

**Interner Verteilerschlüssel:**

- (A)  Veröffentlichung im ABl.  
(B)  An Vorsitzende und Mitglieder  
(C)  An Vorsitzende  
(D)  Keine Verteilung

**Datenblatt zur Entscheidung  
vom 15. Juni 2011**

**Beschwerde-Aktenzeichen:** T 1805/07 - 3.5.06  
**Anmeldenummer:** 02760105.3  
**Veröffentlichungsnummer:** 1428126  
**IPC:** G06F 11/36  
**Verfahrenssprache:** DE

**Bezeichnung der Erfindung:**

Verfahren zur Softwareverifikation für Steuereinheiten und Verifikationssystem

**Anmelder:**

Robert Bosch GmbH

**Stichwort:**

Softwareverifikation/BOSCH

**Relevante Rechtsnormen:**

VOBK Art. 13(1)

**Relevante Rechtsnormen (EPÜ 1973):**

EPÜ Art. 56

**Schlagwort:**

"Erfinderische Tätigkeit - Hauptantrag und Hilfsanträge 1 und 2 (nein)"

"Zulässigkeit - Hilfsantrag 3 (nein)"

**Zitierte Entscheidungen:**

-

**Orientierungssatz:**

-



Aktenzeichen: T 1805/07 - 3.5.06

**ENTSCHEIDUNG**  
der Technischen Beschwerdekammer 3.5.06  
vom 15. Juni 2011

**Beschwerdeführer:** Robert Bosch GmbH  
Postfach 30 02 20  
D-70442 Stuttgart (DE)

**Vertreter:** -

**Angefochtene Entscheidung:** Entscheidung der Prüfungsabteilung des Europäischen Patentamts, die am 2. Juli 2007 zur Post gegeben wurde und mit der die europäische Patentanmeldung Nr. 02760105.3 aufgrund des Artikels 97(1) EPÜ 1973 zurückgewiesen worden ist.

**Zusammensetzung der Kammer:**

**Vorsitzender:** D. H. Rees  
**Mitglieder:** S. Krischer  
C. Heath

## Sachverhalt und Anträge

I. Die Beschwerde richtet sich gegen die Entscheidung der Prüfungsabteilung, verkündet in der mündlichen Verhandlung am 19. Juni 2007 und zugestellt mit Schreiben vom 2. Juli 2007, mit der die Patentanmeldung mit der Nummer 02 760 105.3 zurückgewiesen wurde.

Der Gegenstand von Anspruch 1 des einzigen Antrags wurde als nicht erfinderisch erachtet über die Kombination der Dokumente:

D1 K. Oertel: "Code per Klick - Von der Idee zum fertigen Produkt mit ASCET-SD 3.0", Elektronik Industrie, April 1999, Seiten 95-96, XP2262103.

D2 D. Hoffman: "A Taxonomy of Test Oracles", Internet-Artikel, 16. Januar 1998, Seiten 1-4, XP2262104, gefunden im Internet:  
<URL:<http://fsoftwarequalitymethods.com/SQM/Abstracts/ATaxonomyabstract.PDF>> [gefunden am 19. November 2003].

II. Beschwerde wurde am 16. August 2007 eingelegt. Die Gebühr wurde am selben Tag entrichtet. Die Begründung ist am 25. September 2007 eingegangen. Anbei wurde der Anspruchssatz des aktuellen Hauptantrags eingereicht.

III. In einer Ladung zur mündlichen Verhandlung teilte die Kammer dem Beschwerdeführer ihre vorläufige Meinung mit, wonach die Entscheidung zu bestätigen sei. Zu dem in Anspruch 1 des Hauptantrags hinzugefügten Merkmal der Bestimmung der Softwarecodeabdeckung wurde folgendes Dokument aus dem Prüfungsverfahren angeführt:

- D3 B. Milne: "Entry-level system integrates microprocessor development tools", Electronic Design, Penton Publishing, Cleveland, OH, USA, Bd. 36, Nr. 2, 21. Januar 1988, Seiten 43-44, 46, 48, XP112094.
- IV. In einem Schreiben, eingegangen am 16. Mai 2011, wurden neue Anspruchssätze der drei Hilfsanträge ohne Argumente eingereicht.
- V. In einem Schreiben, eingegangen am 8. Juni 2011, wurde eine Argumentationslinie für den Hauptantrag mitgeteilt.
- VI. Am 15. Juni 2011 fand die mündliche Verhandlung statt, in welcher der Beschwerdeführer das folgende Dokument einführte:
- D5 Robert Bosch GmbH: "DS Software Life Cycle Model";  
Papierausdruck von Vortragsfolien; 17. Januar 2005;  
Seiten 11 und 12.
- Dieses Dokument solle laut Beschwerdeführer nur das Hintergrundwissen erläutern; es stelle keinen Stand der Technik dar.
- VII. Der Beschwerdeführer beantragt, die angefochtene Entscheidung aufzuheben und ein Patent zu erteilen auf der Basis des Hauptantrags (Ansprüche 1-9) oder alternativ auf Basis der Hilfsanträge 1-3 (jeweils mit Anspruch 1 und weiteren Ansprüchen aus dem Hauptantrag angepasst übernommen), als Beschreibung Seite 1 eingegangen am 25. Mai 2006, und Seiten 2-15 wie veröffentlicht, und als Zeichnung die Blätter 1-3 wie veröffentlicht.

VIII. Anspruch 1 des Hauptantrags lautet wie folgt (*Kursivschrift* hinzugefügt zum Markieren der Unterschiede zum erstinstanzlich zurückgewiesenen Antrag):

"Verfahren zur Verifikation von Softwarefunktionen für eine Steuereinheit, mit Verwendung eines Simulationsmodells zur Abbildung der Softwarefunktionen und der Steuereinheit, *dadurch gekennzeichnet, dass* aus dem identischen Simulationsmodell zum einen für eine erste Experimentalsteuereinheit und zum zweiten für eine zweite Seriensteuereinheit der Softwarecode für die Softwarefunktionen automatisch generiert wird, *wobei die Experimentalsteuereinheit einen Zugriff auf das Verhalten der Softwarefunktion ermöglicht, wobei* identische Eingangsgrößen für die Softwarefunktionen auf beiden Steuergeräten zum Einsatz kommen und die sich daraus ergebenden Ausgangsgrößen beider Steuereinheiten zeitsynchron erfasst werden, *wobei durch Vergleich der Ausgangsgrößen beider Steuereinheiten die Softwarefunktionen verifiziert werden, und dass zusätzlich bei der Verifikation die durchlaufenen Pfade im Softwarecode und/oder in den Softwarefunktionen zur Bestimmung der Softwarecodeabdeckung auf der Experimentalsteuereinheit erfasst werden.*"

IX. Anspruch 1 des Hilfsantrags 1 lautet wie folgt (*Kursivschrift* hinzugefügt zum Markieren der Unterschiede zum Hauptantrag):

"Verfahren zur Verifikation von Softwarefunktionen für eine Steuereinheit, mit Verwendung eines Simulationsmodells zur Abbildung der Softwarefunktionen und der Steuereinheit, *wobei* aus dem identischen Simulationsmodell zum einen für eine erste Experimentalsteuer-

einheit und zum zweiten für eine zweite Seriensteuer-  
einheit Softwarecode für die Softwarefunktionen auto-  
matisch generiert wird, dadurch gekennzeichnet, dass der  
*Softwarecode für die Softwarefunktionen jeweils für  
Experimentalsteuereinheit oder Seriensteuereinheit mit  
unterschiedlichen Erzeugungsmitteln generiert wird,*  
wobei die Experimentalsteuereinheit einen Zugriff auf  
das Verhalten der Softwarefunktion ermöglicht, wobei  
identische Eingangsgrößen für die Softwarefunktionen auf  
beiden Steuergeräten zum Einsatz kommen und die sich  
daraus ergebenden Ausgangsgrößen beider Steuereinheiten  
zeitsynchron erfasst werden, wobei durch Vergleich der  
Ausgangsgrößen beider Steuereinheiten die Software-  
funktionen verifiziert werden, und dass zusätzlich bei  
der Verifikation die durchlaufenen Pfade im Softwarecode  
und/oder in den Softwarefunktionen zur Bestimmung der  
Softwarecodeabdeckung auf der Experimentalsteuereinheit  
erfasst werden."

- X. Anspruch 1 des Hilfsantrags 2 lautet wie folgt (*Kursiv-  
schrift* hinzugefügt zum Markieren der Unterschiede zum  
Hilfsantrag 1):

"Verfahren zur Verifikation von Softwarefunktionen für  
eine Steuereinheit, mit Verwendung eines Simulations-  
modells zur Abbildung der Softwarefunktionen und der  
Steuereinheit, wobei aus einem identischen Simulations-  
modell zum einen für eine erste Experimentalsteuer-  
einheit und zum zweiten für eine zweite Seriensteuer-  
einheit Softwarecode für die Softwarefunktionen automa-  
tisch generiert wird, wobei die Experimentalsteuer-  
einheit einen Zugriff auf das Verhalten der Software-  
funktion ermöglicht, dadurch gekennzeichnet, dass der  
Softwarecode für die Softwarefunktionen jeweils für

Experimentalsteuereinheit oder Seriensteuereinheit mit unterschiedlichen Erzeugungsmitteln generiert wird, wobei der Softwarecode in hardwareabhängigen Softwarecode und hardwareunabhängigen Softwarecode unterteilt ist, wobei identische Eingangsgrößen für die Softwarefunktionen auf beiden Steuergeräten zum Einsatz kommen und die sich daraus ergebenden Ausgangsgrößen beider Steuereinheiten zeitsynchron erfasst werden, wobei durch Vergleich der Ausgangsgrößen beider Steuereinheiten die Softwarefunktionen verifiziert werden, und dass zusätzlich bei der Verifikation die durchlaufenen Pfade im Softwarecode und/oder in den Softwarefunktionen zur Bestimmung der Softwarecodeabdeckung auf der Experimentalsteuereinheit erfasst werden."

- XI. Anspruch 1 des Hilfsantrags 3 lautet wie folgt (*Kursivschrift* hinzugefügt zum Markieren der Unterschiede zum Hilfsantrag 2):

"Verfahren zur Verifikation von Softwarefunktionen für eine Steuereinheit, mit Verwendung eines Simulationsmodells zur Abbildung von Softwarefunktionen und der Steuereinheit, dadurch gekennzeichnet, dass aus dem Simulationsmodell mit unterschiedlichen Erzeugungsmitteln hardwareabhängige Software für eine Experimentalsteuereinheit und eine Seriensteuereinheit generiert wird, wobei die Experimentalsteuereinheit einen Zugriff auf das Verhalten der Softwarefunktion ermöglicht, wobei hierdurch eine Softwarecodeabdeckung auf der Experimentalsteuereinheit ermittelt werden kann, wobei zur Verifikation des Zeitverhaltens der Seriensteuereinheit während eines Funktionstests identische Eingangsgrößen für die Steuereinheiten verwendet werden

*und die Ausgangsgrößen der Steuereinheiten zeitsynchron erfasst und verglichen werden."*

XII. Am Ende der mündlichen Verhandlung verkündete der Vorsitzende die Entscheidung der Kammer.

## **Entscheidungsgründe**

### 1. *Zulässigkeit der Beschwerde*

Die Beschwerde ist zulässig, siehe obige Abschnitte I und II.

### 2. *Zulässigkeit der Anträge und ursprüngliche Offenbarung*

2.1 Anspruch 1 des Hilfsantrags 1 bzw. 2 enthält zusätzlich zu Anspruch 1 des Hauptantrags die Merkmale des abhängigen Anspruchs 3 bzw. der Ansprüche 3 und 6 des Hauptantrags. Sie stellen zudem nach Ansicht der Kammer einen ernsthaften Versuch dar, die erhobenen Einwände zu beheben.

Daher werden die Hilfsanträge 1 und 2 zum Verfahren zugelassen.

2.2 Hilfsantrag 3 unterscheidet sich von den vorhergehenden Anträgen durch eine sehr starke Umformulierung des kennzeichnenden Teils. Dabei wurden unter anderem die Adjektive "identisch" (vor "Simulationsmodell" in Zeile 3 von Anspruch 1 des Hauptantrags) und "automatisch" (vor "generiert" in Zeile 6) gelöscht. Außerdem wurde der Ausdruck "zur Verifikation des Zeitverhaltens" hinzugefügt.

2.2.1 Die im Schreiben vom 16. Mai 2011 angegebenen Beschreibungspassagen enthalten nichts, das als Basis für diese Änderungen infrage käme. Auch in der mündlichen Verhandlung wurde keine Basis angegeben, insbesondere nicht für eine Offenbarung der Verifikation des Zeitverhaltens. Selbst in der von der Kammer angeführten Passage auf Seite 10, Absatz 3, Zeilen 3-8 wird nur eine *Spezifi-*

*kation* und *Simulation* des Zeitverhaltens offenbart, nicht eine *Verifikation*. Nach Auffassung der Kammer stellt die *Simulation* des Zeitverhaltens eine *Voraussetzung* des zeitsynchronen Ausgabevergleichs dar. Das Zeitverhalten selbst ist aber nicht Gegenstand der *Verifikation*.

Der Beschwerdeführer charakterisierte überdies in der mündlichen Verhandlung die sehr starke Umformulierung des Hilfsantrags 3 als *Klarstellung*, die bzgl. der erfinderischen Tätigkeit nichts hinzufügen möchte.

2.2.2 Im Lichte dieser Erwägungen übt die Kammer ihr Ermessen aus, Hilfsantrag 3 gemäß Artikel 13(1) der Verfahrensordnung der Beschwerdekammern (VOBK) nicht zum Verfahren zuzulassen.

### 3. *Erfinderische Tätigkeit von Anspruch 1*

#### 3.1 Hauptantrag

3.1.1 Dokument D1 wird wie im gesamten bisherigen Verfahren als nächstliegender Stand der Technik angesehen. Das darin beschriebene Entwicklungssystem ASCET-SD der Fa. ETAS wird auch in der Beschreibung mehrmals erwähnt.

3.1.2 Der erstinstanzlich zurückgewiesene Anspruch 1 unterscheidet sich - zusammengefasst ausgedrückt - von D1 durch einen "synchronen Ausgabevergleich": d.h. identische Eingangsgrößen werden beiden Steuergeräten eingegeben, und die sich daraus ergebenden Ausgaben werden synchron erfasst und verglichen.

Dies wurde vom Beschwerdeführer nicht bestritten.

3.1.3 In der angefochtenen Entscheidung wird der synchrone Ausgabenvergleich aus D2 herausgelesen und mit dem Verfahren aus D1 kombiniert. Die Experimentalsteuereinheit des Anspruchs wird mit dem Test-Orakel aus D2 identifiziert.

In der Beschwerdebeurteilung wird auf Seite 2, letzter Satz eingeräumt, dass in D2, Seite 3 beschrieben wird, dass "reales System und Oracle parallel betrieben werden können".

3.1.4 Im Schreiben vom 8. Juni 2011, Seite 4 und in der mündlichen Verhandlung argumentiert der Beschwerdeführer, dass die Integration des Test-Orakels aus D2 in das Verfahren aus D1 einen *unterschiedlichen Algorithmus* für die Ermittlung der erwarteten Ausgabe auf der Experimentalsteuereinheit erfordern würde, d.h. einen anderen als den in der Seriensteuereinheit benutzten. Die Aussagekraft der Tests sei gemäß D2 umso größer, je weniger das Test-Orakel mit der zu testenden Funktion gemeinsam hätte. Deshalb würde der Fachmann einen anderen Algorithmus wählen. Außerdem spreche D2 von der "function under test (FUT)" (siehe D2, Seite 2, Abschnitt 1). Dies zeige, dass es in D2 um einen *Funktions-Test* gehe, im Gegensatz zum *System-Test* der Anmeldung. Siehe dazu auch das während der mündlichen Verhandlung eingereichte Dokument D5. Darin werde der Funktions-Test "Unit Test" genannt und wurde von hand mit "FUT" markiert, während der System-Test mit "SUT" ("system under test") gekennzeichnet wurde.

Ein weiterer Unterschied von D2 zum Anspruch sei, dass ein Orakel eine *Spezifikation der Testfälle* erfordere, die alle erwarteten Ausgangsgrößen in Abhängigkeit der

Eingangsgrößen abdecken. (Schreiben vom 8. Juni 2011, Seite 4, Absatz 2).

3.1.5 Diese Interpretation von D2 ist jedoch eine verengte Darstellung. Zum einen offenbart D2 in der Aufzählung auf Seite 1 die unterschiedlichsten Arten von Test-Orakel. So wird als Punkt 6 folgende Methode aufgeführt, um erwartete Ergebnisse zu erzeugen:

- "• Same version of software on a different hardware platform"

Das ist also ein mögliches Test-Orakel (zu dessen Definition siehe Seite 1, Abschnitt 2 "Background:", Zeilen 1 und 4).

Dieses Test-Orakel ist offensichtlich so zu verstehen, dass der gleiche Programmtext (mit identischer Versionsnummer) genommen wird und aus ihm für eine andere Hardware-Plattform ausführbarer Code als konkretes Test-Orakel erzeugt wird. Dies bedeutet, dass nicht nur kein unterschiedlicher Algorithmus als Test-Orakel genommen werden muss, sondern sogar vorgeschlagen wird, diese Implementierung des Algorithmus, also den gleichen Programmtext dafür zu nehmen. Ebenso verlangt es der Anspruch ab Zeile 3:

"dass aus dem identischen Simulationsmodell zum einen für eine erste Experimentalsteuereinheit und zum zweiten für eine zweite Seriensteuereinheit der Softwarecode ... automatisch generiert wird"

3.1.6 Es ist außerdem klar, dass für diese Variante die Testfälle keineswegs spezifiziert werden müssen.

3.1.7 Zum andern spricht D2 in der Tat von einer "function under test", aber D2 thematisiert gar nicht den Moment im Software-Lebenszyklus, zu welchem getestet wird. So sagt D2 nicht, ob das Orakel für einen "Unit Test", "Integration Test" oder "System Test" verwendet wird, sondern erläutert nur, was gegeneinander getestet werden kann.

3.1.8 Daher würde es ein Fachmann als offensichtlich ansehen, das Merkmal des synchronen Ausgabenvergleichs aus D2 in das Verfahren von D1 zu integrieren, um die in der angefochtenen Entscheidung angegebene technische Aufgabe zu lösen, nämlich (siehe Abschnitt II, 1c)):

"Wie kann die Korrektheit der Software eines Seriensteuergeräts getestet werden, wenn ein funktionierendes Referenzsystem vorliegt."

3.1.9 Der *aktuelle Anspruch 1* enthält gegenüber dem erstinstanzlich zurückgewiesenen zwei zusätzliche Merkmale:

- die Ermöglichung des Zugriffs auf das Verhalten der Softwarefunktion durch die Experimentalsteuereinheit und
- die Erfassung der Codeabdeckung der durchlaufenen Pfade im Softwarecode und/oder in den Softwarefunktionen (aus dem ursprünglichen Anspruch 4).

3.1.10 Auch diese beiden Merkmale vermögen indessen nicht, die erfinderische Tätigkeit herzustellen. Zum einen ist der Zugriff auf das Software-Verhalten durch die Experimentalsteuereinheit ebenfalls in D1 offenbart, und zwar auf Seite 95, linke Spalte, Zeile 14 (*Kursivschrift* hinzugefügt): "... das sogenannte zielsystem-identische

Prototyping eingeführt. Dabei kann die gesamte Arithmetik und das Zeitverhalten auf dem Entwicklungssystem spezifiziert, simuliert und mit großer Genauigkeit ausgegeben werden.",

und Zeile 25: "Mit ASCET-SD lässt sich das Verhalten von Software realistisch untersuchen. Das bedeutet, dass das Echtzeitverhalten des Produkts nicht simuliert, sondern ausgeführt wird. Das Verhalten ist somit zielsystemidentisch und das Zeitverhalten der Applikation verifizierbar."

- 3.1.11 Zum ändern findet sich das zweite neu hinzugefügte Merkmal, also die Erfassung der Codeabdeckung, zwar weder in D1 noch in D2, aber - wie im Internationalen Vorläufigen Prüfungsbericht, Abschnitt 2.3 zum ursprünglichen Anspruch 4 gesagt - in Dokument D3. Darüberhinaus wird es als eine jedem Fachmann geläufige Methode angesehen, um ausreichendes Testen sicherzustellen.
- Die Anmeldung selbst bezeichnet die Codeabdeckungsanalyse als "etablierte Testtechnologie" (Seite 2, Zeilen 6 und 7).

- 3.1.12 Die Erfassung der Codeabdeckung ist darüber hinaus inhaltlich *unabhängig* vom synchronen Ausgabevergleich der beiden Einheiten. Seine Integration in Anspruch 1 eröffnet eine *zweite technische Teilaufgabe*, nämlich:

"die Zuverlässigkeit der Verifikation erhöhen".

- 3.1.13 Die Argumentation im Schreiben vom 6. Juni 2011, Seite 5, Abschnitt "2. Zweite technische Teilaufgabe", dass keine zweite technische Teilaufgabe vorläge, da es denkbar sei, dass die Seriensteuereinheit über den *Trigger (301)* aus

Figur 3 die Codeabdeckungsanalyse in der Experimentalsteuereinheit starte, hat die Kammer nicht überzeugt. Diese Annahme ist spekulativ und durch nichts in der Beschreibung gestützt. Außerdem würde eine solche Erklärung des Steuersignalflusses nichts daran ändern, dass die Codeabdeckungserfassung die Zuverlässigkeit der Verifikation zu erhöhen ermöglicht (zweite technische Teilaufgabe), aber nicht die Korrektheit der Software eines Seriensteuergeräts testet, wenn ein funktionierendes Referenzsystem vorliegt (erste technische Teilaufgabe). Es liegt also eine zweite Teilaufgabe vor.

3.1.14 Zur Lösung dieser zweiten Teilaufgabe würde der Fachmann es als offensichtlich ansehen, die Erfassung der Codeabdeckung aus D3 in das Verfahren aus D1 einzubauen.

3.1.15 Es wird in der Beschwerdebegründung argumentiert (Seite 2, Absätze 1-2), dass die Seriensteuereinheit nicht genügend Ressourcen habe, die durchlaufenen Pfade mitzuprotokollieren. Daher erfasse die Erfindung die Pfade auf der Experimentalsteuereinheit. Wenn diese eine ausreichende Codeabdeckung während der Verifikation aufweise, könne daraus geschlossen werden, dass auch die synchron getestete Seriensteuereinheit eine solche aufweise.

3.2 Allerdings stellt die Beschreibung die nicht beanspruchte Option, dass die Codeabdeckungsanalyse auf der *Seriensteuereinheit* ablaufen könne, gleichrangig neben die beanspruchte Option (Seite 12, Absatz 2, Zeile 4). In Absatz 4, Zeile 1 wird dies wiederholt und ab Zeile 3 wird ausgeführt, dass die nicht beanspruchte Option *vorteilhaft* sei, da dann die Codeabdeckungsanalyse auch die HW-abhängige Plattformsoftware umfasse.

3.3 Die Kammer stimmt zu, dass es ein übliches Bestreben ist, die Kosten - und damit auch die Ressourcen - der Seriensteuereinheit auf ein Minimum zu reduzieren. Es kann also durchaus sein, dass die Ressourcen einer geplanten Seriensteuereinheit nicht ausreichen, die Codeabdeckungserfassung auszuführen. In so einem Fall, in welchem man zusätzlich eine Experimentalsteuereinheit zur Hand hat, die nicht ebenso ressourcenbeschränkt ist, betrachtet es die Kammer als offensichtlich für einen Fachmann, die Codeabdeckungserfassung von der bevorzugten Plattform, der Seriensteuereinheit, zur zweiten Möglichkeit, der Experimentalsteuereinheit, zu übertragen.

3.3.1 Daher ist der Gegenstand von Anspruch 1 des Hauptantrags nicht erfinderisch im Sinne von Artikel 56 EPÜ.

3.4 Hilfsantrag 1

3.4.1 Anspruch 1 von Hilfsantrag 1 enthält gegenüber Anspruch 1 des Hauptantrags zusätzlich das Merkmal aus Anspruch 3 des Hauptantrags (identisch mit dem ursprünglichen Anspruch 3), nämlich dass der Softwarecode für die Softwarefunktionen jeweils für Experimentalsteuereinheit oder Seriensteuereinheit mit unterschiedlichen Erzeugungsmitteln generiert wird.

3.4.2 Dieses Merkmal ist jedoch unvermeidlich, da sowohl im beanspruchten Verfahren als auch in demjenigen von D1 zwei unterschiedliche Hardware-Plattformen (eine für die Experimentalsteuereinheit und eine für die Seriensteuereinheit) benutzt werden. Das bedeutet, dass zwei Codeerzeugungsmittel notwendig sind (für jede Plattform ein

eigenes).

3.4.3 Daher ist der Gegenstand von Anspruch 1 des Hilfsantrags 1 nicht erfinderisch im Sinne von Artikel 56 EPÜ.

3.5 Hilfsantrag 2

3.5.1 Anspruch 1 von Hilfsantrag 2 enthält gegenüber Anspruch 1 des Hilfsantrags 1 zusätzlich einen Teil des Merkmals aus Anspruch 6 des Hauptantrags (identisch mit dem ursprünglichen Anspruch 7), nämlich dass der Softwarecode in hardwareabhängigen Softwarecode und hardwareunabhängigen Softwarecode unterteilt ist. Es fehlt die Formulierung aus Anspruch 6, dass nur der hardwareunabhängige Softwarecode Softwarefunktionen zur Steuerung von Betriebsabläufen bei einem Fahrzeug dient. Dies hätte jedoch nicht in Anspruch 1 des Hilfsantrags 1 gepasst, da dort nicht von einem Fahrzeug die Rede ist.

3.5.2 Ein Fachmann weiß, dass es normalerweise in jedem Computer hardwareabhängige und hardwareunabhängige Software gibt. Treiber sind beispielsweise hardwareabhängig, während Programme in der Programmiersprache Java hardwareunabhängig sind. Somit ist es offensichtlich den Softwarecode in hardwareabhängigen und hardwareunabhängigen zu unterteilen.

3.5.3 Daher ist der Gegenstand von Anspruch 1 des Hilfsantrags 2 nicht erfinderisch im Sinne von Artikel 56 EPÜ.

**Entscheidungsformel**

**Aus diesen Gründen wird entschieden:**

Die Beschwerde wird zurückgewiesen.

Die Geschäftsstellenbeamtin:

Der Vorsitzende

B. Atienza Vivancos

D. H. Rees