



□ Fevzi Belli

(E-Mail: bellif@upb.de)
 vertritt das Fachgebiet Angewandte Datentechnik an der Universität Paderborn, Institut für Elektrotechnik und Informationstechnik. Er beschäftigt sich mit Techniken des systematischen Testens, der Zuverlässigkeit und der Fehlertoleranz.



□ Axel Hollmann

(E-Mail: hollmann@adt.upb.de)
 ist Mitarbeiter im Institut für Elektrotechnik und Informationstechnik an der Universität Paderborn. 2007 schloss er sein Studium der Informatik an der Universität Paderborn ab. Seine Forschungsinteressen liegen im Bereich der Softwarequalitätssicherung, Modellbasiertes Testen und Mutationsanalyse.

Ereignisbasiertes Testen eingebetteter Systeme – Vergleich mit Klassifikationsbäumen anhand eines automotiven Beispiels

Mechanische Steuerkomponenten, wie die in modernen Automobilen, werden mehr und mehr durch elektronische verdrängt. Diese sind meistens mit einem Steuergerät gekoppelt. Um die Funktionsfähigkeit dieser Steuergeräte zu gewährleisten, ist ein systematisches Testen notwendig. Die Automatisierung der Tests soll nicht nur die Qualität steigern, sondern auch die Effizienz erhöhen, um die Kosten zu reduzieren. Während die eigentliche Durchführung der Tests mittlerweile einen hohen Automatisierungsgrad erreicht hat, werden die Testfälle oft manuell erstellt. Dieser Beitrag beschreibt den Einsatz von Ereignis-Sequenz-Graphen, die aus einer formalen Beschreibung eines Steuergerätes die Erzeugung von Testfällen ermöglichen. Diese Methode wird mit der Klassifikationsbaummethode anhand einer Fallstudie verglichen.

Einführung und Motivation

Die Komplexität der elektronischen Systeme im Automobil wächst bei gleichzeitiger Verringerung der Entwicklungszeit. Die Integration der Steuergeräte zu einem funktionierenden Gesamtsystem wird immer schwieriger, da die Abhängigkeiten der Geräte voneinander immer höher werden. Dieser Artikel betrachtet die Erfahrungen bei der Anwendung einer graphbasierten Methode (*Ereignis-Sequenz-Graphen*) zur Ermittlung von Testschritten auf ihre Anwendbarkeit im Testprozess für automotiv Anwendungen. Zudem erfolgt ein Vergleich der Leistungsfähigkeit mit der manuellen Klassifikationsbaummethode anhand eines Fallbeispiels eines Fahrerassistenzsystems zur automatischen Abstandsregelung.

In dem Fallbeispiel kommt ein Testprozess, wie in **Tabelle 1** dargestellt, zur Anwendung. Das betrachtete Steuergerät hängt direkt von der Korrektheit der Signale vier weiterer Steuergeräte ab. Etwa 2000 Anforderungen werden hierbei zu 400 Testzielen zusammengefasst, die in 4000 Testschritten umgesetzt werden. Die in der Tabelle angegebenen Aufwände sind

empirisch für das Projekt ermittelt worden. Die Vermutung liegt nahe, dass eine Optimierung der Anzahl der Testschritte im Testentwurf direkt Zeit bei der Umsetzung der folgenden Prozessschritte einspart.

Methoden zum Testentwurf

In der Praxis ist noch immer die *intuitive und erfahrungsbasierte Testfallermittlung* weit verbreitet. Dabei werden vom Testingenieur die Eingabegrößen so variiert, dass die Ausgangsgrößen die zu erfüllenden Kriterien aufweisen. Der Erfolg dieser Vorgehensweise, sowie die Anzahl und

Effizienz der aufgestellten Testschritte und -fälle hängen stark von der Erfahrung des Testingenieurs ab. Sie sollte eher dazu benutzt werden, um die nach systematischen Methoden aufgestellten Testfälle sinnvoll zu ergänzen. Zudem wird das Aufstellen von Negativ-Testfällen meistens vernachlässigt, da die Anforderungen in der Regel nur positiv formuliert sind.

Ebenfalls weit verbreitet ist die *Klassifikationsbaummethode*. Sie ermöglicht über eine Klassifizierung aller Aspekte des Testobjekts die Komplexität zu reduzieren. Die Grundlage für diese Methode bildet die

Prozess	Durchführung	Aufwand
Testplanung	manuell	gering 5%
Testspezifikation	manuell / geringe Toolunterstützung	mittel 20%
Testentwurf	manuell	mittel 20%
Testimplementierung	manuell	hoch 30%
Testdurchführung	automatisch	gering 5%
Testauswertung	manuell / geringe Toolunterstützung	mittel 20%

Tabelle 1: Testprozess und Aufwände

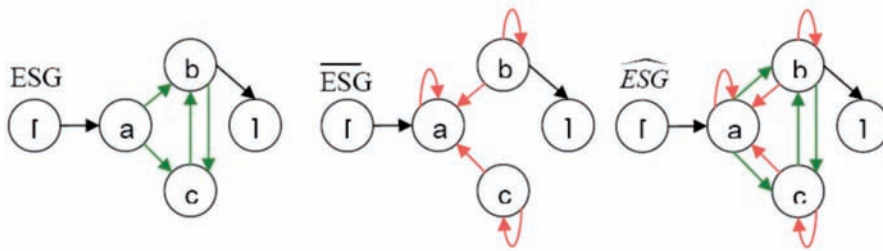


Abbildung 1: Ereignis-Sequenz-Graph, inverser und kompletierter Graph

Systemspezifikation. Bei der Erstellung eines Klassifikationsbaums wird zunächst die Komplexität des in der Spezifikation verankerten Problems zerlegt. Durch die Analyse der Spezifikation werden einzelne für den Test relevante Aspekte (Klassifikationen) der Problemstellung herausgefiltert. Mehrere Klassifikationen können zu einer Komposition zusammengeführt werden. Der Wertebereich einzelner Klassifikationen wird betrachtet und in sinnvolle Klassen zerlegt. Die Klassen werden so gewählt, dass alle Werte aus diesen als äquivalent betrachtet werden können (Äquivalenzklassen). Diese Äquivalenzklassen können nach der Betrachtung zusätzlicher Bedingungen weiter klassifiziert werden. Durch die wiederholte Bildung von Unterklassifikationen wird der Klassifikationsbaum erstellt. Beschreibt ein Klassifikationsbaum ein dynamisches Verhalten eines Systems, so können die Blätter als Ereignisse aufgefasst werden. Diese sind gleichzeitig der Kopf einer Kombinationstabelle, wobei jede Zeile einen Testfall spezifiziert.

Ereignis-Sequenz-Graphen (ESG) werden benutzt, um das Verhalten eines Systems und die Interaktion mit einem Benutzer bzw. der Umgebung zu modellieren. Ereignisse sind Eingaben des Benutzers und der Umgebung oder stellen Antworten des Systems dar. Ein Ereignis-Sequenz-Graph $ESG=(V,E)$ (siehe [1]) besteht aus einer endlichen Menge von *Ereignissen* V und einer endlichen Menge von *Kanten* $E \subseteq V \times V$. Eine Kante $(v,v') \in E$ gibt an, dass das Ereignis v' dann ausführbar ist, wenn das Ereignis v ausgeführt wurde. Die Pseudoknoten [und] markieren zulässige *Start-* bzw. *Endereignisse*. Der Graph beinhaltet somit alle gültigen Ereignissequenzen, die als Testfälle für *Positivtests* eingesetzt werden können: „Funktioniert das System so, wie ich es mir wünsche?“

Der *inverse* Graph $\overline{ESG}=(V, \overline{E})$ ist gegeben durch die Menge der Ereignisse V und die Kantenmenge $\overline{E}=(V \times V) \setminus E$. Er repräsentiert

das ungültige Verhalten des Systems und wird zur systematischen Generierung von *Negativtests* benutzt: „Kann das System so funktionieren, wie es nicht erwünscht bzw. wie es zu vermeiden ist?“. **Abbildung 1** zeigt ein kleines Beispiel mit dem Graphen $ESG=(V,E)$ mit $V=\{a,b,c\}$ und $E=\{(a,b),(a,c),(b,c),(c,b)\}$, den inversen Graphen \overline{ESG} mit $V=\{a,b,c\}$ und $\overline{E}=\{(a,a),(b,a),(b,b),(c,a),(c,c)\}$ und dem *komplettierten* Graphen, der beide Graphen – sowohl den ESG als auch den \overline{ESG} – beinhaltet.

Anforderungen an die Testmethode

Das ESG-Konzept wurde bereits für das Testen von graphischen Oberflächen erfolgreich eingesetzt. Das Testen von Steuergeräten besitzt allerdings andere Rahmenbedingungen. Die Anforderungen sind nur in textueller Form festgeschrieben. Die Steuergeräte sind physikalisch über CAN-Bus, Spannungsversorgung und mechanische Halterung mit dem Fahrzeug verbunden. Die Sensorik und die Ein-

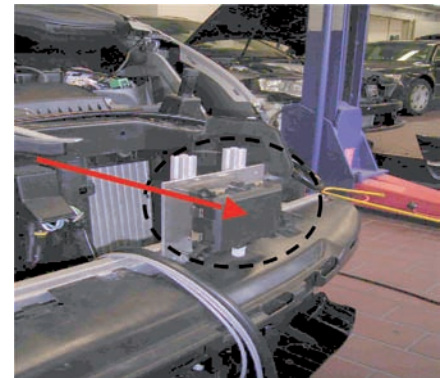


Abbildung 2: Adaptive Cruise Control (ACC)

bindung des Steuergerätes in die Umwelt werden in dem Fallbeispiel nicht betrachtet, sondern nur der Teil, der für die Fahrzeugumgebung relevant ist. Da der Test der Fahrzeugintegration für fast alle Steuergeräte ähnlich ist, eignet sich diese Testmethode für einen Vergleich zur Anwendung von ESGs.

Testobjekt

Das Steuergerät ist ein Radarsensor mit 24GHz Technik (**Abbildung 2**). Es ist über den CAN-Bus mit fünf weiteren Steuergeräten und so indirekt mit mehr als 7 Sensoren gekoppelt und wird über die Standardspannungsversorgung mitversorgt. Vom Gesamttest wurde ein Minimalfall (8 Zustände, abhängig von 4 Eingangsgrößen) und ein Maximalfall (ca. 20 sog. Testziele, abhängig von über 20 Eingangsgrößen, zentraler Zustandsautomat) ausgewählt. **Abbildung 3** zeigt

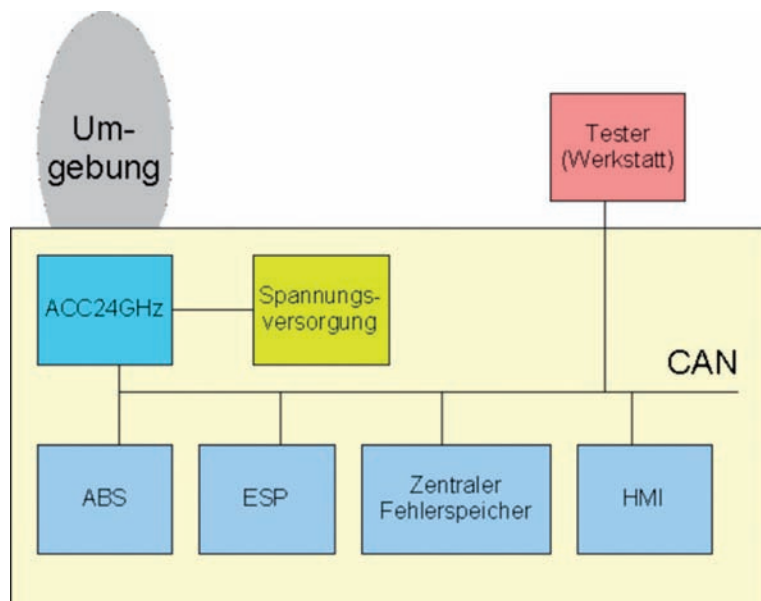


Abbildung 3: Abhängigkeit der Steuergeräte über das CAN-Netzwerk

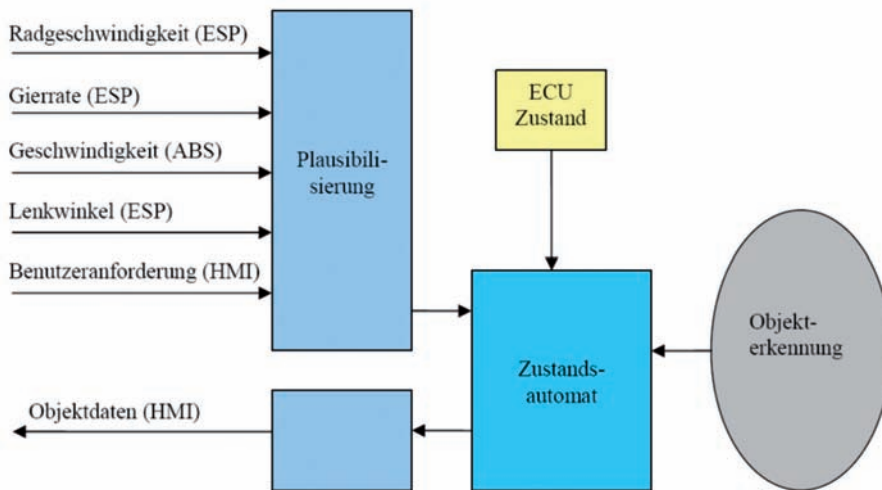


Abbildung 4: Ein- und Ausgangssignale

die physikalischen Abhängigkeiten des Testobjekts zu den übrigen Steuergeräten. Der zentrale Zustandsautomat ist ein besonders komplexes und umfangreiches Testobjekt, da er in direkter Beziehung zu den Signalen aller aufgezeigten Steuergeräte steht. Aufgrund der sich kontinuierlich ändernden Anforderungen während der Entwicklung wurden zahlreiche Erweiterungen vorgenommen, die sich mit einem Zustandsautomaten nicht konsequent modellieren lassen. **Abbildung 4** zeigt eine Zusammenfassung der für den Zustandsautomaten relevanten Signale. Grundsätzliche Randbedingung für die Durchführung der Tests war, dass nicht auf interne Signale oder Daten zugegriffen werden soll.

Testumgebung

Um die Durchführung der Integrationstests automatisieren zu können, wurde ein Integrationstestplatz eingerichtet (**Abbildung 5**). Mit Hilfe dieses Integrationstestplatzes werden unterschiedliche Situationen, die auf der Strasse auftreten können, simuliert und in den Test eingebunden. Mit Hilfe der SPS (Speicherprogrammierbare Steuerung) und programmierbaren Netzgeräte werden Kurzschlüsse, Fehlpotenziale sowie unterschiedliche Versorgungsspannungskurven am Bus sowie am zutestenden Steuergerät realisiert. Der gesamte Testablauf wird von einem Hostrechner mit der Softwarekomponente CANoe überwacht und durchgeführt. Außer der Überwachung und Steuerung des Testprozesses

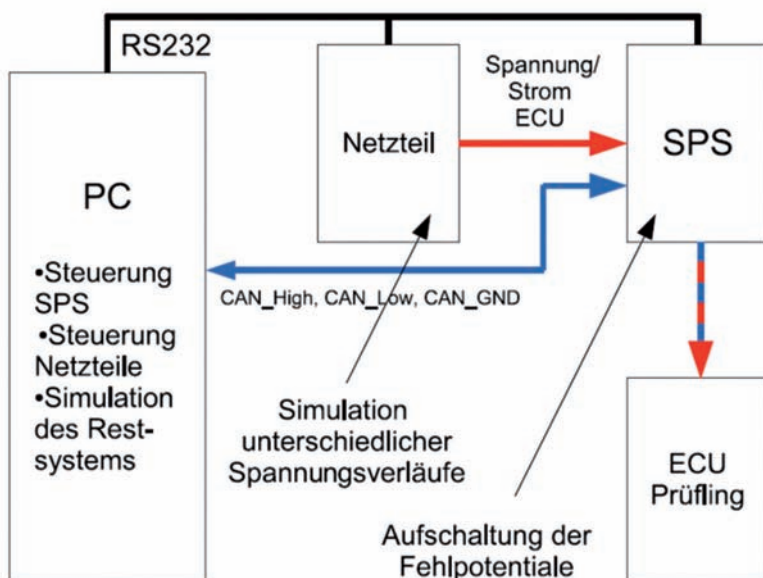


Abbildung 5: Schematischer Aufbau des Integrationstestplatzes

wird mit Hilfe von CANoe die imaginäre Vernetzung des Steuergerätes im Auto (Restbussimulation) ohne Vorhandensein von realen Steuergeräten, die unmittelbar mit dem Prüfling über CAN-Bus gekoppelt sein müssen, realisiert. Am Ende eines Tests erstellt CANoe einen Testbericht, in dem alle durchgeführten Testschritte der einzelnen Testfälle mit kurzen Beschreibungen aufgelistet sind. Der Testbericht dient als Grundlage für die spätere Testauswertung.

Anwendung der ESG-Methode und Testfallgenerierung

Im Rahmen einer Diplomarbeit (siehe [2]) wurden mit Hilfe der ESG-Methode Testfälle aufgestellt und auf der beschriebenen Testumgebung durchgeführt. Dabei wurden die Original-Lastenhefte für die Erstellung der Testspezifikation mit herangezogen.

Als Grundlage für die Knoten des ESGs dienen die system- und testumgebungsspezifischen Ereignisse. Die systemspezifischen Ereignisse sind die Ereignisse, die auf der Kommunikationsebene (Netzwerkebene) geschehen. Damit sind die Signale gemeint, deren Eingang bzw. Ausgang als Ereignis interpretiert werden. Die testumgebungsspezifischen Ereignisse beschreiben dagegen die äußeren Einflüsse auf das System. Zu solchen Ereignissen gehören die Aufschaltungen von Fehlpotenzialen sowie Kurzschlüsse im Gesamtsystem. Jedem Ereignis wird ein Programmcode zugeordnet, der beim Auftreten des Ereignisses ausgeführt wird.

Um die Zusammenhänge zwischen einzelnen Ereignissen besser verstehen zu können, müssen einzelne Punkte der Systemspezifikation mit Hilfe von Kombinations- oder Zustandsübergangstabellen zusammengeführt und damit die Aufgabenstellung präzisiert werden. Anhand dieser Tabellen wird dann entschieden, ob einem Ereignis ein oder mehrere Knoten im ESG zugeordnet werden müssen. Die Kantenbildung erfolgt anhand der Systemspezifikation und/oder durch logische Zusammenhänge. Die Erkenntnisse, die bei der Präzisierung der Aufgabenstellung (Aufstellung der Kombinations-, Zustandsübergangstabellen usw.) gewonnen wurden, müssen bei der Kantenbildung berücksichtigt werden. Ausgehend von einem Knoten (Startknoten bevorzugt) im ESG werden zwischen ihm und allen seinen direkten Nachfolgern Kanten angelegt. Diese Prozedur wird solange wiederholt, bis alle Knoten mit ihren

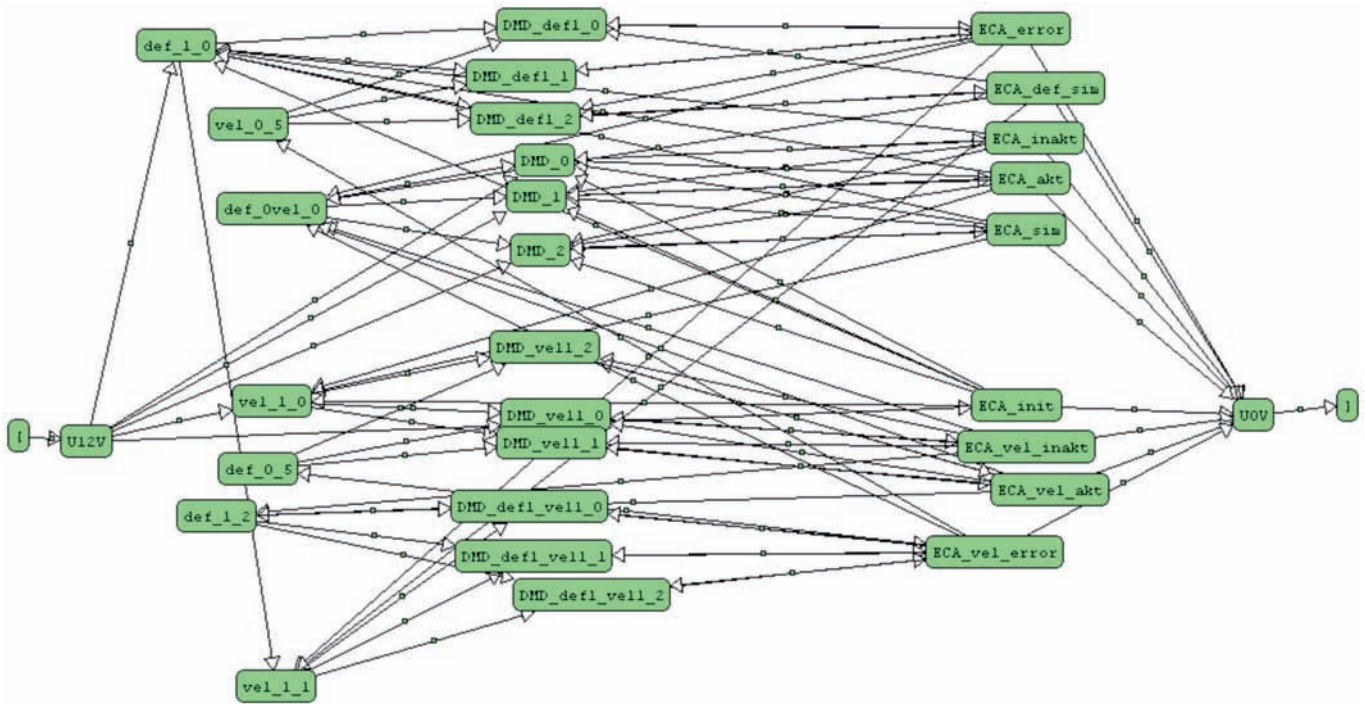


Abbildung 6: Beispiel eines Ereignis-Sequenz-Graphen zur Überprüfung der Zustandsübergänge des ACC

direkten Nachfolgern verbunden sind. Beim Anlegen der Kanten dürfen einzelne Knoten nicht isoliert, sondern müssen im Zusammenhang mit allen ihren Vorgängern betrachtet werden. **Abbildung 6** zeigt ein Beispiel. Aus diesen Modellen wurden automatisch Testfälle generiert, so dass sämtliche

Ereignispaare (Kanten und damit alle Knoten) überdeckt werden. Diese Testsequenzen wurden mittels des *Chinese Postman Algorithmus* minimiert. Zudem wurden Testfälle aufgestellt, um die ungültigen Ereignispaare des inversen Graphen abzudecken.

Manueller Testprozess mit Hilfe der Klassifikationsbaummethode

Zur selben Zeit hat ein Team von zwei Personen einen vollständigen Test nach der manuellen Methode, basierend auf Klassifikationsbäumen [3], spezifiziert und imple-

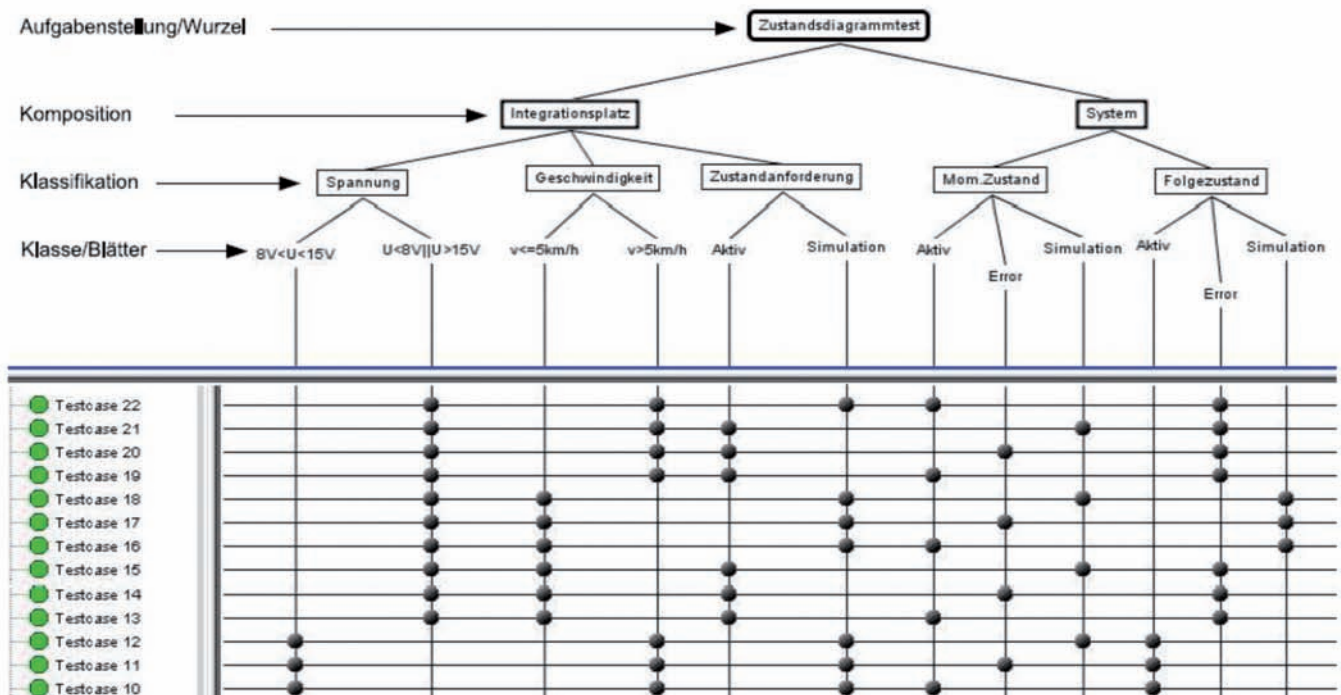


Abbildung 7: Beispiel Klassifikationsbaum

	Äquivalenz- klassenmethode	ESG
Stunden zur Aufstellung der Testspezifikation	120	152
Anzahl Testfälle	26	11
Anzahl Testschritte	240	62

Tabelle 2: Stundenvergleich

mentiert. Ein Beispiel ist in **Abbildung 7** gegeben. Bei der Erstellung der Testspezifikation zeigte sich, dass die Findung der Testziele relativ schnell und zügig möglich war. Aufwändig war dagegen die Verfeinerung der Testziele in einzelne Testschritte und die Pflege bei Änderungen. Die Zusammenhänge waren nicht mehr intuitiv verfolgbar.

Bewertung der Anwendbarkeit der ESG-Methode und Ausblick

Die Anzahl der benötigten Stunden, die für die Erstellung der Testspezifikation ge-

braucht wurden, nahm mit der ESG-Methode ca. 26 % mehr Zeit in Anspruch (siehe **Tabelle 2**). Die Anzahl der Testfälle konnte bei der ESG-Methode jedoch mehr als halbiert werden – die Anzahl der Testschritte reduzierte sich sogar fast um den Faktor vier.

Mit Hilfe der Ereignissequenzgraphen wurden zudem alle Fehler gefunden, die mit dem manuellen Klassifikationsbaum-Testprozess gefunden wurden. Zusätzlich zu diesen Fehlern wurde ein weiterer, erheblicher (da sicherheitskritischer) Fehler in der Implementierung des zentralen Zustandsautomaten gefunden (Fehler Nr. 3).

1	Befindet sich das System im Fehlerzustand, die Geschwindigkeit sinkt unter 5 km/h und es erfolgt eine Simulationsaufforderung, bleibt das System im Fehlerzustand anstatt in den Simulationszustand zu wechseln.
2	Beim Aufschalten des Fehlers wechselt das System nicht rechtzeitig in den Fehlerzustand.
3	Das System befindet sich im Fehlerzustand und die Fehlerbedingung verschwindet. Dann wechselt das Gerät nicht rechtzeitig in den angeforderten Zustand <i>Aktiv</i> [Fehler gefunden mit Hilfe von negativen Testfällen].
4	Wenn sich das System im Fehlerzustand befindet und die Fehlerbedingung verschwindet, wechselt das Gerät nicht rechtzeitig in den angeforderten Zustand <i>Inaktiv</i> [Fehler gefunden mit Hilfe von negativen Testfällen].
5	Das System wechselt nicht rechtzeitig aus dem Zustand <i>Init</i> in <i>Aktiv</i> oder <i>Inaktiv</i> .

Tabelle 3: Fehlerliste

Tabelle 3 gibt einen Überblick über die mit der ESG-Methode gefundenen Fehler.

Abschließend lässt sich sagen, dass die Zeit für die Implementierung der Testfälle und die Kosten, die durch die Testdurchführung entstehen, mit Hilfe der ESG-Methode erheblich reduziert wurden. Mit dem Ereignis-Sequenz-Graphen ist eine Methode gegeben, die die Automatisierung der Testvorbereitung ermöglicht und zudem das Regressionstesten vereinfacht. Weitere Schritte zur Effizienzsteigerung beispielsweise durch die Entwicklung eines Rahmenprogramms, das die generierten Testfälle aus den ESGs in ein von CANoe lesbares Format wie z. B. XML umsetzt, sind zudem möglich. ■

Literatur

1. Belli, F.; Budnik, C.J.; White, L.: Event-Based Modelling, Analysis and Testing of User Interactions: Approach and Case Study, Software Testing, Verification and Reliability, 16(1), S. 3-32, 2006
2. Beidinger, F.: Bewertung einer graphenbasierten Methode für den Testentwurf. Diplomarbeit Universität Paderborn, Fachgebiet Angewandte Datentechnik.
3. Grochtmann, M.; Grimm, K.: Classification Trees for Partition Testing. Software Testing, Verification and Reliability, 3(2), S. 63-82, John Wiley Verlag, 1993.