung der verdächtigen Komponenten. Dadurch bekommt jeder Benutzertyp seinen Favoriten, unabhängig von unterschiedlichen Präferenzen [19].

Die oben genannten Anforderungen stellen ein klassisches Problem der Anfrageoptimierung dar. Als Maximum-Vektor-Problem wurde dieses Problem in [74] [62] identifiziert, zu dessen Lösung der Algorithmus entwickelt wurde. In dieser Arbeit wurde der Branch-And-Bound-Algorithmus, der in [109], [110] vorgestellt wird, in Java implementiert und für die Berechnung der Skyline im Diagnosevorgang verwendet. Dieser Algorithmus beruht auf dem gleichen Lösungsansatz wie der Nearest-Neighbor-Algorithmus [73] und ist für den höher dimensionaleren Suchraum geeignet [21].

---

<sup>3</sup>Algorithm BBS (R-tree R)

Begin BNN

1.  $S = \emptyset$  /\*list of skyline points\*/
2. insert all entries of the root  $R$  in the heap
3. while heap not empty
4. remove top entry  $e$
5. if  $e$  dominated by some point in  $S$  discard  $e$
6.     else /\* $e$  is not dominated\*/
7.         if  $e$  is an intermediate entry
8.             for each child  $e_i$  of  $e$
9.                 if  $e_i$  is not dominated by some point in  $S$
10.                     insert  $e_i$  into heap
11.         else /\* $e$  is a data point\*/
12.             insert  $e_i$  into  $S$
13. end while



## 4.4 Anwendungsbeispiele

In diesem Abschnitt werden Beispiele gezeigt, die aus der Evaluierung in den Werkstätten entstanden sind. Nachfolgend wird auf einzelne Algorithmen mit Beispielen eingegangen. Dabei existieren Unterschiede zwischen den Vorgehensweisen in dieser Arbeit im Vergleich zu existierenden Lösungen der funktionalen Klassifikation (vgl. [111]), weil das Diagnosewissen nicht nur in modellbasierter, sondern auch in heuristischer Form vorliegt. Zur Erläuterung der Algorithmen werden in dem ersten und zweiten Diagnosebeispiel die Diagnoseschritte verbal beschrieben. Zur Veranschaulichung der vorgestellten Diagnosealgorithmen werden in dem dritten Beispiel die Schritte im Diagnosevorgang visualisiert. Die Beispiele sind reale Diagnosefälle, die aus den Feldtestdaten entnommen wurden.

### 4.4.1 Anwendungsbeispiel mit existierenden Fehlercodes

Als erstes Beispiel wird eine kurze Diagnose eines Teils des Luftsystems (Abbildung 4.2) mit eindeutigem Fehlercode ausgewählt. Mit diesem Beispiel wird die Notwendigkeit des Algorithmus E verdeutlicht. Dabei stehen folgende Diagnoseinformationen zur Verfügung:

- Es existiert ein Fehlercode: P0112.
- Die Kundenbeanstandung lautet „Schlechte Leistung bei Kaltstart“.

Durch den Fehlercode P0112, der im ersten Beispiel direkt auf die Komponente Temperatursensor zeigt, beinhaltet die Tabelle für komponentenbezogene Diagnoseparameter dieses Beispiels folgende Einträge:

$DTC^{input}$	Komponentenbezogene Diagnoseparameter
P0112	{ <i>Temperatursensor</i> }

Tabelle 4.14: Darstellung der Menge  $COM^{DTC,-OK} = \{Temperatursensor\}$  des Fehlercodes P0112 in Tabellenform

Bevor die Kundenbeanstandung „Schlechte Leistung bei Kaltstart“ mit dem Algorithmus A analysiert wird, werden durch den Algorithmus E die Prüfschritte des Temperatursensors vorgeschlagen (Abbildung 4.1). In diesem konkreten Beispiel wird die Empfehlung befolgt und die Komponente Temperatursensor geprüft (der Inhalt des Komponententests beziehungsweise der Komponentenprüfung ist in Abbildung 3.20 dargestellt und in Abschnitt 3.7 beschrieben). Hierbei wird zuerst versucht, den Istwert des Sensors im Ansaugtrakt auszulesen. Als Ergebnis liefert der Sensor einen Wert von  $-40^{\circ}\text{C}$ . Dieser Wert ist unplausibel, da die Temperatur in der Werkstatt höher als  $-40^{\circ}\text{C}$  ist. Dieses Messungswissen ist fahrzeugspezifisch und direkt an die Fahrzeugkomponente gebunden (Kapitel 3 Abschnitt 2). Die Erklärung für die Abweichung ist entweder Kabelbruch, ein loser Sensorstecker oder Wackelkontakt. Mit dieser Erklärung, die in der Komponentenprüfung steht, wurde in diesem Fall eine Prüfung durchgeführt und ein Wackelkontakt als Fehler festgestellt. Als Reparaturmaßnahme wurde der Kontakt des Temperatursensors erneuert.

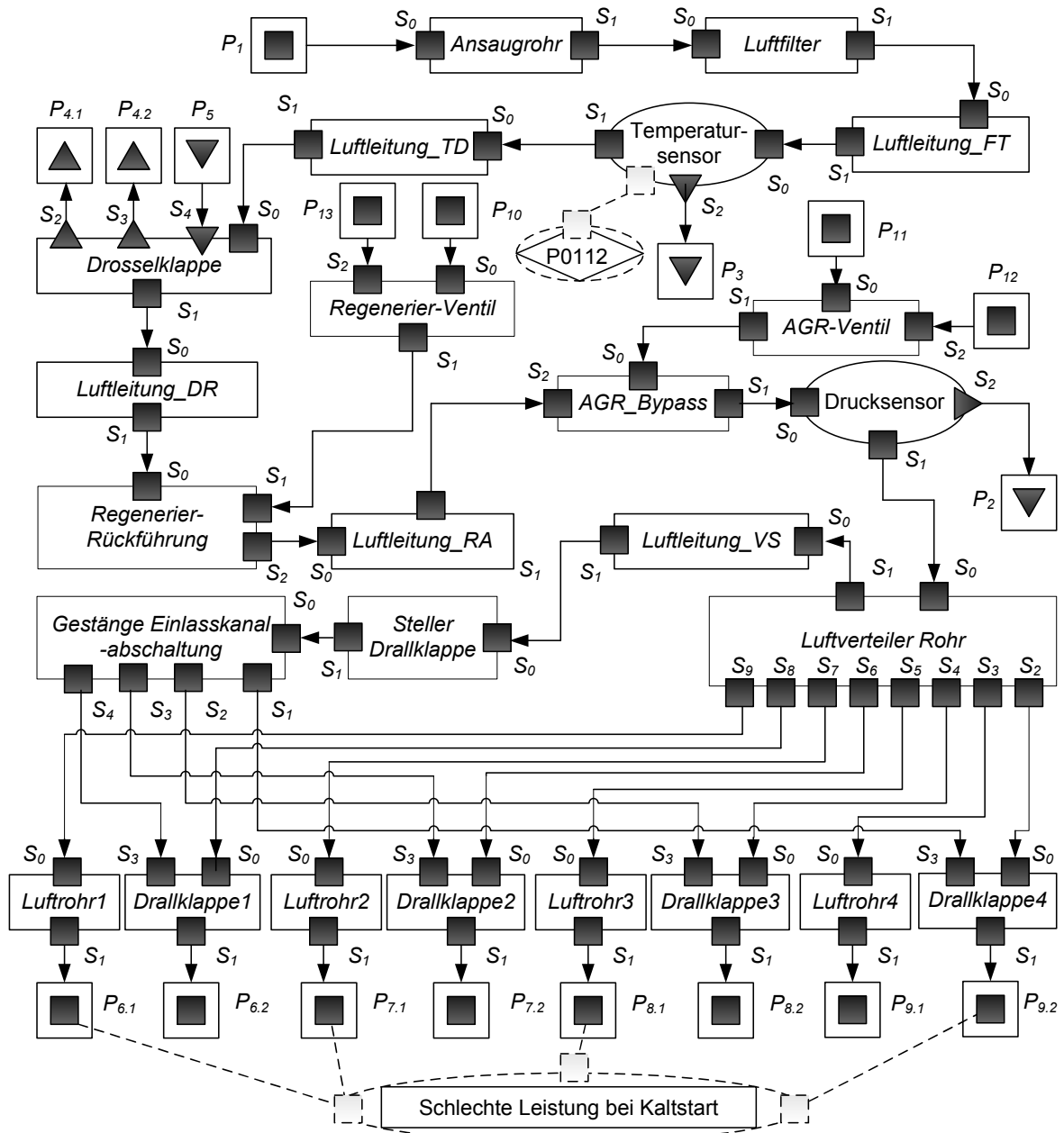


Abbildung 4.2: Modell des LuftsysteMS ME7.6.x mit Fehlercode und Kundenbeanstandung für das erste Beispiel

#### 4.4.2 Anwendungsbeispiel mit Kundenbeanstandung, Sichtprüfung und ohne Fehlercode

Anhand des obendargestellten Luftsystems wird ein weiterer Diagnosevorgang, der relativ komplex ist, als zweites Beispiel herangezogen. Folgende Diagnoseinformationen stehen zur Verfügung:

- Fehlercode ist nicht vorhanden.
- Die Kundenbeanstandung lautet „Schlechte Gasannahme, Übergangsfehler“.
- Es existiert keine Leckage in dem System (Sichtprüfung).

In diesem Anwendungsbeispiel existieren keine Fehlercodes, daher sind die Analyseergebnisse  $COM^{DTC,-OK}$  der verfügbaren Diagnoseinformationen nach der Anwendung des Algorithmus E eine leere Menge:

$$COM^{DTC,-OK} = \emptyset$$

Mit dieser Vorbedingung wird der Algorithmus A gestartet. Die Anwendung des Algorithmus A auf das Beispiel Luftsystem mit der Sichtprüfung „keine Leckage“ (Tabelle 4.15) und der Kundenbeanstandung „Schlechte Gasannahme, Übergangsfehler“ (Tabelle 4.16) ergibt folgende Diagnoseinformationen:

Diagnoseparameter	$ComTEST^{input}$
	Sichtprüfung: Keine Leckage
<i>Luftrohr[1..4].Leckage</i>	$\{N\}$
<i>Luftverteiler Rohr.Leckage</i>	$\{N\}$
...	...
<i>Ansaugrohr.Leckage</i>	$\{N\}$

Tabelle 4.15: Darstellung der Sichtprüfung „Keine Leckage“ in Tabellenform

Diagnoseparameter	$OBS^{input}$
	Schlechte Gasannahme, Übergangsfehler
<i>Luftsystem.P<sub>6,2</sub>.Luft.<math>\dot{m}</math></i> <sup>4</sup>	$\{L\}$
<i>Luftsystem.P<sub>7,2</sub>.Luft.<math>\dot{m}</math></i>	$\{L\}$
<i>Luftsystem.P<sub>8,2</sub>.Luft.<math>\dot{m}</math></i>	$\{L\}$
<i>Luftsystem.P<sub>9,2</sub>.Luft.<math>\dot{m}</math></i>	$\{L\}$

Tabelle 4.16: Darstellung der Kundenbeanstandung „Schlechte Gasannahme, Übergangsfehler“ in Tabellenform

<sup>4</sup>Massenfluss :=  $\dot{m}$

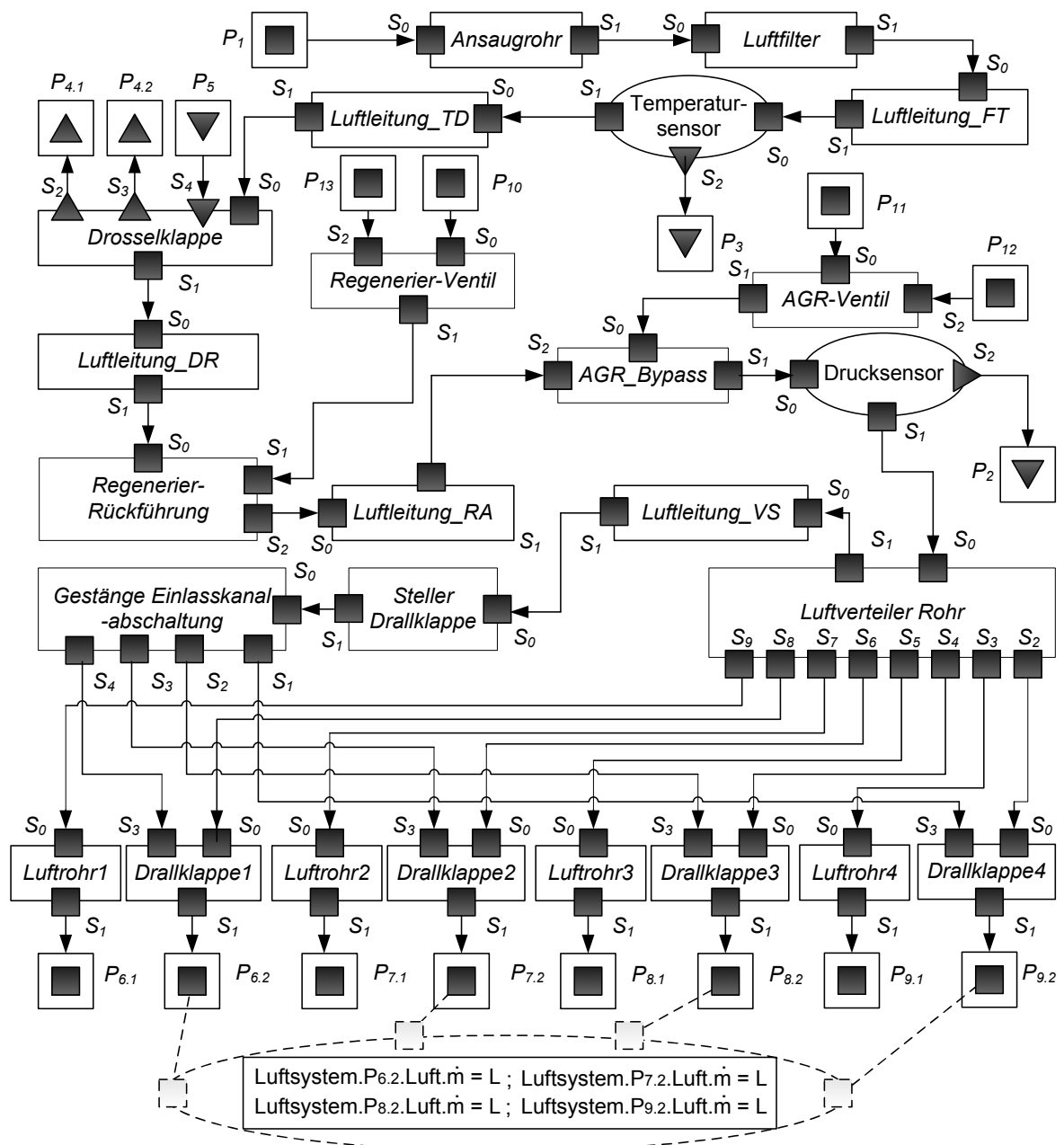


Abbildung 4.3: Modell des Luftsystems ME7.6.x mit Kundenbeanstandung für das zweite Beispiel

Im folgenden Abschnitt wird die Funktionsweise der Algorithmen P und B anhand des zweiten Diagnosebeispiels erläutert. Der Algorithmus P besteht aus vier Schritten. Zur Veranschaulichung der Schritte dieses Algorithmus können nicht die einzelnen Komponenten des Modells (Abbildung 4.3) durchgegangen, sondern nur der Hauptpfad und die relevanten Komponenten beschrieben werden.

<i>Drallklappe<sub>1</sub></i>										
$S_3^{in}$			$S_0^{in}$				$S_1^{out}$			
<i>Transl</i> <sup>5</sup>			<i>Luft</i>				<i>Luft</i>			
<i>f</i>	<i>s</i>	<i>m</i>	...	...	...	<i>t</i>	...	<i>m</i>	...	<i>t</i>
N	N	H	...	...	...	N	...	H	...	...
N	L	H	...	...	...	N	...	H	...	...
...										
L	L	H	...	...	...	N	...	H	...	...
N	N	N	...	...	...	N	...	N	...	...
N	L	N	...	...	...	N	...	L	...	...
N	N	L	...	...	...	N	...	L	...	...
...										
N	L	L	...	...	...	N	...	L	...	...

Tabelle 4.17: Beispiel eines Abschnitts der reduzierten Verhaltenstabelle der Komponente *Drallklappe<sub>1</sub>* mit *Drallklappe<sub>1</sub>.leakage = N*

Anhand der Diagnoseinformation in Tabelle 4.15 wird die Reduzierung der Verhaltenstabelle der Systemkomponenten mittels Schritt P1 des Algorithmus P gestartet. Im Beispiel der Komponente *Drallklappe<sub>1</sub>* wird die Verhaltenstabelle dieser Komponente mit der Diagnoseinformation *Drallklappe<sub>1</sub>.leakage = N* reduziert. Es gilt:

$$P_1: \text{tab}_{\text{Drallklappe}_1} \xrightarrow{\text{Drallklappe}_1, \text{leakage}=N} \text{tab}_{\text{Drallklappe}_1}^{P_1}$$

Analog zur *Drallklappe<sub>1</sub>* werden die Verhaltenstabellen aller Systemkomponenten mit dem Ergebnis der Sichtprüfung (Tabelle 4.15) reduziert.

Nach der ersten Reduzierung der Verhaltenstabellen der Systemkomponenten durch die Projektion der Wertebelegungen der Diagnoseparameter wird die Rückwärtspropagierung gestartet (Schritt P2). In Beispiel zwei wird die Diagnoseinformation aus Tabelle 4.16 verwendet. Diese Tabelle ist das Ergebnis der Analyse des Algorithmus A. Für die Systemschnittstelle *P<sub>6.2</sub>* gilt:

$$\text{Luftsystem.P}_{6.2}.\text{Luft}.\dot{m} = L$$

Wegen der gerichteten Verbindung von Ausgangsschnittstelle *S<sub>1</sub>* der Komponente *Drallklappe<sub>1</sub>* zur Ausgangsschnittstelle *P<sub>6.2</sub>* des Systems kann durch die Rückwärtspropagierung folgende Schlussfolgerung erstellt werden:

<sup>5</sup>*Transl* : Translationsbewegung; *f*: Kraft; *s*: Weg

$$Drallklappe_1 \cdot S_1 \cdot Luft \cdot \dot{m} = L.$$

<i>Drallklappe<sub>1</sub></i>										
$S_3^{in}$		$S_0^{in}$				$S_1^{out}$				
<i>Transl</i>		<i>Luft</i>				<i>Luft</i>				
<i>f</i>	<i>s</i>	<i>m</i>	...	<i>t</i>	...	<i>leakage</i>	...	<i>m</i>	...	<i>t</i>
N	L	N	...	...	...	N	...	L	...	...
N	N	L	...	...	...	N	...	L	...	...
...										
N	L	L	...	...	...	N	...	L	...	...

Tabelle 4.18: Beispiel eines Abschnitts der reduzierten Verhaltenstabelle der Komponente *Drallklappe<sub>1</sub>* mit  $Drallklappe_1.leakage = N$  und  $Drallklappe_1.S_1^{out}.Luft.m = L$

Die Weiterreduzierung der Verhaltenstabelle der Komponente *Drallklappe<sub>1</sub>* kann mit dieser Wertebelegung durchgeführt werden. Als Ergebnis entsteht eine neue reduzierte Verhaltenstabelle 4.18.

Analog zur Komponente *Drallklappe<sub>1</sub>* werden die Verhaltenstabellen der Komponenten *Drallklappe<sub>2</sub>*, *Drallklappe<sub>3</sub>* und *Drallklappe<sub>4</sub>* im gleichen Schritt reduziert. Der Vorgang der Rückwärtspropagierung wird bis zu den Eingangsschnittstellen des Luftsystems fortgesetzt. Dabei werden verschiedene Komponenten des Luftsystems verdächtigt, welche in ihren reduzierten Verhaltenstabellen mindestens eine Zeile mit dem Eintrag:

$$(com_i^{out}.Luft.m = L) \wedge ((com_i.var_j^{faul} = \neg N) \vee (com_i.var_j^{faul} = T))$$

beinhalten. Zum Beispiel existieren durch die Rückwärtspropagierung der Wertebelegung der Schnittstelle  $S_0$  der Komponente *Drallklappe<sub>i</sub>* mit  $i \in 1, 2, 3, 4$  die folgenden Einträge in der reduzierten Verhaltenstabelle (Tabelle 4.19) der Komponente *Drosselklappe*:

<i>Drosselklappe</i>											
$S_4^{in}$		$S_0^{in}$				$S_1^{out}$			$S_2^{out}$		$S_3^{out}$
<i>Steuerung</i>		<i>Luft</i>				<i>Luft</i>			<i>Messsignal</i>		<i>Messsignal</i>
<i>st</i>	<i>m</i>	...	<i>t</i>	...	...	<i>m</i>	...	<i>t</i>	<i>ms</i>	<i>ms</i>	
L	L	...	...	...	T	...	L	...	...	...	...
N	N	...	...	...	T	...	L	...	...	...	...
...											
L	N	...	...	...	F	...	L	...	...	...	...

Tabelle 4.19: Beispiel eines Abschnitts der reduzierten Verhaltenstabelle der Komponente *Drosselklappe* mit  $Drosselklappe.klemmt = N$  und  $Drosselklappe.schwergaengig = T$

$S_0^{in}$				$S_1^{out}$ $S_2^{out}$ $S_3^{out}$ $S_4^{out}$								
<i>Transl</i>		...	<i>mechanischerFehler</i>	...	<i>Transl</i>		<i>Transl</i>		<i>Transl</i>		<i>Transl</i>	
<i>f</i>	<i>s</i>				<i>f</i>	<i>s</i>	<i>f</i>	<i>s</i>	<i>f</i>	<i>s</i>	<i>f</i>	<i>s</i>
N	N	...	T	...	N	L	N	L	N	L	N	L
N	L	...	T	...	N	L	N	L	N	L	N	L
...												
N	N	...	N	...	N	N	N	N	N	N	N	N
L	N	...	F	...	L	N	L	N	L	N	N	L

Tabelle 4.20: Beispiel eines Abschnitts der reduzierten Verhaltenstabelle der Komponente *Gestänge Einlasskanalabschaltung*

Die Rückwärtspropagierung der Wertebelegung von Schnittstelle  $S_3$  der Komponente *Drallklappe* reduziert die Verhaltenstabelle der Komponente *Gestänge Einlasskanalabschaltung* zur Tabelle 4.20.

Auf die Menge der verdächtigen Komponenten mit ihren reduzierten Verhaltenstabellen wird der Diagnostik-Algorithmus  $B$  zur Erzeugung der Bewertungspunktzahl angewendet. In diesem Beispiel erhalten die Komponenten *Drosselklappe* und *AGR-Ventil* die höchste Punktzahl, weil ihre reduzierten Verhaltenstabellen die meisten Zeilen mit der belegten internen Fehlervariable beinhalten. Das heißt, diese Komponenten haben die meisten Fehlerzustände (vergleiche Algorithmus  $B$ ). Die Komponente *Drallklappe<sub>1</sub>*, *Drallklappe<sub>2</sub>*, *Drallklappe<sub>3</sub>* und *Drallklappe<sub>4</sub>* sowie *Gestänge Einlasskanalabschaltung* bekommen geringere Punktzahlen als die oben genannten Komponenten, weil die Anzahl der Zeilen mit belegten internen Fehlervariablen in ihren reduzierten Verhaltenstabellen kleiner ist als die Anzahl der Zeilen der Komponente *Drosselklappe* beziehungsweise *AGR-Ventil*. Es existieren drei einfache Komponententests für die verdächtigen Komponenten, die aus den Ergebnissen der Algorithmen  $P$  und  $B$  entstanden sind:

- Test der Komponente *Drosselklappe* durch Signalspannungs-, Widerstandsprüfung sowie Prüfungen der Drosselklappen-/Pedalpositionssensoren A und B
- Test der Komponente *AGR-Ventil* durch Prüfung der Spannungsversorgung, des Komponentenwiderstands und des Istwerts des Ladedrucks
- Stellgliedtest für folgende Komponenten der Drallklappengruppe :
  - *Drallklappe<sub>1</sub>*, *Drallklappe<sub>2</sub>*, *Drallklappe<sub>3</sub>* und *Drallklappe<sub>4</sub>*
  - *Gestänge Einlasskanalabschaltung*
  - *Steller Drallklappe* (Drallklappenantrieb)

Im Luftsystem existiert nur der *Drucksensor*, welcher mit seinem Istwert über den aktuellen Ladedruck informiert. Da ein direkter Zusammenhang zwischen dem Ladedruck und der aktuell angesaugte Luftmasse in Form der Relationsgleichung

$$Drucksensor_1^{out}.Luft.\dot{m} \stackrel{qual}{=} Drucksensor_0^{in}.Luft.druck \stackrel{qual}{-}$$



$Drucksensor_1^{out}.Luft.druck,$

existiert, eignet sich dieser Messwert zur Prüfung der Funktionsluftfüllung aller Komponenten des Luftsystems, die vor dem Drucksensor liegen, wie zum Beispiel: *Drosselklappe* und *AGR – Ventil*. Die Priorisierung der Messaktionen und einfachen Komponententests durch den Algorithmus *K* ergibt folgende Empfehlungen:

- Istwert des Drucksensors auslesen und mit Sollwert vergleichen
- Stellgliedtest für Komponenten der Drallklappengruppe durchführen
- Komponententest der Komponente *Drosselklappe* durchführen
- Komponententest der Komponente *AGR – Ventil* durchführen

Die Positionierung des Auslesens des Drucksensor-Istwerts als erste Handlungsempfehlung liegt darin begründet, dass mit der Auswertung dieses Istwerts die meisten Komponenten einbeziehungsweise ausgeschlossen werden können und die entstehenden Kosten bei der Ausführung der Aktion sehr gering sind. Aus dem gleichen Grund befindet sich der Stellgliedtest an zweiter Stelle. Der Test der Komponente *Drosselklappe* ist als Drittes, weil die Funktion der Komponente *Drosselklappe* größeren Einfluss auf den Luftmassenfluss hat als die Komponente *AGR – Ventil*.

Die Auswahl der komplizierten Tests beziehungsweise Prüfungen der restlichen Komponenten erfolgt durch den Algorithmus *S*. Hierbei entsteht eine Liste mit folgenden Eintragungen:

- Komponententest der Komponente *Luftfilter*
- Komponententest beziehungsweise Verstopfungsprüfung der Rohre im Luftsystem (Dichtheitsprüfung fällt wegen den existierenden Diagnoseinformationen aus)

Der Test des Luftfilters auf Verstopfung hat höhere Priorität als die Komponententests der Rohre im Luftsystem, weil die Wahrscheinlichkeit einer Verstopfung der Komponente *Luftfilter* höher ist als die Verstopfungswahrscheinlichkeit der Rohre. Zudem sind die Kosten des Luftfiltertests geringer als die Komponententests der Rohre, da der Ein- und Ausbau dieser Komponente geringeren Aufwand erfordert.

In diesem Anwendungsbeispiel befolgt der Nutzer die generierten Handlungsempfehlungen und führt die Druckmessung mittels des systemeigenen Drucksensors durch. Diese Aktion ist der erste Schritt des iterativen Diagnoseprozesses. Dabei muss der Messwert zuerst auf Plausibilität geprüft werden. In diesem Beispiel ist das Messergebnis des Ansaugdrucks plausibel und im Sollbereich. Die neu gewonnene Information wird wie folgt formalisiert:

$Drucksensor_0^{in}.Luft.druck \stackrel{qual}{=} N.$

Mit der neuen Diagnoseinformationen wird der Algorithmus *A* erneut ausgeführt. Danach werden die Algorithmen *P* und *B* zur Verdachtsgenerierung und -überprüfung angestoßen. Die Rückwärtspropagierung der Attribute im Modell entfernt alle Zeilen der Verhaltenstabelle der Komponenten, die vor dem Drucksensor liegen und in deren Verhaltenstabelle der Luftmassenfluss  $Komponenten_x^{out}.Luft.m$  durch Druckabweichung nicht in Ordnung sein könnte. Damit reduziert sich die Anzahl der Zeilen einer Komponente, die vor dem Drucksensor liegt.

Die Verhaltenstabellen der Komponenten des Luftsystems, die hinter dem Drucksensor liegen, können hingegen kaum reduziert werden, da alle Zeilen mit Wertebelegungen des Drucks als „in Ordnung“ im Eingang und einer Abweichung am Ausgang nicht entfernt werden können.

Auf Basis der neuen reduzierten Verhaltenstabelle wird durch die Algorithmen  $K$  und  $S$  die Liste der Messaktionen und einfachen Komponententests sowie die Liste komplizierter Komponententests neu sortiert. Dabei entsteht durch den Algorithmus  $K$  eine neue Liste der Messaktionen und einfachen Komponententests, die der alten Liste entspricht:

- Stellgliedtest für Komponenten der Drallklappengruppe (*Steller Drallklappe*, *Gestänge Einlasskanalabschaltung*, *Drallklappe<sub>1</sub>*, *Drallklappe<sub>2</sub>*, *Drallklappe<sub>3</sub>*, *Drallklappe<sub>4</sub>*) durchführen
- Komponententest der Komponente *Drosselklappe* durchführen
- Komponententest der Komponente *AGR – Ventil* durchführen

Die Liste der komplizierten Tests beziehungsweise Prüfungen der restlichen Komponenten wird neu sortiert:

- Verstopfungsprüftest: *Luftleitung\_VS*
- Verstopfungsprüftest: *Luftverteiler Rohr*
- Komponententest der Komponente *Luftfilter*
- Komponententest der Komponente *Regenerier – Ventil*
- Komponententest beziehungsweise Verstopfungsprüfung der Komponenten vor Drucksensor: *Ansaugrohr*, *Luftleitung\_FT*, *LuftleitungvTD*, *Luftleitung\_DR*, *Regenerier Rückführung*, *Luftleitung\_RA*, *AGR – Bypass*

Im zweiten Iterationschritt dieses Beispiels wird der Stellgliedtest für Komponenten der Drallklappengruppe (*Steller Drallklappe*, *Gestänge Einlasskanalabschaltung*, *Drallklappe<sub>1</sub>*, *Drallklappe<sub>2</sub>*, *Drallklappe<sub>3</sub>*, *Drallklappe<sub>4</sub>*) durchgeführt. In diesem Beispiel wurde der Stellgliedtest nicht bestanden. Damit erhöht sich der Verdächtigungsgrad der Komponenten der Drallklappengruppe im Unterschied zu den übrigen Komponenten. Des Weiteren existieren nur noch Komponententests sowohl in der Liste der Messaktionen und einfachen Komponententests als auch in der Liste der komplizierten Tests beziehungsweise Prüfungen. Wegen dieser Tatsache werden die beiden Listen zusammengefasst:

- Komponententest: *Gestänge Einlasskanalabschaltung*
- Komponententests: *Drallklappe<sub>1</sub>*, *Drallklappe<sub>2</sub>*, *Drallklappe<sub>3</sub>*, *Drallklappe<sub>4</sub>*
- Komponententest: *StellerDrallklappe*
- Komponententest: *Luftleitung\_VS*
- Komponententest: *LuftverteilerRohr*

Alle weitere Komponenten des Luftsystems (*Drosselklappe*, *AGR – Ventil*, *Luftfilter*, *Regenerier – Ventil*, *Ansaugrohr*, *Luftleitung\_FT*, *Luftleitung\_TD*, *Luftleitung\_DR*, *Regenerier Rückführung*, *Luftleitung\_RA*, *AGR – Bypass*) werden standardmäßig weiterhin verdächtig aufgeführt, allerdings mit niedriger Priorität (in der Benutzeroberfläche sind die Komponenten durch grüne Farbe markiert, das heißt dass diese Komponenten wahrscheinlich in Ordnung sind). Die Priorisierung der Komponentenstests erfolgt durch den Algorithmus *S*. Zum Beispiel befindet sich der Test der Komponente *Gestänge Einlasskanalabschaltung* an erster Stelle der Prüfliste, weil nicht nur die Kosten dieses Tests im Vergleich zu den Testkosten der Komponente *Steller Drallklappe* sowie der Komponenten *Drallklappe<sub>1</sub>*, *Drallklappe<sub>2</sub>*, *Drallklappe<sub>3</sub>*, *Drallklappe<sub>4</sub>* geringer ist, sondern auch die Fehlerwahrscheinlichkeit des mechanischen Fehlers dieser Komponente höher ist als die übrigen Komponenten der Drallklappengruppe. An zweiter Stelle steht die Prüfung der *Drallklappe* von 1 bis 4, weil die Fehlerwahrscheinlichkeit dieser Komponenten höher ist als die der Komponente *Steller Drallklappe*.

Im dritten Iterationschritt wird die Komponente *Gestänge Einlasskanalabschaltung* geprüft. Hierbei wurde festgestellt, dass diese Komponente einen mechanischen Fehler hat (die Verbindung zur *Drallklappe* von 1 bis 4 ist lose). Damit ist der Fehler identifiziert und die Anleitung zur Reparatur wird dem Werkstattpersonal angezeigt.

#### 4.4.3 Anwendungsbeispiel mit Kundenbeanstandung

Das dritte Beispiel betrifft die automatische Leuchtweitenregulierung (Abbildung 4.4). Das ALWR1.0-System ist ein automatisches Leuchtweitenregulierungssystem und hat zwei Aufgaben: Die Vermeidung der Blendung anderer Verkehrsteilnehmer infolge der Verwendung von Gasentladungslampen und die optimale Wegbeleuchtung für den Fahrer. Die Funktion des ALWR-Steuergeräts ist die Korrektur der Leuchtweiten des Scheinwerfers links und rechts. Diese Regelung erfolgt in Abhängigkeit vom Neigungswinkel des Fahrzeugs zur Straße und von der Fahrzeuggeschwindigkeit. Der Neigungswinkel (auch Nickwinkel oder Längsneigung genannt) wird aus Messwerten des Achssensors vorn und hinten berechnet. Der Sensor ist ein Winkelsensor und misst die Relativbewegung zwischen Fahrwerk und Karosserie. Zur Steuerung werden zwei Schrittmotoren verwendet, welche durch Stellmechaniken an die Scheinwerfer gebunden sind. In diesem letzten Anwendungsbeispiel werden die Diagnoseschritte visualisiert und damit ein Diagnosevorgang unter Verwendung der oben beschriebenen Algorithmen komplett erläutert. Im Startvorgang der Werkstattdiagnose wurden Kunden- und Fahrzeugdaten erfasst und die Kundenbeanstandung aufgenommen. Dabei wurde folgende Diagnoseinformation erhoben:

- Die Kundenbeanstandung lautet „Leuchtweite zu hoch“.

Anhand der Fahrzeugidentifikation und der Kundenbeanstandung wurde das relevante Diagnosemodell geladen. Dabei wird das betroffene System beziehungsweise die betroffene Funktion für den Nutzer dargestellt. Aufgrund des spezifischen Aufbaus des mechatronischen Systems wurde im nächsten Schritt der Fehlerspeicher aus dem Steuergerät ausgelesen (Abbildung 4.4). Als Ergebniss wurde in diesem Beispiel kein Fehlercode ausgegeben ( $COM^{DTC,-OK} = \emptyset$ ). Mit den Diagnoseinformationen:

- Fehlercode ist nicht vorhanden,

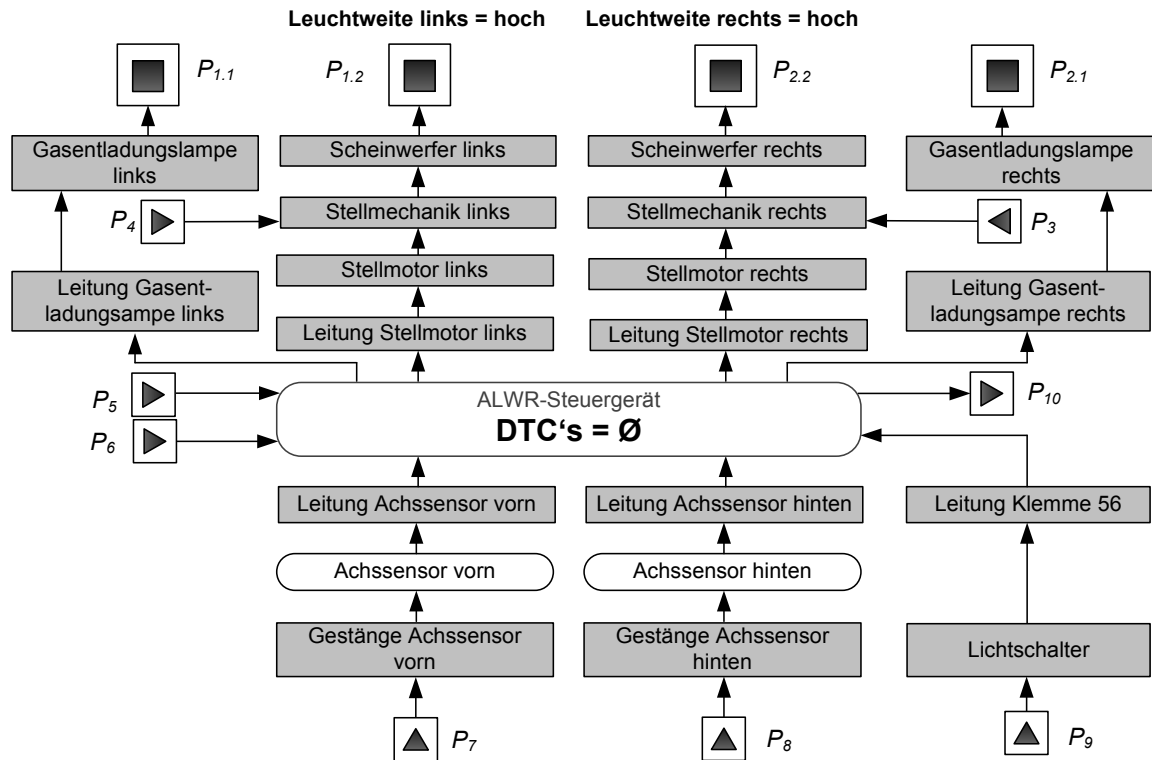


Abbildung 4.4: Auslesen der Fehlercodes mittels Steuergerät

- Die Kundenbeanstandung lautet „Leuchtwerte zu hoch“,

wurden die Algorithmen  $E$  und  $A$  gestartet. Hierbei wurde zuerst die verbale Beschreibung der Kundenbeanstandung in Modellgrößen mittels der Inhalte des Diagnosemodells überführt:

- $(P_{1.2}.Drehwinkel = hoch) \wedge (P_{2.2}.Drehwinkel = hoch)$ .

Im nächsten Schritt konnten folgende Komponenten des Systems durch die Algorithmen  $P$  und  $B$  ausgeschlossen werden (Abbildung 4.5):

- Gasentladungslampe links,
- Leitung Gasentladungslampe links,
- Gasentladungslampe rechts,
- Leitung Gasentladungslampe rechts,
- Leitung Klemme 56,
- Lichtschalter.

Die Ausschließung ist durch die Nichtbeeinflussung beziehungsweise Nichtbeteiligung dieser Komponenten an der Steuerung beziehungsweise an der Regulierung des Drehwinkels begründet. In diesem System existieren zwei systemeigene Sensoren und ein Stellgliedtest für die Steuerung der Scheinwerfer links und rechts. Zur Koordinierung der Sensoren-, Messsätze und einfachen Tests wurde der Algorithmus  $K$  angewendet. Als Ergebnis wurde eine Handlungsempfehlung mit folgenden Einträgen ausgegeben:

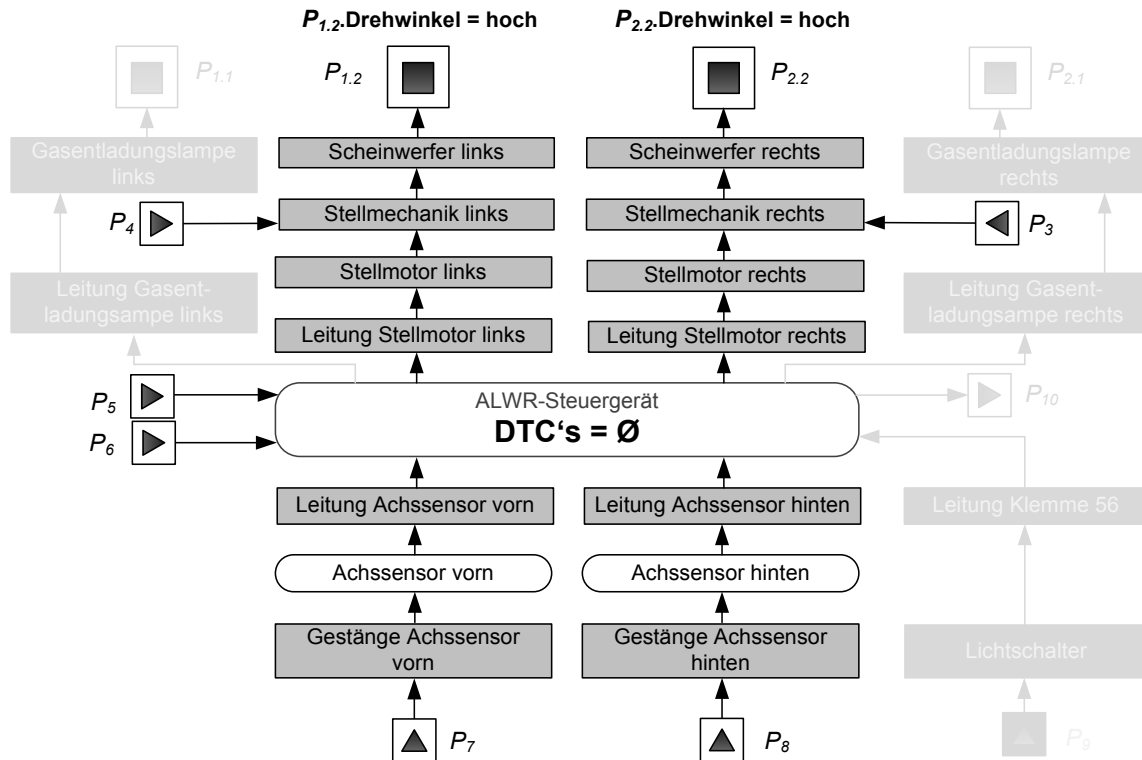


Abbildung 4.5: Ermittlung der verdächtigen Komponenten mittels Kundenbeanstandung

- Durchführung des Stellgliedtests für folgende Komponenten:
  - Scheinwerfer links, Stellmechanik links, Stellmotor links und Leitung Stellmotor links,
  - Scheinwerfer rechts, Stellmechanik rechts, Stellmotor rechts und Leitung Stellmotor rechts,
- Istwert des Achssensors vorn und hinten auslesen und mit Sollwert vergleichen.

Eine Priorisierung zur Ausschließung der verdächtigen Komponenten wird möglich durch die möglichen ausgeführten Aktionen und deren hypothetische Ergebnisse, da der Aufwand der Durchführung der Aktionen gleichgesetzt werden kann. Hierbei wurde der Stellgliedtest ausgeführt. Das Ergebnis des Stellgliedtests galt als bestanden und damit konnten folgende Komponenten ausgeschlossen und in der dargestellten Wirkkette ausgeblendet werden (Abbildung 4.6):

- Scheinwerfer links, Stellmechanik links, Stellmotor links und Leitung Stellmotor links,
- Scheinwerfer rechts, Stellmechanik rechts, Stellmotor rechts und Leitung Stellmotor rechts.

Da in diesem Beispiel die Handlungsempfehlung streng verfolgt werden sollte, war der nächste Schritt „Achssensor-Istwert auslesen“. Aufgrund der parallelen Bearbeitung der Diagnoseinformationen mittels der entwickelten Algorithmen konnten die Istwerte der beiden Achssensoren vorne und hinten abgefragt werden. Es existieren folgende Sensorinformationen:

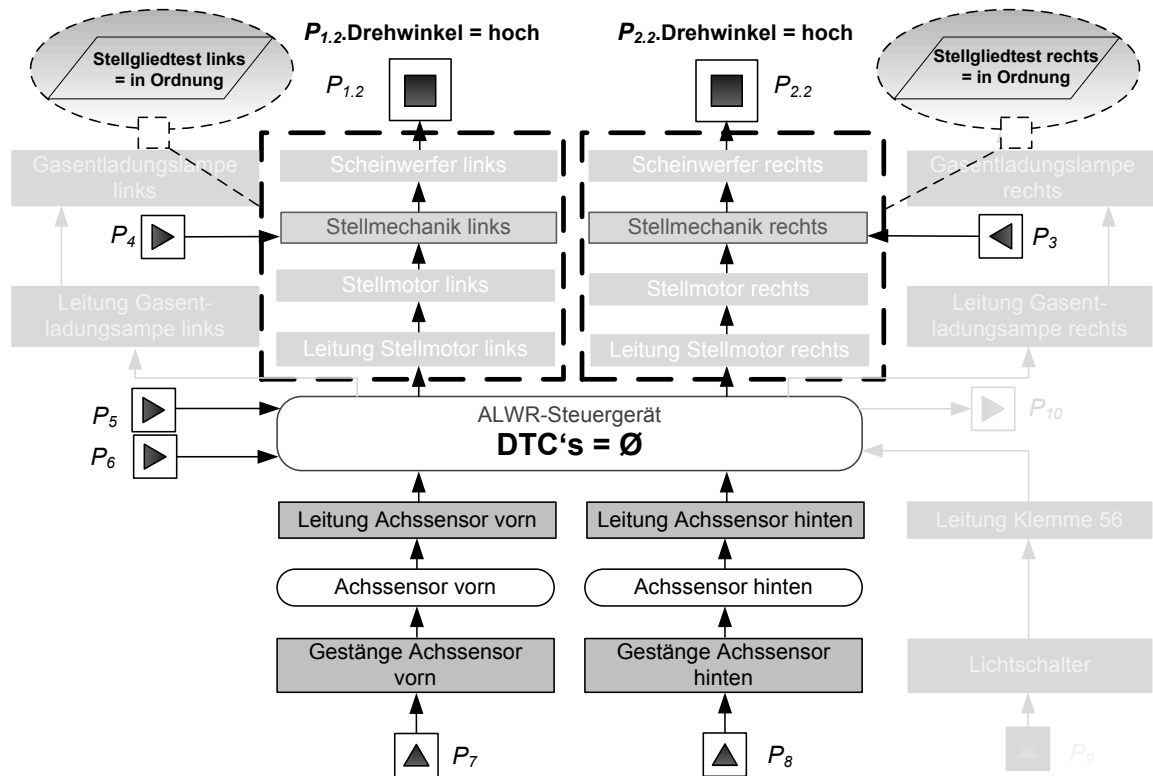


Abbildung 4.6: Ergebnisse des Stellgliedertests

- *Achssensor vorne:*
  - Istwert = 22,5% ,
  - Istwert  $\in$  Sollbereich = {12,5;...; 50%},
- *Achssensor hinten:*
  - Istwert = 32,8%,
  - Istwert  $\notin$  Sollbereich = {50;...; 87,5%}.

In Abbildung 4.7 sind die Ergebnisse der Messungen und deren Auswertungen dargestellt. Mit diesen Informationen („Ist-/Sollwert Abweichung“) wurde die Analyse mittels der Algorithmen  $P$  und  $B$  erneut gestartet. Nach der Analyse gehören folgende Komponenten zur Verdächtigungsliste:

- *ALWR-Steuergerät, Leitung Achssensor hinten, Achssensor hinten, Gestänge Achssensor hinten.*

Die Komponenten, die sich nicht in der Verdächtigungsliste befanden, wurden ausgeblendet (Abbildung 4.7). Zur Gruppierung beziehungsweise Priorisierung der verdächtigen Komponenten wurde der Algorithmus  $S$  erneut ausgeführt. Als Ergebnis entstand eine Liste mit folgenden Einträgen:

- *Gestänge Achssensor hinten,*
- *Leitung Achssensor hinten,*

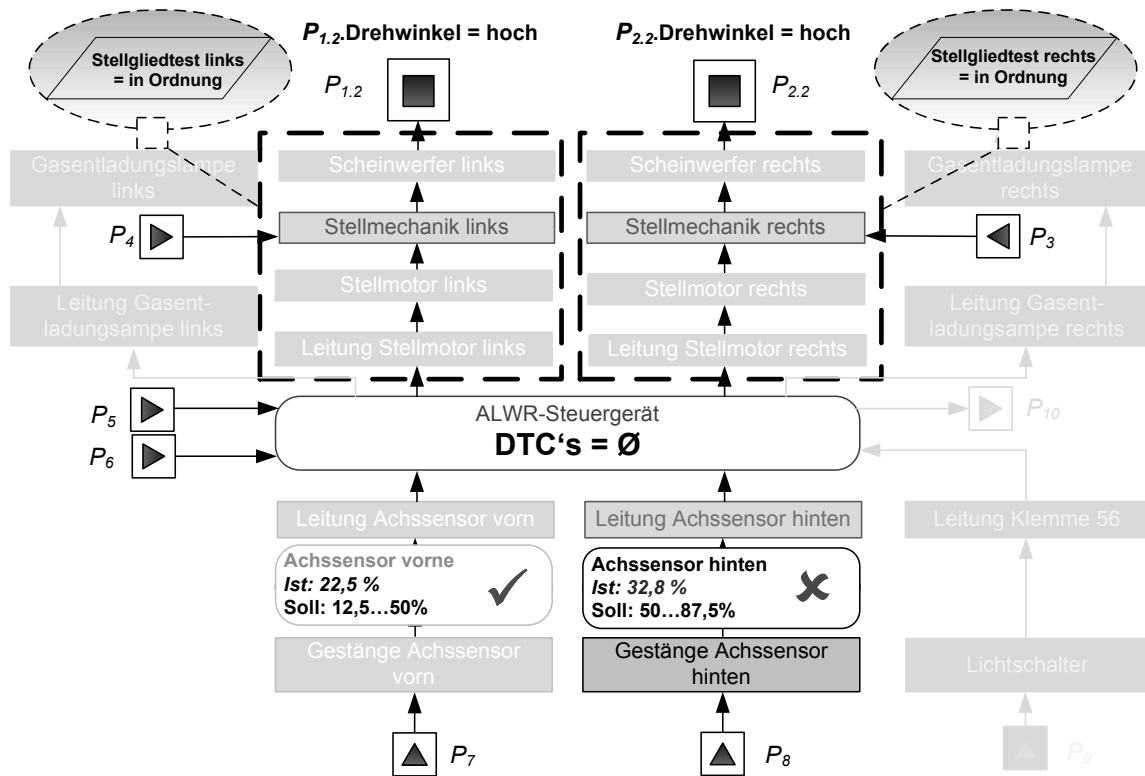
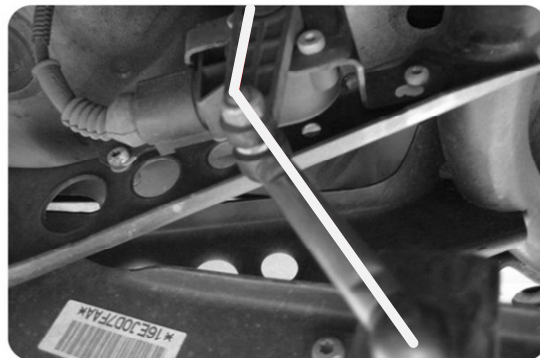


Abbildung 4.7: Auslesen der Sensor-Istwerte

- *Achssensor hinten,*
- *ALWR-Steuergerät.*

Abbildung 4.8: Istzustand der Einbaulage der Komponente *Gestänge Achssensor hinten*

Zur Prüfung der Komponenten *Gestänge Achssensor hinten* (Abbildung 4.8) wurde die Diagnose- und Prüfungsinformation der Komponente *Gestänge Achssensor hinten* (Abbildung 4.9) abgerufen.

Es mussten drei Prüfschritte durchgeführt werden:

- *Befestigung der Komponente mit dem Achssensor,*
- *Leichtgängigkeit der Komponente,*

**Funktion: Gestänge hinten**

Die Komponente **Gestänge** überträgt die Federwegänderung des Fahrzeugs bei Beladung vom Querlenker auf die Komponente **Achssensor**.

Ebenso wird die Neigungsänderung des Fahrzeugs bei Beschleunigungs- und Verzögerungsvorgängen auf die Komponente **Achssensor** übertragen.

**Warnhinweis: Gestänge hinten**

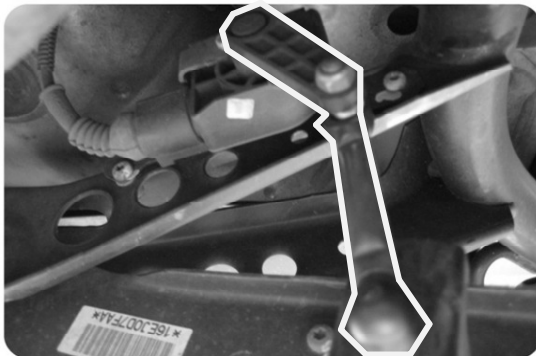
Die Komponente **Gestänge** kann nur bei angehobenem Fahrzeug ausgebaut werden. Die Hebeeinrichtungen müssen für die Last geeignet und nachweislich in sicherem Zustand sein.

Nach Austausch der Komponente **Gestänge** muss die Funktion **Grundeinstellung Leuchtweitenregulierung** durchgeführt werden.

**Prüfung: Gestänge hinten**

Korrekte Befestigung der Komponente **Achssensor** sowie die Leichtigängigkeit der Komponente **Gestänge** kontrollieren.

Die Komponente **Gestänge** darf nicht deformiert sein und die Einbaulage muss der in der Abbildung exakt entsprechen. Die Federwegänderung muss von der Komponente **Gestänge** spielfrei auf die Komponente **Achssensor** übertragen werden.

**Einbaulage: Gestänge hinten****Arbeitswerte: Gestänge hinten**

Positionsnummer:	1
Positionstyp:	AW
Artikelnummer:	0 307 859 310
Bezeichnung:	Gestänge Achssensor hinten
Menge:	0,20
Mehrwertsteuer:	19,00%
Rabatt:	0,00%
Stundenver. Satz:	40,00 Euro
Positionspreis:	8,00 Euro

Abbildung 4.9: Komponentenbezogenes Diagnosewissen



- *Prüfung der Einbaulage.*

Die Prüfung lieferte das Ergebnis „falsche Lage Gestänge Achssensor hinten“. Als Reparaturmaßnahme wurde die Komponente *Gestänge Achssensor hinten* wieder in die richtige Lage gebracht. Um den Diagnosevorgang abzuschließen, wurde nach der Reparatur die Existenz der Kundenbeanstandung „Leuchtweite zu hoch“ überprüft. Nach der Feststellung, dass keine Kundenbeanstandung bezüglich der Leuchtweiteregulierung mehr existierte, war der Diagnose- und Reparaturvorgang beendet.

## 5 Wissenserhebung mit DMB-System

Der Wissenserhebungsansatz, der in Kapitel 3 beschrieben ist, wurde in dieser Arbeit umgesetzt. Die entwickelte Modellierungapplikation wird im Folgenden als DMB-System (Diagnostic Model Builder System) bezeichnet. Im ersten Abschnitt dieses Kapitels wird auf den Aufbau von DMB-System und auf die Aktivitäten zur Erstellung eines Diagnosemodells eingegangen. Im zweiten Abschnitt werden die verschiedenen Rollen bei der Modellerstellung definiert. Anschließend wird die entwickelte Diagnoselaufzeitumgebung aus Benutzersicht kurz beschrieben. Abschließend wird die Qualitätssicherung in Form des Modelltests vorgestellt.

### 5.1 Modellerstellungsprozess, Rollen und Systemarchitektur

#### 5.1.1 Modellerstellungsprozess

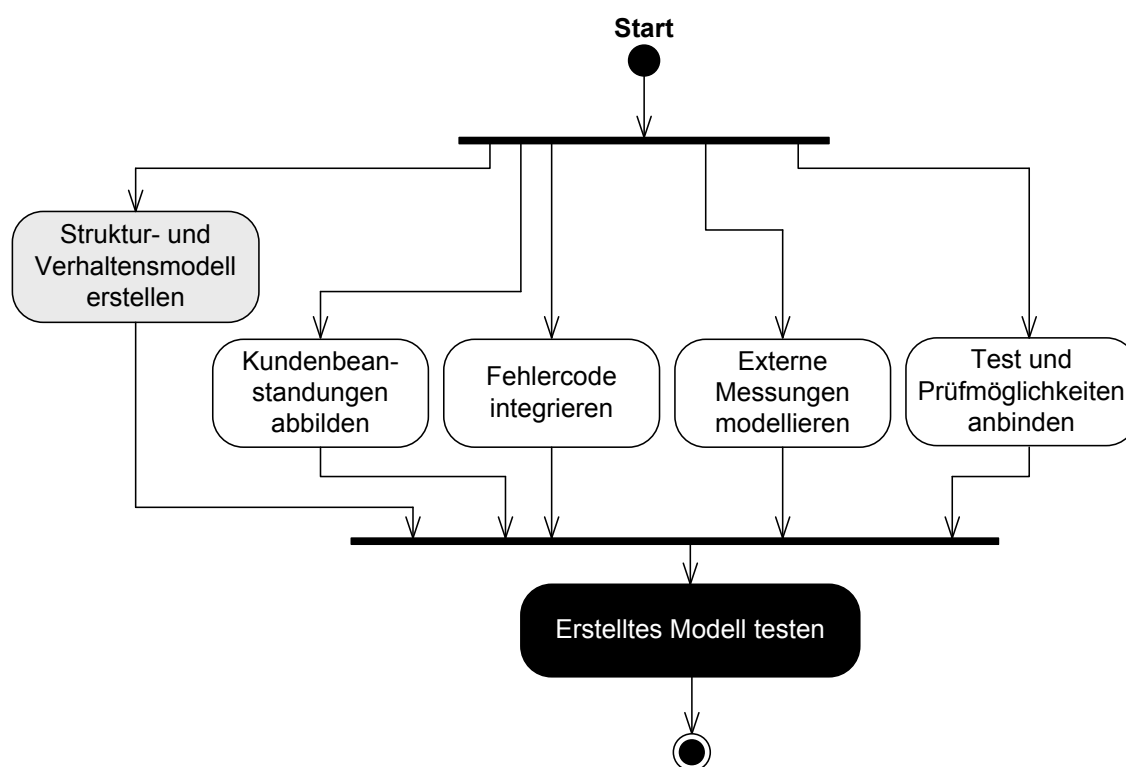


Abbildung 5.1: Die sechs Phasen bei der Modellierung eines Systems

Aufgrund der Anforderungsanalyse und des Konzepts zur Modellierung des Diagnosemodells kann der gesamte Modellierungsprozess in sechs Phasen aufgeteilt werden (Abbildung 5.1). Nachfolgend werden die einzelnen Phasen der Modellierung vorgestellt. Zunächst erfolgt in der ersten Phase eine Darstellung der Systemgrenze, der systemeigenen Komponenten, der Subsysteme sowie deren Verbindungen (Abbildung 5.2). Darüber hinaus gilt es, die systemeigenen

Komponenten in Sensoren, Aktoren, Steuergeräte und passive Komponenten zu unterscheiden.

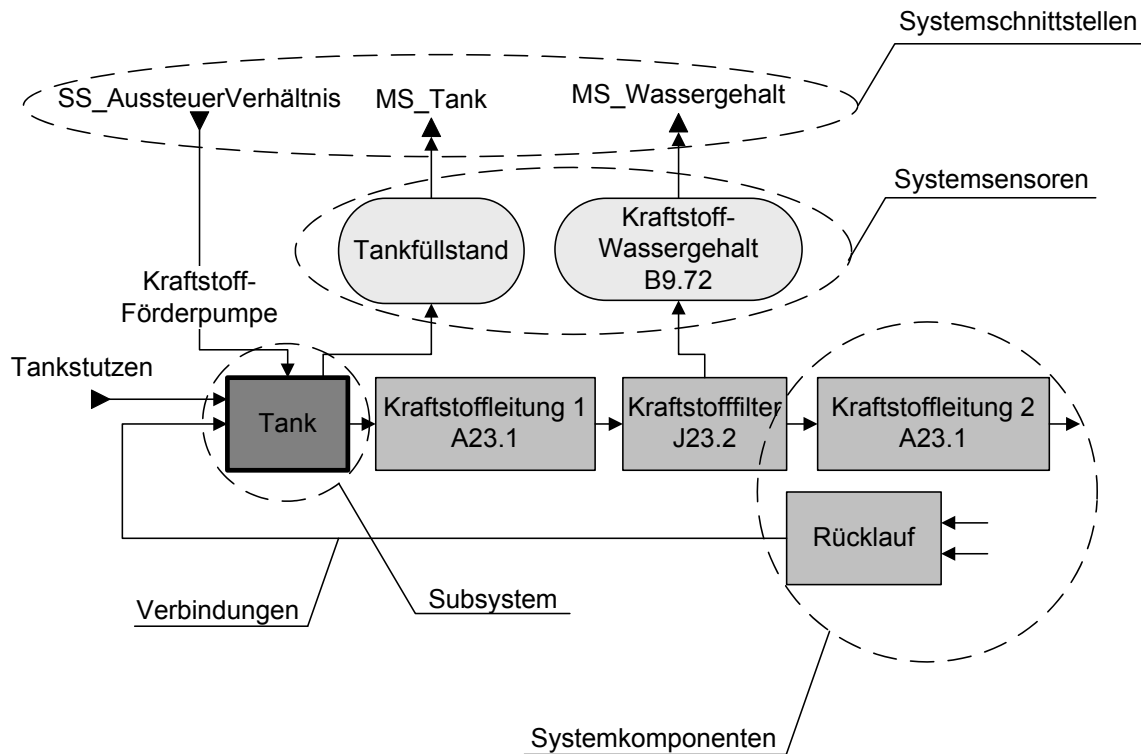


Abbildung 5.2: Erste Phase: Struktur- und Verhaltensmodellierung

Um die Hauptaktivitäten bei der Erstellung eines Struktur- und Verhaltensmodells zu beschreiben, wird in Abbildung 5.3 eine mögliche Prozessvariante, die bei diesem Erstellungsvorgang auftreten kann, skizziert. Die einzelnen, dargestellten Schritte sind im Folgenden kurz beschrieben:

- „Struktur- und Verhaltensmodell anlegen“: Um eine Diagnose durchzuführen, muss ein Struktur- und Verhaltensmodell angelegt werden.
- „Modell aus Bibliothek importieren“: In diesem Schritt können einzelne Modelle aus der Modellbibliothek ausgewählt und importiert werden.
- „Komponentenschablone importieren“: Komponenten, die nicht in der Modellbibliothek enthalten sind, müssen modelliert werden. Dazu wird die vordefinierte Schablone aus der Toolbar per Mausklick in das Modell importiert.
- „Komponentenschnittstellen festlegen“: Um ein Komponentenmodell zu erstellen, muss der Modellierer die Schnittstellen definieren. Es werden nicht nur Art und Richtung der Schnittstelle festgelegt, sondern auch die Transportobjekte und deren Attribute, welche sich in dieser Schnittstelle befinden, spezifiziert.
- „Variablen festlegen“: Dieser Schritt beinhaltet die Definition aller internen Funktions-, Zustands- und Fehlervariablen der Komponente.

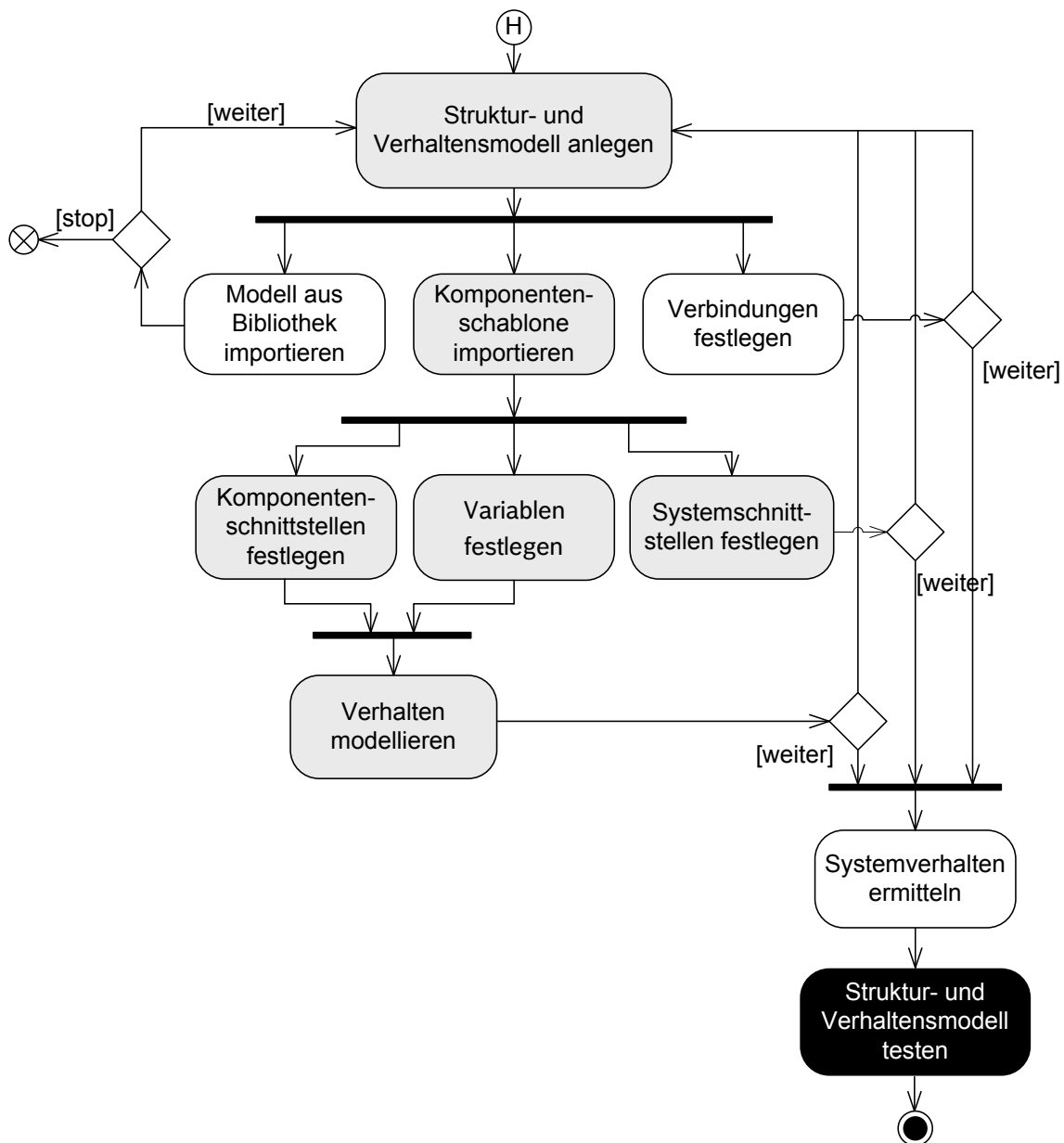


Abbildung 5.3: Detaillierte Aktivitäten zur Erstellung des Struktur- und Verhaltensmodells

- „Verhalten modellieren“: Nach der Definition der Schnittstellen und der internen Variablen kann der Modellierer das Verhalten der Komponente in Form einer oder mehrerer qualitativer relationaler Gleichungssysteme modellieren. Aus diesen Gleichungssystemen wird die Verhaltenstabelle automatisch erzeugt.
- „Systemschnittstellen festlegen“: In diesem Schritt wird die Systemgrenze definiert.
- „Komponentenverbindungen festlegen“: Hier werden die Interaktionen und Beziehungen zwischen Komponenten modelliert.
- „Systemverhalten ermitteln“: Die Modellierung des Gesamtverhaltens des Systems kann auf zwei Weisen abgebildet werden. Entweder wird das Systemverhalten aus dem Verhalten der systemeigenen Komponenten und aus der Topologie des Systems generiert beziehungsweise abgeleitet oder das Gesamtverhalten wird manuell definiert.

- „Struktur- und Verhaltensmodell testen“: Nach der Erstellung des Modells kann durchgehend das Modell gegen die zuvor definierten Testfälle getestet werden. Wenn das Modell alle Testfälle bestanden hat, ist dieser Modellierungsschritt beendet.

Kundenbeanstandungen werden als virtuelle Komponenten in der zweiten Phase in das Modell eingebunden (Abbildung 5.4). Hierbei erfolgt ein Mapping der Beanstandungen auf physikalische Größen des Modells. Dies bedeutet, dass die Kundenbeanstandung an physikalische Größen der Systemein- und Systemausgänge gebunden wird. Zusätzlich können verdächtige Systemkomponenten, die als Ursache des Symptoms beziehungsweise der Kundenbeanstandung dank Expertenwissen bekannt sind, in Verbindung mit der zuvor abgebildeten Kundenbeanstandung gebracht werden.

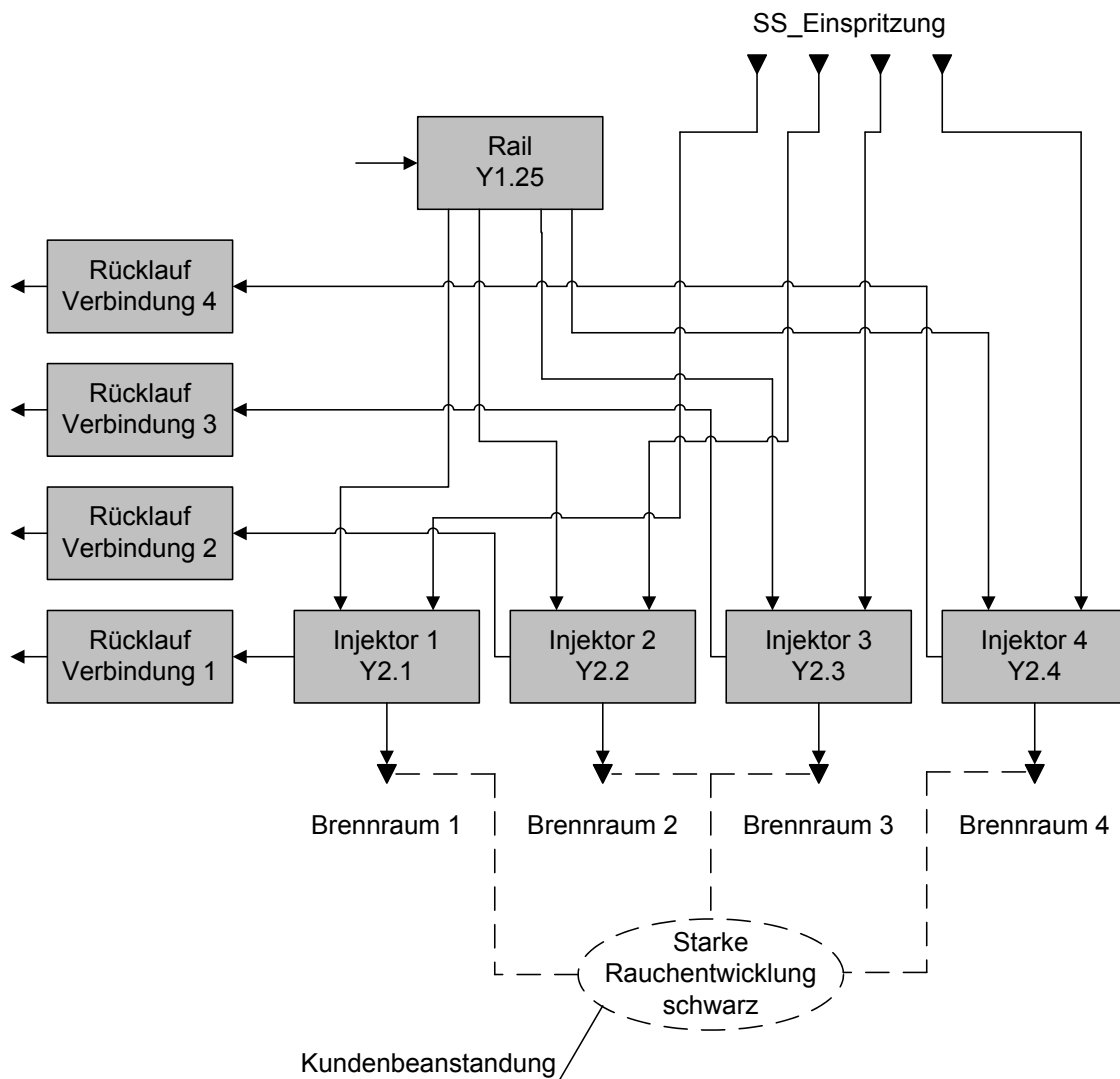


Abbildung 5.4: Zweite Phase: Modellierung der Kundenbeanstandungen

Die Abbildung der Informationen der Onboard-Diagnose als virtuelle Komponenten erfolgt in der dritten Phase (Abbildung 5.5). Dabei gilt es, wie in Kapitel 3 beschrieben, zwischen zwei Arten von Fehlerseichereinträgen zu unterscheiden. Zum einen existieren Fehlercodes, die direkt auf Komponenten zeigen. Zum anderen können die Fehlercodes auf physikalische Größen (zum Beispiel „Kraftstoffmenge zu niedrig“) hinweisen. Darüber hinaus erfolgt eine

Anbindung der virtuellen Komponenten an die realen Komponenten mit Hilfe von virtuellen Verbindungen.

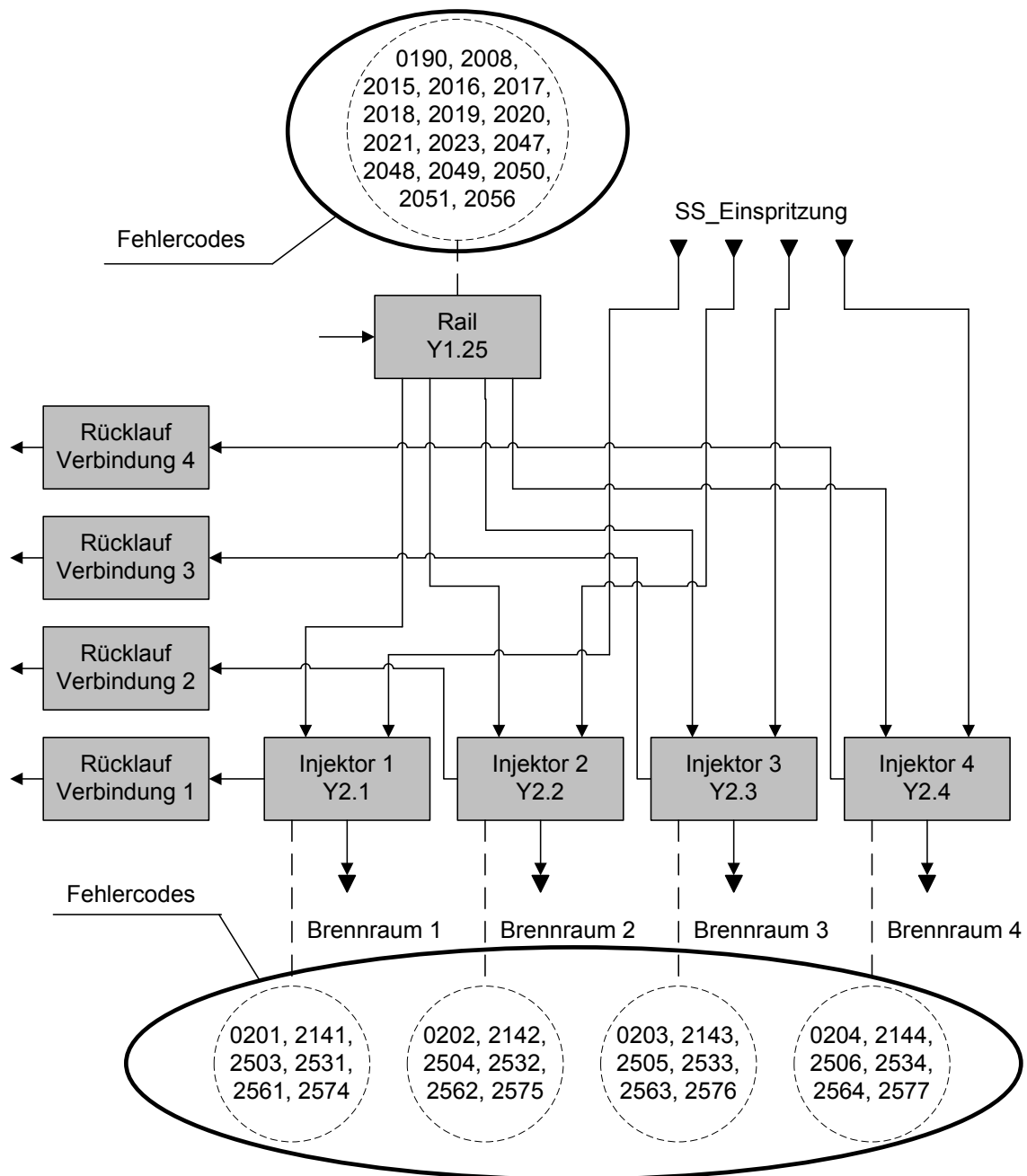


Abbildung 5.5: Dritte Phase: Fehlercodes in das Modell integrieren

In der vierten Phase werden systemexterne Komponenten wie zum Beispiel externe Messungen modelliert und in das Modell integriert. Diese Phase der Modellierung ist in Abbildung 5.6 dargestellt. Hierbei sind die virtuellen Verbindungen getrennt nach systemexternen und -eigenen Komponenten anzulegen.

Die Test- beziehungsweise Prüfmöglichkeiten der systemeigenen Komponenten- und Stellgliedtests werden in der fünften Phase dem Modell hinzugefügt (Abbildung 5.7). Dabei wird der Test der einzelnen Komponente direkt an die Komponente gebunden.

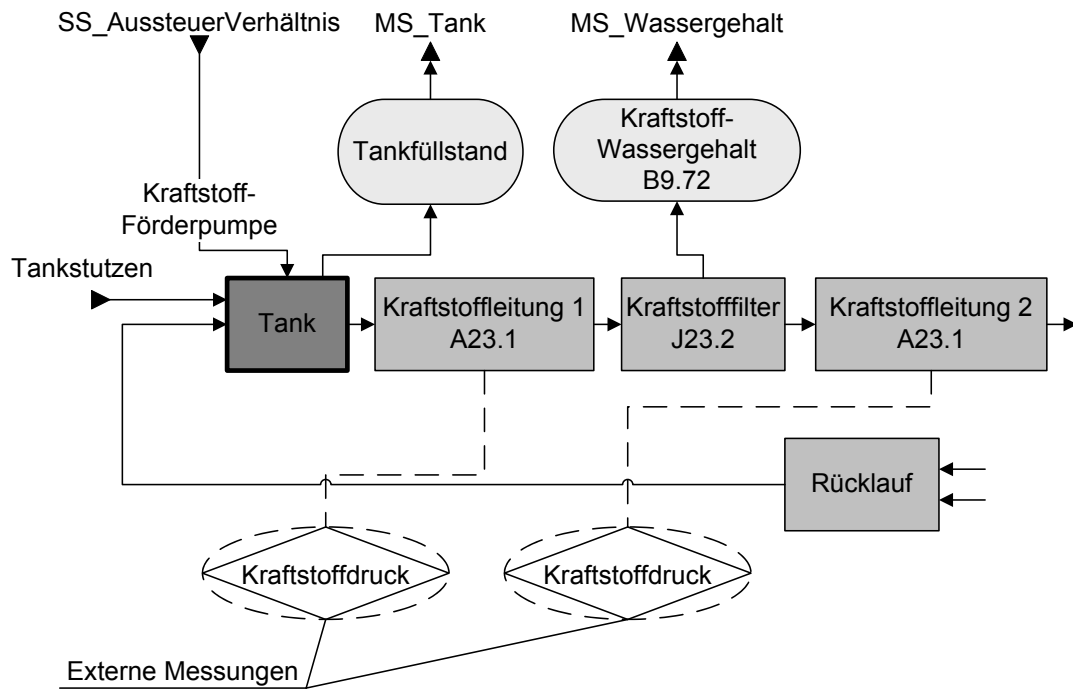


Abbildung 5.6: Vierte Phase: Abbildung der externen Messungen

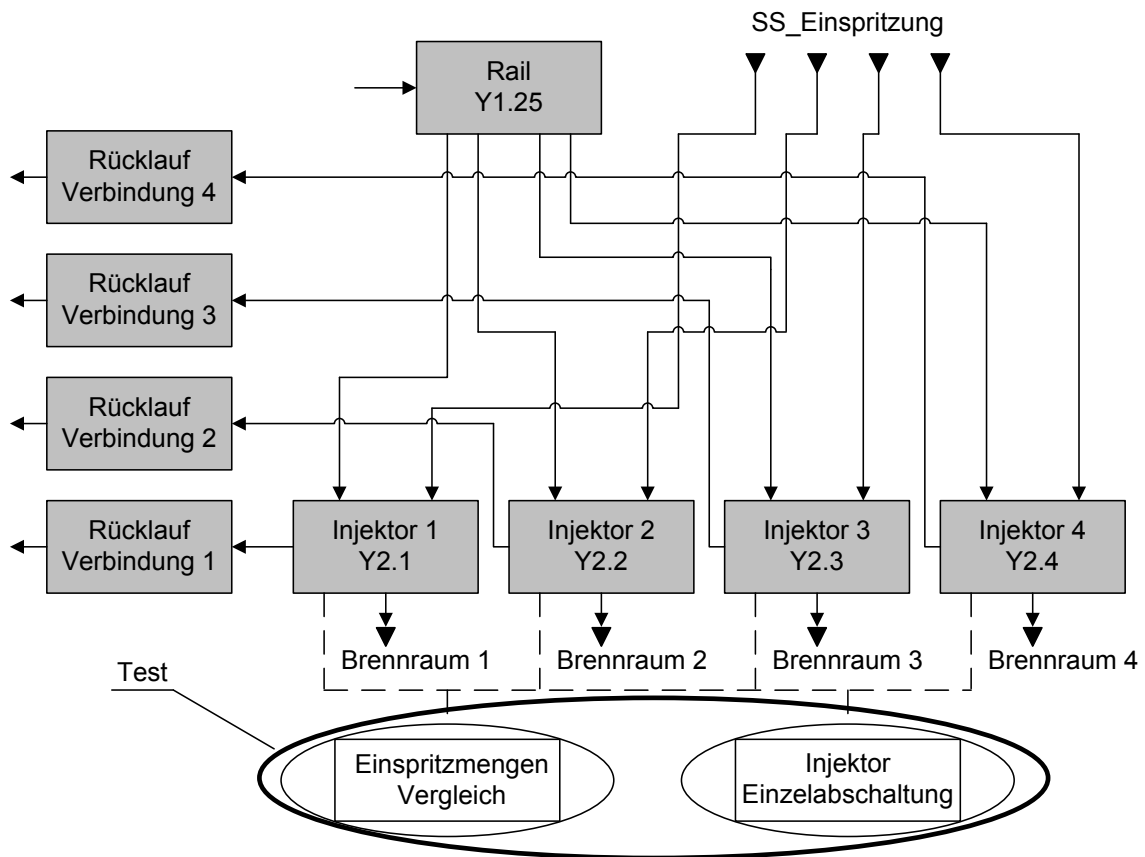


Abbildung 5.7: Fünfte Phase: Komponenten- und Stellgliedtests anbinden

Demgegenüber muss der Stellgliedtest an eine Komponentengruppe gebunden werden. Die Integration des Komponententests kann durch eine Datenbankabfrage anhand des Identifizierungscode der Komponente automatisch erfolgen. Die Anbindung beim Stellgliedtest funktioniert analog. Der Unterschied ist, dass nicht ein Identifizierungscode einer Komponente abgefragt wird, sondern alle Identifizierungscode abgeglichen werden müssen.

Dabei gilt es zu berücksichtigen, dass bei den Phasen 2 - 5 diese Reihenfolge nicht zwingend eingehalten werden muss. Obendrein kann, je nach verfügbaren Informationen, die zweite, dritte oder vierte Phase auch weggelassen werden. Dies führt allerdings unmittelbar zu einer geringeren Diagnoseleistung, da das Werkstattpersonal so nicht mit allen vorhandenen Symptomen und diagnoserelevanten Informationen arbeiten kann. Um die zweckgebundenen Diagnoseanforderungen des erstellten Modells zu überprüfen, wird das Modell im sechsten Schritt gegen Testfälle getestet. Auf die Erstellung und die Ausführung der Testfälle sowie die Auswertung der Testergebnisse wird am Ende dieses Kapitell detailliert eingegangen. Auf Grund der unterschiedlichen Wissens- und Informationsanforderungen in verschiedenen Phasen der Diagnosemodellerstellung ist es möglich, das Diagnosewissen in einem vollständigen Diagnosemodell mit mehreren Personen zu erheben und danach zusammenfügen. Die verschiedenen Rollen in der Wissenserhebung werden im nächsten Abschnitt beschrieben.

### 5.1.2 Rollen bei der Modellerstellung

Mit der Aufteilung des Erstellungsprozesses und der Möglichkeit, einige Phasen parallel zu bearbeiten, können verschiedene Rollen von Beteiligten übernommen werden, um das Diagnosemodell zu vervollständigen. In Abbildung 5.8 ist eine mögliche Rollenverteilung, die bei der Modellerstellung auftreten kann, dargestellt. Die Beteiligten können in folgenden Rollen auftreten:

- Erste Rolle (Person 1): Datenbankbetreuer - verantwortlich für die Datenverwaltung in den verschiedenen Datenbanken wie Modellbibliothek-, Basisparameter- und Optimierungsparameterdatenbank.
- Zweite Rolle (Person 2): Modellierer beziehungsweise Modellverantwortlicher - erstellt das Struktur- und Verhaltensmodell und ist verantwortlich für die Betreuung und Pflege des erstellten Modells.
- Dritte Rolle (Person 3): Diagnoseparameterbetreuer - ermittelt und pflegt die Sollwerttabelle, Inhalte und Interpretationsmöglichkeiten der Fehlercodes, die Testmöglichkeiten der systemeigenen Komponenten sowie die Stellgliedtests. Damit ist seine Zuständigkeit die Beschaffung der Basisparameter.
- Vierte Rolle (Person 4): Optimierungsdatenbetreuer - sammelt das Wissen über Ausfallwahrscheinlichkeit, Test- und Prüfkosten der Komponenten, Subsysteme und Systeme.
- Fünfte Rolle (Person 5): Testfallspezifizierer - definiert die Testfälle für das Basisdiagnosemodell und das Diagnosemodell.
- Sechste Rolle (Person 6): Tester - nutzt das erstellte Modell und die zuvor definierten Testfälle unter Verwendung der entwickelten Modellüberprüfungsapplikation, um Testprotokolle zu erzeugen.



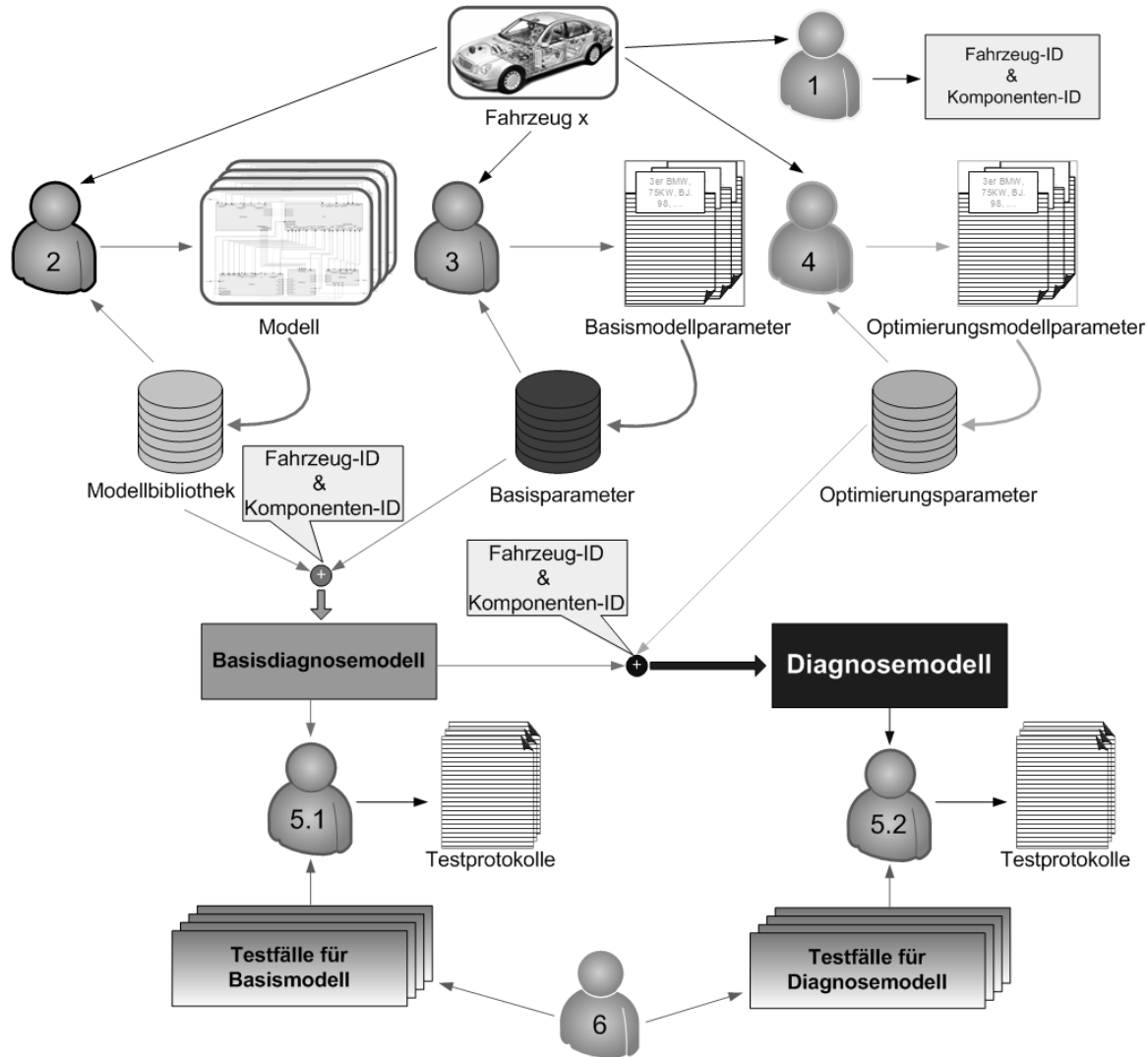


Abbildung 5.8: Rollen der Beteiligten bei der Modellerstellung

### 5.1.3 Systemarchitektur DMB-System

Ausgehend von verschiedenen Anforderungen an DMB-System, die im Kapitel 2 beschrieben wurden, von dem im Kapitel 3 entwickelten Lösungskonzept und dem oben beschriebenen Modellerstellungsprozess sowie der Aufteilung der unterschiedlichen Rollen bei der Modellerstellung kann eine Systemarchitektur von DMB-System festgelegt werden. In diesem Abschnitt wird die Architektur des entwickelten Systems erläutert. Eine detaillierte Beschreibung der Systemarchitektur ist bedingt durch die umfangreiche Funktionalität des Systems im Rahmen dieser Arbeit nicht möglich. Nachfolgend werden lediglich die Hauptsegmente und die wichtigsten Systemschnittstellen erläutert.

In Abbildung 5.9 wird die Struktur des Systemaufbaus in der ersten Ebene der Dekomposition dargestellt. Dabei ist DMB-System das entwickelte System. Um ein Diagnosemodell erstellen zu können, greift DMB-System auf vier verschiedene Datenbanken zu:

- Die Identifizierungsdatenbank enthält die Identifikationsinformationen, die für die Zusammenführung des Diagnosewissens entsprechend der oben beschriebenen Modeller-

stellungsphasen benötigt werden.

- Die Wiederverwendung des Diagnose- beziehungsweise des Modellierungswissens wird unter Verwendung der Modellbibliothek realisiert. Eine Datenbank, die ein Content-Management-System (CMS) besitzt, übernimmt serverseitig die Verwaltung der Modellbibliothek. Clientseitig unterstützt das System die Erzeugung des neuen Diagnosemodells durch Bereitstellung des existierenden Diagnosewissens. DMB-System beinhaltet zusätzliche Funktionen, um das Modell in der Bibliothek zu verwalten, wenn die Rolle des Benutzers zugelassen ist.
- Die Testfalldatenbank liefert die Testfälle für die Modellprüfung. Die Testfälle werden durch Testfallspezifizierer mittels einer Testplattform von DMB-System definiert. Die Testergebnisse und deren Auswertung werden archiviert und an die Testfalldatenbank zurückgeschickt.
- Wenn ein Diagnosemodell fertig gestellt und geprüft ist, kann der Nutzer mittels einer Schnittstelle von DMB-System das Modell in die Modelldatenbank hochladen. Diese Modelldatenbank verfügt über ein Enterprise-Content-Management-System (ECMS), das verschiedene Aufgaben wie zum Beispiel Verwaltung, Speicherung, Archivierung und Bereitstellung des Diagnosewissens in Form von Modellen übernimmt.

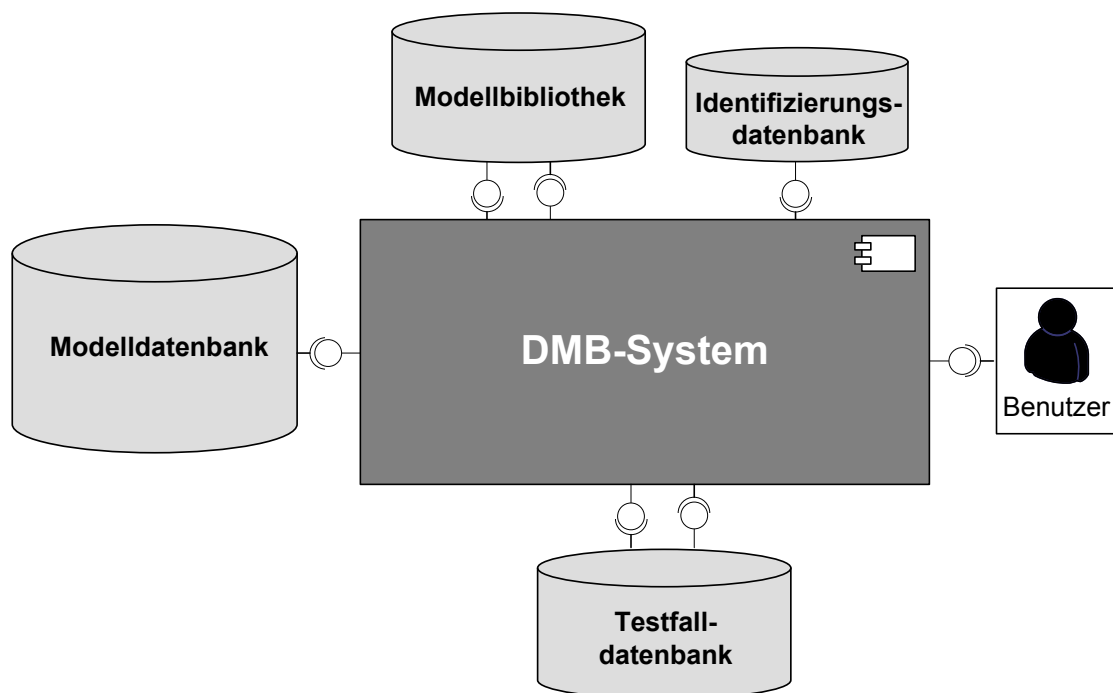


Abbildung 5.9: DMB-System und seine Umgebung

Die Zerlegung von DMB-System ist in Abbildung 5.10 dargestellt. DMB-System besitzt insgesamt neun Hauptsegmente beziehungsweise Pakete:

- Knowledge Reuse: Die Wiederverwendbarkeit des Diagnosemodellwissens wird mittels dieses Pakets unterstützt. Hierbei existiert eine Reihe von Funktionen, wie zum Beispiel:
  - Import des Komponenten- beziehungsweise Subsystemmodells

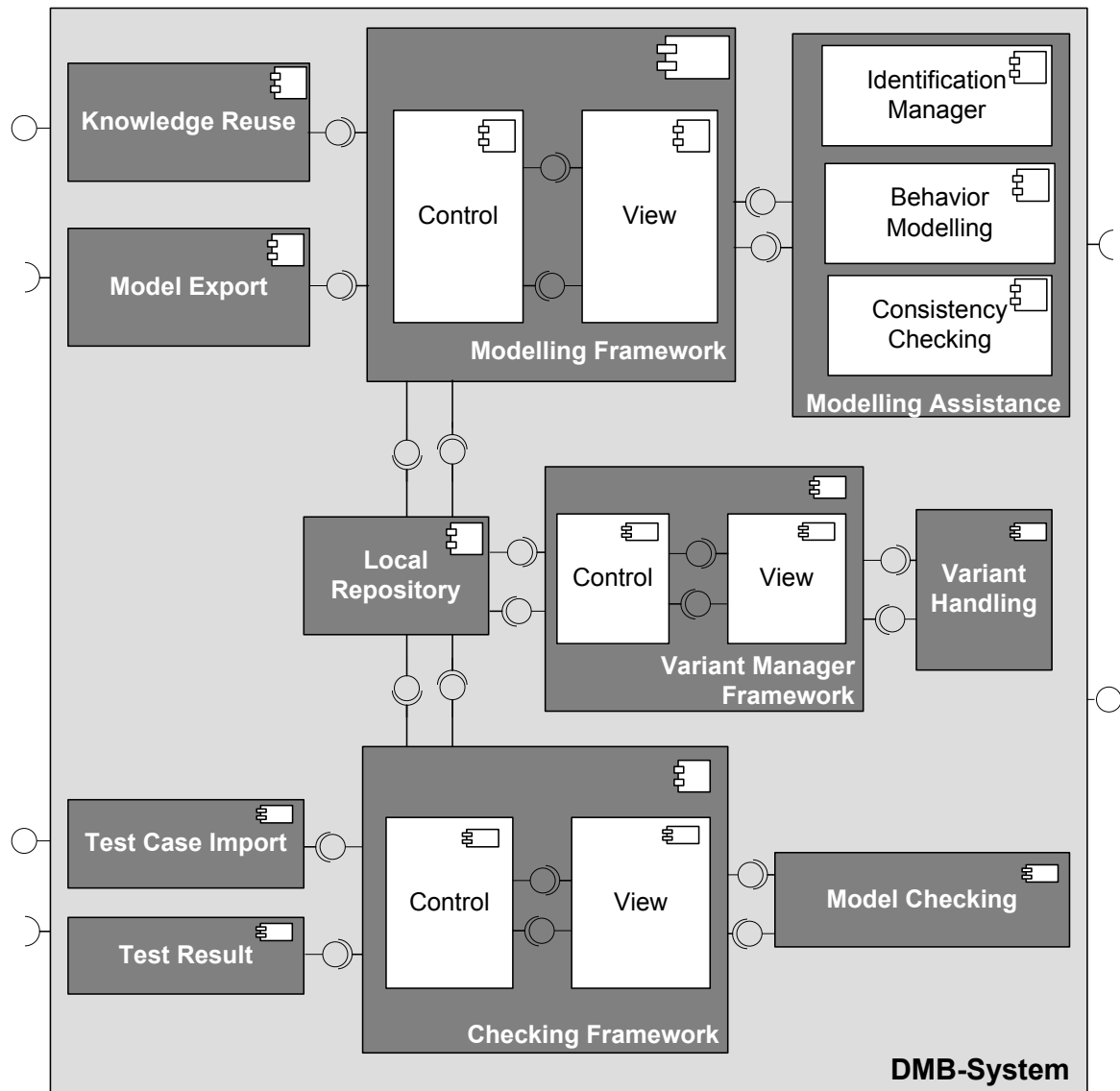


Abbildung 5.10: Dekomposition von DMB-System in Segmente

- Spezialisierung des vorhandenen Komponentenmodells
- Automatische Erzeugung des Komponentenverhaltens aus dem Simulink-Modell und Import in das Diagnosemodell.
- **Model Export:** Das erstellte Modell wird mittels dieses Pakets in die Modelldatenbank hochgeladen. Das Paket verfügt über eine zusätzliche Extraktionsfunktion, die einen Teil des Modells, das für verwandte Systeme gültig ist, extrahiert und für die Modellbibliothek bereitstellt.
- **Modelling Framework:** Die Architektur dieses Segments basiert auf dem Model-View-Controller-Muster. Da das Segment „Datenmodell“ die Daten und die Geschäftslogik enthält, die nicht nur für das Paket „Modelling Framework“ sondern auch für das Paket „Checking Framework“ relevant sind, besteht das Paket „Modelling Framework“ aus zwei Segmenten:
  - **View:** Das Paket präsentiert die Daten des zu erstellenden Diagnosemodells und

nimmt die Benutzereingabe entgegen.

- Control: Die Verwaltung der Präsentationsschichten und der Auswertung beziehungsweise Weiterverarbeitung der Interaktionen zwischen dem Benutzer und Modelling Framework sind die Aufgaben dieses Pakets. Dabei ist die Manipulation der Daten abhängig von den Benutzerinteraktionen und der vordefinierten Geschäftslogik.
- Modelling Assistance: Dieses Segment unterstützt die Zuordnung des Diagnosewissens und die Modellierung des physikalischen Verhaltens und besteht aus drei Hauptpaketen:
  - Identification Manager: Da das Diagnosewissen aus verschiedenen Quellen stammt und durch unterschiedliche Personen zusammengestellt wird, stellt die Verwaltung des Diagnosewissens neue Anforderung an die Wissenszusammenführung. Die Methoden, die in diesem Paket implementiert sind, unterstützen die Individualisierung und Konsistenzerhaltung von komponenten-, system- beziehungsweise fahrzeugtypenbezogenen Diagnoseinformationen, zum Beispiel Sollwertbereichen oder herstellerspezifischen Fehlercodes.
  - Behavior Modelling: Dieses Paket wurde entwickelt, um die Erzeugung der Verhaltenstabelle aus qualitativen Relationsbeschreibungen zu automatisieren.
  - Consistency Checking: Durch die Wiederverwendung des Modellwissens in Form von Modellbibliotheken und die Möglichkeit der Zusammenführung der Teilmodelle entsteht möglicherweise eine Inkonsistenz, wie zum Beispiel bei der Namensgebung der Komponenten im Diagnosemodell. Dieses Paket überprüft die Konsistenz des Modells und gibt Rückmeldung an den Modellierer.
- Local Repository: Wenn das Modell in Bearbeitung ist und noch nicht in die Modelldatenbank hochgeladen wurde, erfolgt die Erhaltung der Daten des Modells lokal. Dieses Paket enthält die Modelldaten und die Geschäftslogik für die Modellierungs- und die Modellprüfungsvorgänge.
- Test Case Import: Die erstellten beziehungsweise erzeugten Testfälle aus der Werkstattinformationsrückführung können mit den bereitgestellten Funktionen dieses Pakets in das Checking Framework importiert werden.
- Checking Framework: Die Architektur dieses Segments ist analog zur Architektur des Segments Modelling Framework und enthält zwei Teilsegmente:
  - View: Das Paket präsentiert nicht nur die Daten des erstellten Diagnosemodells, sondern auch die Daten der definierten Testfälle. Dieses Segment nimmt hauptsächlich die Benutzereingabe bezüglich der Testdurchführung entgegen.
  - Control: Das Paket verwaltet die Präsentation der Modelldaten und der Daten der Testfälle. Die Manipulation der Daten ist ausschließlich auf Testfälle beschränkt.
- Model Checking: Die Ausführung der erstellten und importierten Testfälle werden durch Funktionen dieses Segments realisiert. Die entstehenden Testergebnisse nach jedem Testlauf werden mit Hilfe der Methoden in diesem Segment als Testreport erstellt, visualisiert und gesichert.
- Test Result: Dieses Segment ermöglicht dem Tester, die erstellten Testreporte in der Testfalldatenbank zu archivieren.

- Variant Handling Framework: Da alle Frameworks in DMB-System auf dem Model-View-Controller-Muster basieren, besteht dieses Frameworks auch aus zwei Teilsegmenten: View und Control.
- Variant Handling: Dieses Segment beinhaltet verschiedene Algorithmen, die zum einen die Abstraktion der Struktur beziehungsweise des Verhaltens von vorhandenen Modellen ermöglichen und zum anderen den Abstand zwischen Modellen ermitteln.

## 5.2 Das entwickelte DMB-System

Im Rahmen der Realisierung wurde eine Modellierungssapplikation zur Erhebung und Verwaltung des Diagnosewissens, das für eine Durchführung der Diagnose in der freien Werkstatt erforderlich ist, entwickelt. Eine grobe Beschreibung der Funktionen der entwickelten Applikation zur Modellierung der Systemkomponente wurde in Abschnitt 1.3.3 beschrieben. Demzufolge werden in diesem Abschnitt nur das Speicherformat des erstellten Modells sowie die entwickelte Umgebung zur Qualitätssicherung und der Variantenmanager erläutert.

### 5.2.1 Speicherformat des Modells

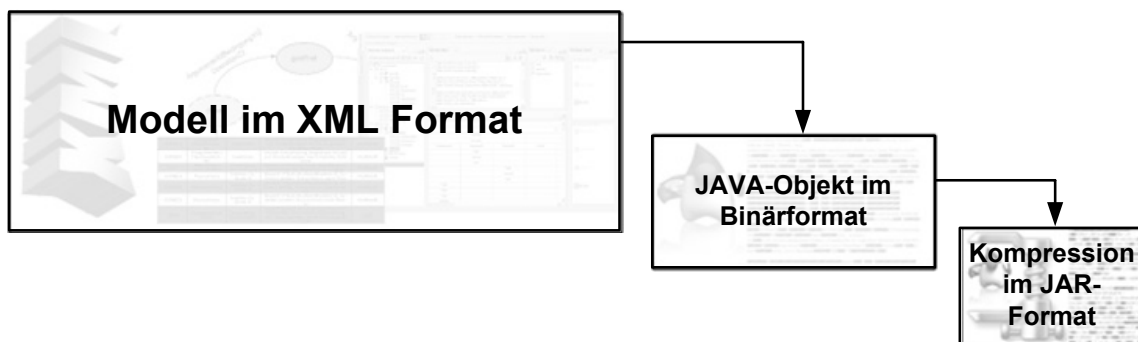


Abbildung 5.11: Ergebnisse der Reduktion des Speicherbedarfs

Der Austauschbarkeit und Flexibilität wegen wurde das mit DMB-System erstellte Modell im XML-Format gespeichert. Dieses Format führt zu einem relativ großen Speicherbedarf und längeren Ladezeiten beim Diagnosevorgang und ist daher für freie Werkstätten mit schwacher Rechnerinfrastruktur nicht geeignet. Hier ist eine Lösung zur Reduktion des Speicherbedarfs nötig. In dieser Arbeit wurde das XML-Format durch ein Binärformat mit einer Komprimierung für das Diagnoselaufzeitsystem ersetzt. Aus Tabelle 5.1 geht hervor, dass ein durchschnittlicher Komprimierungsfaktor von circa 200 erreicht wurde.

Modell	Speicherbedarf	Faktor
OpelCorsa1.4KraftstoffV0.7.mod	37.995 KB	
OpelCorsa1.4KraftstoffV0.7.obj	2.257 KB	16.8
OpelCorsa1.4KraftstoffV0.7.jar	196 KB	193.8
MB220CdiKraftstoffV1.3.mod	35.500 KB	
MB220CdiKraftstoffV1.3.obj	2.714 KB	13.1
MB220CdiKraftstoffV1.3.jar	200 KB	177.5

OpelCorsa1.4LuftV0.6.mod	31.899 KB	
OpelCorsa1.4LuftV0.6.obj	2.416 KB	13.2
OpelCorsa1.4LuftV0.6.jar	143 KB	223.1
<hr/>		
MB220CdiLuftV1.2.mod	31.431 KB	
MB220CdiLuftV1.2.obj	2.434 KB	12.9
MB220CdiLuftV1.2.jar	143 KB	219.8
<hr/>		
Durchschnittlicher Verbesserungsfaktor		203.6

Tabelle 5.1: Ergebnisse der Reduktion des Speicherbedarfs

### 5.2.2 Qualitätssicherung

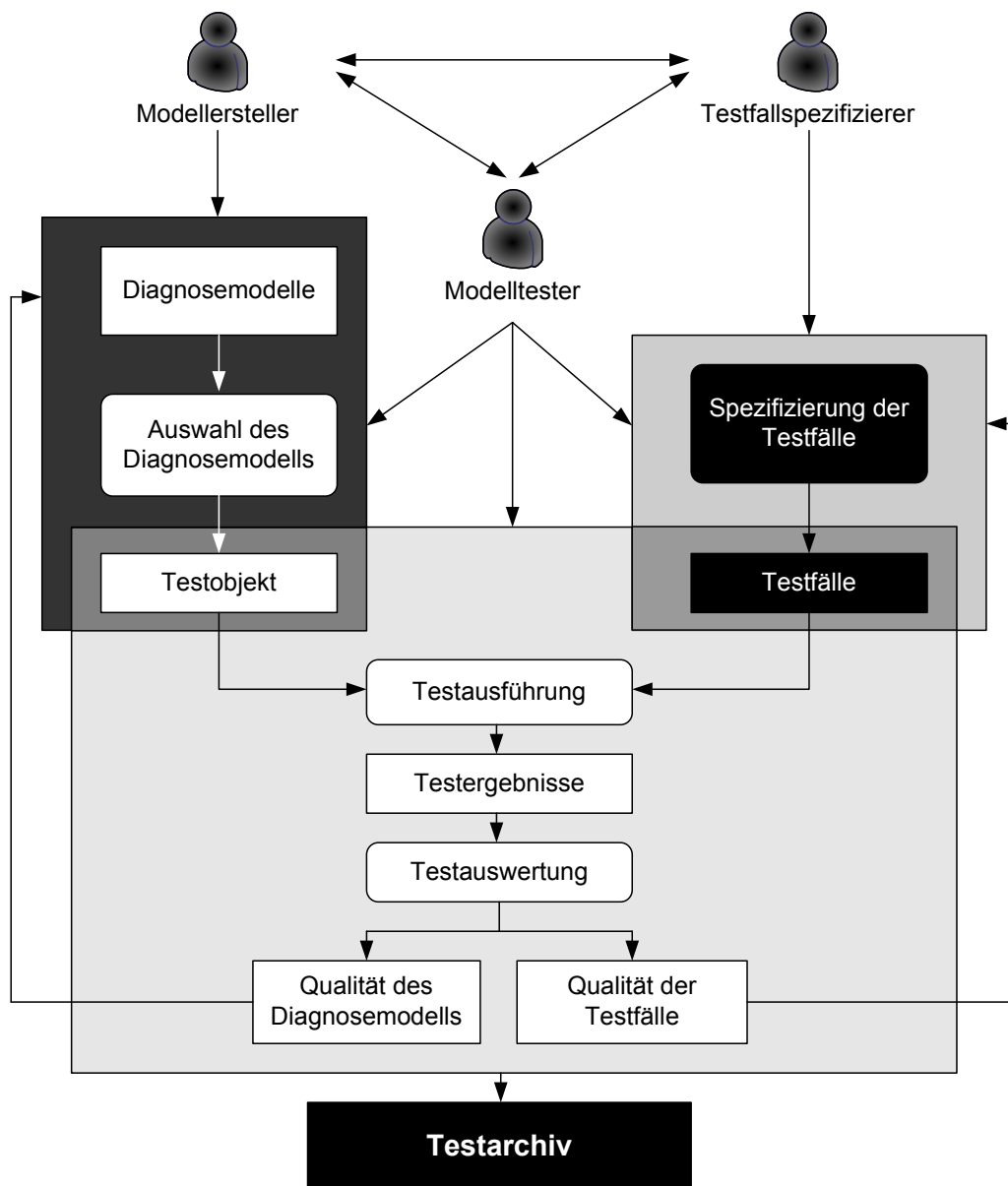


Abbildung 5.12: Testplattform und Testrollen

Für die verschiedenen Teststufen, die in 3.11 beschrieben sind, wurde in dieser Arbeit eine Testplattform entwickelt. Mit den entwickelten Funktionen unterstützt diese Plattform alle Testphasen in verschiedenen Entwicklungsständen des Modells. Für die Plattform sind drei Benutzerrollen (Abbildung 5.12) vorgesehen:

- Modellersteller: Seine Rolle ist die Erstellung des Testobjekts beziehungsweise des Diagnosemodells. Dabei kann das erstellte Modell ein Komponenten-, Teilsystem- oder Systemmodell sein.
- Testfallspezifizierer: Analog zum Standard IEEE 829 ist die Aufgabe des Testfallspezifizierers die Spezifizierung und Dokumentierung der Testfälle.
- Modelltester: Anhand der Testfälle führt der Tester mittels der entwickelten Testplattform die Überprüfung des Testobjekts durch. Die Ergebnisse der Testausführung werden vom Tester erfasst und ausgewertet.

Luftansaugrohr									
in					internal variable		out		
PortW0					Leckage	Verstoffung	PortE0		
Luft							Luft		
Druck	Massenfluss	O2	Staubgehalt	Temperatur			Druck	Massenfluss	O2
high	high				normal	high	normal	high	
high	high				high	high	normal	high	
high	high				normal	high	low	high	
high	high				high	high	low	high	
normal	high				normal	high	low	high	
normal	high				high	high	low	high	
high	high				normal	high	high	high	
high	high				high	high	high	high	
low	high				normal	high	low	high	
low	high				high	high	low	high	
high	high				high	normal	normal	high	
high	high				high	normal	low	high	
normal	high				high	normal	low	high	
high	high				high	normal	high	high	
low	high				high	normal	low	high	
normal	high				normal	high	normal	high	
normal	high				high	high	normal	high	

Abbildung 5.13: Ergebnisse des Komponententests

In dieser Arbeit wird bei der Modellerzeugung entsprechend des testgetriebenen Softwareentwicklungsansatzes ein testgetriebener Modellentwicklungsansatz unterstützt. Demzufolge wird der Komponententest vom Modellersteller selbst durchgeführt. In dieser Teststufe übernimmt der Modellersteller alle drei oben erwähnten Rollen. Beim Integrationstest sollte die Rolle zwischen Modelltester und Testfallspezifizierer getrennt werden; die Rollen des Modellerstellers und des Modelltesters können in dieser Teststufe zusammengefasst werden. In

der Systemteststufe sollten die drei Rollen komplett getrennt werden, da der Test nicht nur in einer simulierten Werkstattumgebung stattfindet, sondern auch in der realen Werkstatt durchgeführt werden kann.

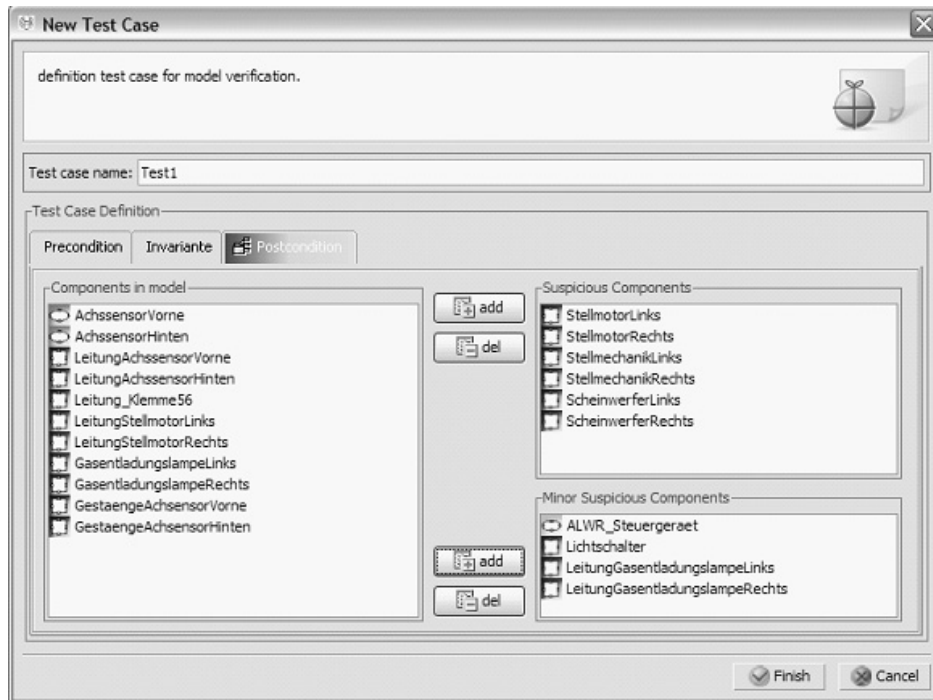


Abbildung 5.14: Maske zur Spezifizierung der Modelltestfälle

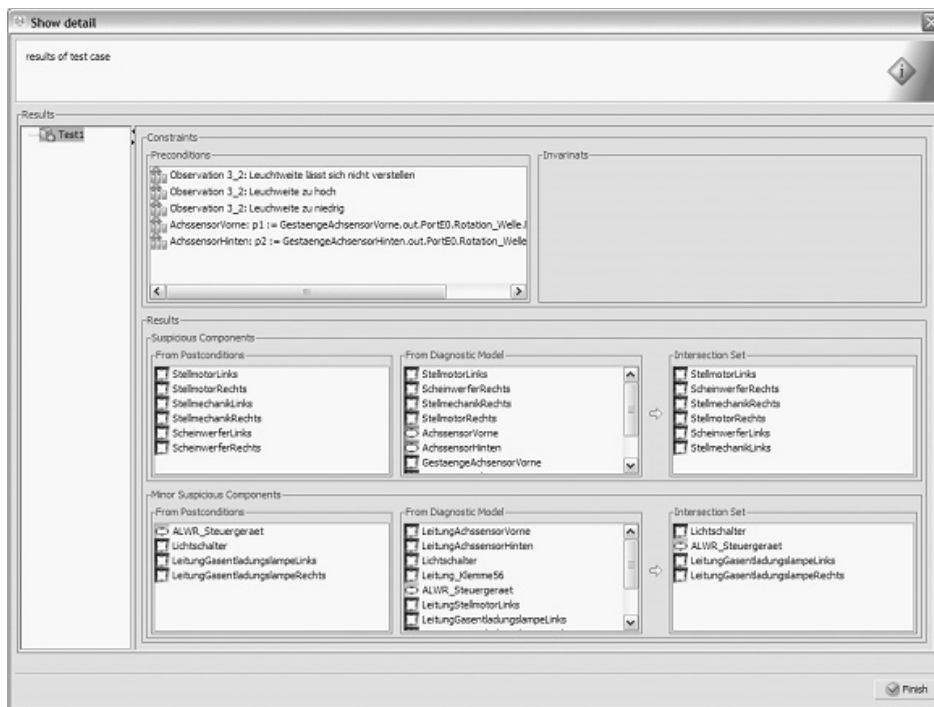


Abbildung 5.15: Maske zur Darstellung der Modelltestergebnisse

In Abbildung 5.13 sind die Ergebnisse eines Komponententests dargestellt. Hier kann der



Modellierer die Verhaltenstabelle, die aus den von ihm zuvor definierten Verhaltensrelationen generiert wurde, prüfen.

Die zweite Testphase des Modells in dieser Arbeit ist der Integrationstest. In Abbildung 5.14 ist die Maske zur Definition des Testfalls im Integrationstest abgebildet. Die letzte Testphase des Modells in dieser Arbeit ist der Systemtest. Neben der Möglichkeit, die Überprüfung des Modells direkt am Versuchsträger durchzuführen, wurde in dieser Arbeit ein Testmodul in der Testplattform entwickelt. Dieses Modul ermöglicht den Automatisierungstest in der Systemtestphase. In Abbildung 5.15 wurden die Testergebnisse in Form von Schnittmengen für den Systemtester dargestellt. Detaillierte Ergebnisse wurden in dieser Arbeit in einem Testprotokoll gespeichert.

### 5.2.3 Variantenmanager

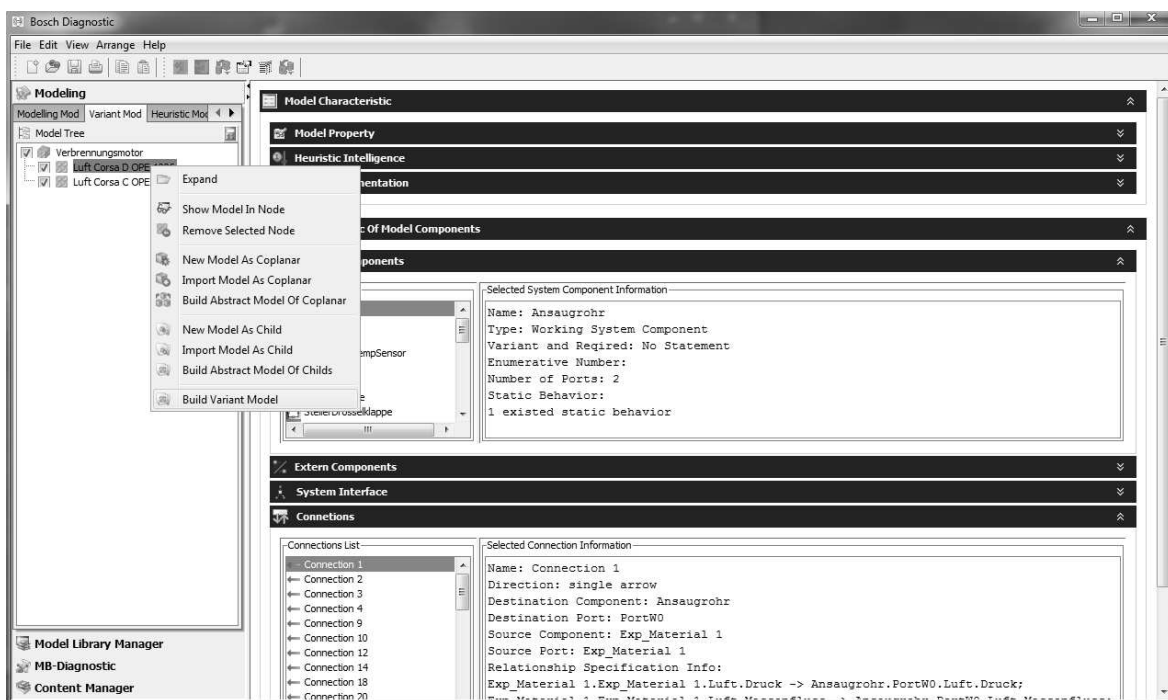


Abbildung 5.16: Variantenmanagement-Modul in DMB-System

Die theoretische Grundlage zur Behandlung der Variantenvielfalt, die in Kapitel 3 Abschnitt 11 beschrieben ist, wurde in dieser Arbeit in DMB-System in das Variantenmanagement-Modul implementiert. In Abbildung 5.16 ist ein Screenshot dieses entwickelten Moduls dargestellt. Mit der graphischen Oberfläche kann der Benutzer die erstellten Modelle abstrahieren und generalisieren. Des Weiteren kann der Nutzer neue Modelle durch Vererbung beziehungsweise Variantenbildung erzeugen.

Die Verwaltung der erstellten Modelle in der Modellbibliothek erfolgt zum einen durch Versionierung und Kategorisierung mittels existierender Datenbankfunktionen und zum anderen durch Verfolgung der Beziehungen zwischen den Modellen, welche zum Beispiel voneinander abgeleitet sein können oder über ein gemeinsames Abstraktmodell verfügen.

Die Berechnung der Distanz zwischen Modellen erleichtert verschiedene Entscheidungen bei der Analyse und Verwaltung der Varianten. Mit der Distanzberechnung können Entschei-

dungen über generische und spezielle Eigenschaften der Diagnosemodelle, wie zum Beispiel Schnittstellen, interne Variablen oder Verhalten getroffen werden. Die Grundlage der Distanzberechnung ist in Abschnitt 11, Kapitel 3 beschrieben und die Implementierung erfolgt durch das Modul *Variant Handling* in DMB-System (Abbildung 5.17).

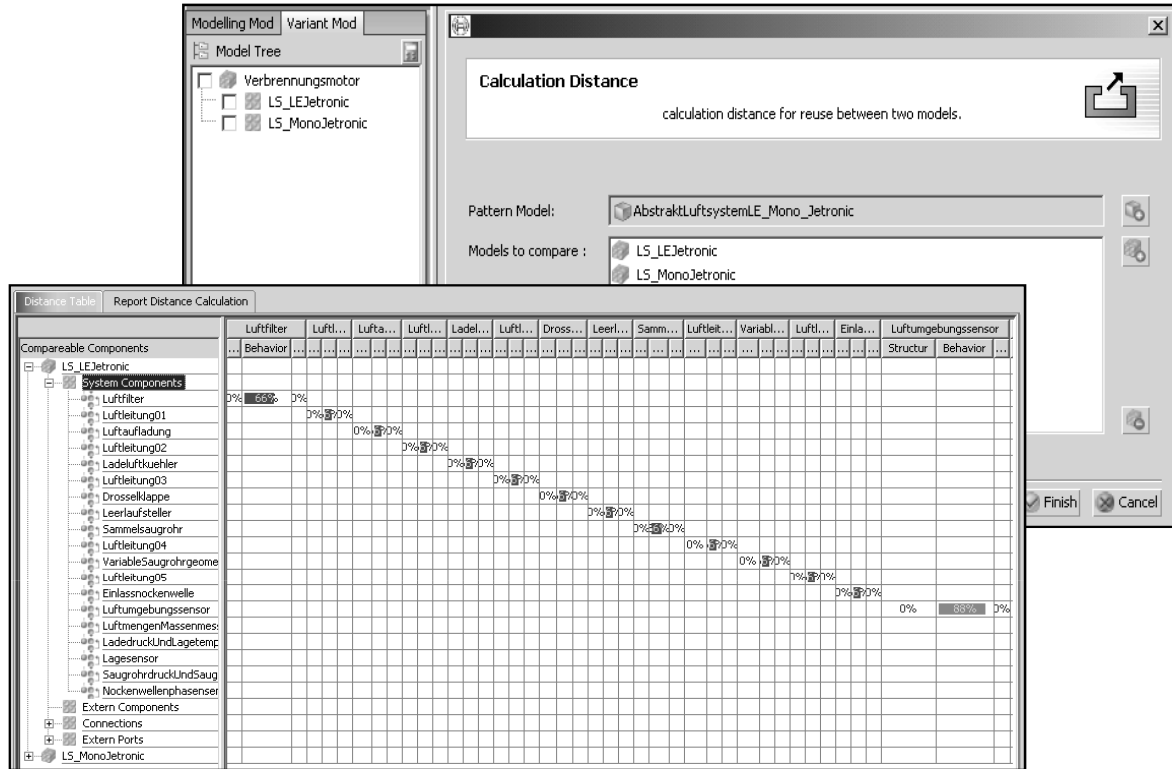


Abbildung 5.17: Berechnung der Distanz zwischen Modellen

## 6 Diagnose mit DRT-System

Der in Kapitel 4 beschriebene Diagnoseansatz wurde in dieser Arbeit umgesetzt. Das daraus entstandene Diagnoselaufzeitsystem wird im Folgenden als DRT-System (Diagnostic Runtime System) bezeichnet. Im ersten Teil dieses Kapitels wird auf den Aufbau des entwickelten DRT-Systems eingegangen. Danach wird die entwickelte Diagnoselaufzeitumgebung aus Benutzersicht beschrieben und anhand zweier kleiner Beispiele die Vorgehensweise im Diagnosevorgang erläutert. Abschließend werden verschiedene Migrationskonzepte vorgestellt und bewertet.

### 6.1 Diagnoseprozess und Systemarchitektur

Da der Diagnoseprozess direkten Einfluss auf der Entwicklung der Systemarchitektur hat, wird in diesem Abschnitt zuerst der Diagnoseprozess in der freien Werkstatt mit DRT-System erläutert. Anschließend wird die entwickelte Systemarchitektur beschrieben.

#### 6.1.1 Diagnoseprozess mit DRT-System

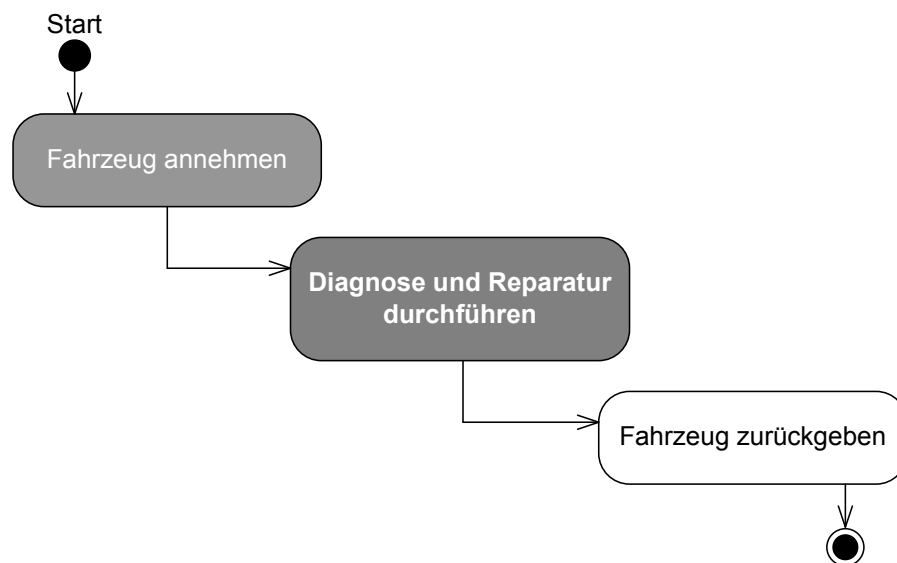


Abbildung 6.1: Die drei Hauptschritte im Werkstattdiagnoseprozess

Ein symptomorientierter Diagnoseprozess in einer Werkstatt wird ausgelöst, indem sich ein Kunde wegen einer beziehungsweise mehrerer Beanstandungen bei der Werkstatt meldet. Hierbei kann der Diagnoseprozess in der freien Werkstatt in drei Prozessabschnitte gegliedert werden (Abbildung 6.1).

Im ersten Prozessabschnitt „Fahrzeug annehmen“ werden Aktivitäten (Abbildung 6.2) durchgeführt, die für die erste Diagnoseiteration zur Erzeugung der Diagnosehypothese notwendig sind. Die Fahrzeugannahme beginnt mit der Erfassung der Kunden- und Fahrzeugdaten.

Dazu werden die Beanstandungen des vorliegenden Fahrzeugs des Kunden mittels einer vordefinierten Symptomliste aus der Datenbank des Diagnosesystems aufgenommen. Zusätzlich können die Umgebungsinformationen aufgenommen werden. Diese diagnoserelevanten Informationen können als zusätzliche Messungen betrachtet werden. Alle Informationen über die Fahrzeughistorie werden dokumentiert. Die Informationen über frühere Reparaturen und den alten Zustand der verbauten Komponenten dienen zur Optimierung der Fehlersuche. Weitere Diagnoseinformationen liefern die Steuergeräte durch die Eigendiagnosefähigkeiten. Um die verbauten Steuergeräte nach möglichen vorhandenen Fehlercodes auszulesen, wird der Diagnosetester an das Fahrzeug angeschlossen. Darüber hinaus kann das Werkstattpersonal einige Funktionskontrollen durchführen, zum Beispiel durch eine Probefahrt oder Sichtkontrolle, durch Überprüfung der Kabel-, Steck- oder Schlauchverbindungen.

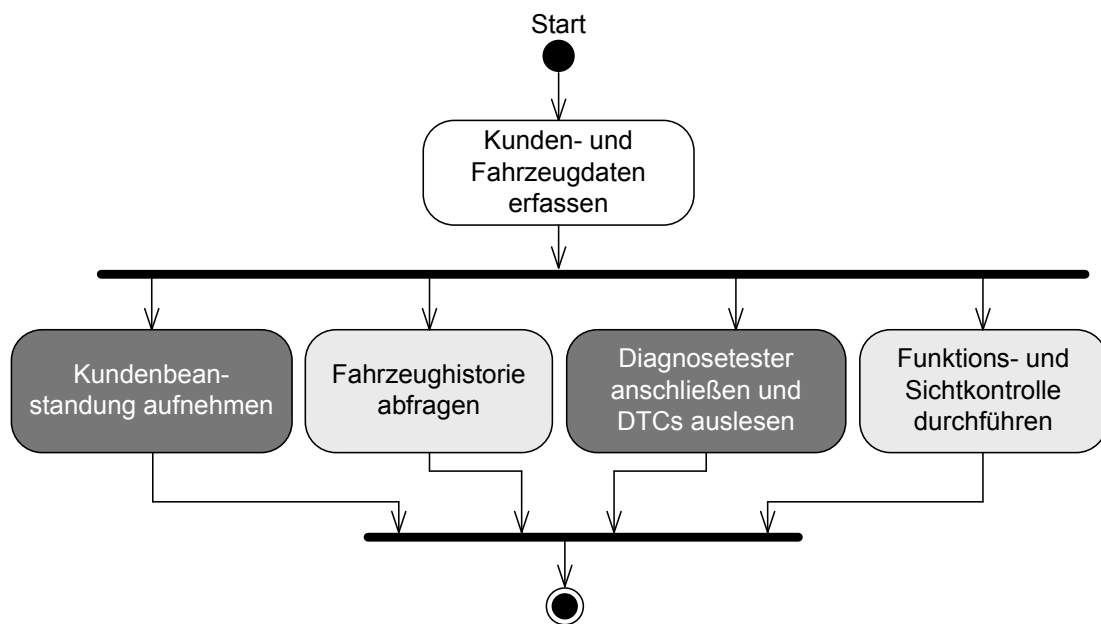


Abbildung 6.2: Aktivitäten bei der Fahrzeugannahme

Der nächste Prozessabschnitt „Diagnose und Reparatur durchführen“ beginnt mit der Analyse der Diagnoseinformationen, die im ersten Schritt gesammelt wurden. Eine Handlungsempfehlung wird anhand der vorhandenen Informationen ausgegeben. Hierbei können verschiedene Aktivitäten durchgeführt werden (Abbildung 6.3):

- Abfragen der Istwerte von Systemsensoren
- Durchführung von Messungen mit externen Messgeräten
- Ausführung von Stellgliedtests
- Prüfung der Systemkomponenten.

Die Ergebnisse von Messungen, das Auslesen von Istwerten sowie Stellgliedtests dienen zur Verfeinerung der Diagnosehypothese. Die Komponentenprüfung bestätigt fehlerhafte Komponenten oder schließt sie endgültig aus. Anschließend werden die defekten Komponenten ausgetauscht beziehungsweise die Reparatur nach Vorgabe abgearbeitet. Danach werden vom Werkstattmitarbeiter mit Hilfe des Diagnosetesters alle Fehlercodes aus den Steuergeräten

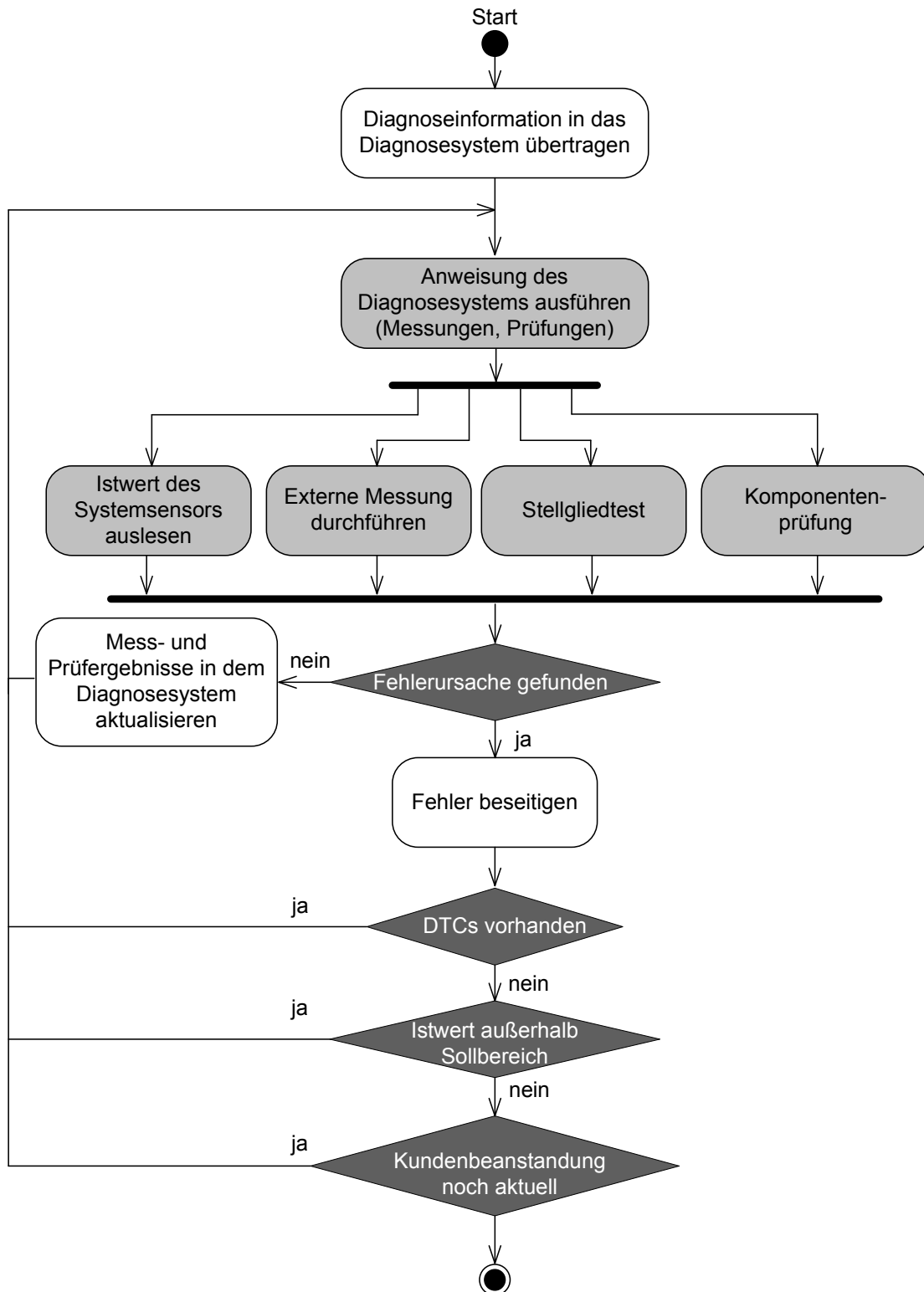


Abbildung 6.3: Aktivitäten bei der Diagnosedurchführung und Reparatur

gelöscht, da durch die Arbeiten am Fahrzeug eventuell neue Fehlercodes entstanden sind. Der nächste Schritt ist die Prüfung der Existenz der vorliegenden Beanstandungen zum Beispiel durch eine Probefahrt beziehungsweise Funktionstests. Abweichende Istwerte, die während der Diagnose gefunden werden, sollen mit Hilfe eines Diagnosetesters nochmals ausgelesen

und mit dem Sollwertbereich verglichen werden. Ein Wiedereinstieg in den zweiten Prozessabschnitt ist notwendig, wenn Symptome, Fehlercodes oder Sollwertabweichungen weiterhin existieren.

Der letzte Prozessabschnitt startet mit der Übergabe des reparierten Fahrzeuges und dem Diagnoseprotokoll vom Werkstattpersonal an den Kunden. Anschließend können die Prüfschritte und die Reparatur beziehungsweise der Austausch der Komponente aus dem Diagnoseprotokoll dokumentiert, archiviert und gegebenenfalls mit Zustimmung des Werkstattpersonals an den Hersteller des Diagnosetools gesendet werden.

### 6.1.2 Systemarchitektur DRT-System

In diesem Abschnitt wird die Systemarchitektur von DRT-System in rudimentärer Form unter Verwendung von UML-Notation beschrieben. Die im Kapitel 2 beschriebene Problematik in der freien Werkstattdiagnose und die daraus resultierenden Anforderungen sowie das in Kapitel 3 und 4 entwickelte Lösungskonzept ermöglichen es, eine Systemarchitektur zu entwerfen. Dabei existieren zwei Hauptanforderungen an die Systemarchitektur:

- Bei entsprechender Realisierung soll DRT-System die Anforderungen der freien Werkstattdiagnose erfüllen.
- Das entwickelte Diagnoselösungskonzept in DRT-System soll den Servicebereich bezüglich der Qualitätssicherung geeignet unterstützen.

Ausgehend von diesen Anforderungen werden mittels des Dekompositionsansatzes das System und die Schnittstellen zwischen den einzelnen Segmenten beziehungsweise Bestandteilen des Systems erläutert. In der ersten Ebene der Dekomposition des Systems wird die Struktur des Systemaufbaus in Abbildung 6.4 grob dargestellt. Die adäquate Systemarchitektur beinhaltet sieben Teilsysteme, die in Folge beschrieben werden. Dabei ist DRT-System das zu entwickelnde System. Um eine Diagnose erstellen zu können, greift DRT-System auf fünf Wissensdatenbanken zu:

- Die erste Datenbank ist die Modelldatenbank, in der die erstellten Modelle abgelegt sind.
- Die Fahrzeugparameterdatenbank enthält die Sollwertbereiche, Test- und Prüfanweisungen, Arbeitswerte, etc. Mit der Modelldatenbank und der Fahrzeugparameterdatenbank wird nach der Fahrzeugidentifikation das Diagnosemodell für das konkrete Fahrzeug erstellt.
- Die Übersetzungsdatenbank beinhaltet verschiedene Schlüssel zur Decodierung der Diagnoseinformationen, die aus den Fahrzeugsteuergeräten stammen.
- Die Ersatzteildatenbank liefert die Informationen über die Verfügbarkeit der Ersatzteile im Lager der Werkstatt, Ersatzteilkataloge mit entsprechenden Bestellformularen und die benötigte Zeit, um die Teile zu liefern.
- Um das Wissen vom Feld zurückzuführen, existiert eine weitere Datenbank, die DRT-System die Rückmeldeinformationen liefert:

- Die Rückmeldungsdatenbank bekommt Informationen über Fehler beziehungsweise aufgelaufene Bugs von DRT-System während des Einsatzes in Form von Trouble-Tickets.
- Wenn ein Diagnosevorgang abgeschlossen ist, kann der Nutzer mittels einer Schnittstelle von DRT-System das Diagnoseprotokoll an die Rückmeldungsdatenbank senden. Diese Informationen können zu verschiedenen Zwecken, zum Beispiel zur Verbesserung der Modelle, zur Auswertung der Ausfallwahrscheinlichkeit, etc. genutzt werden.

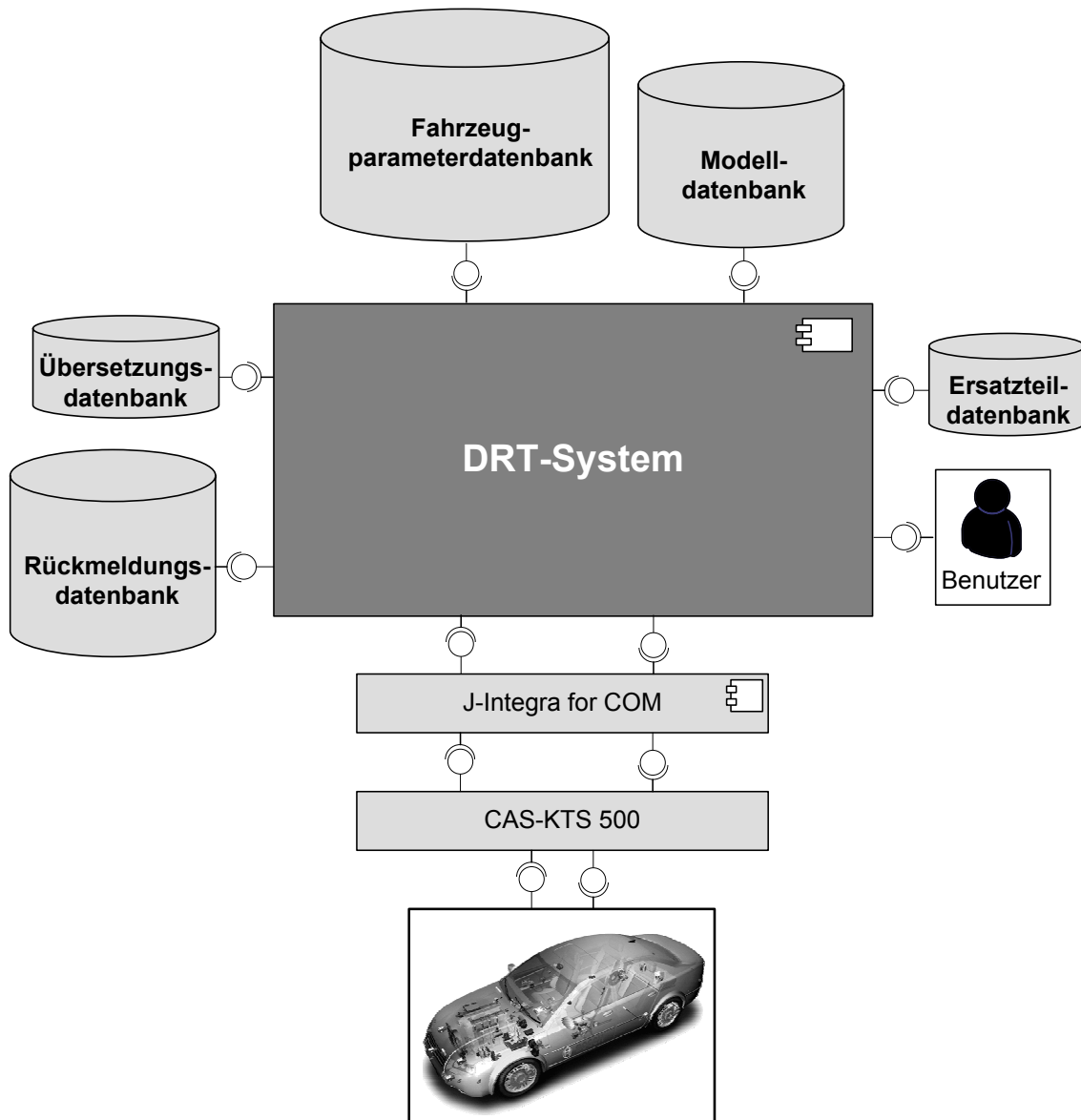


Abbildung 6.4: DRT-System und seine Umgebung

Neben der Benutzerschnittstelle, die die Interaktion zwischen dem Benutzer und DRT-System ermöglicht, benötigt das System eine weitere Schnittstelle, über die das entwickelte System mit dem Fahrzeug kommuniziert. Hierbei steuert DRT-System mit Hilfe von „J-Integra for COM“ die Testerschnittstelle der Software des CAS-KTS500 (CAS - Computer Aided Service - Eigendiagnose-Software von Bosch; KTS500 - Steuergeräte-Diagnose-Tester) an, um damit die

Messungen mit systemeigenen Sensoren sowie Stellgliedtests anzustoßen. Die Ergebnisse der Messungen und Tests werden auch von „J-Integra for COM“ an DRT-System weitergereicht. „J-Integra for COM“ ist ein Produkt von Intrinsic Software International Inc. und ermöglicht die Integration von komplexen Softwaresystemen über Systemplattformen hinweg. Mit dieser Komponente lässt sich aus DRT-System heraus auf COM-Komponenten zugreifen.

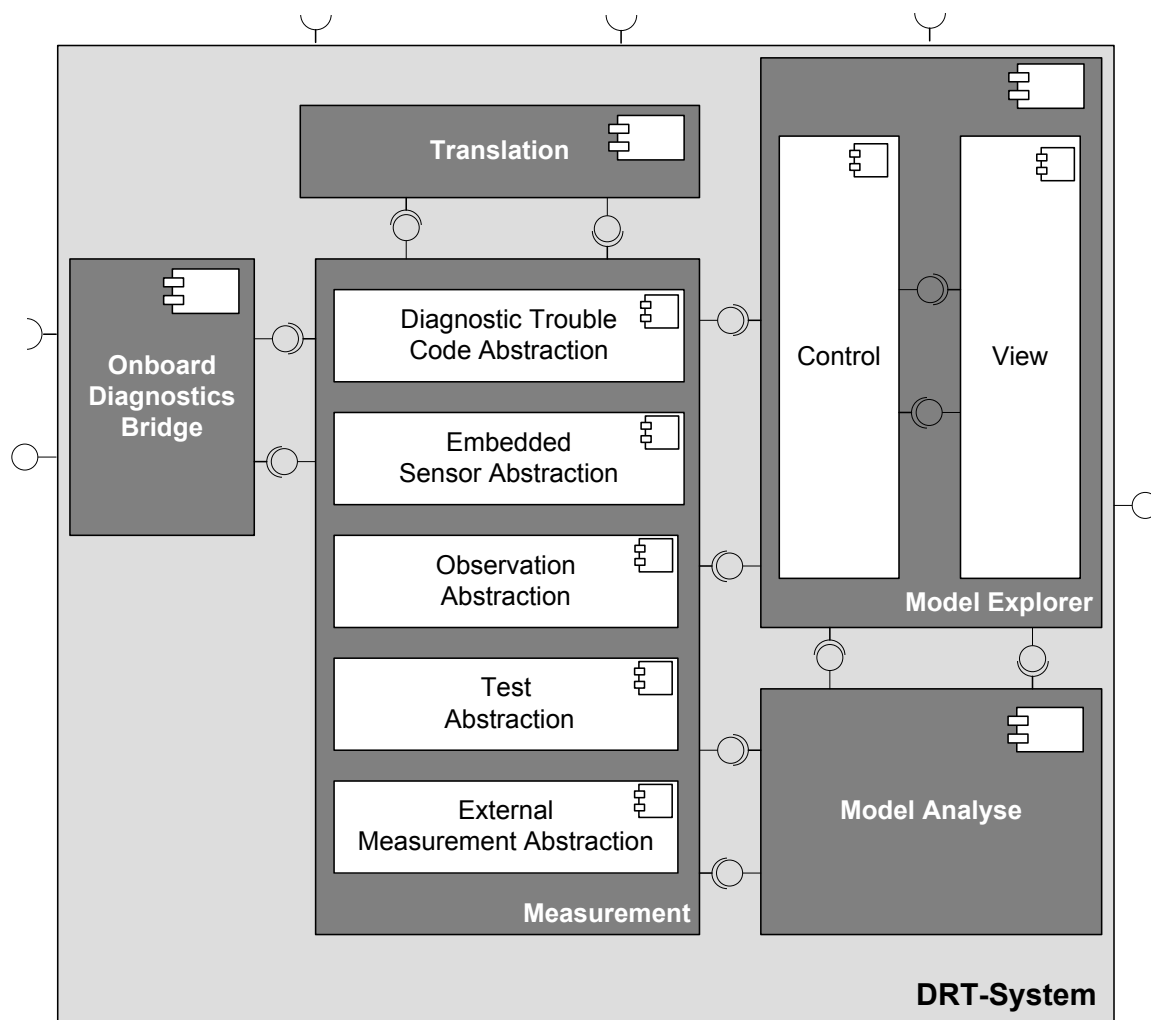


Abbildung 6.5: Dekomposition von DRT-System in Segmente

In Abbildung 6.5 ist die Zerlegung von DRT-System dargestellt. DRT-System besitzt insgesamt fünf Segmente beziehungsweise Pakete:

- **Model Analyse**: Dieses Paket ist in der Lage, die erzeugten Modelle mit den eingegebenen Diagnoseinformationen zu analysieren. Das Paket stellt Methoden für die Fehleranalyse, die Erzeugung von Listen der verdächtigen Komponenten und den optimalen Arbeitsablauf dem Paket „Model Explorer“ zur Verfügung.
- **Model Explorer**: Dieses Paket stellt das Benutzerinterface für das Diagnosetool dar. Es basiert auf zwei Teilsegmenten: „Control“ und „View“. Es unterstützt die Interaktionen zwischen dem Benutzer und dem Diagnosetool im Diagnosevorgang und steuert und protokolliert den Ablauf der Fehlersuche.



- Onboard Diagnostics Bridge: Diese Brücke ist nötig, um die Methodenaufrufe aus der DRT-System-Umgebung in die Fahrzeugumgebung mittels „J-Integra for COM“ weiterzuleiten und zurückzuerhalten.

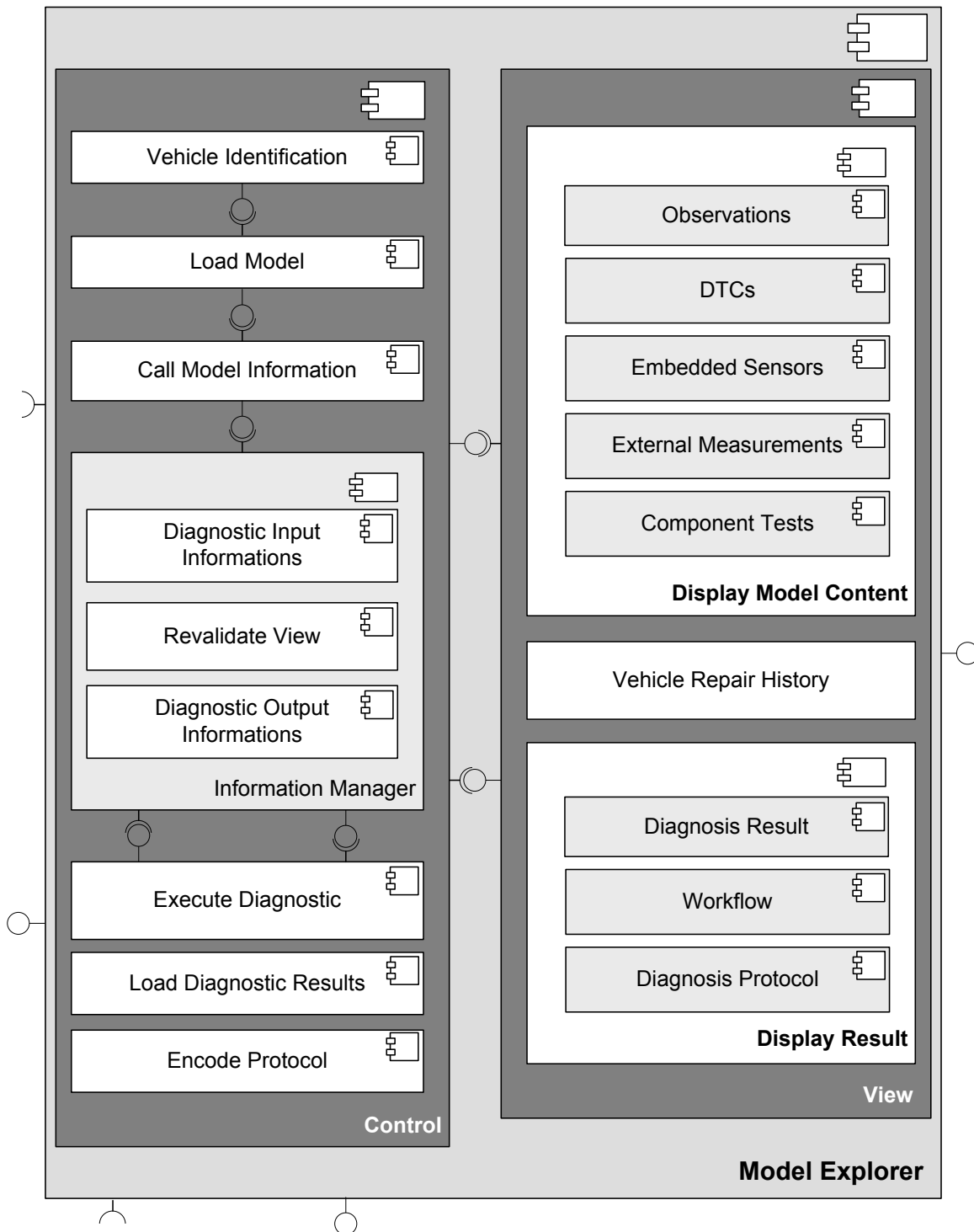


Abbildung 6.6: Dekomposition des Segments „Model Explorer“ in Pakete

- Translation: Das Paket ermöglicht es, die kodierten Textbausteine der CAS-Schnittstelle mit Hilfe einer Übersetzungsdatenbank in Klartext zu übersetzen. Deshalb besitzt das Paket Schnittstellen zur Übersetzungsdatenbank und Schnittstellen zum Measurement-

Paket. Die Messergebnisse und ausgelesenen Fehlercodes werden vom Measurement-Paket über eine Schnittstelle an das Translation-Paket weitergereicht und im Klartext zurückgegeben.

- **Measurement:** Das Measurement-Paket beinhaltet fünf Segmente. Dabei führen das Paket „Embedded Sensor Abstraction“ die angeforderten Istwert-Messungen, das Paket „Diagnostic Trouble Code Abstraction“ die Fehlerspeicherabfrage und das Paket „Test Abstraction“ die Stellgliedtests eigenständig aus und bedienen sich jeweils der Schnittstelle des Pakets „Onboard Diagnostics Bridge“. Die Interpretation der Kundenbeanstandungen, die von dem Paket „Model Explorer“ aufgenommen wurden, ist Aufgabe des Pakets „Observation Abstraction“. Die Ergebnisse der Komponententests, die durch den Benutzer ausgeführt und durch das Paket „Model Explorer“ aufgenommen wurden, werden in dem Paket „External Measurement Abstraction“ verarbeitet und an das Paket „Model Analyse“ weitergereicht.

Das Segment beziehungsweise Paket „Model Explorer“ besitzt zwei Hauptpakete: „Control“ und „View“ (Abbildung 6.6). Das Paket „Control“ stellt sämtliche Funktionsmodule bereit, die zur Durchführung und Protokollierung des Diagnosevorgangs in der freien Werkstatt benötigt werden. Dabei stellt dieses Paket nicht nur die Verbindung zu den oben genannten Datenbanken her, sondern verwaltet auch die Diagnoseinformationen, die vom Benutzer mittels des Paketes „View“ eingegeben wurden und jenen, die aus der Onboard-Diagnosefunktion des Fahrzeugs zur Verfügung stehen. Zudem verfügt dieses Paket über Kodierungsverfahren, über die das Diagnoseprotokoll mit Zustimmung des Benutzers an die Rückmeldedatenbank gesendet wird. Das Paket „Control“ besitzt natürlich weitere Funktionen wie Fahrzeugidentifikation, Laden des Modells, Anstoßen der Modellanalyse und Abholen der Analyseergebnisse sowie deren Erklärungen für das vorgelegte Problem. Das Paket „View“ repräsentiert das Diagnosewissen, das sich im Modell befindet, und die Diagnoseergebnisse nach jedem Arbeitsschritt, der in dem Diagnosevorgang durchgeführt wurde. Die bislang grobe Strukturbeschreibung der Architektur von DRT-System wird im Folgenden um eine detaillierte Betrachtung aus Benutzersicht ergänzt. Hierbei werden die Ergebnisse der Umsetzung der Systemarchitektur auf die Wahrnehmung des Benutzers reduziert.

## 6.2 Diagnose mit DRT-System

Die entwickelten Algorithmen, die in Kapitel 4 beschrieben sind, und die Systemarchitektur, auf die in Abschnitt 6.1 eingegangen wurde, wurden in dieser Arbeit als Grundlage für die Entwicklung eines Diagnosetools genommen. Im Abschnitt 1.3.4 wurden die Ergebnisse der Realisierung und die Bedienung von DRT-System aus Benutzersicht grob vorgestellt. Deshalb wird in diesem Abschnitt nur die Diagnose mit der entwickelten Applikation beschrieben.

In Tabelle 6.1 und Tabelle 6.2 wurden zwei Versuche mit ihren Abläufen unter Benutzung von DRT-System dargestellt.

	Beobachtung/Messung/Test/Prüfung	Handlung und Ergebnis
1	Kundenbeanstandung	schlechte Gasannahme und Übergangsfehler
2	DTCs	keine

3	Istwert Saugrohr-Drucksensor	normal
4	Drosselklappen Istwert Positionssensor 1, 2	normal
5	Istwert Ansaugluft-Temperatursensor	normal
6	Stellgliedtest Drallklappen	in Ordnung
7	Unterdruckleitung von Drallklappe	Leckage gefunden

Tabelle 6.1: Ablauf eines Diagnoseversuchs an einem ME7.6-Benzinsystem mit DRT-System

	Beobachtung/Messung/Test/Prüfung	Handlung und Ergebnis
1	Kundenbeanstandung	Kraftstoffverbrauch zu hoch und Abgas ist schwarz
2	DTCs	P0110
3	Ladeluft-Temperatursensor	Signal unplausibel → Unterbrechung

Tabelle 6.2: Ablauf eines Diagnoseversuchs an einem CP3-Dieselsystem mit DRT-System

Zu Beginn werden die Einstiegsinformationen aus Kundenbeanstandungen und ausgelesenen Fehlercodes durch Diagnosealgorithmen ausgewertet. Die Ergebnisse der Auswertung bestehen aus einer Liste der verdächtigen Komponenten und einer Liste der Messungs- oder Prüfungsvorschläge. Die verdächtigen Komponenten sind Komponenten gemäß Kapitel 4, die die eingegebenen Symptome am ehesten durch ihren Defekt beziehungsweise ihr Fehlverhalten verursachen können. Die Komponenten erhalten eine Gewichtung gemäß ihrer Fähigkeit, bei ihrem Ausfall diese Symptome hervorzurufen, und ihrer Ausfallwahrscheinlichkeit, die durch statistische Erhebungen entstanden sind. Die Handlungsempfehlung ist eine Liste der Messungen und Prüfungen, mit welcher der Benutzer die zunächst noch sehr umfangreiche Liste der verdächtigen Komponenten eingrenzen kann. Im ersten Beispiel existiert kein Fehlercode, sondern es gibt nur zwei Kundenbeanstandungen. Mit diesen Einstiegsinformationen berechnet DRT-System den nächsten optimalen Schritt „Istwert des Saugrohr-Drucksensors auslesen“.

Bei der nächsten Iteration soll die am höchsten gewichtete Messung oder Prüfung abgearbeitet werden. Im ersten Beispiel (Tabelle 6.1) wird durch einen Doppelklick die hinter dem Sensor liegende Anleitung geöffnet (Abbildung 6.7). In der Anleitung befinden sich Informationen, unter welchen Randbedingungen die Messungen erfolgen, wie die Messungen ablaufen und zu welchem Sollwertergebnis die Messungen gelangen sollen. Die erstellten Anleitungen weisen ein Textfeld für die direkte Eingabe von Messwerten aus, so dass auch ohne Ansteuerung der Tester das Ergebnis der Prüfung eingegeben werden kann. Diese Möglichkeit ist für die manuelle Messung mittels Oszilloskop, Voltmeter, etc. erforderlich. Sollte in das Formular kein oder ein ungültiges Ergebnis eingetragen werden, wird automatisch über systemeigene Sensoren der Diagnosetester angesteuert und die Messwerte automatisiert übernommen. Das gewonnene Ergebnis der Prüfungen kann vom Diagnose-Algorithmus nicht direkt verwendet werden, da es sich um Realwerte handelt. Diese Werte müssen nun bewertet werden, so dass dem Algorithmus die Tendenz (niedrig, normal oder hoch) übergeben werden kann. Die Bewer-

tung kann teilweise automatisch erfolgen, wenn Grenzwerte vorliegen. In den anderen Fällen muss die Bewertung vom Werkstattpersonal erfolgen.

### **Saugrohr Drucksensor**

**Beschreibung des Istwertes: der Wert informiert über die aktuellen Saugrohrdruck.**

**Bitte Verbindung zum Diagnoseanschluss im Fahrzeug herstellen.**

**Saugrohrdruck Status (Maximaler Anzeigebereichs!)**

**Sollwert: 0 ... 150 kPa Istwert:  kPa**

**Motor ausgeschaltet. Sämtliche Verbraucher ausgeschaltet.**

**Sollwert: 90 ... 110 kPa Istwert:  kPa**

**Motor im Leerlauf. Sämtliche Verbraucher ausgeschaltet.**

**Sollwert: 25 ... 50 kPa Istwert:  kPa**

Abbildung 6.7: Beispiel: Instruktion zum Auslesen eines Systemsensor-Istwertes

Im Gegensatz zum ersten Beispiel gibt es bei dem zweiten Beispiel einen Fehlercode, der direkt auf eine Komponente zeigt. In diesem Fall ist der zweite Schritt der Diagnose die Prüfung des Ladeluft-Temperatursensors. Analog zu den Messungen liegen hinter den Systemkomponenten die jeweiligen Prüfanleitungen (Abbildung 6.8). In diesen Anleitungen stehen auch die Randbedingungen sowie die Schritte der Prüfungen. Per Doppelklick wird der Komponententest aufgerufen und kann abgearbeitet werden. Im Unterschied zur Messung sind die Ergebnisse der Komponentenprüfung immer mit „in Ordnung“ oder „nicht in Ordnung“ zu bewerten. Wenn das Ergebnis mit „in Ordnung“ bewertet wurde, werden alle internen Fehlervariablen der geprüften Komponente auf „normal“ gesetzt.

Das Ergebnis der Messung beziehungsweise Prüfung wird an die Diagnosealgorithmen übergeben, es werden neue Listen berechnet und die nächste Iteration beginnt. Bei jeder Iteration wird das Modell mit immer umfangreicheren Diagnoseinformationen berechnet. Der Vorgang wird so lange fortgeführt, bis keine weitere Messung oder Prüfung in der Handlungsempfehlungsliste mehr vorgeschlagen werden kann und sich damit die Liste der verdächtigen Komponenten nicht weiter eingrenzen lässt. Falls die Liste der Handlungsempfehlungen komplett abgearbeitet wurde und keine Komponenten bei den bisherigen Prüfungen als nicht fehlerhaft identifiziert wurden, wird die Liste der verdächtigen Komponenten in der Reihenfolge der Gewichtung abgearbeitet, bis die defekte Komponente gefunden ist.

Im Beispiel des CP3-Dieselsystems wurde die Komponente Ladeluft-Temperatursensor geprüft. Da diese Komponente ein Systemsensor ist, existieren eine Reihe elektrischer Prüfungen. Im Unterschied zu systemeigenen Sensoren gibt es bei den Aktoren nicht nur elektrische Prüfungen, sondern auch Stellgliedtests, die mit Steuergerätefunktionen geprüft werden können. Im ersten Beispiel sollten die Istwerte weiterer verfügbarer Sensoren ausgelesen werden, bevor DRT-System eine Komponentenprüfung vorgeschlug. Die Prüfanleitung der Unterdruckleitung der Drallklappe ist hierbei sehr einfach gehalten. Aus den definierten möglichen Fehlern kann die Prüfanweisung generiert werden. Wenn das Komponentenmodell keinen vordefinierten Fehler beinhaltet, wird eine allgemeingültige Prüfanweisung für passive Komponenten der bestimmten Domäne aufgerufen. Natürlich ist es auch möglich in den Listen weiter unten



### 6.3 Migration in das bestehende Laufzeitsystem

Aus wirtschaftlicher Sicht ist es eine große Umstellung vom fall- beziehungsweise regelbasierten Diagnoseansatz zum modellbasierten Ansatz. Zum einen fordert die Einbettung des Modellierungstools in den vorhandenen Wissenserhebungsprozess eine Erweiterung und ein Umdenken der vorhandenen Mitarbeiter. Zum anderen wurde der Umstieg auf das neue Diagnosetool noch keiner kompletten ergonomischen Überprüfung unterzogen. Trotz seiner intuitiven Bedienung bedarf es möglicherweise einer Umschulung oder kurzen Einführung für die Werkstattmitarbeiter. Bedingt durch die gleichzeitigen erforderlichen Änderungen bei der Wissenserhebungs- und Diagnosevorgehensweise entstehen verschiedene Einführungszenarien, welche mit Aufwand, Nutzen und Risiko verbunden sind. Es existieren drei Szenarien:

- **Totalumstieg:** Bei einem Totalumstieg (Abbildung 6.9) werden existierende Autoren- und Diagnosesysteme durch das neue Wissenserhebungs- und Diagnosetool ersetzt. Zunächst wird das neue Tool komplett entwickelt. Nach der Schulung der Autoren erfolgt die komplette Umstellung aller Fehlersuchanleitungen auf die modellbasierte Diagnose. Anschließend erfolgt die Markteinführung der neuen Testersysteme bei den Kunden.

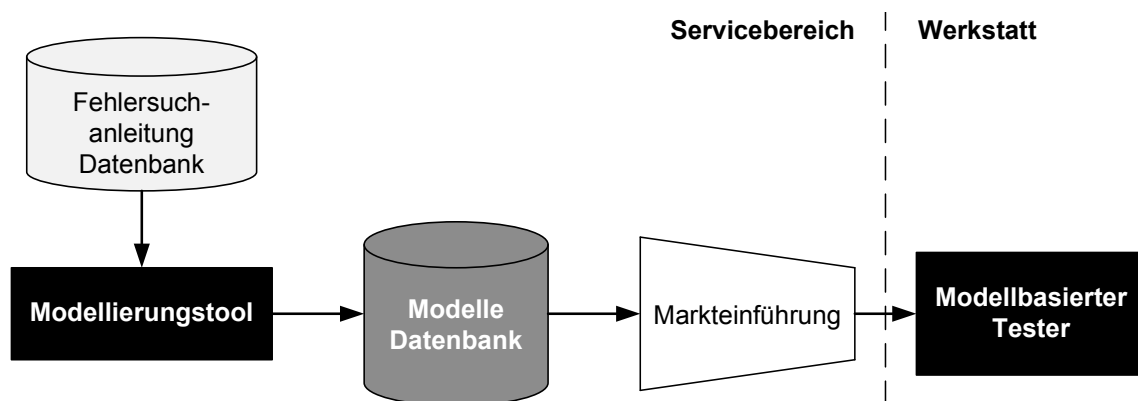


Abbildung 6.9: Szenario für den Totalumstieg

- **Parallelbetrieb:** Beim Parallelbetrieb (Abbildung 6.10) wird das neue Wissenserhebungs- und Diagnosetool parallel zu den existierenden Autoren- und Diagnosesystemen betrieben. Nachdem die neuen Tools komplett entwickelt sind, werden die neu zu erstellenden, geführten Fehlersuchen mit dem neuen, modellbasierten System erzeugt. Das neue Diagnosetool läuft parallel zum alten auf der gleichen Hardware. Nach der Fahrzeugbeziehungsweise Systemidentifikation wird entweder das alte oder das neue Diagnosetool angezeigt.
- **Integration:** Bei der Integration in die existierende Tool-Landschaft baut das neue Wissenserhebungstool auf die existierenden Datenbanken auf. Hier lässt sich zum Beispiel das Textbaustein-Konzept für die Erstellung der System- oder Komponentenbeschreibungen und Prüfanweisungen weiterverwenden. Neben dieser Integration bestehender Softwarekomponenten müssen hingegen einige entscheidende Softwarekomponenten nach dem Schichtkonzept des Wissenserhebungstools entwickelt werden (zum Beispiel automatisierte Einbindung der Steuergerätediagnose und Fehlercodes oder automatisierte datenbankgestützte Sollwert-Einbindung). Im Diagnosetool sind mehrere Varianten der Integration möglich.

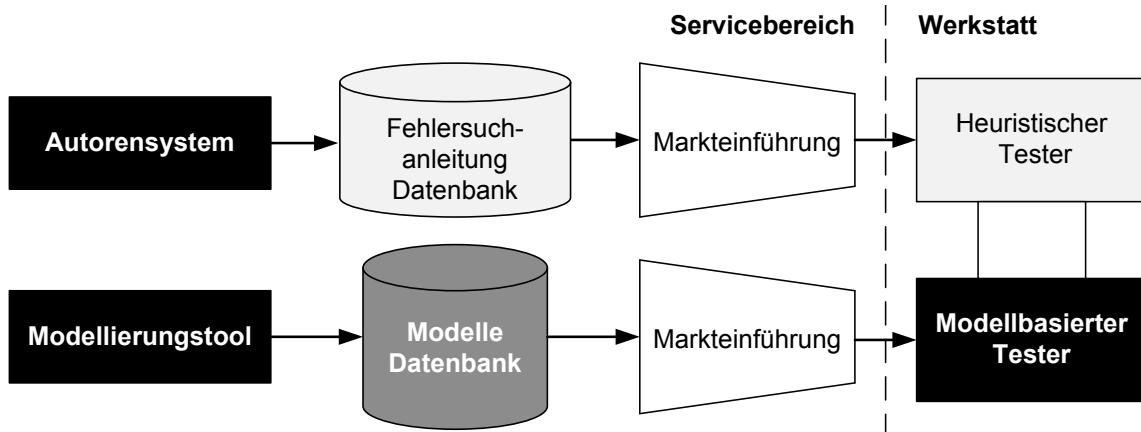


Abbildung 6.10: Szenario für den Parallelbetrieb

Nach Abwägung des Aufwands und insbesondere des Risikos der verschiedenen Szenarien kommt nur das integrative Konzept als Einführungsszenario infrage. Hierbei kann die Integration wiederum in drei Szenarien aufgeteilt werden, wobei sich die drei integrativen Konzepte nur in der Entwicklung und Einführung des Diagnosesystems in der Werkstatt unterscheiden:

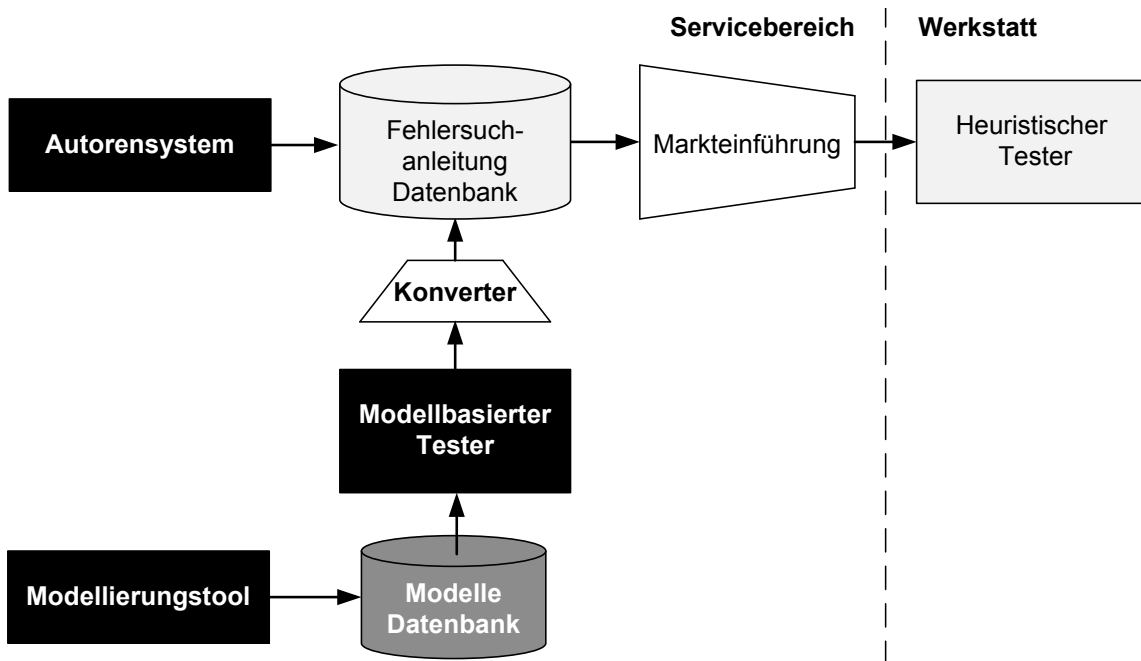


Abbildung 6.11: Integration mit dem Konvertierungsansatz

- Integration durch ein Konvertierungstool (Abbildung 6.11): Eine der Integrationsmöglichkeiten ist die Generierung der Fehlersuchanleitung aus Modellen mittels Analysealgorithmen und deren Import in die Fehlersuchanleitungsdatenbank. Hierbei wird das bestehende Diagnosetool nicht verändert und die Fehlersuchanleitung bleibt im alten Format. Dieses Konzept hat den Vorteil, dass kein Einführungs- und Entwicklungsaufwand bei dem Diagnosetool anfällt. Leider ist dieser Vorteil auch der Nachteil, da eine dynamische Optimierung der Fehlersuche unter Verwendung der Diagnosemodelle nicht ausgenutzt werden kann.

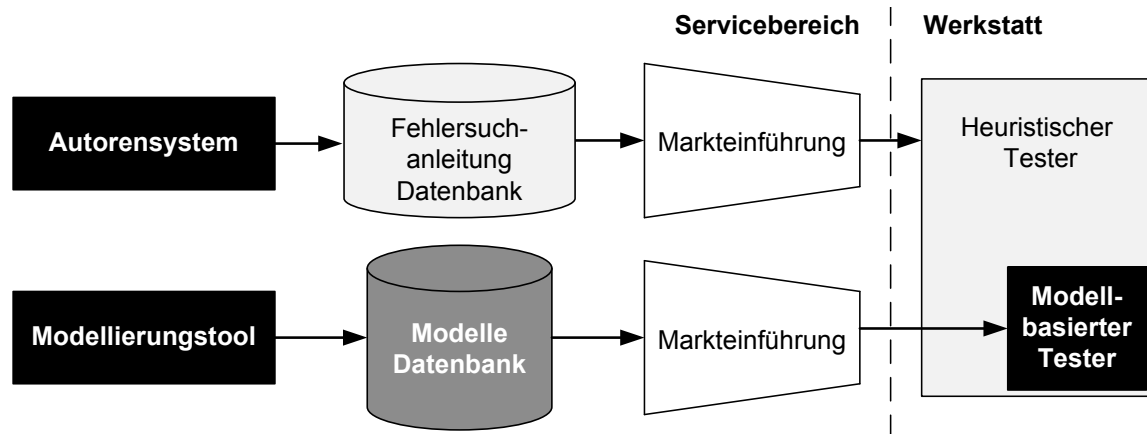


Abbildung 6.12: Integration mit dem Plug-in-Ansatz

- Plug-in-Ansatz (Abbildung 6.12): Ein weiterer Ansatz ist der Plug-in-Ansatz, bei welchem für bestehende Diagnosetools ein zusätzliches Plug-in für die Analyse und Darstellung der Analyseergebnisse des Diagnosemodells entwickelt und integriert wird. Der Vorteil dieses Ansatzes ist die verbesserte Diagnoseunterstützung durch den modellbasierten Ansatz für ein neues System beziehungsweise einen neuen Fahrzeugtyp. Weiterhin benötigt dieser Ansatz einen geringeren Entwicklungsaufwand, weil das Framework durch das vorhandene Diagnosetool bereits gegeben ist. Der Nachteil des Plug-ins ist, dass die modellbasierte Diagnose ohne ein vorhandenes Diagnosetool nicht funktionsfähig ist. Dies bedeutet, dass die Altlast immer weiter mitgenommen werden muss, selbst wenn das Diagnosewissen später überhandnimmt.

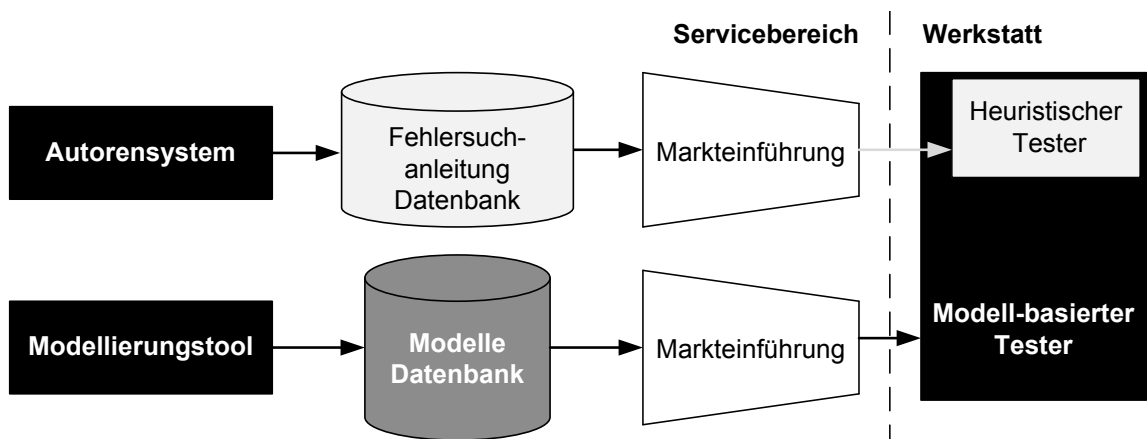


Abbildung 6.13: Integration mit der Entwicklung eines neuen Diagnosetools

- Integration durch ein neu erstelltes Diagnosetool (Abbildung 6.13): Der letzte Integrationsansatz erfolgt durch die Entwicklung eines neuen Diagnosetools, das die modellbasierte Diagnose vollständig unterstützt und die Möglichkeit der Integration des vorhandenen Diagnosetools anbietet.

Aus den vorgestellten Integrationsansätzen wurde der letzte Ansatz durch Fach- und Führungskräfte des Servicebereichs als sinnvoll und realisierbar bewertet. In der Entwicklung der



neuen Diagnosegeneration ab 2012 wird der entwickelte Ansatz umgesetzt und in die Tool-Landschaft integriert. Dabei ist die vorgestellte Lösung ein Alleinstellungsmerkmal (USP - unique selling point) des neuen Produkts.

## 7 Evaluierung des Lösungsansatzes

Dieser Abschnitt beschreibt die Vorgehensweise zur Evaluierung der in Kapitel 3 und 4 erarbeiteten Lösungen auf zwei Gebieten:

- Aufwandsabschätzung für die Erstellung und Pflege von Diagnosemodellen mit dem entwickelten Modellierungstool
- Diagnosegüte der geführten Fehlersuche unter Verwendung des erstellten Diagnosemodells und des Diagnosetools.

Anhand der Ergebnisse sollen die Leistungsfähigkeit der Lösungen bewertet sowie ihre Stärken und Schwächen erfasst werden, um die weiterführenden Arbeiten planen zu können. Mit den Ergebnissen soll langfristig auch untersucht werden, wie sich das entwickelte Expertensystem in die bestehende Struktur des Servicebereichs einführen ließe.

### 7.1 Evaluierung der Wissenserhebung

Die Bewertungskriterien der Aufwandsabschätzung für die Modellierung können in zwei Kategorien aufgeteilt werden ([51]):

- Den Erstellungsaufwand eines neuen Systems oder Subsystems
- Das Einpflegen einer neuen Variante eines bestehenden Systems beziehungsweise Subsystems.

Die Beherrschung der Variantenvielfalt kann durch diese Aufteilung untermauert werden, da bei existierenden Redaktionssystemen auch die Möglichkeit zur Wiederverwendung des Diagnosewissens existiert. Um einen Vergleich zwischen der DMB-Applikation und existierenden Redaktionssystemen durchzuführen, erfolgt zunächst die Modellierung von Beispielsystemen, die auch für die Analyse der Diagnosegüte benötigt werden. Während der Modellierung wird der Aufwand festgehalten, der für die Modellierung beziehungsweise Erstellung eines neuen Systems und für dessen Variantenbildung anfällt. Anschließend erfolgt eine Abschätzung der Gültigkeit der Wiederverwendbarkeit für die Gesamtwissensbasis. Abschließend wird der Aufwand pro System beziehungsweise pro Subsystem abgeleitet. Dabei wird der Aufwand für nicht modellierte Anteile der Fehlersuchanleitung (wie zum Beispiel Sicherheitshinweise, Bildrecherche, Fahrzeugerprobung, Administration) mit dem gleichem Aufwand wie bei den vorhandenen Redaktionssystemen bewertet. Analog zu existierenden Redaktionssystemen gibt es in der Modellierungsphase zwei Arten von Projekten: Leit- und Folgeprojekte. Im Unterschied zum Leitprojekt wird im Folgeprojekt das vorhandene Modell in der Modelldatenbank verwendet. Das Leitprojekt wird in zwei Arten unterschieden: Leitprojekt mit Grundmodell

und Leitprojekt mit Variantenmodell. Dabei wird das Variantenmodell aus dem bestehenden Grundmodell abgeleitet. Die Wissenserhebung in einem Leit- und Folgeprojekt besteht aus folgenden Schritten ([37], [38], [53]):

- Leitprojekt:
  - Grundmodell
    - \* Systemgliederung in Subsysteme und Komponenten
    - \* Anlage fehlender Komponentenmodelle in der Modelldatenbank
    - \* Anlage der Modellstruktur mit Komponentenmodellen aus der Modelldatenbank
    - \* Parametrisierung der Modelle mit fahrzeugspezifischen Daten
  - Modellvariante
    - \* Ableitung des neuen vom bestehenden Modell
    - \* Anlage fehlender Komponentenmodelle in der Modelldatenbank
    - \* Ergänzung des bestehenden Modells um fehlende Komponentenmodelle
    - \* Parametrisierung der Modelle mit fahrzeugspezifischen Daten
- Folgeprojekt:
  - Parametrisierung der Modelle mit fahrzeugspezifischen Daten (Istwerte, DTCs, Abbildungen, etc.).

Für existierende Redaktionssysteme wurden Referenzdaten erfasst. Diese Erfassung erfolgte durch die Befragung von Servicemitarbeitern. Hierbei wurden Arbeitsschritte und Aufwand aufgenommen für:

- Anlegen eines neuen Systems (Leitprojekt)
- Variantenbildung eines Systems und Aktualisierung von Komponentendaten (Folgeprojekt).

Bei der ersten Evaluierung mit DMB-System wurde zunächst mit der Modellierung eines CP3-Diesel-Systems begonnen. Die Diagnoseinformationen für die Modellerstellung wurden in Interviews mit Werkstattmitarbeitern und Fachexperten gesammelt. Danach wurden die Informationen mit Hilfe von Fachliteratur ergänzt, abschließend die Modelle verschiedenen Reviews unterzogen. Die Prüfanweisungen, Testanleitungen, Soll- & Istwerte, Fehlercodes und Abbildungen für die Komponenten wurden aus existierenden Komponentendatenbanken entnommen und aufbereitet. Der entstehende Aufwand wurde dabei erhoben und in Tabelle 7.1 dargestellt. Die Komponenten-, Subsystem- und Systemmodelle wurden von Grund auf neu angelegt und stehen zur Fertigstellung und Prüfung für weitere Modelle in einer Modelldatenbank zur Verfügung.

Als weiteres System wurde ein ME7.6-Benzin-System modelliert. Dabei konnte auf bereits modellierte Komponenten aus der Datenbank zurückgegriffen werden. Durch diese Wiederverwendung konnten circa 70% der Komponenten der Datenbank entnommen werden. Bei den verbleibenden Komponenten konnte wiederum eine Einsparung von 50% für die Erfassung von Systemen und Subsystemen sowie 40% bei Komponenten in Struktur und Verhalten protokolliert werden. Somit war der Erstellungsaufwand des ME7.6-Benzin-Systems deutlich geringer als bei dem CP3-Diesel-Modell.

		CP3-Diesel mit DMB-System		
		Aufwand in h	Anzahl d. Komp.	Komp. in h
1	Wissenserhebung			
1.1	Erfassung System und Subsystem	35,0	40	0,88
1.2	Erfassung der Komponenten	57,6	57	1,01
1.3	Integration von Kundenbeanstandungen und Test	25,9	22	1,18
1.4	Erfassung von Sollwert und DTCs	21,5	24	0,9
2	Wissensprüfung			
2.1	Verifizierung gegen Testfälle	13,9		
2.2	Evaluierung in der Werkstatt	18,5		
3	Administration			
3.1	Allgemeine Verwaltung	9,2		
Summe		181,6	86	2,11

Tabelle 7.1: Aufwandserhebung bei der Erstellung des CP3-Diesel-Systems

		ME7.6-Benzin mit DMB-System		
		Aufwand in h	Anzahl d. Komp.	Komp. in h
1	Wissenserhebung			
1.1	Erfassung System und Subsystem	15,0	19	0,79
1.2	Erfassung der Komponenten	30,6	8	3,83
1.3	Integration von Kundenbeanstandungen und Test	29,0	13	2,23
1.4	Erfassung von Sollwert und DTCs	27,6	30	0,92
2	Wissensprüfung			
2.1	Verifizierung gegen Testfälle	10,3		
2.2	Evaluierung in der Werkstatt	16,5		
3	Administration			
3.1	Allgemeine Verwaltung	13,7		
Summe		142,7	86	1,66

Tabelle 7.2: Aufwandserhebung bei der Erstellung des ME7.6-Benzin-Systems

Für die abgeleitete Modellierung des Benzinsystems vom Dieselsystem wurde der Aufwand in Tabelle 7.2 erhoben ([52], [54], [151]).

Um ein reines Folgeprojekt bewerten zu können, wurde ein CP3-Diesel-System für vier und sechs Zylinder verglichen. Hierfür waren zum Beispiel circa 15% der Komponenten in Struktur und Verhalten und 10% der Prüfanleitungen, Sollwerte, Istwertabfragen für einen Sechszylindermotor zu ergänzen oder zu ändern. Dabei ergab sich der in Tabelle 7.3 aufgelistete

Aufwand für das Folgeprojekt des CP3-Diesel-Systems.

		CP3-Diesel mit DMB-System	
		Leitprojekt in h	Folgeprojekt in h
1	Wissenserhebung		
1.1	Erfassung System und Subsystem	35,0	10,4
1.2	Erfassung der Komponenten	57,6	8,7
1.3	Integration von Kundenbeanstandungen und Test	25,9	7,2
1.4	Erfassung von Sollwert und DTCs	21,5	2,2
2	Wissensprüfung		
2.1	Verifizierung gegen Testfälle	13,9	9,8
2.2	Evaluierung in der Werkstatt	18,5	15,7
3	Administration		
3.1	Allgemeine Verwaltung	9,2	6,3
Summe		181,6	60,3

Tabelle 7.3: Aufwandserhebung des CP3-Diesel-Systems bei Leit- und Folgeprojekt

Es soll im Zuge der Evaluierung der Aufwand für die Erstellung der Diagnosemodelle mittels DMB-System mit dem Aufwand zur Erstellung der geführten Fehlersuchanleitung unter Zuhilfenahme existierender Autorensysteme gegenübergestellt und bewertet werden. Da das Diagnosewissen bei DMB-System anhand von Modellen präsentiert, in existierenden Autorensystemen hingegen in Form von Fehlersuchbäumen mit Textbausteinen dargestellt wird, ist ein direkter Vergleich nicht möglich. Es existieren jedoch viele Aktivitäten bei der Wissenserhebung, die gleich sind:

- Die Erfassung der eingebauten Systeme, Subsysteme und Komponenten: Struktur, Einbaulage, Funktionsbeschreibung, Pinbelegungen, Abbildung, etc.
- Onboard-Diagnose-Informationen: Fehlercodes, Stellgliedtests
- Die Informationen über Sollwertbereiche und Möglichkeiten zur Istwertabfrage der Systemsensoren beziehungsweise externen Messungen
- Die Prüfanweisungen der Komponenten oder Systeme
- Wissensprüfung und Administration.

Einige Aktivitäten, die - bedingt durch den angewendeten Ansatz - unterschiedlich sind:

- Die Struktur- und Verhaltensmodellierung mittels DMS-System
- Die Erstellung der Prüfabläufe in existierenden Autorensystemen.

		Leitprojekt I (CP3-Diesel)		Leitprojekt II (ME7.6-Benzin)	
		DMB- System	Autoren- systeme	DMB- System	Autoren- systeme
1	Wissenserhebung				
1.1	Erfassung System und Subsystem	35,0	⊖ 17,6	15,0	⊖ 21,1
1.2	Erfassung der Komponenten	57,6	⊖ 25,6	30,6	⊖ 36,0
1.3	Integration von Kundenbeanstandungen und Test	25,9	⊖ 25,9	29,0	⊖ 29,0
1.4	Erfassung von Sollwert und DTCs	21,5	⊖ 21,5	27,6	⊖ 27,6
1.5	Erstellung von Prüfabläufen		⊖ 43,0		⊖ 55,3
2	Wissensprüfung				
2.1	Verifizierung gegen Testfälle	7,9	⊖ 13,9	6,3	⊖ 10,3
2.2	Evaluierung in der Werkstatt	18,5	⊖ 18,5	16,5	⊖ 16,5
3	Administration				
3.1	Allgemeine Verwaltung	15,2	⊖ 9,2	17,7	⊖ 13,7
Summe		181,6	⊖ 175,2	142,7	⊖ 210,0

Tabelle 7.4: Aufwandserhebung der Leitprojekte im Vergleich

In Tabelle 7.4 werden die Aktivitäten und deren Aufwand bei der Erstellung der Leitprojekte einander gegenüber gestellt. Anhand des Aufwands der beiden Leitprojekte mit DMB-System lässt sich erkennen, dass für den Aufbau der Komponenten- und Systemmodelle ein Initialaufwand nötig ist. In nachfolgenden Leitprojekten mit dem selben Umfang wie die beiden Leitprojekte sollte nach Fertigstellung der Modellbibliothek mit mindestens 15% Aufwandsreduzierung beim gesamten Umfang des Projekts beziehungsweise 48% bei der Modellierungsaktivität zu rechnen sein. Dies entspricht gegenüber dem Aufwand für Leitprojekte gemäß der Auswertung der durchschnittlichen Erstellzeiten der geführten Fehlersuche in existierenden Autorensystemen für Diesel-Systeme einer Einsparung von circa 18% und 32% bei Benzin-Systemen ([39], [7]).

In der zweiten Evaluierung ist ein weiteres System - ALWR1.0-System - als Leitprojekt modelliert worden. Das Ziel der Modellierung eines ALWR1.0-Systems ist die Überprüfung der Übertragbarkeit des Modellierungsansatzes auf weitere Systeme beziehungsweise Domänen im Fahrzeug, da das System viele elektrische Komponenten und wenige Sensoren besitzt.

		Leitprojekt ALWR 1.0	
		DMB- System	Autoren- systeme
1	Wissenserhebung		
1.1	Erfassung System und Subsystem	11,6	⊖ 10,4
1.2	Erfassung der Komponenten	13,4	⊖ 18,3
1.3	Integration von Kundenbeanstandungen und Test	21,7	⊖ 21,7
1.4	Erfassung von Sollwert und DTCs	37,8	⊖ 37,8

1.5	Erstellung von Prüfabläufen		⊙ 32,5
2	Wissensprüfung		
2.1	Verifizierung gegen Testfälle	14,7	⊙ 23,8
2.2	Evaluierung in der Werkstatt	17,2	⊙ 17,2
3	Administration		
3.1	Allgemeine Verwaltung	15,4	⊙ 12,3
Summe		131,8	⊙ 174,0

Tabelle 7.5: Aufwandserhebung des ALWR1.0-Systems bei Leitprojekten

Aufgrund des symmetrischen Aufbaus des ALWR1.0-Systems ist die Wiederverwendung bei der Modellierung nicht nur auf Komponentenebene, zum Beispiel elektrischen Leitungen oder Achssensoren möglich, sondern es können Teilsysteme wiederverwendet werden. Beispielsweise kann nach Fertigstellung des Wirkkettenmodells des linken Scheinwerfers dieses Teilmodells sofort für die Wirkkette des rechten Scheinwerfers verwendet werden. Die Anpassung des Modells beschränkt sich auf die Namensänderung. Aus diesem Grund ist selbst bei Leitprojekten eine Einsparung von circa 24,3% möglich (Tabelle 7.5), obwohl in der Modellbibliothek vorher keine Komponentenmodelle beziehungsweise Subsystemmodelle des ALWR1.0-Systems existieren ([40]).

Der Vergleich bei reinen Folgeprojekten gestaltet sich schwierig, da die Abweichung unterschiedliche Auswirkungen bei der Wissenserhebung hervorruft. Es reicht von der Änderung einer Abbildung der Einbaulage bis hin zu Umstrukturierungen der geführten Fehlersuche in existierenden Autorensystemen. Um einen durchschnittlichen Aufwand für die Folgeprojekte zu erhalten, wurde von verschiedenen Servicebereichen die durchschnittliche Ratio für ein Folgeprojekt auf circa 50% geschätzt.

		Folgeprojekt	
		DMB-System	Autorensysteme
1	Wissenserhebung		
1.1	Erfassung System und Subsystem	10,4	⊙ 7,0
1.2	Erfassung der Komponenten	8,7	⊙ 9,0
1.3	Integration von Kundenbeanstandungen und Test	7,2	⊙ 9,0
1.4	Erfassung von Sollwert und DTCs	2,2	⊙ 7,0
1.5	Erstellung von Prüfabläufen		⊙ 14,0
2	Wissensprüfung		
2.1	Verifizierung gegen Testfälle	9,8	⊙ 12,0
2.2	Evaluierung in der Werkstatt	15,7	⊙ 15,0
3	Administration		
3.1	Allgemeine Verwaltung	6,3	⊙ 7,0
Summe		60,3	⊙ 80,0

Tabelle 7.6: Aufwandserhebung des ALWR1.0-Systems bei Leitprojekten

Aus Tabelle 7.6 geht hervor, dass in einem Folgeprojekt mit einer möglichen Einsparung von 24,6% bei Verwendung von DMB-System bei der Wissenserhebung zu rechnen ist.

Anhand der exemplarischen Modellierung des CP3-Diesel-Systems und des ME7.6-Benzin-Systems kann als Hochrechnung der Aufwand zur Modellierung der Common-Rail-Dieselsysteme und Saugrohreinspritzung-Ottomotoren mit einer Abdeckung von 90% ermittelt werden. Hierbei wird zunächst berechnet, wie viele Modelle für die Beschreibung des Komponenten- und Systemverhaltens notwendig sind. Abschließend wird der Aufwand für die Erstellung der Gesamtmodelle bestimmt.

Die Common-Rail-Dieselsysteme (ab 1998) können in 12 Modellen in Form von Leitprojekten abgebildet und somit diagnostiziert werden ([55]). Es sind circa 100 Komponentenmodelle erforderlich. Da ein Komponentenmodell im Durchschnitt 2,1 Personenstunden benötigt (siehe Tabelle 7.1), beträgt die Erstellungszeit aller benötigten Komponentenmodelle circa 210 Personenstunden. Für die Erfassung der Systeme und Subsysteme sowie die Strukturerstellung der einzelnen Modelle kann ein Aufwand von circa 22,5 Personenstunden pro Modell veranschlagt werden. Insgesamt müssen 480 Personenstunden für die Erstellung der Common-Rail-Modelle aufgewendet werden. Die erstellten Modelle der Common-Rail-Dieselsysteme lassen sich mit geringfügigen Änderungen für die Diagnose von Benzin-Direkt-Einspritz-Systemen (BDE) wieder verwenden.

Mit 16 Modellen können mehr als 90% aller Systeme mit Saugrohreinspritzung (ab 2003) bei Verwendung von DMB-System abgedeckt werden. Für die Modellierung der System-Varianten sind circa 110 Komponenten erforderlich. Hier lassen sich die Komponenten des Diesel-Systems wieder verwenden, so dass maximal 30 Komponenten neu erstellt werden müssen. Wenn hier von einer Erstellzeit von circa 2,1 Personenstunden pro Komponente ausgegangen wird, so wird die Modellbibliothek in circa 63 Personenstunden erweitert. Für die Erzeugung eines Systemmodells werden circa 22,5 Personenstunden veranschlagt, so dass für die Erstellung der Benzin-Saugrohreinspritzung-Modelle insgesamt circa 423 Personenstunden erforderlich sind.

Die Evaluierung hat gezeigt, dass sich mit DMB-System ein Teil des Aufwands für die Wissenserhebung einsparen lässt. Die Höhe der Einsparung wurde durch die Analyse der Erstellprozesse und eine exemplarische Modellierung ermittelt. Die Aussage konnte durch den grundlegenden Vergleich des Erstellprozesses ermittelt und für drei Systeme verifiziert werden. Hochgerechnet auf den Gesamtdatenbestand ist die Ersparnis nur für gleiche Verhältnisse von Leit- und Folgeprojekten gültig. Für die Wissenserhebung mit DMB-System sind allerdings wesentlich weniger Leitprojekte erforderlich. Dafür steigt der Anteil der Folgeprojekte. Die Folgeprojekte erfordern einen geringeren Aufwand. Deshalb wird ein erheblich höheres Einsparpotential bei der Erstellung vieler Systeme vermutet.

Da die Modellerstellung durch einen eingeschränkten Expertenkreis (6 Personen) erfolgte, wurde ein dreitägiger Modellierungsworkshop durchgeführt, um weitere qualifizierte Beurteilungen des entwickelten Modellierungsansatzes durch Expertise der Fehlersuchanleitetter und Systemverantwortlichen zu bekommen. Die Praxistauglichkeit des Ansatzes ist durch die Erstellbarkeit der Modelle durch vorhandene Mitarbeiter mit deren Wissen verifiziert worden. Eine weitere Verifizierung erfolgte durch die Prüfung, ob und wie das Wissen in den existierenden Fehlersuchanleitungen als Modell abgebildet und integriert werden kann. Dabei diente das Modell des automatischen Leuchtweitenregulierungssystems als Beispiel, bei welchem wie folgt vorgegangen wurde ([138]):



- Vorstellung des Modellierungstools (DMB-System)
- Strukturmodellierung, Verhaltensmodellierung
- Integration des heuristischen Diagnosewissens
- Modellprüfung zwecks Qualitätssicherung
- Live-Demo am Fahrzeug mit dem erstellten Modell.

Nach Abschluss des Workshops wurde von den Teilnehmern ein Bericht verfasst. In diesem wurden folgende Aussagen getroffen:

- Die Modellierung ist mit vorhandenem Fachwissen machbar, die Übernahme der Modellierungsaufgaben ist von den Experten gewollt.
- Es existiert ein hoher Initialaufwand zur Erstellung der Modelle. Der Aufwand kann durch Verwendung einer Modellbibliothek deutlich reduziert werden. Die Modellbibliothek soll aufgebaut und kontinuierlich erweitert werden.
- Die Verwendung der bestehenden Daten zur Erstellung der Diagnosemodelle ist möglich.
- Komponenten- und Systemwissen ist für die Modellierung erforderlich und es existiert kein Datenkaufpotenzial. Das Wissen für die Modellierung betrifft hauptsächlich die Verhaltensbeschreibung und die Vernetzung der Komponenten, sprich die Systemstruktur. Da diese Informationen nicht einfach eingekauft werden können und somit auch Wettbewerbern nicht zur Verfügung stehen, ist diese Vorgehensweise ein Alleinstellungsmerkmal für die neue Diagnose.
- Je besser die Onboard-Diagnose die auftretenden Fehlerfälle lokalisiert, desto weniger Aufwand muss in die geführte Fehlersuche investiert werden. Es ist zum heutigen Zeitpunkt so, dass aufgrund begrenzter Rechen- und Speicherkapazitäten nur gesetzlich vorgeschriebene Diagnosefunktionen beziehungsweise die wichtigsten Diagnosefunktionen in Steuergeräten ausgeführt werden. Es ist zu erwarten, dass aus Kostengründen auch zukünftig keine vollständige Onboard-Diagnose zur Verfügung stehen wird.
- Der Einsatz des entwickelten Ansatzes ist sinnvoll, wenn die Symptome durch Kunden erlebbar sind. Insbesondere dann, wenn nicht auf Fehlercodes für den Einstieg in die Fehlersuche zurückgegriffen werden kann und daher die Kundenbeanstandungen als Einstiegsmöglichkeit zur Verfügung stehen, ist der entwickelte Ansatz nützlich.
- Die unerfahrenen Anwender werden besser geführt und erhalten durch Visualisierung der Systemstruktur ein besseres Systemverständnis.
- Die entwickelte Lösung hat das Potenzial zur Beherrschung der Systemkomplexität. Die Systemkomplexität spiegelt sich in dem zunehmenden Vernetzungsgrad der Komponenten wieder. Da die Abbildung der Systemstruktur und damit auch der Vernetzung im Diagnosemodell erfolgt und diese automatisch ausgewertet wird, wird die Vernetzung in der Diagnose beherrschbar.

- Die Fahrzeugbetrachtung aus Funktionssicht und nicht aus Systemsicht ist durch den neuen Ansatz möglich. Bisher werden häufig einzelne Systeme unabhängig voneinander betrachtet, obwohl viele Funktionen im Fahrzeug mittlerweile systemübergreifend realisiert sind. Durch die explizite Modellierung der Verbindungen zwischen den unterschiedlichen Systemen können durch den entwickelten Ansatz auch Wechselwirkungen, die sich auf das Fehlerbild einer bestimmten Funktion auswirken, berücksichtigt werden.

Als Empfehlungen für die Leitung des Servicebereichs wurden folgende Schlussanmerkungen gegeben:

- „Trotz mehr Initialaufwand bei der Erstellung sehen wir, dass der Mehrnutzen durch den entwickelten Ansatz und die Key Features den Mehraufwand deutlich übersteigt.“
- „Modell wird durch den Ersteller visuell aufgebaut, dadurch Übersicht über System erkennbar. Die Darstellung der Wirkkette könnte zentrales Element für die Anleitung werden, insbesondere durch: Einblendung der Einbaulagen, Funktionsbeschreibungen, Steckerbelegungen, Aufruf der Prüfschritte, optische Darstellung der Prüfreihefolge usw.“
- „Die bisherige Programmfunktionalität bei der Erstellung der Fehlersuchanleitungen wird sich in 5 Jahren nicht mehr tragen. Mit dem Zusammenspiel all dieser Bausteine für eine dynamische, geführte Fehlersuche könnten die Servicebereiche einen neuen Produktlebenszyklus bei der Fehlerdiagnose einleiten.“

## 7.2 Evaluierung der Diagnose

Die Leistungsfähigkeit der Diagnose, welche durch DRT-System (Diagnostic Runtime System) bereitgestellt wird, wurde mit verschiedenen Referenzsystemen verglichen. Die Evaluierung erfolgte am Beispiel eines Benzin- und eines Dieselmotorsystems, dabei umfasste das Motorsystem die Subsysteme Luftaufbereitung-, Einspritz-, Zünd-, Verbrennungs-, Abgas- und Kühlsystem. Es wurde darauf geachtet, allgemein gültige Modelle zu erstellen, die nicht für den Feldtest optimiert wurden. Für die Referenzsysteme wurden Systeme und deren Anleitungen der geführten Fehlersuche, die bezüglich der Diagnosegüte von guter Qualität sind, ausgewählt. Das heißt, alle für die Anleitung relevanten Daten liegen vor, sind geprüft und es gibt keine Beanstandungen bezüglich der Diagnosequalität auf Seiten der freien Werkstätten. Die Anleitungen der geführten Fehlersuche sollten es dem Werkstattpersonal ermöglichen, leicht und systematisch Fehlerursachen zu finden. Dabei sollten keine Falschursachen ermittelt werden, da inkorrekte Diagnosen nicht nur zu erhöhten Kosten- und Zeitaufwendungen führen, sondern sich auch negativ auf das Vertrauen in DRT-System und somit in die neu entwickelte Lösung auswirken. Durch die geführte Fehlersuche sollten alle Fehler gefunden werden, um eine vollständige Diagnose zu ermöglichen. Dies ist besonders wichtig, um Folgefehler aufgrund nicht gefundener Defekte und somit höhere Kosten zu vermeiden. Des Weiteren war es wichtig, mit Hilfe der geführten Fehlersuche alle Fehler mit möglichst geringem Aufwand zu finden. Mit der oben genannten Prämisse lassen sich folgende Bewertungskriterien ableiten, die im Rahmen einer Versuchsdurchführung erfasst wurden ([56]):

- Anzahl der Fehldiagnosen - das Diagnosesystem identifiziert eine falsche Fehlerursache

- Anzahl der nicht gefundenen Ursachen - das Diagnosesystem kann die Ursache des Fehlers nicht identifizieren
- Anzahl der erfolgreichen Diagnosen - das Diagnosesystem identifiziert eindeutig den zuvor injizierten Fehler
- Anzahl der durchgeführten Prüfschritte - die Anzahl der Prüfschritte, die benötigt werden, um die Fehlerursache zu lokalisieren
- Durchführungszeiten - die Zeit zur Durchführung der gegebenen Prüfschritte bzw. Prüfanweisungen.

Die Auswahl der zu injizierenden Fehler unter Berücksichtigung verschiedener Fehlerarten erfolgte in Abstimmung mit Werkstattmitarbeitern. Zudem wurden die erzeugten Fehler erst vor Ort von den unabhängigen Werkstattmitarbeitern festgelegt. Somit waren ähnliche Ergebnisse auch bei Vergleichen mit anderen Fahrzeugsystemen zu erwarten. Folgende Fehler wurden in den CP3-Dieselmotor eingebaut:

- D1: Mechanischer Defekt des Druckregelventils (über Ersatzlast)
- D2: Kraftstoffleitung zwischen Kraftstofffilter und Hochdruckpumpe verengt
- D3: Ansteuerung der Zumesseinheit unterbrochen
- D4: Verstopfung zwischen HFM-Sensor und Turbolader.

Für den ME7.x-Ottomotor wurden ebenfalls vier Fehler injiziert:

- O1: Ansteuerung des Injektors 1 unterbrochen
- O2: Signal des HFM-Sensors verfälscht (R-800Ohm)
- O3: Signal des Ansaugluft-Temperatur-Sensors auf konstant 120°C verfälscht (R-120Ohm)
- O4: Signal der Lambdasonde verfälscht (Übergangswiderstand).

Versuch	Eingebauter Fehler	DRT-System			
		Nr.	Symptom, Messung, Prüfung	Zeit[min]	Ergebnis
O4	Signal der Lambdasonde verfälscht, Übergangswiderstand Signalleitung				
		1	Leerlaufprobleme (Drehzahl, Abgas)		
		2	Eigendiagnose	2	keine DTC
		3	Luftmassenmesser Istwert	2	i.O.
		4	Lambdasonde B1	2	Signal der

				Lambdasonde entspricht nicht den Vorgaben
	5	Lambdasonde B1 Signalspannung prüfen	3	Übergangs- widerstand Signalleitung

Tabelle 7.7: Protokoll des Versuchs mit DRT-System für den ME7.x-Ottomotor

Für die Versuchsdurchführung wurde zunächst mit DRT-System diagnostiziert und anschließend der Fehler anhand der Vorgaben der existierenden Diagnosetools gesucht. Um den Aufwand für die Prüfzeiten zu ermitteln, wurde die Dauer jedes Prüfschrittes ermittelt beziehungsweise geschätzt. Die Zeiten hängen stark vom jeweiligen System oder Fahrzeug ab. Die Prüfzeit eines Prüfschrittes ist für beide Diagnosesysteme gleich und wird für beide Diagnosesysteme gleich bewertet.

In Tabelle 7.7 stehen die detaillierten Prüfungsschritte, die bei dem Diagnosevorgang eines ME7.x-Ottomotors mit dem Symptom „Leerlaufprobleme (Drehzahl, Abgas)“, um die Ursache beziehungsweise die defekten Komponente zu finden, benötigen wurden. Hierbei brauchte das entwickelte DRT-System fünf Schritte und insgesamt neun Minuten, dagegen benötigten existierende Diagnosesysteme im Durchschnitt dreizehn Schritte und fünfunddreißig Minuten (Tabelle 7.8). Die Unterschiede bei den Prüfungsschritten und Prüfzeiten liegen daran, dass in dem entwickelten Ansatz die einzelnen Systeme nicht unabhängig voneinander betrachtet werden. Die Wechselwirkungen zwischen den Systemen wurden durch die Modellierung der Verbindungen zwischen den unterschiedlichen Systemen berücksichtigt. Damit erfolgt die Diagnose aus Funktionssicht und nicht aus Systemsicht.

Versuch	Eingebauter Fehler	Existierendes Diagnosesystem			
O4	Signal der Lambdasonde verfälscht, Übergangswiderstand Signalleitung	Nr.	Symptom, Messung, Prüfung	Zeit[min]	Ergebnis
		1	Leerlaufprobleme (Drehzahl, Abgas)		
		2	Eigendiagnose	2	keine DTC
		3	Istwerte Drossel-	2	i.O.
		3	klappensteuereinheit		
		4	Drosselklappen- Stellmotor	1	i.O.
		5	Luftmassenmesser Istwert	2	i.O.
		6	Stellgliedtest Drallklappen	2	i.O.
		6	Unterdruckleitung	2	i.O.

von Drallklappe				
8	Temperatursensor Istwert	2	i.O.	
9	Einspritzzeit	4	i.O.	
10	Einspritzventile	6	i.O.	
11	Kraftstoffdruck Istwert	2	i.O.	
12	Kraftstoffverdunst. -Regenerationsventil	5	i.O.	
13	Lambdasonde B1	2	Signal der Lambdasonde entspricht nicht den Vorgaben	
14	Lambdasonde B1 Signalspannung prüfen	3	Übergangs- widerstand Signalleitung	

Tabelle 7.8: Protokoll des Versuchs mit einem existierenden Diagnosesystem für den ME7.x-Ottomotor

Tabelle 7.9 gibt die entscheidenden Versuchsergebnisse wider. Dabei ist die Durchführungszeit der Messungen beziehungsweise Prüfschritte in Minuten angegeben und die Einsparung als prozentuale Ersparnis der Prüfzeit von DRT-System gegenüber Referenzsystemen.

Versuch	DRT-System		Referenzsysteme	
	Anzahl der Prüfschritte	Durchführungs- zeit [min]	Anzahl der Prüfschritte	Durchführungs- zeit [min]
D1	6	9	14	26
D2	6	18	5	24
D3	3	5	3	5
D4	4	7	6	13
O1	3	4	3	4
O2	7	12	8	25
O3	7	18	5	19
O4	5	9	14	35

Tabelle 7.9: Zusammenfassung der ersten Evaluierungsergebnisse

Es sind folgende Aussagen festzustellen:

- Alle Diagnosesysteme fanden alle Fehler. Dies war allerdings nur bei Einstieg in die Diagnose mit korrekter Kundenbeanstandung der Fall. Eine falsche Kundenbeanstan-

ung führte dazu, dass die Fehlerursache mit Hilfe der Referenzdiagnosesysteme nicht gefunden werden konnte. Bei DRT-System war die Fehlersuche nicht erfolglos, die dafür benötigte Zeit jedoch bedeutend länger.

- Es kam zu keiner Fehldiagnose während der Evaluierung. Das bedeutet, es wurde keine intakte Komponente als fehlerhaft ausgewiesen und ausgetauscht.

Aus den Ergebnissen geht hervor, dass bei der Diagnose mit DRT-System eine mittlere Reduzierung der Prüfzeit um 45% (Tabelle 7.9) gegenüber den Referenzsystemen erreicht wurde. Der maximale Wert lag bei 74% und der minimale Wert bei 0%. Die Reduzierung der Prüfzeit hat in der Werkstatt eine immense Kostenminimierung zur Folge.

Die Reduzierung der Prüfzeit ergibt sich durch eine optimale Reihenfolge der Prüfschritte, die dynamisch von DRT-System berechnet und dem Werkstattmitarbeiter vorgeschlagen wird. Daher lässt sich auch die minimale Reduzierung der Prüfzeit (Versuch D3 und O1) erklären: In diesen Fällen zeigte ein Fehlercode auf die defekten Komponenten. Dieser Fehlercode wurde frühzeitig abgerufen und der Prüfvorgang nach der Komponentenprüfung abgeschlossen. Hierbei ergaben sich keine Unterschiede in den Vorgehensweisen der Systeme.

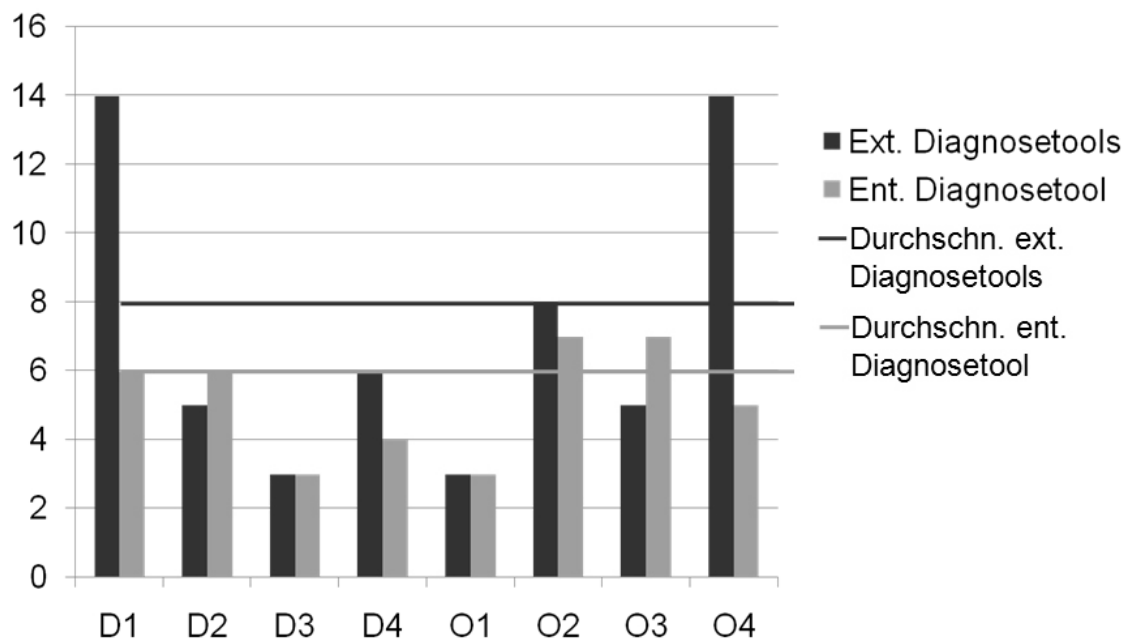


Abbildung 7.1: Ergebnisse der Prüfschritte bei der Evaluierung

In einer Vielzahl der Fälle hatte der optimierte Prüfablauf auch eine Reduzierung der Prüfschritte zur Folge. Dies muss aber nicht immer der Fall sein (Abbildung 7.1 und 7.2 - Versuch D2 und O3), da auch mehrere kleine Schritte statt eines komplexen, lang andauernden Prüfschrittes zum Ziel führen.

Bei dem Diagnosevorgang war festzustellen, dass in den Referenzsystemen schnell zur Prüfung von Komponenten übergegangen wird. DRT-System hingegen versucht zunächst über die Erfassung von Istwerten und anderen Messergebnissen die defekte Komponente einzugrenzen, bevor Komponentenprüfungen den Verdacht verifizieren. Dieses Vorgehen lässt sich in Referenzsystemen sehr schwer realisieren, da in den meisten Referenzsystemen Fehlersuchbäume verwendet werden. Damit ein Referenzsystem für jedes Messergebnis eine entsprechende Handlung vorschlagen kann, würde ein Suchbaum schnell sehr komplex werden.

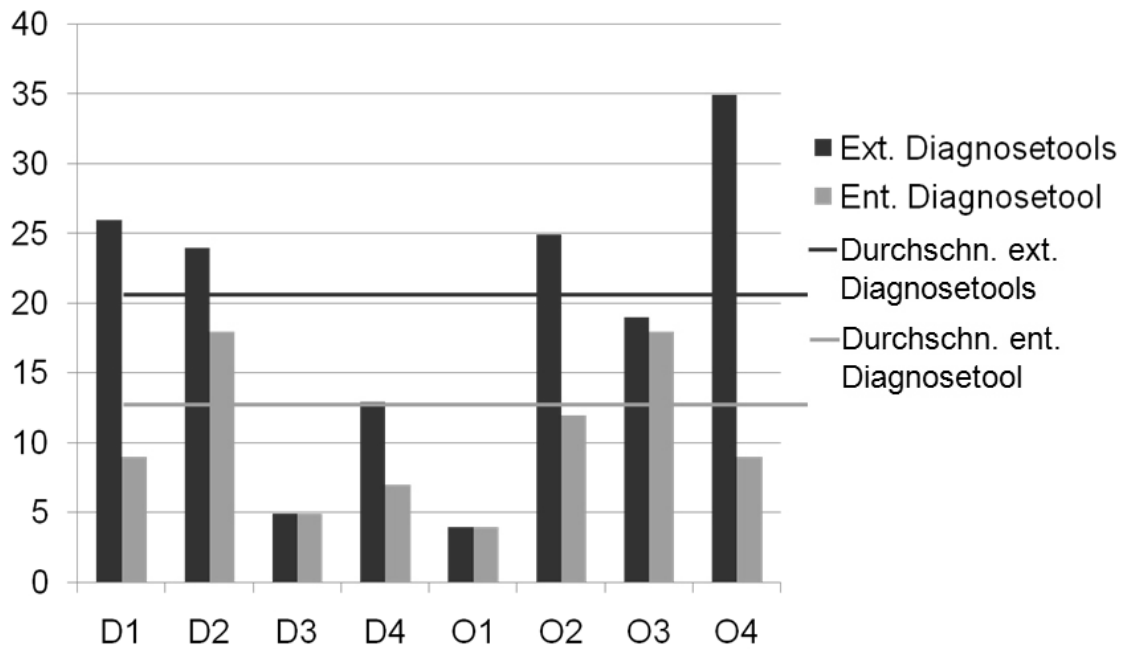


Abbildung 7.2: Ergebnisse der Prüfzeit bei der Evaluierung

Als ein weiterer Vorteil stellte sich heraus, dass DRT-System jeden Prüfschritt vorgab und erst nach vollständiger Abarbeitung der vorgeschlagenen Prüfschritte zur Liste der verdächtigen Komponenten mit ihren Komponenten-Tests überging. Im Gegensatz dazu wird in den Referenzsystemen nach jedem Diagnoseschritt eine ähnliche Liste mit den weiteren Fehlermöglichkeiten angeführt. Hier obliegt es dem Werkstattmitarbeiter, ob er dieser Liste oder dem nächsten Prüfschritt folgen soll. Der Umstand, dass alle Prüfschritte vom System vorgegeben werden, ermöglicht es auch Personal, das weniger mit dem fehlerhaften System vertraut ist, effizient eine Diagnose durchzuführen.

Im Anschluss an die erste Evaluierung wurde die Leistungsfähigkeit von DRT-System durch die beteiligten Experten und Werkstattmitarbeitern folgendermaßen beurteilt:

- Die Diagnosegüte beziehungsweise Diagnosequalität hinsichtlich der Diagnosegenauigkeit und -vollständigkeit wurden anerkannt und herausgehoben.
- DRT-System ist für die freie Werkstatt gut geeignet, da die Erfahrung und das Systemwissen des freien Werkstattpersonals nicht ausreicht, um bei der Vielfalt der Fahrzeuge, die in die freie Werkstatt kommen, jeden Fehler ohne weitere Hilfsmittel zu entdecken. Insbesondere die in der Evaluierung erzeugten Fehler ohne Fehlercodeeintragung sind nur von einem erfahrenen Werkstattmeister, der sich tagtäglich mit solchen Fehlern auseinandersetzen muss und sich mit dem Autotyp bzw. den eingebauten Systemen auskennt, zu finden. Das Personal der freien Werkstatt jedoch, das in drei Monaten vielleicht ein- oder zweimal mit solchen Fehlern konfrontiert wird, hat keine Chance den Fehler zu finden und zu beheben.
- Der Vorteil des DRT-Testers im Gegensatz zu heutigen Testern sind die Vorschläge der Handlungsempfehlung und die Liste der verdächtigen Komponenten, die für den Diagnoseschritt relevant sind. Das Werkstattpersonal muss nicht überlegen, welche Istwerte ausgelesen, beziehungsweise welche Komponente getestet werden sollte. Die dadurch

gesparte Zeit ist sehr wichtig, weil sie die Kundenzufriedenheit steigert und Kosten reduziert.

- DRT-System kann ohne konkrete Soll- und Ist-Werte eine Diagnose für existierende Systeme erstellen. Die Plausibilisierung der Messergebnisse obliegt dabei dem Werkstattmitarbeiter. Die nötige Diagnosetiefe kann deshalb nicht immer garantiert, aber immerhin können neue Fahrzeuge beziehungsweise Systeme damit diagnostiziert werden.
- Es existiert kein Tool, das eine solche Diagnoseleistung, insbesondere eine systemübergreifende Diagnose erbringt. Bei einigen existierenden Diagnosesystemen ist eine systemübergreifende Diagnose nur möglich, wenn Fehlercodes gesetzt worden sind. Hierbei versuchen solche Systeme die Fehlercodes miteinander zu kombinieren. Als Ergebnisse erhält man die verdächtigen Systeme oder Komponenten, die sich mit Hilfe der vorhandenen Fehlercodeeinträge am besten erklären lassen.

Als kritische Punkte wurden die Bedienung und die Übersichtlichkeit der Testeroberfläche angesehen. Für den Einsatz in der Werkstatt müsste die Bedienung einfacher gestaltet und die Übersichtlichkeit verbessert werden. Die erste Evaluierung hat gezeigt, dass der entwickelte Diagnoseansatz und DRT-System in der Werkstatt die Diagnoseanforderungen bezüglich Korrektheit und Vollständigkeit erfüllt haben. Bei der Kostenminimierung bezüglich der Diagnosedauer und damit dem Aufwand der Fehlersuche konnte gegenüber Referenzsystemen durch die Reduzierung der Prüfzeit eine erhebliche Einsparung erreicht werden. Diese erzielte Kostenminimierung und die verbesserte Nutzerführung mit optimierter Abfolge der Prüfschritte führen zu einer schnelleren Diagnose, von der in erster Linie die Werkstätten profitieren und damit die Marktpositionierung der Diagnosehersteller sichern.

Die zweite Evaluierung ([41]) sollte weitere qualifizierte Beurteilungen der Leistungsfähigkeit, Praxistauglichkeit und des Kundennutzens des entwickelten Ansatzes durch Werkstattmitarbeiter und Fachexperten mit umfangreichen Werkstatterfahrungen bringen. Hierfür wurden insgesamt achtzehn Teilnehmer mit unterschiedlichem Profil ausgesucht:

- Mitarbeiter aus freien Werkstätten und Vertragswerkstätten sowie der Hotline
- Trainer aus dem Diagnoseschulungszentrum
- Autoren der Fehlersuchanleitungen
- Mitarbeiter verschiedener Gruppen und Abteilungen des Servicebereichs.

Die drei Diagnosemodelle, die zur Verfügung stehen, decken folgende Fahrzeuge ab ([42]):

System	Typ	Variante
ALWR1.0	VW-Touran	1.4 TSI; 1.6[i,FSI]; 1.9 TDI; 2.0 [i,Eco Fuel,FSI,TDI]
	VW-Golf V	1.4[i,Variant,FSI,FSI Variant,TSI,TSI Variant]
		1.6[i,Variant,FSI]; 1.9 [TDI,TDI Variant,TDI 4Mot]
		2.0 [FSI,FSI 4Mot,GTI,GTI Variant,SDI]
	VW-Golf Plus	2.0 [TDI,TDI Variant,TDI 4Mot]; 3.2 R32 4 Mot
SEA-Altea	1.4 [i,FSI,TSI]; 1.6[i,FSI]; 1.9 TDI; 2.0 [FSI,TDI]	
		1.4[i,TSI]; 1.6; 1.8 TSI; 1.9 TDI
		2.0 [FSI,T FSI,T FSI Free,TDI,TDIFree]



	SEA-Leon	1.4[i,TSI]; 1.6; 1.8 TSI; 1.9 TDI; 2.0 [FSI,TFSI,TDI]
	SEA-Toledo	1.4[i,TSI]; 1.6;1.8 TSI;1.9 TDI; 2.0[FSI, TDI,TFSI]
	SKO-Octavia	1.4[i,Combi]; 1.6[i,FSI,Combi,FSI Comb] 1.9 [TDI, TDI 4X4, TDI Comb]; 2.0 [Combi RS] 2.0[FSI Comb, TDI Comb, FSI 4X4, RS, TDI]
CP3	CDI Vito	109[kW: 65, 70] ; 111[kW:80, 85]; 115
	CDI Sprinter	209; 211; 213; 215; 309; 311; 313; 315 411; 415; 509; 511;515
	C 200	CDI[Sportcoupe, i kW:75,90,100, T-Modell kW:75,90,100]
	C 220	CDI[Sportcoupe, i kW:100,125, T-Modell kW:110,125]
	CLC 200	CDI[kW:90,110]
	CLK 220	CDI Coupe[kW:110]
	E 200	CDI[T-Modell, kW:75,90,100]
	E 220	CDI[T-Modell kW:100,110,125, i kW:100,110,125]
	Viano	CDI 2.0[kW:80,85,110]
ME 7.6.x	Agila Twinport	1.0; 1,2
	Corsa Twinport	1.0[C,D]; 1.2i[C,D]; 1.4[C,D]
	Astra Twinport	1.2 C; 1.2 i; 1.4 [G,H]
	Astra	1.4 Caravan Tw[G,H]; 1.4 GTC Twinpo[H]
	Sonstiges	Combo 1.4i; Meriva 1.4i; Tigra 1.4i

Tabelle 7.10: Liste der abgedeckten Fahrzeuge

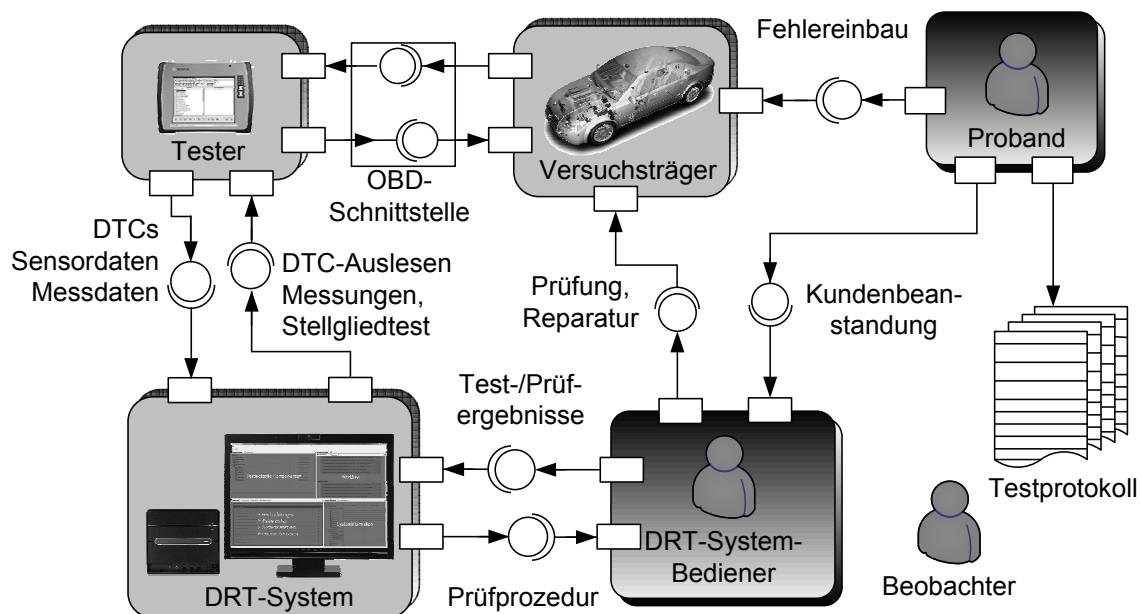


Abbildung 7.3: Szenario des zweiten Feldtests

In Abbildung 7.3 ist das Szenario beziehungsweise die Vorgehensweise des zweiten Feldtests dargestellt. Die Aktivitäten können grob in drei Phasen aufgeteilt werden:

- Erste Phase: Der Proband baute reale Fehler in Abwesenheit des Bediener von DRT-System in den Versuchsträger ein. Die Kundenbeanstandungen wurden von dem Pro-

banden benannt und in ein Protokoll eingetragen.

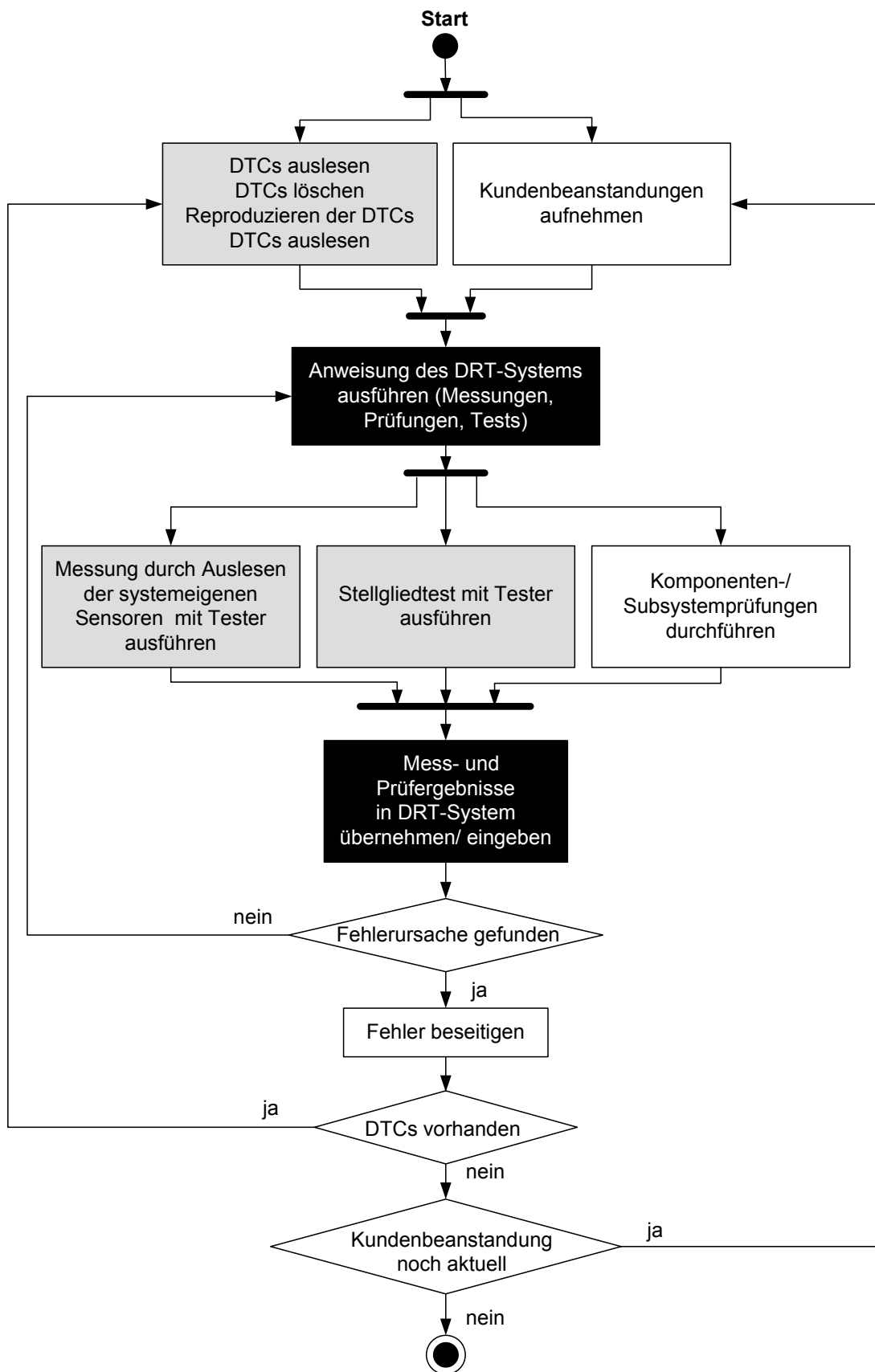


Abbildung 7.4: Diagnoseablauf bei dem zweiten Feldtest

- Zweite Phase: Die Diagnose erfolgte mit DRT-System (Abbildung 7.4). Dabei befolgte der Bediener genau die Anweisungen von DRT-System und der Proband protokollierte die Schritte. Das Auslesen der Fehlercodes sowie der Istwerte der Systemsensoren und die Durchführung des Stellgliedertests erfolgte mittels eines Diagnosetesters, welcher durch die OBD-Schnittstelle an das Fahrzeug gebunden wurde. Bei dem Diagnosevorgang wurde durch den Beobachter die Objektivität der Versuchsdurchführung und deren Ergebnisse garantiert.
- Dritte Phase: Nach Ablauf eines Versuchs wurde der Diagnosevorgang durch den Probanden mit einem Fragebogen (Abbildung 7.5) bewertet.

**Fragebogen-Nr.:** 1

**Name:** Dennis Behreucht

**Datum:** 30.07.09

**Uhrzeit:** 15:10

**Ort:** P1

**Fahrzeug:** Opel Corsa 1.2D

- Beschreibung der Komponentenmanipulation:  
*Verstopfung des Ansaugkanals vor der Dieseleinspritzung*
- Beschreibung des Fehlerbildes / der Symptome / der Kundenbeanstandung:  
*Motor startet und geht gleich wieder aus.*

**Beschreibung der Vorgehensweise bei der Fehlersuche:**

ExpD

Nr	Prüfung	Ergebnis	Bemerkung
1	Kundenbeanstandung	i.O.	
2	DTC	keine	
3	Istwert Durchmesser	n.i.O.	
4	Dieseleinspritzung geprüft, elektrische	i.O.	
5	Drehmoment Dieseleinspritzung	i.O.	
6	Luftdruck vor der Dieseleinspritzung	i.O.	
7	Leistungswert von Dk bis Dieseleinspritzung	i.O.	
8	Leistungswert zwischen Dieseleinspritzung und Dieseleinspritzung	n.i.O.	
9			

**Auswertung:** 1

1. Wird die defekte Komponente gefunden?  
Falls nein, warum nicht?

ja  nein

2. Wie lange dauert die Suche?

eher kurz  normal  eher lang

3. Erfolgt die Fehlersuche in der erwarteten Reihenfolge?  
Falls nein, warum nicht?

ja  nein

4. Hätten Sie mit Ihrer Erfahrung die Suche anders durchlaufen?  
Falls ja, bitte begründen an welcher Stelle!

ja  nein

5. Wie bewerten Sie die Reparaturkosten?  
(Bitte einkreisen und ggf. begründen!)

eher niedrig  normal  eher hoch

**Fragebogen-Nr.:** 1

6. Wurden während der Diagnose neue Erkenntnisse gewonnen?  
Falls ja, welche?

ja  nein

7. Wie bewerten Sie den Ablauf insgesamt?  
(Bitte einkreisen und ggf. begründen!)

eher zufrieden  eher nicht zufrieden  weiß nicht

8. Wie bewerten Sie den Nutzen und die Akzeptanz des vorgestellten Diagnoseansatzes? (Bitte einkreisen und ggf. begründen!)

- hilfreich / nicht hilfreich / weiß nicht
- ermöglicht eine / keine Verbesserung des Systemverständnisses

9. Sonstige Anmerkungen:

Abbildung 7.5: Beispiel eines Versuchprotokolls bei der zweiten Evaluierung

Der Evaluierungszeitraum des Diagnosetools am Fahrzeug mit realen Fehlern aus dem Feld betrug zwei Wochen und die Versuche wurden an drei Fahrzeugen durchgeführt:

- Mercedes Benz 220CDI (CP3-DS)
- Opel Corsa 1.2 (ME7.6 -GS)

- Golf V 1.4 Variant (ALWR1.0).

Als Ergebnisse des zweiten Feldtests existieren fünfundvierzig auswertbare Fragebogen mit einunddreißig unterschiedlichen Fehlerfällen. Die vierzehn Fehlerüberlappungen wurden von verschiedenen Probanden eingebaut. Dies lässt sich nicht vermeiden, weil die Versuche unabhängig voneinander durchgeführt wurden und bei diesem Feldtest stets versucht wurde, Fehler ohne Fehlercode zu initialisieren. Die Fehlerfälle enthalten:

- Neun Fehler mit eindeutigem Fehlercode
- Einen Fehler mit nicht eindeutigem Fehlercode
- Einundzwanzig Fehler ohne Fehlercode (in Leerlauf und Teillast)
- Fünf doppelte Fehler.

Bei Fehlerfall mit eindeutigen Fehlercodes wurde im ersten Durchlauf des Versuchs eine Diagnose ohne Berücksichtigung der existierenden Fehlercodes aufgestellt, das heißt, es wurden nur die Kundenbeanstandungen beziehungsweise Symptome als Diagnoseeinstieg herangezogen und die Fehlercodes nicht berücksichtigt. Im zweiten Durchlauf wurden die Fehlercodes als Einstieg in die Diagnose verwendet. Diese Vorgehensweise diente zur Bestätigung des Vorteils des entwickelten Ansatzes; dass der Diagnosevorgang auch ohne Fehlercodes beziehungsweise mit unbekanntem Fehlercodes funktioniert. Diese Eigenschaft ist wichtig für die freie Werkstattdiagnose, weil die Fehlercodes, die nicht abgas- oder sicherheitsrelevant sind, von Fahrzeugherstellern nicht freigegeben werden müssen. Die Fehler mit nicht eindeutigem Fehlercode „Grundeinstellung nicht möglich“ entstanden bei den Versuchen mit dem ALWR1.0-System. Bei solcher Art des Fehlercodes erfolgte die Diagnose ohne dessen Berücksichtigung. Die doppelten Fehler wurden unabhängig voneinander erzeugt. Das bedeutet, dass keine Folgefehler in diesem Feldtest existierten.

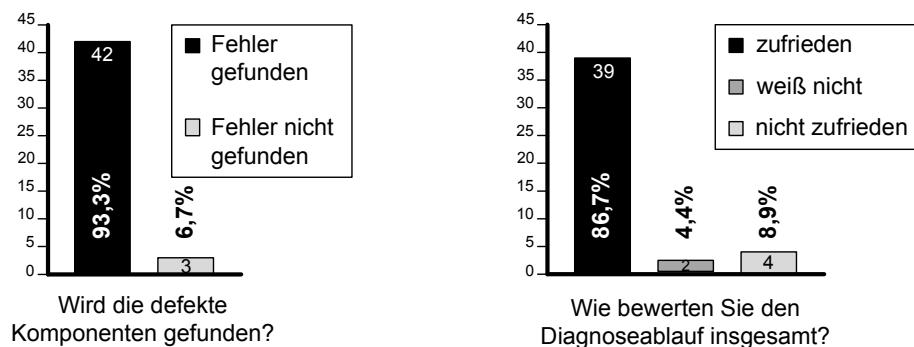


Abbildung 7.6: Feldtestergebnisse zur Auswertung der Diagnosefähigkeit

In drei Fällen wurden die defekten Komponenten nicht gefunden (Abbildung 7.6). Folgende Gründe sind für die erfolglosen Fehlersuchen verantwortlich ([43]):

- Nicht gefundene Komponenten sind nicht in verfügbaren Unterlagen enthalten und daher nicht im Modell abgebildet, zum Beispiel: Es existiert keine Luftleitung zwischen Drucksensor und Saugrohr im CP3-System.

- Die Festlegung der Systemgrenze von Modellierer und Proband sind unterschiedlich, zum Beispiel beinhaltet das Diagnosemodell nicht die elektrische Sicherung für das ALWR-Steuergerät und den Steckverbinder zwischen Batterie und dem ALWR-Steuergerät.

Die Ergänzung der fehlenden Komponenten wurde vor Ort vorgenommen. Nach der Erweiterung der Modelle funktionierten die Diagnosevorgänge problemlos und die defekten Komponenten wurden gefunden. Die Unzufriedenheit mit dem Diagnoseablauf beruhte zum einen auf den nicht gefundenen defekten Komponenten und zum anderen auf nicht eindeutige Fehlercodes, da die Probanden, die aus Vertragswerkstätten kamen, eine präzise Erklärung in Form von Fehlerpfad und Fehlerart des Fehlercodes erwarteten. Diese Problematik ist bei der freien Werkstattdiagnose bekannt und kann nur durch gesetzliche Regelungen verbessert werden, da die Fahrzeughersteller großes Interesse haben, ihr eigenes Diagnosegerät zu verkaufen. Ein weiteres Defizit ist die Darstellung der Einbaulage der Komponenten, hier soll in Zukunft die Einbaulage mit Bildern dargestellt werden.

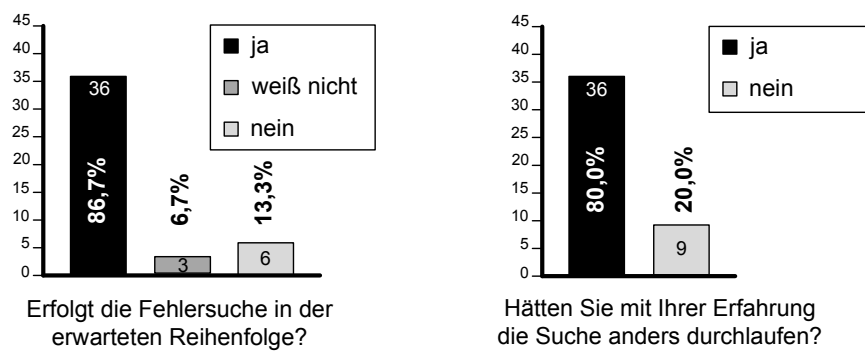


Abbildung 7.7: Feldtestergebnisse zur Überprüfung der vorgeschlagenen Prüfschritte

Wegen der fehlenden Komponenten im Modell und des Verzichts auf das Wissen über Ausfallwahrscheinlichkeiten wurden in sechs Fällen die Diagnosedauer als „lang“ und die erwartete Reihenfolge als „nicht optimal“ bewertet (Abbildung 7.7, 7.8). In den meisten Fällen, in denen die Dauer der Diagnose im Vergleich zum Werkstattalltag als normal empfunden wurde, erzeugte der Fehler einen eindeutigen Fehlercode. Bei Fehlern mit nicht eindeutigen Fehlercode beziehungsweise ohne Fehlercode ist die Stärke des entwickelten Ansatzes durch die erwartete Reihenfolge und die damit verbundene Diagnosedauer deutlich zu erkennen.

Die Reparaturkosten, die hier abgefragt wurden (Abbildung 7.8), ergeben sich aus der Diagnosedauer und den Prüfmittelkosten (zum Beispiel: Lecksuchspray, elektronische Ersatzsicherung, Überbrückungskabel, etc.).

Aus diesem Grund stimmen die Ergebnisse bei der Reparaturkostenfrage mit den Ergebnissen der Diagnosedauer fast zu hundert Prozent überein. Die Fälle mit höheren Reparaturkosten sind analog zur längeren Diagnosedauer durch nicht modellierte Komponenten, Verzicht auf Erfahrungswissen und Ungenauigkeit der Diagnosewissensbasis über Fehlercodes begründet.

Folgende Erkenntnisse wurden während der Diagnose gewonnen (Abbildung 7.9):

- Die Darstellung der Wirkkette in der Erklärungskomponente ist nicht nur für Werkstattpersonal, das wenig Erfahrung mit dem zu diagnostizierenden System hat, sondern auch für erfahrene Mitarbeiter eine große Hilfe.

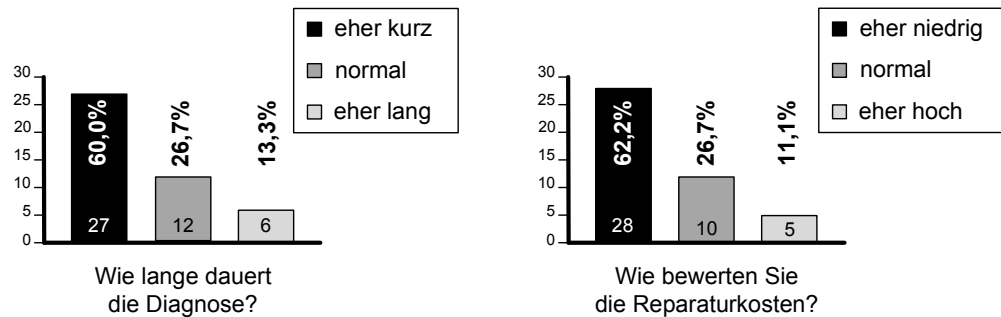


Abbildung 7.8: Feldtestergebnisse zur Auswertung der Effektivität des entwickelten Ansatzes

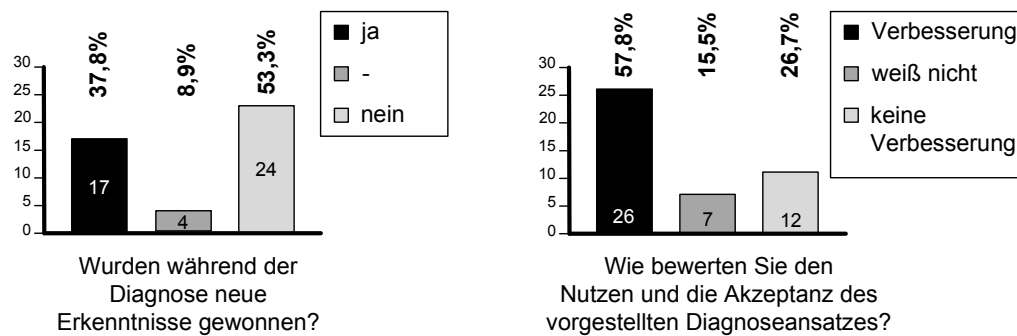


Abbildung 7.9: Feldtestergebnisse zur Überprüfung des Nutzens und der Akzeptanz des entwickelten Ansatzes

- Die vorgeschlagenen Messungen beziehungsweise Prüfungen beschleunigen die Fehlersuche, da der Nutzer nicht selbst suchen und entscheiden muss, welche Schritte sinnvollerweise als nächste durchzuführen wären.

Der Nutzen und die Akzeptanz des entwickelten Diagnoseansatzes wurden als Verbesserung hinsichtlich Diagnosetiefe, Effizienz, Benutzerführung und Fehlererklärung bewertet. Aus neu eingebauten Fehlern mit eindeutigen Fehlercode, die direkt auf die defekte Komponente zeigten und drei Fällen, in denen die defekte Komponente nicht im Modell enthalten war, entstanden zwölf Fälle, die keine Verbesserung durch den vorgestellten Ansatz hervorbringen konnten. Sieben Fälle, bei denen mit „weiß nicht“ geantwortet wurde, stammen zum Teil aus der Befragung der Vertragswerkstattmitarbeiter und zum Teil von Autoren, die die Fehlersuchanleitungen schreiben und durch ihre Erfahrung die fehlerhafte Komponente möglicherweise schneller hätten finden können. In vielen Fällen, in denen die Komponentenfehler keine Fehlercodes hervorriefen, mussten selbst die Vertragswerkstattmitarbeiter und die Fehlersuchanleitungsautoren zugeben, dass der entwickelte Ansatz eine große Verbesserung und Erleichterung bei der Fehlersuche gebracht hat.

## 8 Zusammenfassung und Ausblick

In diesem Kapitel werden in Abschnitt 8.1 die Ergebnisse der Arbeit zusammengefasst. Dabei wird die Lösung aus verschiedenen Sichtweisen bewertet. Diese Bewertung beruht auf den Ergebnissen der durchgeführten Feldtests, der Modellierungswshops und der Aufwandsabschätzung durch Autoren und Techniker, die die Diagnosemodelle erstellt haben. Obwohl die Arbeit als abgeschlossen gilt, existieren weitere Möglichkeiten, um die Erstellung des Diagnosemodells und den Diagnosevorgang zu verbessern. In Abschnitt 8.2 wird zuerst ein Ausblick über die weiterführende Arbeit im Bereich der Wissenserhebung vorgestellt. Im Anschluss wird eine der Möglichkeiten zur Integration der entwickelten Lösung im Zusammenhang mit der fallbasierten Diagnose kurz erläutert.

### 8.1 Zusammenfassung

Die Grundlage qualitativer Diagnosemodellbildung im Verbund mit heuristischem Wissen, die Mechanismen zur Auswertung dieses Diagnosewissens und die Implementierung wurden in der vorliegenden Arbeit beschrieben. Die Validierung des entwickelten Ansatzes erfolgte in mehreren Schritten und dient als Bewertungsbasis. Es existieren Ergebnisse aus folgenden durchgeführten Aktivitäten:

- Modellerstellung, Aufwandserhebung und Aufwandsabschätzung durch Autoren und Techniker, die die bisherigen Fehlersuchanleitungen erstellt haben, Ende 2007 und Mitte 2009
- Zwei Modellierungswshops Anfang 2008 und Mitte 2009
- Drei Feldtestdurchführungen:
  - Feldtest 2007 mit Diesel- und Benzinsystem
  - Feldtest 2009 mit Licht- und Benzinsystem
  - Feldtest 2009 mit Licht-, Diesel- und Benzinsystem
- Vier Live-Demos im Jahre 2009.

Aus interner Sicht des Servicebereichs existieren eine Reihe von Kriterien zur Bewertung des Lösungsansatzes, die mit der Bewertungsbasis untersucht und ausgewertet wurden:

- Erstellungsaufwand und Variantenvielfalt: Die Zeit und Kosten für Beschaffung, Aufbereitung, Erstellung und Pflege des Diagnosewissens ist selbst bei dem benötigten Initialaufwand vertretbar. Mit sukzessivem Aufbau einer Modellbibliothek verringert sich der Erstellungsaufwand mit der Zeit. Die Modellierung von Varianten erfolgt objektorientiert; durch die Eigenschaften der Generalisierung, Spezialisierung und Vererbung ist der Grad der Wiederverwendung von Modellwissen hoch.

- **Robustheit und Abdeckungsgrad:** Die Sensitivität der erstellten Modelle bezüglich Serienstreuung und deren Übertragbarkeit auf andere Fahrzeuge wurde durch die Feldtests bestätigt. Der Umgang mit fehlenden Informationen und die Integrierbarkeit unterschiedlicher Informationsquellen bei der Modellerstellung wurden durch die Kombination von Modellwissen und heuristischem Wissen bewiesen.
- **Validierungsmöglichkeit und Überprüfbarkeit:** Einzelne Modellteile können mit der entwickelten Überprüfbarkeit separat getestet werden. Das gesamte Diagnosemodell ist im Simulationsmodus validierbar. Die automatisierten Tests des gesamten Diagnosemodells können angestoßen werden, sobald die Testfälle definiert sind. Diese Testfälle können nicht nur durch Experten definiert werden, sondern auch durch Analyse der Feedback-Informationen aus dem Feld gewonnen werden. Eine weitere Validierungsmöglichkeit ist die Prüfung der Auswirkungen von Fehlern am Fahrzeug und deren Übereinstimmung beim Analysemodus, da in diesem Modus Fehler gesetzt werden können.
- **Performance und Ressourcenbedarf:** Der erforderliche Speicherbedarf der Modelle, um eine Abdeckung von 96% aller Fahrzeuge auf dem Markt zu erreichen, beträgt in der aktuellen Abschätzung ca. 4,5 GB. Auf existierender Hardware (2,1 GHz und 2,0 GB RAM) benötigte der Diagnosevorgang pro Auswertungsschritt zwei bis fünf Sekunden.
- **Kompetenzen und Rollen:** Das erforderliche Fachwissen zur Modellerstellung ist in dem existierenden Autorenteam vorhanden. Die externe Beauftragung der Modellbildung ist nach Meinung der Experten, die bereits mit externen Autoren gearbeitet haben, möglich. Die Anpassung und Erweiterung der existierenden Prozesse und Rollen in der Wissenserhebungsphase ist machbar und fordert keine Strukturänderung im Servicebereich.
- **Altdatenmigration und Selbstlernpotenzial:** Die einhundertprozentige Nutzbarkeit bestehender Diagnosedaten ist bestätigt worden. Wegen des hybriden Ansatzes ist die Migration der Altdaten in die erstellten Modelle nahtlos und fordert kaum Aufwand. Mit der Möglichkeit zur Nutzung der heuristischen Informationen, zum Beispiel der Ausfallwahrscheinlichkeit der Komponenten beziehungsweise der Abhängigkeit zwischen Fehlern und Symptomen und der Anpassung der Benutzerführung in Form von Benutzerprofilen hat das Diagnosetool ein großes Selbstlernpotenzial.
- **Wissensaufbau und Wissenserhaltung:** Steigerung des Systemverständnisses der Autoren durch Modellierungsaktivitäten, zum Beispiel Struktur- und Verhaltensmodellierung. Das Diagnosewissen ist besser nachvollziehbar und dokumentierbar durch die Strukturierung des Wissen mittels funktions- beziehungsweise komponentenorientierter Paradigmen.
- **Zukunftssicherheit:** Mit dem Sichtkonzept ist die Flexibilität und Erweiterbarkeit der Datenstrukturen für Zukaufdaten und Feedback-Informationen garantiert. Der Modellierungsansatz ermöglicht verschiedene Sichtweisen auf das Fahrzeug: Systemsicht, Steuergerätesicht und Funktionssicht. Nur mit dieser Möglichkeit ist die Diagnose in der Werkstatt in der Zukunft durchführbar.
- **Vermarktbarkeit:** Die schrittweise Vermarktung einzelner Aspekte des Diagnoseansatzes ist möglich. Die Sichtbarkeit der Veränderungen und des Nutzens für den Kunden im Diagnosevorgang eröffnet weitere Möglichkeiten der Markterschließung.



- **Produktreife und Umsetzungsaufwand:** Es existiert ein Initialaufwand zur Realisierung des Wissenserhebungstools und des Laufzeitsystems (Diagnosetool). Für das Wissenserhebungstool müssen die Datenbankschnittstellen definiert und implementiert werden. Für das Laufzeitsystem müssen die Schnittstellen zum Diagnosetester wegen der veränderten Implementierung und die Bedienungsoberfläche mit neuem Layout umgesetzt werden. Schulungskonzepte und Unterlagen für die Modellerzeugung müssen erstellt werden. Die Erstellung der Modellbibliothek ist notwendig und teilweise für Diesel-, Benzin- und Lichtsystem vorhanden.

Die Bewertung des Diagnoseansatzes aus Markt- beziehungsweise Kundensicht wird anhand verschiedener Kriterien durchgeführt und ausgewertet:

- **Effizienz:** Die Reduzierung der Prüfdauer und damit der Prüfkosten beim Fehlerfinden wurde anhand der Feldtestergebnisse nachgewiesen.
- **Diagnosequalität:** Im Vergleich zu heutigen Diagnosetools ist die Verbesserung der Eindeutigkeit der Diagnoseergebnisse deutlich zu erkennen. Die Vollständigkeit bezüglich der Abdeckung aller Zweige des Fehlersuchbaums durch ein Modell und die Robustheit bezüglich ungenauer und fehlender Informationen ist durch Feldtests bewiesen.
- **Einstiegsmöglichkeit:** Die Diagnoseinformationen in Form von Kundenbeanstandungen, Fehlercodes, Sensorwerten und Messungen können unabhängig voneinander aber auch kombiniert als Einstieg in der Diagnose genutzt werden. Diese Möglichkeit ist für das Werkstattpersonal entscheidend, da es nicht selbst gewichten und entscheiden muss, welche Informationen als Einstieg genommen werden sollen.
- **Auswertung der Diagnoseinformationen:** Die gleichzeitige Auswertung mehrerer Eingaben (Beanstandungen, Fehlercodes, Sensorwerte, Messwerte) ist hilfreich im Diagnosevorgang. Zum einen wird das Überdeckungsproblem zwischen verschiedenen Beanstandungen und zwischen Beanstandungen und Fehlercodes gelöst, zum anderen können Arbeitsschritte wie zum Beispiel Auslesen der Sensorwerte, Ergebnisvergleich zwischen Ist- und Sollwert bei der Messung und Durchführung von Stellgliedtests automatisiert werden.
- **Bedienbarkeit:** Neben der benutzerangepassten Bedienung (Spezialisten-/Anfängermodus) existiert die Möglichkeit, Mess-/Testvorschläge zu überspringen. Damit ist die Individualisierung des Nutzerprofils möglich. Das schnelle Auffinden gewünschter Informationen (Einbaulage, Funktionsbeschreibung, etc.) ist durch den Komponenten- u. funktionsorientierten Modellierungsansatz deutlich verbessert worden. Die Darstellung der Wirkkette ist eine leistungsstarke Erklärungskomponente, die nicht nur für Anfänger, sondern auch für Spezialisten brauchbar ist.
- **Lernkurve des Kunden:** Durch die Visualisierung der Wirkkette baut der Kunde System- und Funktionsverständnis stetig auf. Damit steigt die Akzeptanz für das neue Diagnosetool. Mit dem entwickelten Diagnosetool kann der Benutzer vom Anfänger zum Spezialisten durch Selbstlernen ausgebildet werden.
- **Zukunftssicherheit:** Mit der automatischen Durchführung von vorgeschlagenen Messungen und Stellgliedtests in optimaler Reihenfolge ist die Verbindung von geführter Fehlersuche und Steuergerätediagnose sichergestellt worden. Die Individualisierung der Diagnose durch Berücksichtigung regionaler Unterschiede bei der Ausfallwahrscheinlichkeit,

Werkstattinfrastruktur in Form von vorhandenem Test- und Prüfequipment sowie Reparaturhistorie einzelner Fahrzeuge sichert die Zukunft des entwickelten Ansatzes. Bei vorhandener Falldatenbank ist durch den hybriden Ansatz die Integration der fallbasierten Diagnose möglich.

Durch Evaluation und Auswertung wurde in dieser Arbeit gezeigt, wie eine Diagnoselösung für freie Werkstätten mittels des modellbasierten Ansatzes im Verbund mit heuristischem Diagnosewissen umgesetzt und als Erfolg versprechend betrachtet werden kann. Damit ist der Weg zur Produktumsetzung und dem praktischen Einsatz in freien Werkstätten sichergestellt. Nichtsdestotrotz sind noch viele Hindernisse, nicht nur theoretischer, sondern auch wirtschaftlicher Natur, zu überwinden. Nachfolgend werden weiterführende Ansätze, die die erarbeiteten Ergebnisse in eine neue Offboard-Diagnose-Generation einbringen, erläutert.

## 8.2 Ausblick

Der vorgestellte Ansatz hat sich bereits in der Praxis bewährt. Trotzdem ergeben sich zahlreiche Perspektiven, die Arbeit weiterzuführen und zu verbessern. In diesem Abschnitt werden einige Ansätze zur Erweiterung der Wissenserhebung vorgestellt, die teilweise schon prototypisch umgesetzt wurden. Abschließend wird das Vorhaben, die Lösung auf weitere Domänen zu transferieren, kurz erläutert.

### 8.2.1 Erweiterung des Diagnosewissens

Da die Modellierung in Verbindung mit heuristischem Diagnosewissen unter Berücksichtigung des Erstellungsaufwandes nur einen Teil der Realität in einem Diagnosemodell abbildet, kann das Diagnosetool nicht immer eine exakte Diagnose für die vorhandenen Symptome liefern. Eine Fehlerabdeckung mit der Größenordnung zwischen dreiundachzig und einundneunzig Prozent ist mit vertretbarem Aufwand unter Verwendung des entwickelten Ansatzes möglich. Darüber hinaus ist der Aufwand zu hoch und wirtschaftlich nicht sinnvoll. Nachfolgend wird die Überlegung zur Erweiterung des Diagnosewissens mit fallbasiertem Wissen aus der Feedback-Funktion des Diagnosetools vorgestellt und die Erweiterung durch Anbindung weiterer Modellarten und die Abstraktion des quantitativen Modells kurz erläutert. Dabei soll die Erweiterung zum einen die Diagnosegüte erhöhen und zum anderen die Wissenserhebungskosten nicht erheblich steigern.

#### Erweiterung des Diagnosewissens mit Diagnosefallsammlungen

Bei der Wissenserhebung kann das Diagnosemodell durch Informationen aus relevanten Fallsammlungen angereichert werden. Diese Überlegung ist nicht neu und wurde mehrfach in der Literatur diskutiert [83], [114], [150], [88]. In dieser Arbeit existieren hierfür zwei Möglichkeiten:

- Der Fehlerfall ist durch das Diagnosemodell abgedeckt: Wegen der vorhandenen Abdeckung muss das Diagnosewissen im Modell nicht erweitert werden. Abhängig von der Auftrittswahrscheinlichkeit des Fehlers kann gegebenenfalls die Häufigkeit, die zur Priorisierung in dem Diagnosevorgang verwendet wird, geändert werden.

- Der Fehlerfall ist durch das Diagnosemodell nicht abgedeckt: Die Gründe für die nicht vorhandene Abdeckung sind vielseitig und erfordern größeren Analyseaufwand. Nachfolgend werden einige Möglichkeiten aufgezeigt und der Umgang mit diesen Informationen erläutert:
  - Das vorhandene Symptom (Kundenbeanstandung, Fehlercode, etc.) ist nicht in dem Diagnosemodell enthalten. Wenn dieser Fall auftritt, muss je nach relevantem Grad das Symptom und dessen physikalische Bedeutung in das Diagnosemodell aufgenommen oder als separater Fall bei bestimmten Fahrzeugtypen gespeichert werden.
  - Die Grenze der Sollwertbereiche, die durch systemeigene Sensoren beziehungsweise Messungen ermittelt wurden, stimmen nicht. In diesem Fall muss die Diagnosewissensbasis geprüft und geändert werden. Beachtet werden muss, dass die Änderungen möglicherweise nicht für alle Fahrzeuge dieses Typs gelten.
  - Das Komponentenmodell beinhaltet nicht den vorhandenen Komponentenfehler beziehungsweise das Modell kann das Fehlverhalten nicht erklären. Dieser unbekannt beziehungsweise nicht abgedeckte Fehler muss einen Bewertungsprozess durchlaufen, in dem eine Abwägung zwischen Aufwand und Nutzen erfolgt. Wenn der Aufwand nicht überwiegt, wird das Modell um den Fehler erweitert. Wenn der Nutzen wegen der Auftrittswahrscheinlichkeit des Fehlers gering ist, kann der Fall gespeichert und mit dem Diagnosemodell verlinkt werden.

Die Fehlerlokalisierung im Diagnosevorgang kann wie folgt realisiert werden:

- Zuerst wird die fallbasierte Schlussfolgerung angewandt. Wenn keine bekannten Fälle vorliegen, wird die modellbasierte Schlussfolgerung angewandt.
- Die Modellanalyse und deren Schlussfolgerung werden herangezogen, wenn die vorhandenen Symptome im Modell abgebildet sind.

Die Diagnoseabläufe sind mit der Entscheidung, dass Fallsammlungen zuerst überprüft werden anstatt modellbasiert zu analysieren, nicht festgelegt. Eine Umpositionierung der Schlussfolgerung von Diagnosestrategien ist möglich. Das Werkstattpersonal hat die freie Entscheidung, welche Strategie es zuerst anwenden möchte.

### **Generierung der Verhaltenstabelle mittels Zustandsdiagramm**

Eine Möglichkeit zur Erweiterung des Diagnosewissens im Modell ist die Anbindung verschiedener Modellarten. Dadurch werden zum einen unterschiedliche Betrachtungsweisen der Modellierer zugelassen und zum anderen wird die Möglichkeit geboten, unterschiedliche Aspekte der Realität besser abzubilden, miteinander zu kombinieren und das Modellierungswissen so zusammenzuführen. In dieser Arbeit wird zuerst das Verhalten einer Komponente beziehungsweise eines Systems durch qualitative Relationen beziehungsweise Gleichungen beschrieben und durch Verhaltenstabellen dargestellt. Zusätzlich kann das Verhalten im Modellierungstool mittels Zustandsdiagrammen beschrieben werden. Die daraus generierten Verhaltenstabellen und die aus Relationsgleichungen generierten Tabellen werden durch Abgleichalgorithmen zusammengefügt und für die Modellanalyse zur Verfügung gestellt.

In Rahmen der Arbeit wurden nur kleine Modelle mit Zustandsdiagrammen erzeugt und mit der entwickelten Modellprüfung getestet (Abbildung 8.1). Als Ergebnis kann festgehalten werden, dass sich die beiden Verhaltensmodelle ineinander fügen lassen und der Modellierungsaufwand wie erwartet nicht doppelt so hoch ist wie bei einer verwendeten Modellierungsart. Trotzdem ist der Modellierungsaufwand deutlich höher im Verhältnis zu einem geringfügigen Mehrwert bei der Diagnose, weil sich die Verhalten aus den unterschiedlichen Modellierungsarten sehr stark überdecken. Der große Vorteil liegt nur darin, dass der Modellierer nicht an eine bestimmte Modellierungsart gebunden ist.

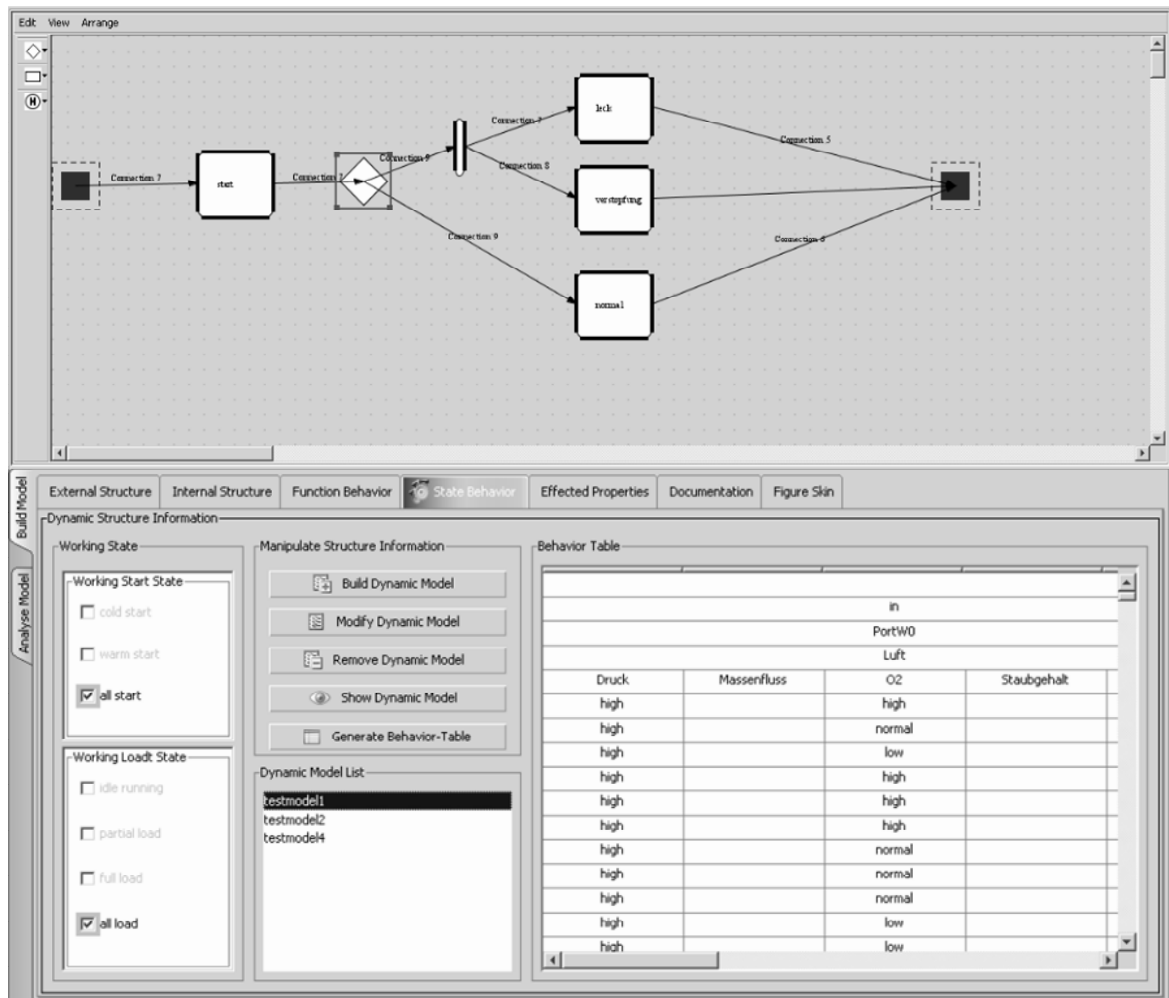


Abbildung 8.1: Prototypische Umsetzung der Generierungsverhaltenstabelle mittels Zustandsdiagramm

### Generierung der Verhaltenstabelle mittels Simulink-Modell

Eine weitere Überlegung ist die Generierung des qualitativen Modells aus einem quantitativen Modell. Aufgrund der statischen Systembetrachtung in der Werkstatt diagnose und der qualitativen Modellbildung sollen die Übertragungsfunktionen aus existierenden analytischen Simulationsmodellen linearisiert werden. Des Weiteren wurden in dieser Arbeit nur drei Betriebspunkte (Leerlauf, Teillast, Vollast) mit zwei Startvorgängen (Warm- und Kaltstart) betrachtet. Daher müssen die Übertragungsfunktionen der Simulationsmodelle nur an diesen

Betriebspunkten berücksichtigt werden. Der Vorteil dieser Vorgehensweise ist die Wiederverwendung des Modellwissens, das zuvor von den Entwicklungsabteilungen beziehungsweise Simulationstool-Herstellern bereitgestellt wurde.

In der Diplomarbeit [85] wurde die Methode zur Abstraktion des Komponentenverhaltens aus Simulink-Modellen untersucht. Zur Evaluierung des entwickelten Verfahrens wurden die stationären Komponentenverhalten der Komponenten des ALWR1.0-Systems aus entsprechenden analytischen Modellen abstrahiert und in DMB-System importiert (Abbildung 8.2). Zur Verifizierung wurden die generierten Komponentenmodelle mittels des entwickelten Modelltestermoduls geprüft. Als Ergebnis kann festgehalten werden, dass die Überführung des quantitativen Modells ins qualitative Modell möglich ist. Der Aufwand bei der Überführung des Normalenverhaltens ist wie erwartet gering. Dagegen beinhalten die Simulationsmodelle in den meisten Fällen keine Fehlverhalten beziehungsweise es wurden kaum Fehlervariablen in der Entwicklung durch die Modellbildung berücksichtigt. Demzufolge ist die Nutzung der generierten Modelle in der Diagnose eingeschränkt.

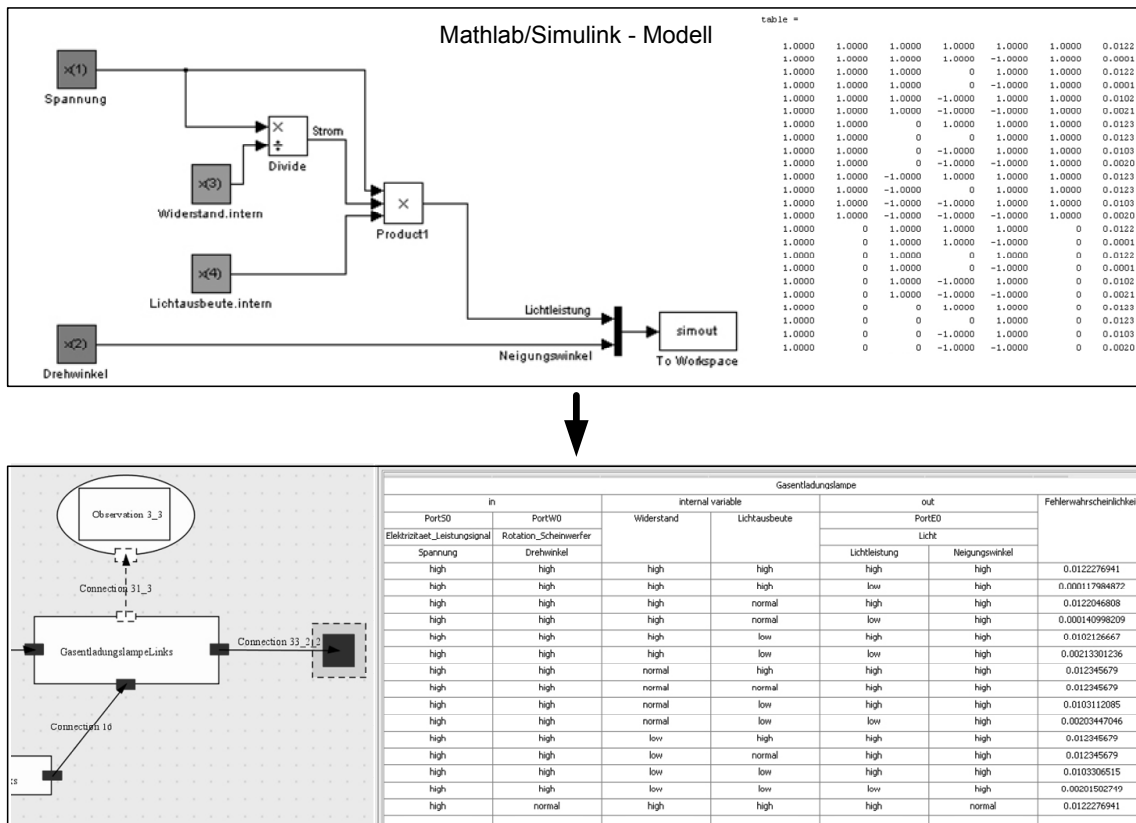


Abbildung 8.2: Prototypische Umsetzung der Generierungstabelle mittels Simulink-Modell

### 8.2.2 Weiterführende Projekte

Nicht nur im Automotivbereich sondern auch im Thermotechnikbereich wird das Gesamtsystem stetig komplexer (Abbildung 8.3). Damit nimmt die Problematik der hohen Anzahl an i.O.-Ausbauten zu (30% i.O.-Ausbauten in Thermotechnikbereich). Bei dem i.O.-Ausbau

werden eine oder mehrere nicht fehlerhafte beziehungsweise In-Ordnung-Komponenten in einem Diagnose- und Reparaturvorgang durch neue Komponenten ersetzt. Die Ursache dieser Problematik gilt für beide Bereiche, zum Beispiel:

- Durch die Weiterentwicklung der Heizungssysteme mit dem Ziel zur Steigerung der Energieeffizienz beziehungsweise Senkung der Energiekosten steigt die Systemkomplexität exponential an,
- Wegen der Kombinationsmöglichkeit der Komponenten beim Aufbau eines Systems existiert eine große Variantenvielfalt.

Neben der Ähnlichkeit der beiden Domänen existieren einige Unterschiede, die bei der Diagnose berücksichtigt werden müssen, zum Beispiel:

- Heizungssysteme reagieren langsamer auf Anregungen als mechatronische Systeme im Fahrzeug, demzufolge ist die Berücksichtigung der Prüfzeit im Diagnosevorgang entscheidend,
- Aufgrund der manuellen Montage eines Heizungssystems können Strukturfehler auftreten. Ein Beispiel für einen Strukturfehler ist das Vertauschen der Ausgangsschnittstellen des Außen- und Innentempersensors beim Einbau.

Wegen der Gemeinsamkeiten aber auch der Unterschiede ist eines der weiterführenden Projekte der Transfer des entwickelten Diagnoseansatzes vom Automotive- zum Thermotechnikbereich.

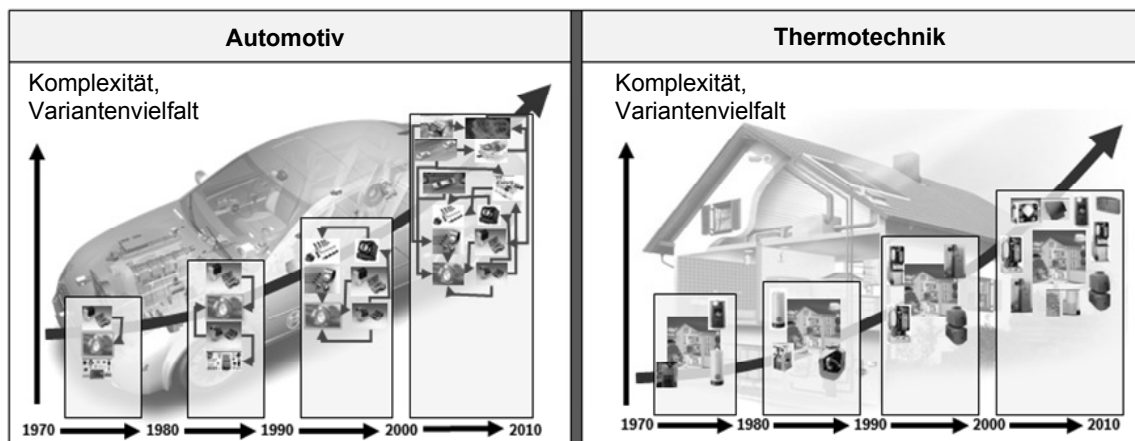


Abbildung 8.3: Parallele Entwicklung der Systemkomplexität im Automotive- und Thermotechnikbereich

Der bestehende Onboard-Diagnoseansatz im Kraftfahrzeugbereich basiert auf der Auswertung von Signalverläufen. Damit ist der Anwendungsbereich nur für Komponenten und nicht für Subsysteme beziehungsweise Systeme geeignet. Eines der Probleme bei der Onboard-Diagnose ist die Auswirkung eines Fehlers in einem System in Bezug zu Symptomen in anderen Systemen. Aktuell wurde kein Lösungsansatz zur systemübergreifenden Diagnose im Steuergerät implementiert, obwohl die Verkopplungen der Systeme in der Onboard-Diagnose berücksichtigt werden müssen. Ein weiterführendes Projekt ist die Bereitstellung von Methoden zur

Lokalisierung fehlerhafter Systeme. Der Lösungsansatz hierzu ist die Zusammenführung des entwickelten Diagnoseansatzes und der vorhandenen Onboard-Diagnose-Funktionen (Abbildung 8.4). Das Verhalten der Systeme und Subsysteme wird durch qualitative relationale Übergangsfunktionen abgebildet, somit können die Lokalisierung fehlerhafter Systeme und Subsysteme sowie die Koordinierung der OBD-Funktionen für die entwickelten Diagnosealgorithmen angewendet werden.

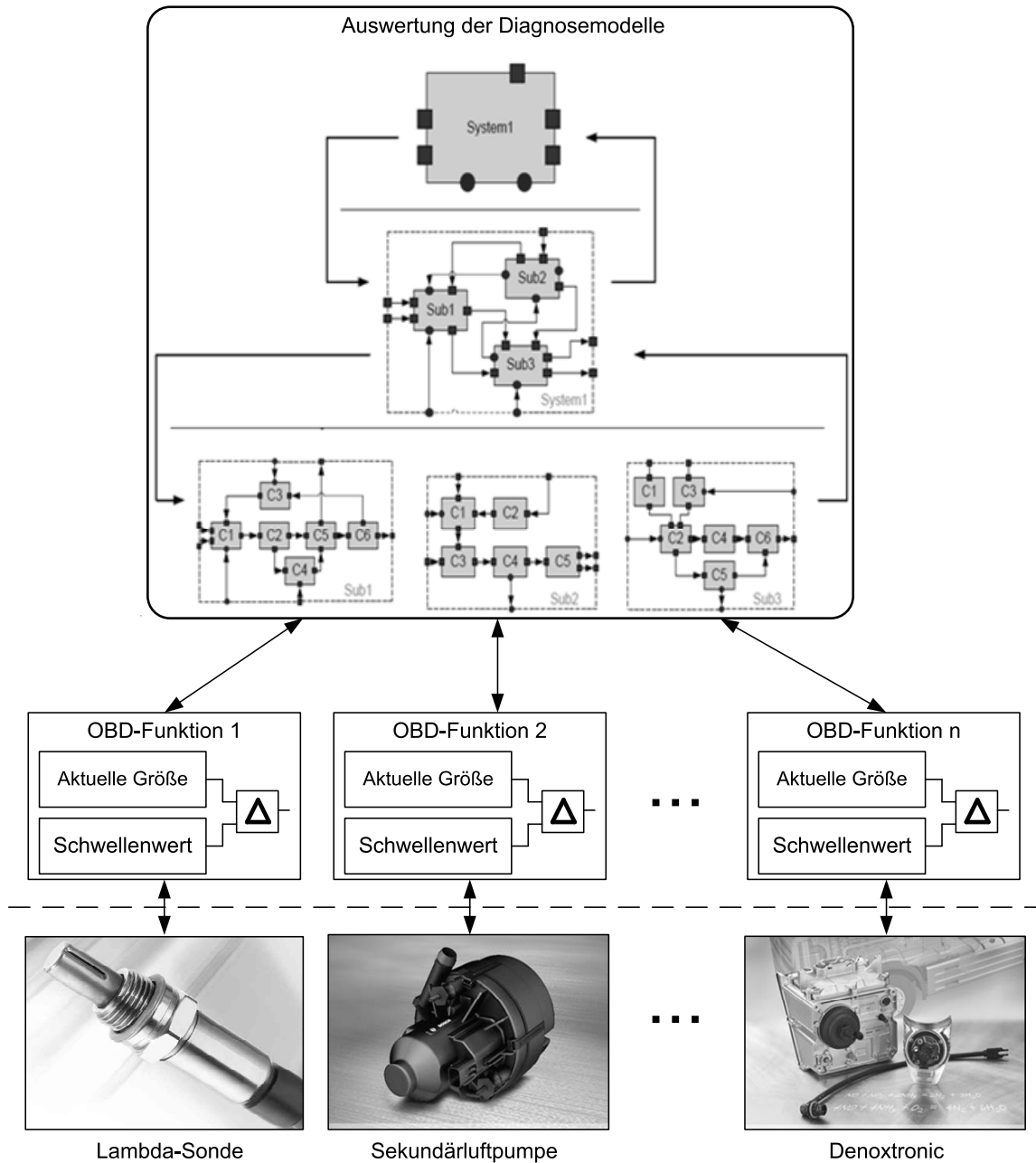


Abbildung 8.4: Einsatzmöglichkeit des entwickelten Ansatzes bei der Onboard-Diagnose

Die Vorteile liegen darin, dass durch eine qualitative Modellierung des Systems und Subsystems der Aufwand der Modellierung relativ gering ist und dennoch eine Diagnoseentscheidung, in welchem System das Fehlverhalten liegen könnte, getroffen werden kann. Bei der Lokalisierung und Bestätigung der fehlerhaften Komponente werden die OBD-Funktionen herangezogen. Die Herausforderung bei diesem Projekt ist die Koordinierung der durchzuführenden

OBD-Funktionen. Die Einschaltbedingungen der OBD-Funktionen können sehr komplex sein und damit kann die OBD-Funktion oft nur in einem bestimmten Betriebspunkt durchgeführt werden. Zum Beispiel können viele OBD-Funktionen des Luftsystems nur im Leerlauf durchgeführt werden. Die Zeit, in welcher sich das System in diesem Betriebspunkt befindet, wird durch ein Start-Stop-System deutlich verkürzt. Daher steht neben der Optimierung des Diagnosealgorithmus bezüglich der Anforderungen der Onboard-Funktion die Priorisierung der durchzuführenden Tests in Form von OBD-Funktionen im Vordergrund.

Zur Lösung von Wissenserhebungsaufgaben benötigen die Mitarbeiter des Service- und Entwicklungsbereichs leistungsfähige und einfach zu bedienende Werkzeuge, die ihre Arbeit unterstützen und erleichtern. Analog brauchen die Mitarbeiter in den Werkstätten ebenso wie die neue Steuergerätgeneration einen leistungsfähigen und effizienten Diagnosealgorithmus bei der Durchführung eines Diagnosevorgangs. Die vorgeschlagenen Lösungsansätze und deren Umsetzung haben in der Evaluierung gezeigt, dass diese Lösung den Anforderungen in diesen beiden Bereichen gerecht wird. Bei der Umsetzung und Weiterentwicklung der Lösungsansätze in der Praxis existieren weitere interessante Herausforderungen.



# Literaturverzeichnis

- [1] AGRAWAL, A.; VIZHANYO, A., KALMAR, Z., SHI, F., NARAYANAN, A., KARSAL, G.: *Reusable Idioms and Patterns in Graph Transformation Languages. Proc. 2nd International Workshop on Graph Based Tools*. Rome , 2004.
- [2] ABI-ANTOUN, M.; ALDRICH, J.; NAHAS, N.; SCHMERL, B.; GARLAN, D.: *Differencing and Merging of Architectural Views*. Institute for Software Research International School of Computer Science Carnegie Mellon University, 2005.
- [3] APPROXIMA ADVANCED TECHNOLOGIES GMBH: *WDS - Das wissensbasierte Diagnosesystem*. Bad Homburg, 2005.
- [4] ALTHOFF, K.-D.; WESS, S.; BARTSCH-SPÖRL, B.; JANETZKO, D.; MAURER, F.; VOSS, A.: *Fallbasiertes Schließen in Expertensystemen: Welche Rolle spielen Fälle für wissensbasierte Systeme?*. KI, Baden-Baden: FBO Verlag, 1992.
- [5] ALTHOFF, K.-D.: *Fallbasierte Systeme und Anwendungen*, Vorlesungsskriptum, WS2010/11, Intelligente Informationssysteme (IIS), 2010.
- [6] ALTHOFF, K.-D.: *Wissensbasierte Systeme*, Vorlesungsskriptum, WS2010/11, Intelligente Informationssysteme (IIS), 2010.
- [7] ANSEL, J.; ENGEL, P.; DANG DUC, N.: *Erstellungszeiten der Fehlersuchanleitungen in Kraftfahrzeug* in: interne Bosch-Berichte. 2007.
- [8] ALANEN, M.; PORRES, I. .: *Difference and Union of Models*. TUCS Turku Centre for Computer Science, 2003.
- [9] BERARD, B.;BIDOIT, M.; FINKEL, A.; LAROUSSINIE, F.; PETIT, A.; PETRUCCI, L.; SCHNOEBELEN,P.: *Systems and Software Verification: Model-Checking Techniques and Tools*. Springer (1 edition), 2001.
- [10] BÄKER, B.: *3. Tagung: Diagnose in mechatronischen Fahrzeugsystemen*. TU Dresden, Institut für Automobiltechnik Dresden - IAD, 2008.
- [11] BIEDERMANN, C.: *Der Vertrieb neuer Automobile in Deutschland: Vor dem Hintergrund der GVO 1400/02 und der aktuellen Marktentwicklung*. Auflage 1, Europäischer Hochschulverlag 2005.
- [12] BAIER, CH.; KATOEN J.-P.: *Principles of Model Checking*. The MIT Press,2008
- [13] BOERGER, E.; STAERK, R.: *Abstract State Machines. A method for High-Level System Design and Analysis*. Springer, 2003.
- [14] BECK, K.: *Test Driven Development by Example*. Addison-Wesley Longman (Amsterdam), 2002

- [15] BORUTZKY, W.: *Bond Graphs - A methodology for modeling multi-disiplinary systems*. bei society for modeling and simulation international, Erlangen 2004.
- [16] BROJ, M.; JONSSON, B.; KATOEN, J.-P.; LEUCKER, M.: *Model-Based Testing of Reactive Systems: Advanced Lectures (Lecture Notes in Computer Science / Programming and Software Engineering)*. Springer (1 edition), 2005.
- [17] BERGMANN, R.; ALTHOFF, K.-D.; BREEN, S.: *Developing Industrial Case-Based Reasoning Applications: The INRECA Methodology (Lecture Notes in Computer Science / Lecture Notes in Artificial Intelligence)*, Springer Berlin Heidelberg, Auflage: 2nd ed., 2003.
- [18] BARDOHL, R.; EHRIG, H.; DE LARA, J.; TAENTZER, G.: *Integrating Meta Modelling Aspects with Graph Transformation for Efficient Visual Language Definition and Model Manipulation*. In M. Wermelinger and T. Margaria, editors, *Proceedings of FASE 2004*. Springer, 2004.
- [19] BORZSONYI, S.; KOSSMANN, D.; STOCKER, K. *The skyline operator*. in: proceedings of the 17th International Conference on Data Engineering, Heidelberg, Germany, April 2001.
- [20] BEGER, T.: *Modellbasierte Diagnose verteilter Systeme auf Basis ihres Kommunikationsverhaltens*. Dissertation, Shaker-Verlag 1999.
- [21] BALKE, W.-T.; GUENTZER, U.; XIN ZHENG, J.: *Approaching the Efficient Frontier: Cooperative Database Retrieval Using High-Dimensional Skylines*. DASFAA, 2005.
- [22] CUNIS R.: *INDIA -Intelligente Diagnose in der industriellen Anwendung*. Shaker Verlag, Aachen, Germany, 2001.
- [23] DIN ISO 13373-1: *Zustandsüberwachung und -diagnostik von Maschinen (Schwingungs-Zustandsüberwachung- Teil 1: Allgemeine Anleitungen)*. DIN Deutsches Institut für Normung e.V., Berlin, 2002.
- [24] DIN ISO 17359: *Zustandsüberwachung und -diagnostik von Maschinen (Allgemeine Anleitungen)*. DIN Deutsches Institut für Normung e.V., Berlin, 2003.
- [25] DIN ISO 13373-2: *Zustandsüberwachung und -diagnostik von Maschinen (Schwingungs-Zustandsueberwachung - Teil 2: Verarbeitung, Analyse und Darstellung von Schwingungsmesswerten)*. DIN Deutsches Institut für Normung e.V., Berlin, 2006
- [26] DIN ISO17359: *Zustandsüberwachung und -diagnostik von Maschinen*. DIN Deutsches Institut für Normung e.V., Berlin, 2007.
- [27] DE KLEER, J.: *Local Methods for Localizing Faults in Electronic Circuits, M.I.T. AI Memo 394*. Cambridge, MA. 1976.
- [28] DE KLEER, J., BROWN, J.S.: *A qualitative physic based on confluences, in: Artificial Intelligence*.1984.
- [29] DE KLEER, J.: *An Assumptionbased TMS, in: Artificial Intelligence*.1986.

- [30] DE KLEER, J., WILLIAMS, B.C.: *Diagnosing Multiple Faults*. in: Artificial Intelligence. 1987.
- [31] DE KLEER, J., WILLIAMS, B.C.: *Diagnosis with Behavioral Modes*, in: *Proc. 11th Intl. Joint Conf. on Artificial Intelligence IJCAI-89*. Detroit, 1989.
- [32] DE KLEER, J., KURIEN, J.: *Fundamentals of Modelbased Diagnosis*. in: Artificial Intelligence. 2003.
- [33] DE LARA, J.; VANGHELUWE, H.: *A Tool for Multi-Formalism Modelling and Meta-Modelling. LNCS 2306*. Springer, 2002.
- [34] DANG DUC, N.; BAUER, N.; ENGEL, P.: *Optimale Testreihenfolge für Werkstattdiagnose*. in: interne Bosch-Berichte CR/AEM. 2005.
- [35] DANG DUC, N.; BAUER, N.; ENGEL, P.: *Allgemeine Test-Reparatur-Strategie*. in: interne Bosch-Berichte CR/AEM. 2005.
- [36] DANG DUC, N.; BAUER, N.; ENGEL, P.: *Sensoreneinsatz in der Werkstattdiagnose*. in: interne Bosch-Berichte CR/AEM. 2005.
- [37] DANG DUC, N.; ENGEL, P. ; DE BOER, G.: *Prozessschritte bei der Wissenserhebung* in: interne Bosch-Berichte CR/AEM. 2006.
- [38] DANG DUC, N.; DE BOER, G.; ENGEL, P. : *Modellbildungskriterien* in: interne Bosch-Berichte CR/AEM-AEH. 2007.
- [39] DANG DUC, N.; ENGEL, P.; WEIDEMANN, D. : *Modellbildungskriterien* in: interne Bosch-Berichte CR/AEM-AEH. 2007.
- [40] DANG DUC, N.; ENGEL, P.; WEIDEMANN, D. : *Aufwand für die Erstellung und die Pflege von Fehlersuchanleitungen und Modelle* in: interne Bosch-Berichte CR/AEM-AEH. 2007.
- [41] DANG DUC, N.; KAERCHER, M. : *Evaluierungsszenario des 2. Feldtests* in: interne Bosch-Berichte CR/AEH. 2008.
- [42] DANG DUC, N.; OPFERMANN, A.; NITSCHKE, R.: *Auswertung und Bewertung des DRT-Systems beim zweiten Feldtest* in: interne Bosch-Berichte CR/AEH. 2009.
- [43] DANG DUC, N.; OPFERMANN, A.; NITSCHKE, R.: *Vorstellung der Ergebnisse in zweitem Feldtest* in: interne Bosch-Berichte CR/AEH. 2009.
- [44] DICKINSON, P.J.; BUNKE, H.; DADEJ, A.; KRAETZL, M. : *Matching graphs with unique node labels*. Pattern Analysis and Applications, 2004.
- [45] DELFMANN, P.: *Adaptive Referenzmodellierung. Methodische Konzepte zur Konstruktion und Anwendung wiederverwendungsorientierter Informationsmodelle*. Diss. Univ. Münster, Berlin 2006.
- [46] DAVIS, R.: *Diagnostic Reasoning Based on Structure and Behavior*, in: *Artificial Intelligence 24/1-3*. 1984.

- [47] DAMIC V. UND MONTGOMERY J.: *Mechatronics by bond graphs: an object-oriented approach to modelling and simulation*. Springer-Verlag Berlin Heidelberg New York, 2004.
- [48] EHRIG, H.; ENGELS, G.; KREOWSKI, H.-J.; ROZENBERG, G.: *Handbook of Graph Grammars and Computing by Graph Transformation. Vol 1. Foundations*. World Scientific, 1999.
- [49] EHRIG, H.; EHRIG, K.; DE LARA, J.; TAENTZER, G.; VARRIO, D.; VARRIO-GYAPAY, S.: *Termination criteria for model transformation*. In *Proc. Fundamental Approaches to Software Engineering (FASE), Lecture Notes in Computer Science* . Springer, 2005.
- [50] EDMUND, M.; CLARKE, JR.; GRUMBERG, O.; PELED, D. A.: *Model Checking*. The MIT Press, 2000.
- [51] ENGEL, P.; DANG DUC, N.; DE BOER, G. : *Evaluierungsprogramm des DMB-Systems* . in: interne Bosch-Berichte CR/AEM. 2006.
- [52] ENGEL, P. : *Auswertung des Modellierungsaufwands* in: interne Bosch-Berichte CR/AEM. 2006.
- [53] ENGEL, P.; DANG DUC, N. : *Modellierbarkeit der Fahrzeug- systeme/funktionen* in: interne Bosch-Berichte CR/AEM-AEH. 2007.
- [54] ENGEL, P.; DANG DUC, N.; WEIDEMANN, D. : *Vergleich der Aufwände für die Erstellung und Pflege von Fehlersuchanleitungen mit existierenden Autorensysteme und DMB-System* in: interne Bosch-Berichte CR/AEM-AEH. 2007.
- [55] ENGEL, P.; DANG DUC, N.; WALTER, M.; WAMBERA, T. : *Auswertung der Modellanzahlen* in: interne Bosch-Berichte CR/AEM-AEH. 2007.
- [56] ENGEL, P.; DANG DUC, N.; WALTER, M.; WAMBERA, T.; SCHABEL, M.; HESS, M.: *Auswertung und Bewertung des DRT-Systems beim erten Feldtest* in: interne Bosch-Berichte CR/AEM. 2007.
- [57] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE: *Testing and Test Control Notation (TTCN-3)*. www.etsi.org, 2007.
- [58] FOWLER, M.: *Refactoring, Improving the Design of Existing Code*. Addison-Wesley-Longman, 1999.
- [59] FOWLER, M.; BECK, K.; BRANT, J.; OPDYKE, W.: *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Longman, Amsterdam; Auflage: illustrated edition, 1999.
- [60] GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J.: *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.
- [61] GENESERETH, M.R.: *The Use of Design Descriptions in Automated Diagnosis, in: Artificial Intelligence, 24/1-3*. 1984.
- [62] GODFREY, P. : *Skyline Cardinality for Relational Processing: How many Vectors Are Maximal?*, FoIKS, 2004.

- [63] GRAF, S.; MOUNIER, L.: *Model Checking Software*. Springer (1 edition), 2004.
- [64] HOLZMANN, G. J.: *The Spin Model Checker: Primer and Reference Manual*. Addison-Wesley Longman (Amsterdam), 2003.
- [65] HATTASS, J.: *Europäischer Kraftfahrzeugvertrieb zwischen den Zielen des EGV und der neuen Wettbewerbspolitik der Kommission: Die Kfz-GVO 1400/2002*. 1. Auflage, Tectum-Verlag 2007.
- [66] HEIMING, B.: *Parallele Prozeßdiagnose auf der Grundlage einer qualitativen Systembeschreibung*. VDI Verlag, Düsseldorf, 2000.
- [67] HAMSCHER, W.: *Readings in Model-based Diagnosis*. (Hrsg. Hamscher, W., Console, L., de Kleer, J.) , San Mateo, California, 1992.
- [68] ISO13374-1:2003: *Condition monitoring and diagnostics of machines (Data processing, communication and presentation - Part 1: General guidelines)*. First edition; Geneva Switzerland 2003.
- [69] ISO13379:2003: *Condition monitoring and diagnostics of machines (General guidelines on data interpretation and diagnostics techniques); First edition*. Geneva Switzerland 2003.
- [70] ISO13372:2004: *Condition monitoring and diagnostics of machines; First edition*. Geneva Switzerland 2004.
- [71] KAYSER, A.: *Komponentenorientierte Modellierung elektrischer und mechatronischer Systeme*. Fortschr.-Ber. VDI Reihe 20 Nr. 368, VDI Verlag 2003.
- [72] KUIPERS, B.: *Qualitative Simulation*. Encyclopedia of Physical Science and Technology, Academic Preas 2002.
- [73] KOSSMANN, D.; RAMSAK, F.; ROST S. *Shooting Stars in the Sky: An Online Algorithm for Skyline Queries*. in: proceedings of the 28 VLDB Conference, Hong Kong, China, 2002.
- [74] KUNG, H.-T.; LUCCIO, F.; PREPARATA, F. P. *On Finding the Maxima of a Set of Vectors*. in: Journal of the ACM 22 (1975), Nr. 4
- [75] KREOWSKI, H.-J.; KUSKE., S.: *Graph transformation units with interleaving semantics*. Formal Aspects of Computing, 1999.
- [76] KERIEVSKY, J.: *Refactoring to Patterns*. Addison-Wesley Longman, Amsterdam, 2004.
- [77] KOSKELA. L.: *Test driven: TDD and Acceptance TDD for Java developers*. Manning Publications, 2007.
- [78] KOCKSKÄMPER, S.; NEUMANN, B.; JOSUB, A.; MÜLLER, H.: *Die Anwendung modellbasierten Schließens bei der Diagnose schiffstechnischer Anlagen*. in: Proc. Expertensysteme 93 (Hrsg. F.Puppe, A.Günter), Springer-Verlag 1993.
- [79] KELLER, U.; WEHREN, J.; NIERE, J.: *A Generic Difference Algorithm for UML Models*. In *Software Engineering*. 2005.

- [80] KÜMMEL, W.: *Technische Strömungsmechanik: Theorie und Praxis*. 3. Auflage, Vieweg+Teubner 2007.
- [81] LUNZER, J.: *Künstliche Intelligenz für Ingenieur. Band 2: Technische Anwendungen* Oldenbourg, München 1995.
- [82] LEHMANN E.: *Time Partition Testing: Systematischer Test des kontinuierlichen Verhaltens von eingebetteten Systemen*. Dissertation, TU-Berlin, 2003.
- [83] LUGER F. G.: *Künstliche Intelligenz . Strategien zur Lösung komplexer Probleme*. Taschenbuch, Pearson Studium Verlag, 2001.
- [84] LINK, J.: *Softwaretests mit JUnit-Techniken der testgetriebenen Entwicklung*. Dpunkt Verlag (2. Auflage), 2005.
- [85] LINNARTZ, M.: *Vertikale Durchgängigkeit in der Diagnose durch funktionale Abstraktion am Beispiel eines mechatronischen Systems*. Diplomarbeit, Hochschule Bonn-Rhein-Sieg in Zusammenarbeit mit der Abteilung CR/AEH der Robert Bosch GmbH, 2009.
- [86] MALIK, A.: *Autorensystem für Anleitung zur Fehlersuche und Reparatur im Kraftfahrzeug: Anforderungsanalyse und Spezifikation, in: Intelligente Diagnose in der industriellen Anwendung.*, Shaker-Verlag, 2000.
- [87] MAUSS, J.: *Analyse kompositionaler Modelle durch Serien-Parrallel-Stern Aggregation*. DISKI 183, infix-Verlag, 1998.
- [88] MENZEL, G.; ZSCHARNACK, H.; UNGER, A.; BÄKER, B.: *Hybride Architekturen für die Offboarddiagnose - Einfluss der Integrationstiefe auf die Diagnoseentwicklung*. Diagnose in mechatronischen Fahrzeugsystemen IV: Neue Verfahren für Test, Prüfung und Diagnose von E/E-Systemen im Kfz von Bernard Bäker und Andreas Unger. Expert-Verlag, 2011.
- bibitemMB88 MEYER, B.: *Object Oriented Software Construction*. Prentice Hall, 1988.
- [89] MORDECHAI, B.-A.: *Principles of the SPIN Model Checker*. Springer London, 2008.
- [90] ERDMANN, M.: *Vertikale Vereinbarungen im Kfz-Sektor: Die neue GVO 1400/2002.*, 1. Auflage, Kovac-Verlag 2006.
- [91] MILI, H. ; MILI, A. ; YACOUB, S. ; ADDY, E.: *Reuse-Based Software Engineering*. New York, 2002.
- [92] MILDE, H.: *Qualitative Analyse von Störungen in elektrischen Systemen.*, DISKI 271, Akademische Verlagsgesellschaft Aka GmbH, Berlin 2003.
- [93] MOZETIC, I.: *Model-Based Diagnosis: An Overview, in: Advanced Topics in Artificial Intelligence*. (Hrsg. Marik, V., Stepankova, O., Trappl, R.), Berlin, 1992.
- [94] MONIN, J.-F.: *Understanding Formal Methods (FACIT)*. Springer (1st Edition), 2003.
- [95] MEFFERT, K.: *JUnit Profi-Tipps-Software erfolgreich testen*. Entwickler.Press (1. Auflage), 2006
- [96] MARTIN, R.C.: *Designing Object Oriented Applications using UML*. 2d. ed., Prentice Hall, 1999.

- [97] MARTIN, R.C.: *Design Principles and Design Patterns*. www.objectmentor.com, 2000.
- [98] MELNIK, S.; GARCIA-MOLINA, H.; RAHM, E.: *Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching*. In 18th Intl. Conference on Data Engineering (ICDE), 2002.
- [99] MÜLLER, T.: *Automatische erfahrungsbasierte Diagnose aus Felddaten mit neuronalen Netzen*. in 12. Internationaler Kongress Elektronik im Kraftfahrzeug, VDI Verlag 2007.
- [100] NYBERG, M.: *Model Based Fault Diagnosis: Methods, Theory, and Automotive Engine Applications, Dissertation*. Linköping, Schweden, 1999.
- [101] OESTEREICH, B.: *Objektorientierte Softwareentwicklung - Analyse und Design mit der UML*. Oldenbourg, Auflage: 5, 2001
- [102] OTTER, M.: *Objektorientierte Modellierung Physikalischer Systeme Teil 1-9*. in: at - Automatisierungstechnik, 1999-2000.
- [103] OTTER, M.; ELMQVIST, H. AND MATTSON S.E.: *The New Modelica MultiBody Library*. Proceedings, 3rd International Modelica Conference, Linköping, Schweden, 2003.
- [104] OTTER, M., ELMQUIST, H.: *Advanced Modelica Tutorial: Developing Modelica Libraries*. in Proc. Modelica, Linköping, Schweden, 2003.
- [105] OTTER, M., ELMQUIST, H.: *Modelica: Language, Libraries, Tools and Conferences*. 2005.
- [106] OLBRICH, R. AND BATTENFELD, D.: *Der markt: Variantenvielfalt und Komplexität - kostenorientierte vs. marktorientierte Sicht*. Springer, Volume 44 / 2005.
- [107] ROBERT BOSCH GMBH: *Ottomotor-Management: Systeme und Komponenten* 3. Auflage, Vieweg 2005.
- [108] PELED, D. A.: *Software Reliability Methods (Texts in Computer Science)*. Springer (1 edition), 2001.
- [109] PAPADIAS, D.;TAO, Y.;FU, G.;SEEGER, B.: *An optimal and progressive algorithm for skyline queries*. in: SIGMOD Conference, 2003.
- [110] PAPADIAS, D.; TAO, Y.; FU, G.; SEEGER, B.: *rogressive Skyline Computation in Database Systems*. in: ACM Transactions on Database Systems 30, 2005.
- [111] PUPPE, F: *Problemlösungsmethoden in Expertensystemen. Studienreihe Informatik*. Springer, Berlin/Heidelberg/New York, 1990.
- [112] PUPPE, F: *Wissensbasierte Diagnosesysteme im Service-Support. Konzepte und Erfahrungen*. Springer, Berlin/Heidelberg/New York, 2003.
- [113] PUPPE, F; GAPPA, U.; POECK, K; BAMBERGER, S.: *Wissensbasierte Diagnose- und Informationssysteme*. Springer, Berlin/Heidelberg/New York, 1996.
- [114] PUPPE, F: *Explizites Wissen versus Black Box-Ansätze in der KI*. Künstliche Intelligenz, Band 25, Heft 1, Springer-Verlag, 2011.

- [115] PIETREK, G.; TROMPETER, J.: *Modellgetriebene Softwareentwicklung. MDA und MDSD in der Praxis*. Entwickler-Press, 2007.
- [116] PFALTZ, J.; NAGL, M.; BOEHLEN, B.: *LNCS: Applications of Graph Transformations with Industrial Relevance Second International Workshop*. In Pfaltz et al. Charlottesville, 2003.
- [117] PAWSON, R.; MATTHEWS, R.: *Naked Objects*. John Wiley and Sons, 2002.
- [118] QUINTARELLI, E.: *Model-Checking Based Data Retrieval: An Application to Semistructured and Temporal Data (Lecture Notes in Computer Science)*. Springer (1 edition), 2004.
- [119] RENSINK, A.; SCHMIDT, A.; VARRIO, D.: *Model checking graph transformations: A comparison of two approaches*. In H. Ehrig, G. Engels, F. Parisi-Presicce, and G. Rozenberg, editors, *ICGT* Springer, 2004.
- [120] RENSINK, A.; DISTEFANO, D.: *Abstract graph transformation*. *Electronic Notes in Computer Science*. Springer, 2006.
- [121] ROBERT V. B.: *Testing Object-Oriented Systems: Models, Patterns, and Tools*. Object Technology Series. Addison-Wesley, 1999.
- [122] ROBERT BOSCH GMBH: *Kraftfahrtechnisches Taschenbuch*. 26. Auflage, Vieweg 2007.
- [123] ROZENBERG, G.: *Hrsg. Handbook of Graph Grammars and Computing by Graph Transformation*. Foundations, Jgg. 1. World Scientific, 1997.
- [124] R.O.S.E. INFORMATIK GMBH: *Systematic Computation of Optimized System Architectures, Reliability Analysis and Diagnostics*. 2005.
- [125] RUSSELL, S.; NORVIG, P.: *Künstliche Intelligenz Ein moderner Ansatz* 2. Auflage, München, 2004.
- [126] REITER, R.: *A Theory of Diagnosis from First Principles*, in: *Artificial Intelligence*, 32/1. 1987.
- [127] SACHENBACHER, M.; STRUSS, P. AND CARLÉN, CLAES M.: *A prototype for model-based on-board diagnosis of automotive systems*. *AI Communications*, 2000.
- [128] STEFFEN, B.; LEVI, G.: *Verification, Model Checking, and Abstract Interpretation*. Springer (1 edition), 2004.
- [129] SOKENOU D.: *UML-basierter Klassen- und Integrationstest objektorientierter Programme*. Dissertation, TU-Berlin, 2006.
- [130] STRUSS, P., DRESSLER, O.: *Physical Negation - Integrating Fault Models into the General Diagnostic Engine*, in: *Proc. Intl. Joint Conf. On Artificial Intelligence*. Detroit, 1989.
- [131] STRUSS, P.: *Structuring of models and reasoning about quantities in qualitative physics* Dissertation, Universität Kaiserslautern, Fachbereich Informatik, 1989.
- [132] STRUSS, P.: *Diagnosis as a Process*, in: *Readings in Modelbased Diagnosis*. (Hrsg. Hamacher, W., Console, L., De Kleer, J.), San Francisco, 1992.



- [133] STRUSS, P.: *What's in SD? Towards a Theory of Modeling in Diagnosis*, in: *Readings in Modelbased Diagnosis* (Hrsg. Hamscher, W.; Console, L.; De Kleer, J.), San Francisco, 1992.
- [134] STRUSS, P.: *Modellbasierte Werkzeuge für Diagnose und Fehleranalyse von Fahrzeugs-ubsystemen in: INDIA - Intelligente Diagnose in der Anwendung/Intelligent Diagnosis for Industrial Applications*. (Hrsg. Holtz, I., Struss, P., Guckenbiehl, Th.), Aachen, 2000.
- [135] STRUSS, P.; PRICE, CH.: *Model-Based Systems in the Automotive Industry*, in: *AI Magazine*. 2003.
- [136] STRUSS, P.: *Modellbasierte Systeme und qualitative Modellierung*, in: *Handbuch der Künstlichen Intelligenz* (Hrsg. Görz G., Rollinger C.-R., Schneeberger J.), Oldenbourg-Verlag 2003.
- [137] STRUSS, P.: *Qualitative Modeling for Diagnosis of Machines Transporting Rigid Objects*. in The 19th International Workshop on Principles of Diagnosis, Blue Mountains, Australia 2008.
- [138] SOHNLE, S.; DANG DUC, N.; OPFERMANN, A.: *Modellierungsworkshop: Überprüfung der Machbarkeit der Modellerstellung und Abschätzung des Erstellungsaufwands* in: interne Bosch-Berichte. 2008.
- [139] STAHL, T.; VÖLTER, M.; EFFTINGE, S.; HAASE, A.: *Modellgetriebene Softwareentwicklung: Techniken, Engineering, Management*. Dpunkt Verlag; Auflage: 2., 2007.
- [140] ZENNER, C. : *Durchgängiges Variantenmanagement in der Technischen Produktionsplanung*. Universität des Saarlandes Schriftenreihe Produktionstechnik Herausgeber: H. Bley und C. Weber, Band 37, 2006
- [141] TELTOW, G.: *Modell- und Distributionsstrategien in der Automobilindustrie: Variantenvielfalt und Verfügbarkeit*. Verlag: Diplomica Verlag GmbH, 1999.
- [142] TAENTZER, G: *LNCS: A Graph Transformation Environment for Modeling and Validation of Software, Lecture Notes in Computer Science* . Charlottesville, 2003.
- [143] THOLE, M.: *Qualitative Analyse analoger Schaltungen* Dissertation, Technische Universität Carolo-Wilhelmina Braunschweig, Fakultät für Maschinenbau und Elektronik, 1998.
- [144] TRÖLTZSCH, U.: *Modellbasierte Zustandsdiagnose von Gerätebatterien*. Universität der Bundeswehr München, Fakultät für Elektrotechnik und Informationstechnik, 2005.
- [145] VDI-RICHTLINIEN: *Einsatz wissensbasierter Diagnosemethoden und -systeme in der Instandhaltung VDI-Handbuch Betriebstechnik (VDI 2889)*. Teil 4, Düsseldorf, 1998.
- [146] VDI-RICHTLINIEN: *Zustandsorientierte Instandhaltung (VDI 2888)*. VDI-Handbuch Betriebstechnik, Teil 4, Düsseldorf; 1999.
- [147] WESTPHAL, F.: *Testgetriebene Entwicklung mit JUnit und FIT*. Dpunkt Verlag (1. Auflage), 2005.
- [148] WALESCHKOWSKI, N.: *Wissensmodellierung und Wissenserwerb am Beispiel der Fahrzeugdiagnose*. KI - Zeitschrift Künstliche Intelligenz, 1995.

- [149] WALESCHKOWSKI, N.: *WDS - A Knowledge Based Diagnostic System*. Technical Paper, Oberursel 2004.
- [150] WALESCHKOWSKI, N.: *Lernfähige wissensbasierte Diagnosesysteme*. Diagnose in mechatronischen Fahrzeugsystemen IV: Neue Verfahren für Test, Prüfung und Diagnose von E/E-Systemen im Kfz von Bernard Bäker und Andreas Unger. Expert-Verlag, 2011.
- [151] WAMBERA, T.; ENGEL, P.; DANG DUC, N.: *Unterschiede zwischen existierenden Autorensysteme und DMB-System* in: interne Bosch-Berichte CR/AEM-AEH. 2007.
- [152] WEBER, R.: *Intelligente Diagnose bei mechatronischen Systemen im Kraftfahrzeug, in: Intelligente Diagnose in der industriellen Anwendung*. Shaker 2000.
- [153] ZEPPENFELD, K.; WOLTERS, R. .: *Generative Software-Entwicklung mit der Model Driven Architecture*. Spektrum Akademischer Verlag; Auflage: 1, 2005.

# Abbildungsverzeichnis

1.1	Verfahren und Strategien für neue Prüftechniksysteme . . . . .	2
1.2	Teilmodell des Kraftstoffsystems . . . . .	4
1.3	Grober Aufbau des Diagnosemodells . . . . .	6
1.4	Entwickelte Modellierungsapplikation (DMB-System) . . . . .	16
1.5	Verhaltenseditor von DMB-System . . . . .	18
1.6	Auswahl des Modells mit DRT-System . . . . .	19
1.7	Entwickeltes DRT-System . . . . .	20
1.8	Maske zur Darstellung der Handlungsempfehlung . . . . .	21
2.1	Prozess der Strategie „Hypothesenerstellung und Hypothesenprüfung“ . . . . .	35
2.2	Architektur des Expertensystems aus funktionaler Sicht . . . . .	36
2.3	Wissenserhebung für die Kfz-Werkstattdiagnose . . . . .	37
2.4	Diagnose- und Reparaturprozess in der Kfz-Werkstatt . . . . .	39
3.1	Kennlinien der Elektrokraftstoffpumpe beim Golf III 1.8 (Quelle: BOSCH Group)	44
3.2	Bestandteile des Basis- und des vollständigen Diagnosemodells . . . . .	47
3.3	Klassendiagramm einer systemeigenen Komponente . . . . .	54
3.4	Beispiel einer qualitativen Übergangsfunktion . . . . .	59
3.5	Modelldarstellung einer Kraftstoffleitung . . . . .	60
3.6	Objektdiagramm eines Einspritzventils . . . . .	64
3.7	Beispielhafte Visualisierung eines Systemsensors . . . . .	66
3.8	Beispielhaftes Objektdiagramm eines Drucksensors . . . . .	66
3.9	Beispielhafte Visualisierung einer Messung . . . . .	67
3.10	Beispielhaftes Objektdiagramm einer manuellen Messung . . . . .	68
3.11	Beispielhafte Notation eines Fehlercodes als virtueller Sensor . . . . .	70
3.12	Beispielhafte Notation eines Fehlercodes als Diagnoseregeln . . . . .	71
3.13	Beispielhaftes Objektdiagramm einiger Fehlercodes . . . . .	72
3.14	Beispielnotation einer Kundenbeanstandung im Lichtsystem . . . . .	74
3.15	Kundenbeanstandung als Diagnoseregeln . . . . .	76

3.16	Beispielhaftes Objektdiagramm einer Kundenbeanstandung . . . . .	77
3.17	Beispielnotation eines Stellgliedtests im Modell des Lichtsystems . . . . .	78
3.18	Beispielhaftes Objektdiagramm eines Stellgliedtests im Modell des Lichtsystems	79
3.19	Beispielnotation eines Komponententests im Modell des Kraftstoffsystems . .	80
3.20	Beispielhaftes Objektdiagramm eines Komponententests im Modell des Lichtsystems . . . . .	80
3.21	Beispielhaftes Objektdiagramm der Systemschnittstellen und Verbindungen .	82
3.22	Vollständige Modelle als Instanzen des physikalischen Modells am Beispiel des Luftsystems-ME7.6.x . . . . .	85
3.23	Generalisierung der Varianten am Beispiel der Luftlader . . . . .	86
3.24	Vererbung der Schnittstelle sowie der internen Variable am Beispiel des Kraftstoffrohrs . . . . .	86
3.25	Modelltransformation am Beispiel eines Teils des Luftsystems mit Lader . . .	87
3.26	Modellerweiterung am Beispiel eines Teils des Lichtsystems . . . . .	88
3.27	Verschiedene Testarten des erstellten Diagnosemodells . . . . .	91
4.1	Diagnostik-Algorithmus $D$ . . . . .	98
4.2	Modell des Luftsystems ME7.6.x mit Fehlercode und Kundenbeanstandung für das erste Beispiel . . . . .	115
4.3	Modell des Luftsystems ME7.6.x mit Kundenbeanstandung für das zweite Beispiel	117
4.4	Auslesen der Fehlercodes mittels Steuergerät . . . . .	125
4.5	Ermittlung der verdächtigen Komponenten mittels Kundenbeanstandung . . .	126
4.6	Ergebnisse des Stellgliedtests . . . . .	127
4.7	Auslesen der Sensor-Istwerte . . . . .	128
4.8	Istzustand der Einbaulage der Komponente <i>Gestänge Achssensor hinten</i> . . .	128
4.9	Komponentenbezogenes Diagnosewissen . . . . .	129
5.1	Die sechs Phasen bei der Modellierung eines Systems . . . . .	131
5.2	Erste Phase: Struktur- und Verhaltensmodellierung . . . . .	132
5.3	Detaillierte Aktivitäten zur Erstellung des Struktur- und Verhaltensmodells .	133
5.4	Zweite Phase: Modellierung der Kundenbeanstandungen . . . . .	134
5.5	Dritte Phase: Fehlercodes in das Modell integrieren . . . . .	135
5.6	Vierte Phase: Abbildung der externen Messungen . . . . .	136
5.7	Fünfte Phase: Komponenten- und Stellgliedtests anbinden . . . . .	136
5.8	Rollen der Beteiligten bei der Modellerstellung . . . . .	138
5.9	DMB-System und seine Umgebung . . . . .	139

5.10	Dekomposition von DMB-System in Segmente . . . . .	140
5.11	Ergebnisse der Reduktion des Speicherbedarfs . . . . .	142
5.12	Testplattform und Testrollen . . . . .	143
5.13	Ergebnisse des Komponententests . . . . .	144
5.14	Maske zur Spezifizierung der Modelltestfälle . . . . .	145
5.15	Maske zur Darstellung der Modelltestergebnisse . . . . .	145
5.16	Variantenmanagement-Modul in DMB-System . . . . .	146
5.17	Berechnung der Distanz zwischen Modellen . . . . .	147
6.1	Die drei Hauptschritte im Werkstattdiagnoseprozess . . . . .	148
6.2	Aktivitäten bei der Fahrzeugannahme . . . . .	149
6.3	Aktivitäten bei der Diagnosedurchführung und Reparatur . . . . .	150
6.4	DRT-System und seine Umgebung . . . . .	152
6.5	Dekomposition von DRT-System in Segmente . . . . .	153
6.6	Dekomposition des Segments „Model Explorer“ in Pakete . . . . .	154
6.7	Beispiel: Instruktion zum Auslesen eines Systemsensor-Istwertes . . . . .	157
6.8	Beispiel: Instruktion einer Komponentenprüfung . . . . .	158
6.9	Szenario für den Totalumstieg . . . . .	159
6.10	Szenario für den Parallelbetrieb . . . . .	160
6.11	Integration mit dem Konvertierungsansatz . . . . .	160
6.12	Integration mit dem Plug-in-Ansatz . . . . .	161
6.13	Integration mit der Entwicklung eines neuen Diagnosetools . . . . .	161
7.1	Ergebnisse der Prüfschritte bei der Evaluierung . . . . .	175
7.2	Ergebnisse der Prüfzeit bei der Evaluierung . . . . .	176
7.3	Szenario des zweiten Feldtests . . . . .	178
7.4	Diagnoseablauf bei dem zweiten Feldtest . . . . .	179
7.5	Beispiel eines Versuchprotokolls bei der zweiten Evaluierung . . . . .	180
7.6	Feldtestergebnisse zur Auswertung der Diagnosefähigkeit . . . . .	181
7.7	Feldtestergebnisse zur Überprüfung der vorgeschlagenen Prüfschritte . . . . .	182
7.8	Feldtestergebnisse zur Auswertung der Effektivität des entwickelten Ansatzes . . . . .	183
7.9	Feldtestergebnisse zur Überprüfung des Nutzens und der Akzeptanz des entwickelten Ansatzes . . . . .	183
8.1	Prototypische Umsetzung der Generierungsverhaltenstabelle mittels Zustandsdiagramm . . . . .	189

8.2	Prototypische Umsetzung der Generierungsverhaltenstabelle mittels Simulink-Modell . . . . .	190
8.3	Parallele Entwicklung der Systemkomplexität im Automotive- und Thermotechnikbereich . . . . .	191
8.4	Einsatzmöglichkeit des entwickelten Ansatzes bei der Onboard-Diagnose . . .	192

# Tabellenverzeichnis

1.1	Beispiel für Aufbau einer Verhaltenstabelle . . . . .	7
1.2	Ein Abschnitt der Verhaltenstabelle des Rohres . . . . .	11
1.3	Ein Abschnitt der reduzierten Verhaltenstabelle mit $Leckage \stackrel{qual}{=} N$ des Rohres	12
1.4	Ein Abschnitt der reduzierten Verhaltenstabelle mit $Leckage \stackrel{qual}{=} N$ des Druckregelventils . . . . .	13
1.5	Weitere Reduzierung der Verhaltenstabelle durch Wertebelegung der Ausgangsschnittstelle des Rohres ( $Rohr.S_1.Kraftstoff.\dot{V} \stackrel{qual}{=} L$ ) . . . . .	14
1.6	Weitere Reduzierung der Verhaltenstabelle des Druckregelventils durch Rückwärtspropagierung (Diagnoseparameter $Druckregelventil.S_5.Kraftstoff.\dot{V} \stackrel{qual}{=} L$ ) . . . . .	14
1.7	Relationen des Volumenstroms ( $\dot{V}$ ) in der reduzierten Verhaltenstabelle des Druckregelventils bei Vorwärtspropagierung (Diagnoseparameter $Druckregelventil.S_0.Kraftstoff.\dot{V} \stackrel{qual}{=} N$ ) . . . . .	15
3.1	Verschiedene reale Komponentenarten im Kraftstoffsystem . . . . .	53
3.2	Systemeigene Komponenten und deren Schnittstellen . . . . .	55
3.3	Einige Transportobjekte und einige ihrer Attribute . . . . .	55
3.4	Rechenbeispiel für qualitative Addition und Subtraktion . . . . .	59
3.5	Aufbau einer Verhaltenstabelle . . . . .	61
3.6	Aufbau einer Beeinflussungsgradtabelle . . . . .	63
3.7	Beispiel Inhalts einer Beobachtungstabelle . . . . .	75
3.8	Testurteil für die nicht verdächtige Menge $COM_j^{OK} = COM_j^{OK,soll} \cap COM_j^{OK,ist}$	95
3.9	Testurteil für die verdächtige Menge $COM_j^{-OK} = COM_j^{-OK,soll} \cap COM_j^{-OK,ist}$	95
4.1	Darstellung der Menge $COM^{DTC,-OK}$ in Tabellenform . . . . .	100
4.2	Darstellung der PARAMOBS in Tabellenform . . . . .	101
4.3	Darstellung der $PARAM^{DTC}$ in Tabellenform . . . . .	101
4.4	Darstellung der $PARAM^{SEN}$ in Tabellenform . . . . .	102
4.5	Darstellung der $PARAM^{MEAS}$ in Tabellenform . . . . .	102
4.6	Darstellung der $PARAM^{ComTEST}$ in Tabellenform . . . . .	102
4.7	Darstellung der $PARAM^{ActTEST}$ in Tabellenform . . . . .	103

4.8	Beispiel eines Abschnitts der vollständigen Verhaltenstabelle $tab_x$ einer Komponente $com_x$ . . . . .	104
4.9	Beispiel eines Abschnitts der reduzierten Verhaltenstabelle $tab_x^{param_{i,x}}$ einer Komponente $com_x$ mit dem Diagnoseparameter $param_{i,x}$ . . . . .	105
4.10	Beispiel eines Abschnitts der vollständigen Verhaltenstabelle $tab_x^{pre}$ einer Komponente $com_x^{pre}$ . . . . .	105
4.11	Beispiel eines Abschnitts der reduzierten Verhaltenstabelle mit den Diagnoseparametern $param_{i,x,j}^{pre}$ und $param_{i,x,j+k}^{pre}$ der Komponente $com_x^{pre}$ . . . . .	106
4.12	Beispiel eines Abschnitts der vollständigen Verhaltenstabelle $tab_x^{post}$ einer Komponente $com_x^{post}$ . . . . .	107
4.13	Beispiel eines Abschnitts der reduzierten Verhaltenstabelle mit den Diagnoseparametern $param_{i,x,j}^{post}$ und $param_{i,x,j+k}^{post}$ der Komponente $com_x^{post}$ . . . . .	107
4.14	Darstellung der Menge $COM^{DTC,-OK} = \{Temperatursensor\}$ des Fehlercodes P0112 in Tabellenform . . . . .	114
4.15	Darstellung der Sichtprüfung „Keine Leckage“ in Tabellenform . . . . .	116
4.16	Darstellung der Kundenbeanstandung „Schlechte Gasannahme, Übergangsfehler“ in Tabellenform . . . . .	116
4.17	Beispiel eines Abschnitts der reduzierten Verhaltenstabelle der Komponente $Drallklappe_1$ mit $Drallklappe_1.leckage = N$ . . . . .	118
4.18	Beispiel eines Abschnitts der reduzierten Verhaltenstabelle der Komponente $Drallklappe_1$ mit $Drallklappe_1.leckage = N$ und $Drallklappe_1.S_1^{out}.Luft.m = L$ . . . . .	120
4.19	Beispiel eines Abschnitts der reduzierten Verhaltenstabelle der Komponente $Drosselklappe$ mit $Drosselklappe.klemmt = N$ und $Drosselklappe.schwergaengig = T$ . . . . .	120
4.20	Beispiel eines Abschnitts der reduzierten Verhaltenstabelle der Komponente $Gestänge Einlasskanalabschaltung$ . . . . .	121
5.1	Ergebnisse der Reduktion des Speicherbedarfs . . . . .	143
6.1	Ablauf eines Diagnoseversuchs an einem ME7.6-Benzinsystem mit DRT-System	156
6.2	Ablauf eines Diagnoseversuchs an einem CP3-Dieselsystem mit DRT-System .	156
7.1	Aufwandshebung bei der Erstellung des CP3-Diesel-Systems . . . . .	165
7.2	Aufwandshebung bei der Erstellung des ME7.6-Benzin-Systems . . . . .	165
7.3	Aufwandshebung des CP3-Diesel-Systems bei Leit- und Folgeprojekt . . . . .	166
7.4	Aufwandshebung der Leitprojekte im Vergleich . . . . .	167
7.5	Aufwandshebung des ALWR1.0-Systems bei Leitprojekten . . . . .	168
7.6	Aufwandshebung des ALWR1.0-Systems bei Leitprojekten . . . . .	168
7.7	Protokoll des Versuchs mit DRT-System für den ME7.x-Ottomotor . . . . .	173



7.8	Protokoll des Versuchs mit einem existierenden Diagnosesystem für den ME7.x-Ottomotor . . . . .	174
7.9	Zusammenfassung der ersten Evaluierungsergebnisse . . . . .	174
7.10	Liste der abgedeckten Fahrzeuge . . . . .	178