

(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES
PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro



(43) Internationales Veröffentlichungsdatum
29. Dezember 2010 (29.12.2010)

PCT

(10) Internationale Veröffentlichungsnummer
WO 2010/149489 A1

(51) Internationale Patentklassifikation:
G06F 11/07 (2006.01) **G05B 23/02** (2006.01)

(21) Internationales Aktenzeichen: PCT/EP2010/057977

(22) Internationales Anmeldedatum:
8. Juni 2010 (08.06.2010)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
10 2009 030 774.5 27. Juni 2009 (27.06.2009) DE

(71) Anmelder (für alle Bestimmungsstaaten mit Ausnahme
von US): **BAYERISCHE MOTOREN WERKE AKTI-
ENGESSELLSCHAFT** [—/DE]; Petuelring 130, 80809
München (DE).

(72) Erfinder; und

(75) Erfinder/Anmelder (nur für US): **KAISER, Dietmar**
[DE/DE]; In der Feldkirchner Au 13, 85368 Moosburg
(DE). **HANEBERG, Michael** [DE/DE]; Liscowstr. 3 A,
81739 Muenchen (DE). **BECHLER, Marc** [DE/DE];
Schinkelstraße 32, 80805 München (DE).

(74) Gemeinsamer Vertreter: **BAYERISCHE MOTOREN
WERKE AKTIENGESSELLSCHAFT**; Patentabteilung
AJ-3, 80788 München (DE).

(81) Bestimmungsstaaten (soweit nicht anders angegeben, für
jede verfügbare nationale Schutzrechtsart): AE, AG, AL,
AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY,
BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DK, DM, DO,
DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP,
KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD,
ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI,
NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD,
SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR,
TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Bestimmungsstaaten (soweit nicht anders angegeben, für
jede verfügbare regionale Schutzrechtsart): ARIPO (BW,
GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ,
UG, ZM, ZW), eurasisches (AM, AZ, BY, KG, KZ, MD,
RU, TJ, TM), europäisches (AL, AT, BE, BG, CH, CY,
CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS,
IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO,
SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM,
GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Fortsetzung auf der nächsten Seite]

(54) Title: METHOD FOR THE COMPUTER-AIDED DETECTION OF ERRORS DURING THE EXECUTION OF ONE OR
MORE SOFTWARE-BASED PROGRAMS IN A SYSTEM OF COMPONENTS

(54) Bezeichnung : VERFAHREN ZUR RECHNERGESTÜTZTEN ERFASSUNG VON FEHLERN BEIM ABLAUF VON EI-
NEM ODER MEHREREN SOFTWARE-BASIERTEN PROGRAMMEN IN EINEM SYSTEM AUS KOMPONENTEN

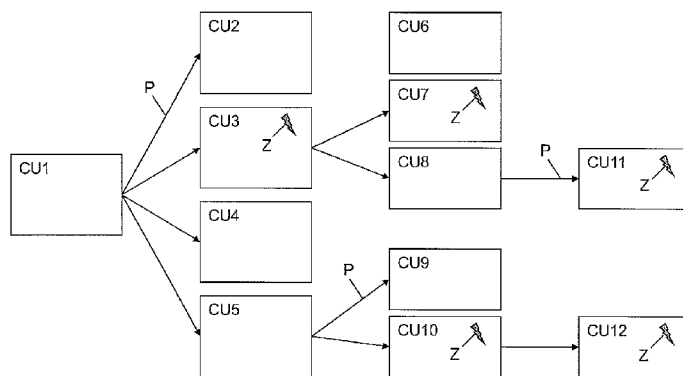


Fig. 2

(57) Abstract: The invention describes a method for the computer-aided detection of errors during the execution of one or more software-based programs in a system of components (CU1, CU2,..., CU12), particularly in a motor vehicle, wherein an identification (I) is assigned to each component (CU1, CU2,..., CU12) and a respective running program is specified by a program identity (a, b), wherein during the execution of the program a component (CU1, CU2,..., CU12) executes a program portion associated therewith (CU1, CU2,..., CU12) and the components (CU1, CU2,..., CU12) call each other reciprocally, at least to some extent. In the method according to the invention, when a component (CUB) is called by another component (CUA), the program identity (a, b) and an error parameter (f) are transmitted from the other component (CUA) to the component (CUB), said error parameter (f) indicating whether an error has been identified beforehand during the execution of the program. If a component (CUA, CUB) identifies an error during the execution of the program portion associated with said component, the component (CUA, CUB) stores an active error entry which contains the program identity (a, b), the identification (i) of

[Fortsetzung auf der nächsten Seite]



WO 2010/149489 A1

**Veröffentlicht:**

— mit internationalem Recherchenbericht (Artikel 21 Absatz 3)

the component (CUA, CUB) and an error status (e), which indicates whether the identified error is the first error during the execution of the program. If a component (CUA) which has called another component receives the identification (p) of the other component (CUB) from the other component (CUB), the component (CUA) stores a passive error entry which contains the program identity (a, b), the identification (i) of the component (CUA) and the identification (p) of the other component (CUB). A component (CUB) which stores one or more active or passive error entries during the execution of the program returns the program identity (a, b) and the identification of the component (CUB), at least once during the execution of the program, to that component (CUA) which has called the component (CUB).

(57) Zusammenfassung: Die Erfindung beschreibt ein Verfahren zur rechnergestützten Erfassung von Fehlern beim Ablauf von einem oder mehreren Software basierten Programmen in einem System aus Komponenten (CU1, CU2,..., CU12), insbesondere in einem Kraftfahrzeug, wobei den Komponenten (CU1, CU2,..., CU12) jeweils eine Identifikation (i) zugeordnet ist und ein jeweiliges ablaufendes Programm durch eine Programmidentität (a, b) spezifiziert ist, wobei im Ablauf des Programms eine Komponente (CU1, CU2,... CU12) einen ihr (CU1, CU2,..., CU12) zugeordneten Programmteil ausführt und sich die Komponenten (CU1, CU2,..., CU12) zumindest teilweise gegenseitig aufrufen. In dem erfindungsgemässen Verfahren werden beim Aufruf einer Komponente (CUB) durch eine andere Komponente (CUA) die Programmidentität (a, b) sowie ein Fehlerparameter (f) von der anderen Komponente (CUA) an die Komponente (CUB) übermittelt, wobei der Fehlerparameter (f) angibt, ob im Ablauf des Programms zuvor ein Fehler erkannt wurde. Falls eine Komponente (CUA, CUB) beim Ausführen des ihr zugeordneten Programmteils einen Fehler erkennt, speichert die Komponente (CUA, CUB) einen aktiven Fehlereintrag, der die Programmidentität (a, b), die Identifikation (i) der Komponente (CUA, CUB) und einen Fehlerstatus (e) enthält, der angibt, ob der erkannte Fehler der erste Fehler im Ablauf des Programms ist. Falls eine Komponente (CUA), welche eine andere Komponente aufgerufen hat, von der anderen Komponente (CUB) die Identifikation (p) der anderen Komponente (CUB) empfängt, speichert die Komponente (CUA) einen passiven Fehlereintrag, der die Programmidentität (a, b), die Identifikation (i) der Komponente (CUA) und die Identifikation (p) der anderen Komponente (CUB) enthält. Eine Komponente (CUB), welche einen oder mehrere aktive oder passive Fehlereinträge im Ablauf des Programms speichert, gibt zumindest einmal im Ablauf des Programms die Programmidentität (a, b) und die Identifikation der Komponente (CUB) an diejenige Komponente (CUA) zurück, durch welche die Komponente (CUA) aufgerufen wurde.

Verfahren zur rechnergestützten Erfassung von Fehlern beim Ablauf von einem oder mehreren software-basierten Programmen in einem System aus Komponenten

Die Erfindung betrifft ein Verfahren zur rechnergestützten Erfassung von Fehlern beim Ablauf von einem oder mehreren software-basierten Programmen in einem System aus Komponenten, insbesondere in einem Kraftfahrzeug.

In einer Vielzahl von technischen Bereichen, insbesondere auch auf dem Gebiet der Automobiltechnik, wird der Anteil an Software in auszuführenden Funktionen der entsprechenden technischen Anlage bzw. des Fahrzeugs immer komplexer. Im Regelfall werden Software-Programme dabei in einem verteilten System aus mehreren Komponenten ausgeführt, wobei hier und im Folgenden der Begriff der Komponente weit zu verstehen ist und jede Einheit mit einer oder mehreren Funktionalitäten beim Programmablauf umfassen kann. Insbesondere ist eine Komponente eine technische Komponente mit technischer Funktionalität bzw. ein technisches Gerät, wie z.B. ein Steuergerät, mit entsprechender Software bzw. ein Teil dieses Geräts. Ebenso können Komponenten auch Funktionen bzw. Objekte im Sinne einer Programmiersprache betreffen, welche Programmteile innerhalb des Systems ausführen. Eine Komponente kann insbesondere auch eine Software-Komponente im Sinne des bekannten Standards AUTOSAR sein.

Software-Fehler treten beim Ablauf eines Programms immer zur Laufzeit und meist sporadisch auf. Ferner stehen Ursache und Wirkung solcher Fehler oft nicht im zeitlichen Zusammenhang. Dadurch wird die Fehlerbehebung beim Auftreten von Software-Fehlern wesentlich erschwert, da die Software-Fehler nicht reproduziert werden können, die Bedingungen ihres Eintretens nicht nachvollzogen werden können und generell unklar ist, wo die Ursache der Fehler zu suchen ist.

Im Bereich der Fahrzeugdiagnose werden derzeit erkannte Fehler in der Form sog. DTCs (Diagnostic Trouble Codes) in entsprechenden technischen Komponenten bzw. Steuergeräten abgelegt. Zwischen den hierdurch generierten Fehlereinträgen gibt es jedoch keine funktionale Verknüpfung und die Fehler lassen sich lediglich über den Zeitpunkt des Auftretens in Beziehung setzen.

Aufgabe der Erfindung ist es deshalb, Fehler beim Ablauf eines Software-Programms in geeigneter Weise zu erfassen, um die erfassten Fehler nachfolgend einer verbesserten Diagnose zum Auffinden der Fehlerursache unterziehen zu können.

Diese Aufgabe wird durch das Verfahren gemäß Patentanspruch 1 bzw. das Verfahren gemäß Patentanspruch 9 bzw. das System gemäß Patentanspruch 11 bzw. die Diagnosevorrichtung gemäß Patentanspruch 13 gelöst. Weiterbildungen der Erfindung sind in den abhängigen Ansprüchen definiert.

In dem erfindungsgemäßen Verfahren ist den Komponenten des Systems jeweils eine Identifikation zugeordnet und ein jeweiliges ablaufendes Programm ist durch eine Programmidentität spezifiziert. Dabei führt im Ablauf des jeweiligen Programms eine Komponente einen ihr zugeordneten Programmteil aus und die Komponenten rufen sich im Programmablauf ferner zumindest teilweise gegenseitig auf.

In dem erfindungsgemäßen Verfahren wird beim Aufruf einer Komponente durch eine andere Komponente die Programmidentität sowie ein Fehlerparameter von der anderen Komponente an die Komponente übermittelt, wobei der Fehlerparameter angibt, ob im Ablauf des Programms zuvor ein Fehler erkannt wurde. Falls eine Komponente beim Ausführen des ihr zugeordneten Programmteils einen Fehler erkennt, speichert die Komponente einen aktiven Fehlereintrag, der die Programmidentität, die Identifikation der Komponente und einen Fehlerstatus enthält, der angibt, ob der erkannte Fehler der erste Fehler im Ablauf des Programms ist. Ein aktiver Fehlereintrag betrifft somit einen Fehler, der in derjenigen Komponente auftritt, in welcher der Fehlereintrag auch gespeichert wird.

Falls in dem erfindungsgemäßen Verfahren eine Komponente, welche eine andere Komponente aufgerufen hat, von der anderen Komponente die Identifikation der anderen Komponente empfängt, speichert die Komponente einen passiven Fehlereintrag, der die Programmidentität, die Identifikation der Komponente und die Identifikation der anderen Komponente enthält. Durch einen passiven Fehlereintrag wird entsprechend spezifiziert, dass der Fehler gemäß dem Fehlereintrag ein Fehler ist, der in einer anderen Komponente als derjenige aufgetreten ist, in welcher der Fehlereintrag gespeichert ist. Der passive Fehlereintrag gibt dabei an, dass zuvor in der Aufrufkette des Programms ein Fehler auf-

getreten ist, wobei durch die Speicherung der Identifikation der anderen Komponente die Aufrufkette, in welcher der Fehler aufgetreten ist, nachverfolgt werden kann.

Erfindungsgemäß gibt eine Komponente, welche einen oder mehrere aktive oder passive Fehlereinträge im Ablauf des Programms speichert, zumindest einmal im Ablauf des Programms die Programmidentität und die Identifikation der Komponente an diejenige Komponente zurück, durch welche die Komponente aufgerufen wurde. Auf diese Weise wird die Propagation von Fehlereinträgen zu deren Nachverfolgung im Programmablauf gewährleistet.

Das erfindungsgemäße Verfahren zeichnet sich dadurch aus, dass durch entsprechend generierte aktive und passive Fehlereinträge sowie der Übermittlung geeigneter Informationen beim gegenseitigen Aufruf der Komponenten Software-Fehler nachverfolgt werden können und auch erkannt werden kann, ob der entsprechende Software-Fehler der erste Fehler im Programmablauf oder ein potentieller Folgefehler ist, der möglicherweise aus dem ersten Fehler resultiert.

In einer bevorzugten Variante der Erfindung sind die einzelnen Komponenten Steuergeräte und/oder Teile von Steuergeräten, welche über entsprechende Schnittstellen miteinander kommunizieren, wobei die aktiven und passiven Fehlereinträge in vorgegebenen Speicherbereichen der Steuergeräte gespeichert werden. Ein bevorzugter Anwendungsfall ist dabei der Ablauf eines Software-Programms in vernetzten Steuergeräten in einem Kraftfahrzeug oder in mehreren miteinander kommunizierenden Kraftfahrzeugen.

In einer weiteren bevorzugten Variante der Erfindung umfasst die Programmidentität eine Identifikation einer auslösenden Komponente, an der das Programm gestartet wird, sowie eine entsprechende Startzeit des Programms.

In einer weiteren Ausführungsform des erfindungsgemäßen Verfahrens werden im Falle, dass nach der Ausführung eines Programmteils durch eine Komponente keine Fehler in den anschließend ausgeführten Programmteilen auftreten, an die Komponente von derjenigen Komponente, welche durch die Komponente aufgerufen wurde, die Programmidentität und ein Parameter zurückgegeben, wobei der Parameter die Fehlerfreiheit der anschließend ausgeführten Programmteile anzeigt und wobei bei Empfang des Parameters

in der Komponente kein passiver Fehlereintrag in der Komponente gespeichert wird. Auf diese Weise erfolgt eine geeignete Rückmeldung in der jeweiligen Komponente, dass nach der Ausführung des ihr zugeordneten Programnteils keine Fehler aufgetreten sind und somit auch kein passiver Fehlereintrag in der Komponente gespeichert werden muss. Der Parameter ersetzt somit die bei Auftreten eines Fehlers zurückgegebene Identifikation der aufgerufenen Komponente.

In einer weiteren Variante des erfindungsgemäßen Verfahrens wird zumindest teilweise im Ablauf des Programms bei jedem Aufruf einer Komponente ein Aufruf-Eintrag in der Komponente gespeichert. Auf diese Weise können auch fehlerfrei ablaufende Aufrufketten in einem Programmablauf erfasst werden. Vorzugsweise erhält ein Aufruf-Eintrag in der Komponente dabei die Identifikation derjenigen Komponente, welche die Komponente aufgerufen hat. Auf diese Weise können auch Aufrufketten von korrekt ablaufenden Programnteilen rekonstruiert werden. Es besteht dabei die Möglichkeit, dass ab dem Auftreten des ersten Fehlers im Ablauf des Programms Aufruf-Einträge in den jeweiligen Komponenten gespeichert werden. Gegebenenfalls können auch während des gesamten Ablaufs des Programms Aufruf-Einträge hinterlegt werden.

In einer weiteren, besonders bevorzugten Ausführungsform des erfindungsgemäßen Verfahrens wird ein jeweiliger Fehlereintrag (d.h. ein aktiver oder ein passiver Fehlereintrag) in einer Komponente durch ein Feld umfassend eine Mehrzahl von Werten beschrieben, wobei in dem Feld die Programmidentität spezifiziert ist und das Feld ferner einen ersten Wert enthält, der die Identifikation der Komponente spezifiziert, welche den Fehlereintrag speichert, sowie einen zweiten Wert, der die Komponente spezifiziert, in welcher der Fehler aufgetreten ist (entspricht bei einem aktiven Fehlereintrag der Komponente, welche den Fehlereintrag speichert) und einen dritten Wert, der angibt, ob der Fehlereintrag ein passiver Fehlereintrag oder ein aktiver Fehlereintrag ist, wobei im Falle eines aktiven Fehlereintrags der dritte Wert ferner spezifiziert, ob der Fehler der erste Fehler im Ablauf des Programms ist. Die Ausdrücke „erster Wert“, „zweiter Wert“ und „dritter Wert“ spezifizieren dabei lediglich entsprechende Arten von Werten und geben nicht zwangsläufig die Stelle an, an der sich der entsprechende Wert im Feld befindet.

In der Variante des erfindungsgemäßen Verfahrens, in der die Programmidentität durch die Identifikation der auslösenden Komponente und die Startzeit festgelegt ist, enthält das

Feld zur Beschreibung des Fehlereintrags zwei Werte zur Spezifikation der Programmidentität, wobei einer der Werte die Identifikation der auslösenden Komponente und der andere der Werte die Startzeit darstellt.

Neben dem soeben beschriebenen Verfahren zur Erfassung von Fehlern betrifft die Erfindung ferner ein Verfahren zum Verarbeiten von Fehlern, welche mit diesem Verfahren erfasst wurden. Dabei werden die gespeicherten aktiven und passiven Fehlereinträge ausgelesen und für jede Programmidentität wird aus den dazugehörigen aktiven und passiven Fehlereinträgen ein Fehlerbaum generiert und ausgegeben, wobei in dem Fehlerbaum zumindest die Aufrufketten von aufeinander folgend aufgerufenen Komponenten enthalten sind, welche wenigstens eine Komponente mit einem aktiven Fehlereintrag enthalten. Dabei sind in dem Fehlerbaum die Komponenten dahingehend gekennzeichnet, ob in der jeweiligen Komponente im Ablauf des Programms kein Fehler oder der erste Fehler oder ein potentieller Folgefehler aufgetreten ist. Aufgrund dieser Information kann in geeigneter Weise nachverfolgt werden, wie die einzelnen Fehler im Fehlerbaum zusammenhängen. Die späteren, nach dem ersten Fehler aufgetretenen Fehler stellen dabei potentielle Folgefehler dar und es wird durch den Fehlerbaum insbesondere vermittelt, aus welchem ursprünglichen Fehler bzw. aus welchen anderen potentiellen Folgefehlern ein entsprechender Folgefehler entstanden sein könnte. Sollten bei der Erfassung von Fehlern ferner auch die oben beschriebenen Aufruf-Einträge enthalten sein, kann der Fehlerbaum gegebenenfalls auch diejenigen Aufrufketten enthalten, in denen alle Komponenten ihren Programmteil fehlerfrei ausgeführt haben.

Eine weitere Variante des Verfahrens zur Rekonstruktion des Fehlerbaums beruht auf einer Erfassung von Fehlern über Fehlereinträge, welche durch die oben beschriebenen Felder spezifiziert werden. Dabei wird der Fehlerbaum derart generiert, dass ausgehend von jeder Komponente mit einem aktiven Fehlereintrag schrittweise über den ersten und zweiten Wert der Felder der Fehlereinträge jeweilige Aufrufketten von aufeinander folgend aufgerufenen Komponenten bestimmt werden.

Neben den oben beschriebenen Verfahren betrifft die Erfindung ferner ein System aus Komponenten, insbesondere in einem Kraftfahrzeug, wobei das System derart ausgestaltet ist, dass Fehler beim Ablauf von einem oder mehreren software-basierten Programmen mit dem oben beschriebenen Verfahren zur Erfassung von Fehlern ermittelt werden.

Darüber hinaus betrifft die Erfindung ein Kraftfahrzeug, welches ein solches System umfasst.

Ferner betrifft die Erfindung eine Diagnosevorrichtung zum Verarbeiten von mit dem oben beschriebenen Verfahren erfassten Fehlern, wobei die Diagnosevorrichtung derart ausgestaltet ist, dass mit der Vorrichtung das oben beschriebene Verfahren zum Verarbeiten entsprechend erfasster Fehler durchführbar ist.

Ausführungsbeispiele der Erfindung werden nachfolgend anhand der beigefügten Figuren detailliert beschrieben.

Es zeigen:

- Fig. 1 eine schematische Darstellung des Austauschs von Informationen in einem Aufruf einer zweiten Komponente durch eine erste Komponente gemäß einer Ausführungsform des erfindungsgemäßen Verfahrens;
- Fig. 2 eine schematische Darstellung eines Systems aus Komponenten und deren Interaktion, auf dessen Basis eine Ausführungsform des erfindungsgemäßen Verfahrens beschrieben wird;
- Fig. 3 bis Fig. 12 schematische Darstellungen, welche beispielhaft einen Ablauf eines Programms in dem System der Fig. 2 und die dabei verwendete Fehlererfassung basierend auf einer Ausführungsform des erfindungsgemäßen Verfahrens zeigen; und
- Fig. 13 eine schematische Darstellung, welche das Auslesen der in dem Programmablauf gemäß Fig. 3 bis Fig. 12 erfassten Fehler und der darauf basierenden Generierung eines Fehlerbaums gemäß einer Ausführungsform des erfindungsgemäßen Verfahrens verdeutlicht.

Nachfolgend werden Ausführungsformen des erfindungsgemäßen Verfahrens basierend auf einem verteilten System aus einer Mehrzahl von Steuergeräten in einem Kraftfahr-

zeug beschrieben, wobei die von den einzelnen Steuergeräten auszuführenden Funktionen mittels Software realisiert sind. Dabei wird ein Ablauf eines Software-Programms betrachtet, bei dem sich die einzelnen Steuergeräte gegenseitig aufrufen, wobei gemäß dem Programmablauf nach Beendigung entsprechender Programmfunktionen in einem Steuergerät ein anderes Steuergerät aufgerufen wird. Die Steuergeräte stellen Komponenten im Sinne von Anspruch 1 dar und werden deshalb im Folgenden auch als Komponenten bezeichnet.

Die einzelnen Steuergeräte weisen zur Kommunikation definierte Schnittstellen auf und kommunizieren untereinander über das Bussystem des Kraftfahrzeugs. Es kann sich bei den Steuergeräten um beliebige Einheiten zur Ausführung von Funktionalitäten im Fahrzeug handeln, beispielsweise um eine elektronische Motorsteuerung, um die Steuerung der Airbags, die Steuerung elektrischer Stellmotoren, z.B. zur Sitzverstellung, die Steuerung der Klimaanlage im Fahrzeug, die Steuerung der Anzeigeeinheiten im Fahrzeug und dergleichen. Ein Programmablauf in diesem System an Komponenten ist beispielsweise die Ansteuerung der einzelnen Stellmotoren einer automatischen Sitzverstellung, welche den Sitz basierend auf einer für eine Person im Fahrzeug hinterlegten vorprogrammierten Sitzeinstellung verstellt.

Bei einem Ablauf eines Software-Programms in dem oben beschriebenen verteilten System aus Komponenten besteht das Problem, dass es keine geeigneten Mechanismen gibt, welche zur Laufzeit des Programms auftretende Software-Fehler derart erfassen, dass eine Fehlerverfolgung zur Ermittlung der Ursache bzw. des Auslösers der einzelnen Fehler möglich ist. Gemäß den nachfolgend beschriebenen Ausführungsformen der Erfindung wird eine Erfassung von Software-Fehlern und deren Hinterlegung in dem Fehlerpeicher der einzelnen Komponenten derart geschaffen, dass beim späteren Auslesen der Fehler ein geeigneter Fehlerbaum generiert werden kann. Die Erfindung umfasst somit zum einen ein Verfahren, bei dem Software-Fehler onboard im Fahrzeug in geeigneter Weise erfasst werden, und zum anderen ein Verfahren, um die erfassten Software-Fehler offboard zu einem Fehlerbaum zusammenzusetzen.

Fig. 1 zeigt allgemein die Kommunikation zwischen einem aufrufenden Steuergerät CUA (CU = Control Unit) und einem aufgerufenen Steuergerät CUB. Im Folgenden wird ein aufrufendes Steuergerät auch als erstes Steuergerät bzw. erste Komponente und ein auf-

gerufenes Steuergerät auch als zweites Steuergerät bzw. zweite Komponente bezeichnet. Es wird dabei im Folgenden davon ausgegangen, dass jede Komponente durch eine eindeutige Identifikation, z.B. eine UUID (UUID = Universal Unique Identifier) oder eine Software-ID, identifizierbar ist und im Programmablauf Aufrufe von einer Komponente durch eine andere auftreten. Darüber hinaus existieren geeignete Mechanismen, um durch die Komponente beim Programmablauf erkannte Software-Fehler in Fehlerspeichern der Komponenten zu hinterlegen. Ferner wird in der hier beschriebenen Ausführungsform der synchrone Aufruf von Komponenten betrachtet, wobei die Erfindung jedoch analog auch auf nicht-synchrone Aufrufe angewendet werden kann.

Gemäß Fig. 1 werden beim Aufruf der zweiten Komponente CUB durch die erste Komponente CUA als zusätzliche Parameter das Parametertripel (a, b, f) übertragen. Der Parameter a bezeichnet dabei die Identifikation derjenigen Komponente, welche ursprünglich das laufende Programm ausgelöst hat. Sofern das Programm an der Komponente CUA beginnt, stimmt der Parameter a mit der Identifikation der Komponente CUA überein. Der Parameter b spezifiziert die Zeit, zu der das Programm bei der auslösenden Komponente gestartet wurde, d.h. zu der die entsprechende Kundenfunktion im Fahrzeug basierend auf dem Programm initiiert wurde. Die Kombination der Parameter a und b charakterisieren somit einen aktuellen Ablauf eines Programms und diese Kombination entspricht einer Programmidentität im Sinne von Anspruch 1. Der Parameter f stellt einen Fehlerparameter dar und gibt an, ob zuvor im Ablauf des Programms bereits ein Fehler in einer Komponente aufgetreten ist. Ist zuvor kein Fehler erkannt worden, wird der Fehlerparameter auf 0 gesetzt, ist ein Fehler aufgetreten, ist der Parameter im nachfolgenden Ablauf des Programms auf 1 gesetzt.

Nachdem die aufgerufene Komponente CUB die entsprechenden Programmfunktionen abgearbeitet hat, wird an die erste Komponente CUA ein Return in der Form eines Parameters r sowie die bereits oben erwähnten Parameter a und b zurückgegeben. Der Return-Wert r ist dabei immer auf 0 gesetzt, wenn in der Komponente CUB kein Fehler erkannt wurde und der Komponente auch nicht von anderen, später aufgerufenen Komponenten das Auftreten eines Fehlers gemeldet wird. In allen anderen Fällen enthält der Parameter r als Wert die Identifikation der Komponente CUB.

Beim Ablauf des Programms werden entsprechende Fehlereinträge in den Fehlerspeichern der einzelnen Komponenten erzeugt, wobei sowohl aktive Fehlereinträge als auch passive Fehlereinträge in den Fehlerspeichern hinterlegt werden. Ein aktiver Fehlereintrag wird dabei dann erzeugt, wenn in der entsprechenden Komponente selbst ein Fehler aufgetreten ist, und ein passiver Fehlereintrag wird dann erzeugt, wenn eine Komponente von einer anderen Komponente als Return-Wert *r* die Identifikation der anderen Komponente empfängt. Als Return-Wert *r* wird somit immer dann die Identifikation der aufgerufenen Komponente übertragen, wenn in dieser Komponente selbst ein Fehler aufgetreten ist oder wenn die aufgerufene Komponente von derjenigen Komponente, die sie selbst aufruft, einen Return-Wert ungleich 0 empfängt.

In den weiter unten beschriebenen Fig. 5 bis 12 sind entsprechende Fehlereinträge der Komponenten als Vektoren mit fünf Werten in eckigen Klammern angegeben, wobei allgemein der erste Wert des Vektors mit *a*, der zweite Wert mit *b*, der dritte Wert mit *i*, der vierte Wert mit *p* und der fünfte Wert mit *e* bezeichnet werden, wie in Fig. 5 entsprechend angedeutet ist. Allgemein kann somit ein Fehlereintrag als Vektor [*a*, *b*, *i*, *p*, *e*] geschrieben werden. Dabei entspricht *a* wiederum der den Programmablauf ursprünglich auslösenden Komponente und *b* der Startzeit, an der das Programm an der auslösenden Komponente gestartet wird. Demgegenüber bezeichnet *i* die Identität der Komponente, die den entsprechenden Fehlereintrag gerade speichert. Durch den Parameter *p* wird die Fehlerherkunft festgelegt. Erkennt eine Komponente selbst einen Software-Fehler, so wird hier die eigene Identität der Komponente eingetragen. Wurde im Return-Wert *r*, den die betrachtete Komponente von der durch sie aufgerufenen Komponente empfängt, die Identität der aufgerufenen Komponente übermittelt, so wird diese Identität für den Parameter *p* verwendet. Über den Parameter *p* wird somit vermittelt, ob ein aktiver Fehlereintrag eines lokal in der betrachteten Komponente aufgetretenen Fehlers vorliegt oder nicht.

Durch den Parameter *e* wird der Fehlertyp angegeben. Wird in einer Aufrufkette von aufeinander folgenden Komponenten ein Software-Fehler zum ersten Mal entdeckt („primärer Fehler“), so wird *e* in der entsprechenden Komponente auf 1 gesetzt. Anschließend wird beim Aufruf der nächsten Komponente der Parameter *f* auf 1 gesetzt und bleibt auch bei allen nachfolgenden Aufrufen von Komponenten auf 1. Wurde eine Komponente bereits mit *f* = 1 aufgerufen, so wird beim Erkennen eines Fehlers in der aufgerufenen Komponente *e* auf 2 gesetzt. Dadurch wird angezeigt, dass ein aufgetretener Fehler ein poten-

tieller Folgefehler ist. Wurde von einer aufgerufenen Komponente ein Fehler an die aufrufende Komponente gemeldet (d.h. wird als Parameter r die Identifikation der aufgerufenen Komponente zurückgegeben), so ist ein passiver Fehlerspeichereintrag vorzunehmen, bei dem e auf 0 gesetzt wird. Dieser Fehlereintrag zeigt somit an, dass ein Fehler von einer anderen Komponente detektiert wurde. Beim Setzen eines passiven Fehlereintrags muss die entsprechende Komponente ferner dann, wenn sie zuvor noch keinen Return-Wert $r \neq 0$ übertragen hat, ihre eigene Identifikation als Return-Wert r an die vorhergehende Komponente zurückgeben. Die Generierung von passiven Fehlereinträgen ist dazu notwendig, dass später die Aufrufreihenfolge der Komponenten bei der Generierung des Fehlerbaums hergestellt werden kann.

Fig. 2 zeigt am Beispiel eines Systems aus zwölf verteilten Steuergeräten CU1, CU2, ..., CU12 den Programmablauf basierend auf einer funktionalen Aufrufkette, bei der die Aufrufe bei der auslösenden Komponente CU1 synchron erfolgen. Die Aufrufe innerhalb des Programmablaufs sind dabei durch entsprechende Pfeile angedeutet, von denen aus Übersichtlichkeitsgründen lediglich einige mit Bezugszeichen P bezeichnet sind. Man erkennt, dass der Programmablauf an der Komponente CU1 als auslösende Komponente startet. Diese Komponente ruft die vier Komponenten CU2, CU3, CU4 und CU5 auf. Anschließend ruft die Komponente CU3 die Komponenten CU7 bis CU8 auf, woraufhin dann die Komponente CU8 die Komponente CU11 aufruft. Ferner ruft die Komponente CU5 die Komponenten CU9 und CU10 auf, woraufhin die Komponente CU10 die Komponente CU12 aufruft. In dem Programmablauf der Fig. 2 treten bei der Abarbeitung der Programmteile in den Komponenten CU3, CU7, CU10, CU11 und CU12 Fehler auf, welche durch entsprechende gezackte Pfeile Z angedeutet sind.

Anhand des Systems der Komponenten der Fig. 2 und des dargestellten Programmablaufs wird nunmehr eine Ausführungsform zur Erfassung von Software-Fehlern basierend auf entsprechenden Fehlereinträgen sowie dem oben beschriebenen Austausch von Parametern (Fig. 1) beschrieben. Im Folgenden bezeichnen die Zahlen 01, 02, ..., 12 jeweils die Identifikation der entsprechenden Komponente CU1, CU2, ..., CU12. Demgegenüber bezeichnet der Buchstabe t den Zeitpunkt des Starts des Programms, d.h. den Auslösezeitpunkt des Programms durch die Komponente CU1.

Gemäß Fig. 3 ruft die Komponente CU1 die Komponente CU2 mit Parametern $a = 01$, $b = t$ und $f = 0$ auf. Gemäß Fig. 4 schickt die Komponente CU2 in Antwort auf den Aufruf den Return-Wert $r = 0$ sowie die Identifikation 01 der auslösenden Komponente und die Auslösezeit t zurück, denn in der Komponente CU2 ist kein Fehler aufgetreten und die Komponente ruft auch keine nachfolgenden Komponenten mehr auf. Gemäß Fig. 5 ruft die auslösende Komponente CU1 die Komponente CU3 auf. In der Komponente CU3 ist lokal ein Fehler aufgetreten. Demzufolge wird der Fehlereintrag $[01, t, 03, 03, 1]$ erzeugt. Der Parameter p ist auf 03 gesetzt, da der Fehler lokal in der Komponente CU3 aufgetreten ist. Der Parameter e ist auf 1 gesetzt, da dies der erste, im Programmablauf auftretende Fehler ist. Beim Aufruf der Komponente CU3 durch die Komponente CU1 wurde die Identität 01 der auslösenden Komponente CU1 sowie die Auslösezeit t übermittelt, die dann als die ersten beiden Werte im Fehlereintrag übernommen werden, wie durch den Pfeil P' in Fig. 5 angedeutet ist. Da nunmehr ein Fehler in der Komponente CU5 aufgetreten ist, wird an die Komponente CU7 neben der Identifikation 01 der auslösenden Komponente CU1 und der Auslösezeit t nunmehr der Parameter $f = 1$ übermittelt. Anschließend führt die aufgerufene Komponente CU7 ihre Programmfunktionen aus, wobei lokal wiederum ein Fehler auftritt. Als Folge wird der Fehlereintrag $[01, t, 07, 07, 2]$ erzeugt, wobei durch den Wert $p = 07$ angezeigt wird, dass es sich um einen lokalen Fehler handelt. Demgegenüber wird durch $e = 2$ angezeigt, dass der aufgetretene Fehler nicht der erste Fehler im Programmablauf ist, was bedeutet, dass es sich um einen potentiellen Folgefehler handeln kann, der aus dem Fehler einer anderen Komponente resultieren kann.

Wie in Fig. 6 angedeutet ist, meldet die Komponente CU7 nach der Ausführung ihrer Programmfunktionen neben der Identifikation 01 der auslösenden Komponente 01 und der Auslösezeit t den Wert $r = 07$ zurück, denn in der Komponente CU7 ist ein Fehler aufgetreten. Daraufhin wird der Fehlereintrag $[01, t, 03, 07, 0]$ in der Komponente CU3 generiert, wobei durch $p = 07$ und $e = 0$ angezeigt wird, dass es sich um einen passiven Fehlereintrag handelt, der auf einen von der Komponente CU7 zurückgehenden Fehler stammt. Schließlich wird die Komponente CU8 durch die Komponente CU3 aufgerufen. Dabei wird neben der Identifikation 01 der auslösenden Komponente und der Auslösezeit t der Parameter $f = 1$ übermittelt. In der Komponente CU8 tritt bei der Programmausführung kein Fehler auf. Schließlich übermittelt die Komponente CU8 die Identifikation 01 der auslösenden Komponente, die Auslösezeit t sowie den Parameter $f = 1$ im Rahmen des Aufrufs der Komponente CU11 an diese Komponente. In der Komponente CU11 tritt wie-

derum ein lokaler Fehler auf. Als Konsequenz wird in der Komponente CU11 der aktive Fehlereintrag [01, t, 11, 11, 2] erzeugt. Dabei zeigt $e = 2$ an, dass es sich in der Komponente CU12 um einen potentiellen Folgefehler handelt.

Wie in Fig. 7 angedeutet ist, wird als Folge des in CU12 aufgetretenen Fehlers die Identifikation 11 der Komponente CU11 neben der Identifikation 01 der auslösenden Komponente und der Auslösezeit t an die Komponente CU8 zurückgegeben. Diese erzeugt wiederum einen passiven Fehlereintrag in der Form [01, t, 08, 11, 0]. Als Folge wird neben der Identifikation 01 der auslösenden Komponente und der Auslösezeit t die Identität 08 der Komponente CU8 an die Komponente CU3 zurückgegeben. Analog erfolgt dann nach der Generierung des passiven Fehlereintrags [01, t, 03, 08, 0] in der Komponente CU3 die Rückgabe der Identifikation 03 der Komponente CU3 zusammen mit der Identifikation 01 der auslösenden Komponente und der Auslösezeit t an die Komponente CU1. Als Folge wird der passive Fehlereintrag [01, t, 01, 03, 0] in CU1 generiert.

Gemäß Fig. 8 ruft die auslösende Komponente CU1 die Komponente CU4 auf und überträgt dabei ihre Identifikation 01, die Auslösezeit t und den Parameter $f = 1$. Da in der Komponente CU4 kein Fehler auftritt und sich das Programm von dort aus nicht weiter fortsetzt, wird gemäß Fig. 9 von der Komponente CU4 neben der Identifikation 01 der auslösenden Komponente und der Auslösezeit t der Parameter $r = 0$ an CU1 zurückgegeben. Als Konsequenz wird in der Komponente CU1 kein weiterer Fehlereintrag generiert.

Gemäß Fig. 10 ruft die Komponente CU1 die Komponente CU5 auf, welche fehlerfrei ihre Programmfunktionalitäten ausführt und anschließend die Komponente CU9 aufruft, welche ebenfalls fehlerfrei ihre Programmfunktionen ausführt. Dabei werden wiederum die Identifikation 01 der auslösenden Komponente, die Auslösezeit t und $f = 1$ übertragen. Da in dieser Programmkette kein Fehler aufgetreten ist, werden auch keine aktiven Fehlereinträge in den Komponenten CU1, CU5 und CU9 erzeugt.

In Fig. 11 ist die Rückgabe des entsprechenden Parameterwerts $r = 0$ von der Komponente CU9 an die Komponente CU5 wiedergegeben. Ferner zeigt Fig. 11 den Aufruf der Komponente CU10 durch die Komponente CU5, wobei wiederum die Identifikation 01 der auslösenden Komponente und die Auslösezeit t sowie der Parameter $f = 1$ übertragen werden. Da in der Komponente CU10 ein lokaler Fehler auftritt, wird dort der aktive Feh-

lereintrag [01, t, 10, 10, 2] hinterlegt. Die Komponente CU10 ruft dann die Komponente CU12 auf und überträgt hierbei wiederum die Identifikation 01 der auslösenden Komponente, die Auslösezeit t sowie den Parameter $f = 1$. In der Komponente CU12 tritt dann bei der Programmausführung ein weiterer lokaler Fehler auf, so dass dort der aktive Fehlereintrag [01, t, 12, 12, 2] gespeichert wird.

Wie sich aus Fig. 12 ergibt, wird nach der Ausführung der Programmfunktionen der Komponente CU12 aufgrund des aufgetretenen Fehlers in der Komponente CU12 neben der Identifikation 01 der auslösenden Komponente und der Auslösezeit t die Identifikation 12 der Komponente CU12 an die Komponente CU10 zurückgegeben. Da der Parameter r ungleich 0 ist, wird nunmehr ein passiver Fehlereintrag in der Form [01, t, 10, 12, 0] in der Komponente CU10 generiert. Aufgrund der Fehlereinträge in CU10 wird der Parameter r = 10 neben der Identifikation 01 der auslösenden Komponente und der Auslösezeit t an die Komponente CU5 übertragen. Da der Parameter r wieder ungleich 0 ist, wird auch in der Komponente CU5 ein entsprechender passiver Fehlereintrag [01, t, 05, 10, 0] gespeichert. Aufgrund dieses passiven Fehlereintrags wird nunmehr der Parameter r = 05 neben der Identifikation 01 der auslösenden Komponente und der Auslösezeit t an die Komponente CU1 zurückgegeben. Auch in der Komponente CU1 wird dann wegen $r \neq 0$ ein passiver Fehlereintrag in der Form [01, t, 01, 05, 0] generiert.

In Fig. 12 ist die Programmausführung dann beendet. Fig. 12 zeigt somit alle im gerade beschriebenen Programmablauf in den einzelnen Komponenten gespeicherten Fehlereinträge. Diese Fehlereinträge werden anschließend mit einer entsprechenden Diagnosevorrichtung aus den Fehlerspeichern der Komponenten ausgelesen. Anschließend wird ein entsprechender Fehlerbaum generiert, wie nachfolgend anhand von Fig. 13 beschrieben wird.

Nach Auslesen der Fehlereinträge ergibt sich zunächst folgende Liste für die Fehlereinträge der einzelnen Komponenten:

CU1: [01, t, 01, 03, 0] [01, t, 01, 05, 0]

CU2: -

CU3: [01, t, 03, 03, 1] [01, t, 03, 07, 0] [01, t, 03, 08, 0]

CU4: -

CU5: [01, t, 05, 10, 0]

CU6: -

CU7: [01, t, 07, 07, 2]

CU8: [01, t, 08, 11, 0]

CU9: -

CU10: [01, t, 10, 10, 2] [01, t, 10, 12, 0]

CU11: [01, t, 11, 11, 2]

CU12: [01, t, 12, 12, 2]

In der obigen Listen sind dabei alle Fehlereinträge mit $e = 2$, welche potentielle Folgefehler darstellen, kursiv wiedergegeben. Ferner ist der Fehlereintrag, der den ersten Fehler im Programmablauf spezifiziert, d.h. für den $e = 1$ gilt, fett gedruckt. Dieser Fehlereintrag ist in der Komponente CU3 hinterlegt. Zur Generierung des Fehlerbaums werden zunächst als Einstiegspunkte entsprechender Aufrufketten diejenigen Komponenten bestimmt, welche einen lokalen Fehlereintrag enthalten, d.h. welche bei sich selbst einen Fehler beim Programmablauf erkannt haben. Dies sind gemäß der obigen Liste alle Komponenten, welche zumindest einen Fehlereintrag enthalten, dessen Parameter e ungleich 0 ist. Dies sind gemäß der obigen Liste der Komponenten CU3, CU7, CU10, CU11 und CU12.

Ausgehend von den Identifikationen der Komponenten mit zumindest einem Fehlereintrag mit $e \neq 0$ wird dann eine entsprechende Aufrufkette derart zusammengesetzt, dass für eine jeweilige Komponente nach denjenigen Komponenten gesucht wird, welche als *vierten Wert im Fehlereintrag die Identifikation der jeweiligen Komponente enthält*. Das heißt, es wird ausgehend von dem dritten Wert i der Fehlereinträge in der jeweiligen Komponente nach einer Komponente gesucht, welche einen Fehlereintrag enthält, dessen vierter Wert dem dritten Wert der jeweiligen Komponente entspricht. Für die Komponente CU12 ergibt sich dann, dass die Komponente CU10 einen solchen Fehlereintrag, nämlich [01, t, 10, 12, 0] enthält. Es wird dann mit der aufgefundenen Komponente analog verfahren, d.h. es wird wieder eine Komponente gesucht, die ausgehend von den Fehlereinträgen der gerade aufgefundenen Komponente einen Fehlereintrag enthält, der einen vierten Wert aufweist, der dem dritten Wert der Fehlereinträge der gerade aufgefundenen Komponente entspricht. Dies wird solange fortgesetzt, bis man schließlich an der auslösenden Komponente CU1 ankommt. Gemäß dem obigen Beispiel wird somit ausgehend von der

gefundenen Komponente CU10 dann nach Komponenten gesucht, welche Einträge enthalten, die als vierten Wert die Identifikation 10 aufweisen. Dies ist gemäß der obigen Liste der Fall für die Komponente CU5. Ausgehend von dieser Komponente wird nach einer Komponente gesucht, welche einen Eintrag enthält, der als vierten Wert die Identifikation 05 aufweist. Dies ist die auslösende Komponente CU1. Auf diese Weise wird die Kette eines lokalen Fehlers geeignet zurückverfolgt.

Gemäß Fig. 13 ergeben sich somit für die einzelnen Identifikationen 03, 07, 10, 11 und 12 der jeweiligen Komponenten CU3, CU7, CU10, CU11 und CU12 die Aufrufketten A1 bis A5. Diese Aufrufketten werden dann in der Reihenfolge umgekehrt und zu einem Fehlerbaum FB zusammengefasst, wie durch den Pfeil P" angedeutet ist. Über die Reihenfolge der in einer Komponente abgelegten Fehlereinträge ergibt sich die Reihenfolge der Aufrufe. Zum Beispiel wird in Komponenten CU1 zuerst ein Fehlereintrag für CU3, danach ein Fehlereintrag für CU5 abgelegt. Daraus ergibt sich, dass CU1 zuerst die Komponente CU3 und danach CU5 aufgerufen hat. In diesem Fehlerbaum werden ausgehend von der Identifikation 01 in der obersten Hierarchieebene Verbindungen zu den gemäß den umgekehrten Aufrufketten in den darunter liegenden Hierarchieebenen vorhandenen Identifikationen gebildet. In dem Fehlerbaum der Fig. 13 schließen sich somit an die Identifikation 01 in umgekehrter Richtung der Aufrufketten A1 bis A5 die Identifikationen 03 und 05 an. Die zweite Hierarchieebene des Fehlerbaums FB enthält folglich die Identifikationen 03 und 05. In gleicher Weise wird dann nach Identifikationen in der nächsten Hierarchieebene gesucht. Gemäß den Aufrufketten der Fig. 13 ergibt sich dabei, dass die Identifikation 03 sowohl mit der Identifikation 07 als auch mit der Identifikation 08 verbunden ist. Demgegenüber ist die Identifikation 05 nur mit der Identifikation 10 verbunden. Das heißt, der Fehlerbaum FB verzweigt sich von der Identifikation 03 zu den Identifikationen 07 und 08. Demgegenüber verzweigt sich der Fehlerbaum von der Identifikation 05 nur zu der Identifikation 10. Für die darauf folgende Hierarchieebene ergibt sich, dass die Identifikation 08 mit der Identifikation 10 und die Identifikation 10 mit der Identifikation 12 verbunden ist. Die Identifikation 07 steht demgegenüber bereits am Anfang der entsprechenden Aufrufkette A2. Somit verzweigt sich der Fehlerbaum ausgehend von der Identifikation 08 nur zur Identifikation 11 und von der Identifikation 10 nur zu der Identifikation 12.

Da für die einzelnen Komponenten bekannt ist, ob in diesen Komponenten kein Fehler ($e = 0$) bzw. ein Fehler zum ersten Mal ($e = 1$) bzw. ein potentieller Folgefehler ($e = 2$) auf-

getreten ist, wird diese Information in dem Fehlerbaum FB geeignet codiert. Dies ist in dem Fehlerbaum der Fig. 13 dadurch angedeutet, dass die dick gedruckte Identifikation 03 die Komponente spezifiziert, an welcher ein Fehler im Programmablauf das erste Mal aufgetreten ist, die kursiv gedruckten Identifikationen 07, 10, 11 und 12 Komponenten spezifizieren, bei denen potentielle Folgefehler aufgetreten sind, und die normal gedruckten Identifikationen 01, 05 und 08 Komponenten spezifizieren, in denen kein Fehler aufgetreten ist.

Wie sich aus der Beschreibung des obigen Ausführungsbeispiels ergibt, kann mit der dargelegten Hinterlegung entsprechender Fehlereinträge beim Programmablauf in geeigneter Weise ein Fehlerbaum generiert werden, aus dem abgelesen werden kann, aus welchen vorhergehenden Fehlern ein potentieller Folgefehler generiert sein könnte. Man erkennt dabei in Fig. 13 insbesondere, dass die Fehler in der Komponente CU7 und CU11 ihre Ursache möglicherweise in dem Fehler in der Komponente CU3 haben und dass die Fehler in den Komponenten 10 und 12 nicht mit den Fehlern in den Komponenten CU3, CU7 und CU11 im Zusammenhang stehen.

Die oben beschriebene Ausführungsform des erfindungsgemäßen Verfahrens kann in geeigneter Weise abgewandelt werden. Insbesondere können alle Aufrufe von Komponenten erfasst werden. In diesem Fall wird bei jedem Aufruf einer Komponente ein entsprechender Eintrag in der Komponente vorgenommen, der in diesem Sinne keinen Fehlereintrag darstellt, sondern lediglich spezifiziert, dass die Komponente aufgerufen wurde. Gemäß dem im Vorangegangenen beschriebenen Format der Fehlereinträge kann ein solcher Eintrag beispielsweise dadurch spezifiziert werden, dass er als dritten und vierten Wert die Identifikation der aufgerufenen Komponente und als fünften Wert den Wert $e = 0$ enthält. Der erste bzw. zweite Wert entspricht dabei wiederum der Identifikation 01 der auslösenden Komponente bzw. der Auslösezeit t . Um dabei eine komplette Rekonstruktion auch von fehlerfrei abgelaufenen Aufrufketten zu gewährleisten, muss ferner bei jedem Aufruf die Identifikation der aufrufenden Komponente in der aufgerufenen Komponente hinterlegt werden.

Gemäß der obigen Abwandlung besteht zum einen die Möglichkeit, dass alle Aufrufe immer ab dem Start des Programms aufgezeichnet werden. Jedoch kann in einer Alternative auch vorgesehen sein, dass die Aufzeichnung aller Aufrufkette erst ab dem ersten Fehler-

fall gestartet wird, d.h. wenn der Parameter f den Wert 1 annimmt. In diesem Fall stehen keine Informationen über die Aufrufketten vor dem Eintreten des ersten Fehlers zur Verfügung, jedoch wird weniger Speicherplatz benötigt, als wenn die Aufrufe ab dem Programmstart erfasst werden.

Das im Vorangegangenen beschriebene Verfahren zum Erfassen von Fehlern und Erzeugung von Fehlerbäumen weist eine Reihe von Vorteilen auf. Insbesondere werden mit dem Verfahren aussagekräftige Fehlerbilder geliefert, bei denen funktional zusammengehörige Software-Fehler zusammengefasst werden können. Neben dem funktionalen Zusammenhang wird dabei unterschieden zwischen dem erstmaligen Auftreten eines Software-Fehlers sowie potentiellen Folgefehlern, die sich daraus ergeben können. Damit ist sichergestellt, dass sporadische Software-Fehler besser nachvollziehbar und reproduzierbar sind. Das Verfahren selbst ist einfach und effizient. Es ist keinerlei Statusinformation in den Komponenten zu halten. Es ist lediglich der Speicherplatz für die Ablage der Fehlerinformation vorzusehen. Das Verfahren arbeitet permanent und somit unabhängig von Fehlervorfällen. Damit ist gewährleistet, dass im Fehlerfall die Buskommunikation im Fahrzeug nicht ansteigt und somit Buslast und Jitter deterministisch bleiben.

Patentansprüche

1. Verfahren zur rechnergestützten Erfassung von Fehlern beim Ablauf von einem oder mehreren software-basierten Programmen in einem System aus Komponenten (CU1, CU2, ..., CU12), insbesondere in einem Kraftfahrzeug, wobei den Komponenten (CU1, CU2, ..., CU12) jeweils eine Identifikation (i) zugeordnet ist und ein jeweiliges ablaufendes Programm durch eine Programmidentität (a, b) spezifiziert ist, wobei im Ablauf des Programms eine Komponente (CU1, CU2, ..., CU12) einen ihr (CU1, CU2, ..., CU12) zugeordneten Programmteil ausführt und sich die Komponenten (CU1, CU2, ..., CU12) zumindest teilweise gegenseitig aufrufen, dadurch gekennzeichnet, dass
 - beim Aufruf einer Komponente (CUB) durch eine andere Komponente (CUA) die Programmidentität (a, b) sowie ein Fehlerparameter (f) von der anderen Komponente (CUA) an die Komponente (CUB) übermittelt wird, wobei der Fehlerparameter (f) angibt, ob im Ablauf des Programms zuvor ein Fehler erkannt wurde;
 - falls eine Komponente (CUA, CUB) beim Ausführen des ihr zugeordneten Programmteils einen Fehler erkennt, die Komponente (CUA, CUB) einen aktiven Fehlereintrag speichert, der die Programmidentität (a, b), die Identifikation (i) der Komponente (CUA, CUB) und einen Fehlerstatus (e) enthält, der angibt, ob der erkannte Fehler der erste Fehler im Ablauf des Programms ist;
 - falls eine Komponente (CUA), welche eine andere Komponente aufgerufen hat, von der anderen Komponente (CUB) die Identifikation (p) der anderen Komponente (CUB) empfängt, die Komponente (CUA) einen passiven Fehlereintrag speichert, der die Programmidentität (a, b), die Identifikation (i) der Komponente (CUA) und die Identifikation (p) der anderen Komponente (CUB) enthält;
 - wobei eine Komponente (CUB), welche einen oder mehrere aktive oder passive Fehlereinträge im Ablauf des Programms speichert, zumindest einmal im Ablauf des Programms die Programmidentität (a, b) und die Identifikation der Komponente (CUB) an diejenige Komponente (CUA) zurückgibt, durch welche die Komponente (CUA) aufgerufen wurde.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Komponenten (CU1, CU2, ..., CU12) Steuergeräte und/oder Teile von Steuergeräten umfassen, welche über entsprechende Schnittstellen miteinander kommunizieren, wobei die aktiven und passiven Fehlereinträge in vorgegebenen Speicherbereichen der Steuergeräte gespeichert werden.
3. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die Programmidentität (a, b) eine Identifikation (a) einer auslösenden Komponente (CU1), an der das Programm gestartet wird, sowie eine Startzeit (b) umfasst.
4. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass im Falle, dass nach der Ausführung eines Programmtails durch eine Komponente (CUA) keine Fehler in den anschließend ausgeführten Programmteilen auftreten, an die Komponente (CUA) von derjenigen Komponente (CUB), welche durch die Komponente (CUA) aufgerufen wurde, die Programmidentität (a, b) und ein Parameter (r) zurückgegeben werden, wobei der Parameter (r) die Fehlerfreiheit der anschließend ausgeführten Programmteile anzeigt und wobei bei Empfang des Parameters (r) in der Komponente (CUA) kein passiver Fehlereintrag in der Komponente (CUA) gespeichert wird.
5. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass zumindest zeitweise im Ablauf des Programms bei jedem Aufruf einer Komponente (CUA, CUB) ein Aufruf-Eintrag in der Komponente (CUA, CUB) gespeichert wird.
6. Verfahren nach Anspruch 5, dadurch gekennzeichnet, dass während des gesamten Ablaufs des Programms oder erst ab dem Auftreten des ersten Fehlers im Ablauf des Programms Aufruf-Einträge in den jeweiligen Komponenten (CUA, CUB) gespeichert werden.
7. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass ein jeweiliger aktiver oder passiver Fehlereintrag in einer Komponente (CUA, CUB) durch ein Feld umfassend eine Mehrzahl von Werten beschrieben wird, wobei in dem Feld die Programmidentität (a, b) spezifiziert ist und das Feld ferner einen

ersten Wert (i) enthält, der die Identifikation (i) der Komponente (CUA, CUB) spezifiziert, welche den Fehlereintrag speichert, sowie einen zweiten Wert (p), der die Komponente (CUA, CUB) spezifiziert, in welcher der Fehler aufgetreten ist, und einen dritten Wert (e), der angibt, ob der Fehlereintrag ein passiver Fehlereintrag oder ein aktiver Fehlereintrag ist, wobei im Falle eines aktiven Fehlereintrags der dritte Wert (e) ferner spezifiziert, ob der Fehler der erste Fehler im Ablauf des Programms ist.

8. Verfahren nach Anspruch 7, wenn abhängig von Anspruch 3, dadurch gekennzeichnet, dass in dem Feld zwei Werte (a, b) zur Spezifikation der Programmidentität enthalten sind, wobei einer der Werte (a) die Identifikation der auslösenden Komponente und der andere der Werte (b) die Startzeit darstellt.
9. Verfahren zum Verarbeiten von mit einem Verfahren nach einem der vorhergehenden Ansprüche erfassten Fehlern, dadurch gekennzeichnet, dass die gespeicherten aktiven und passiven Fehlereinträge ausgelesen werden und für jede Programmidentität (a, b) aus den dazugehörigen aktiven und passiven Fehlereinträgen ein Fehlerbaum (FB) generiert und ausgegeben wird, wobei in dem Fehlerbaum zumindest die Aufrufketten von aufeinander folgend aufgerufenen Komponenten (CU1, CU2, ..., CU12) enthalten sind, welche wenigstens eine Komponente (CU1, CU2, ..., CU12) mit einem aktiven Fehlereintrag enthalten, wobei in dem Fehlerbaum (FB) die Komponenten (CU1, CU2, ..., CU12) dahingehend gekennzeichnet sind, ob in der jeweiligen Komponente (CU1, CU2, ..., CU12) im Ablauf des Programms kein Fehler oder der erste Fehler oder ein späterer Fehler aufgetreten ist.
10. Verfahren nach Anspruch 9, dadurch gekennzeichnet, dass die Fehler mit einem Verfahren nach Anspruch 7 erfasst sind, wobei der Fehlerbaum (FB) derart generiert wird, dass ausgehend von jeder Komponente (CU1, CU2, ..., CU12) mit einem aktiven Fehlereintrag schrittweise über den ersten und zweiten Wert (i, p) der Felder der Fehlereinträge jeweilige Aufrufketten von aufeinander folgend aufgerufenen Komponenten (CU1, CU2, ..., CU12) erzeugt werden.

11. System aus Komponenten (CU1, CU2, ..., CU12), insbesondere in einem Kraftfahrzeug, dadurch gekennzeichnet, dass das System derart ausgestaltet ist, dass Fehler beim Ablauf von einem oder mehreren software-basierten Programmen in dem System mit einem Verfahren nach einem der Ansprüche 1 bis 8 erfasst werden.
12. Kraftfahrzeug, dadurch gekennzeichnet, dass das Kraftfahrzeug ein System aus Komponenten (CU1, CU2, ..., CU12) nach Anspruch 11 umfasst.
13. Diagnosevorrichtung zum Verarbeiten von mit einem Verfahren nach einem der Ansprüche 1 bis 8 erfassten Fehlern, wobei die Diagnosevorrichtung derart ausgestaltet ist, dass ein Verfahren nach Anspruch 9 oder 10 mit der Vorrichtung durchführbar ist.

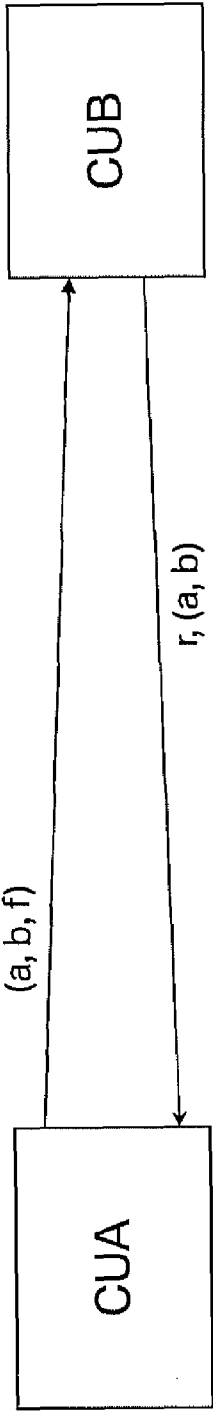


Fig. 1

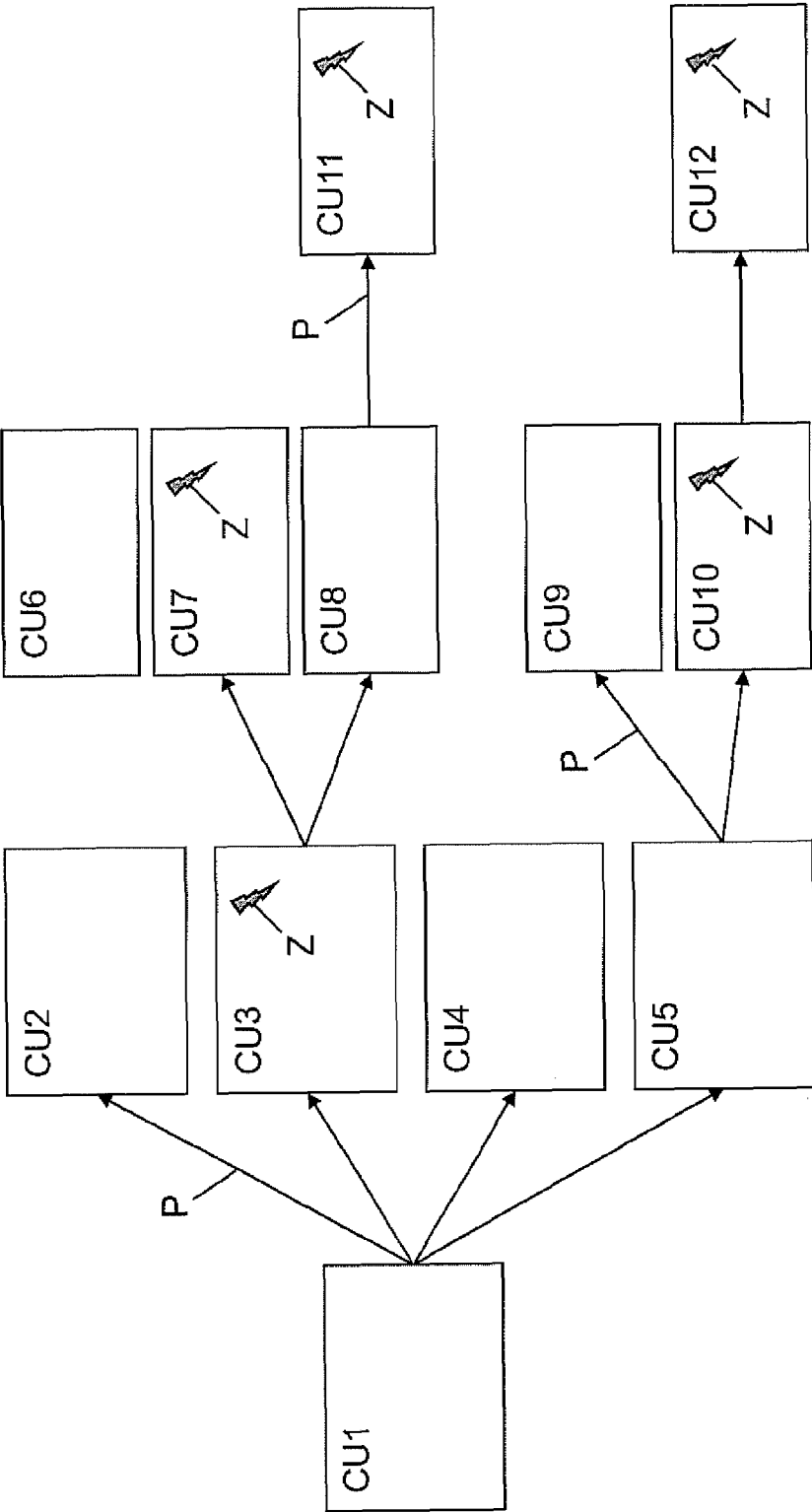


Fig. 2

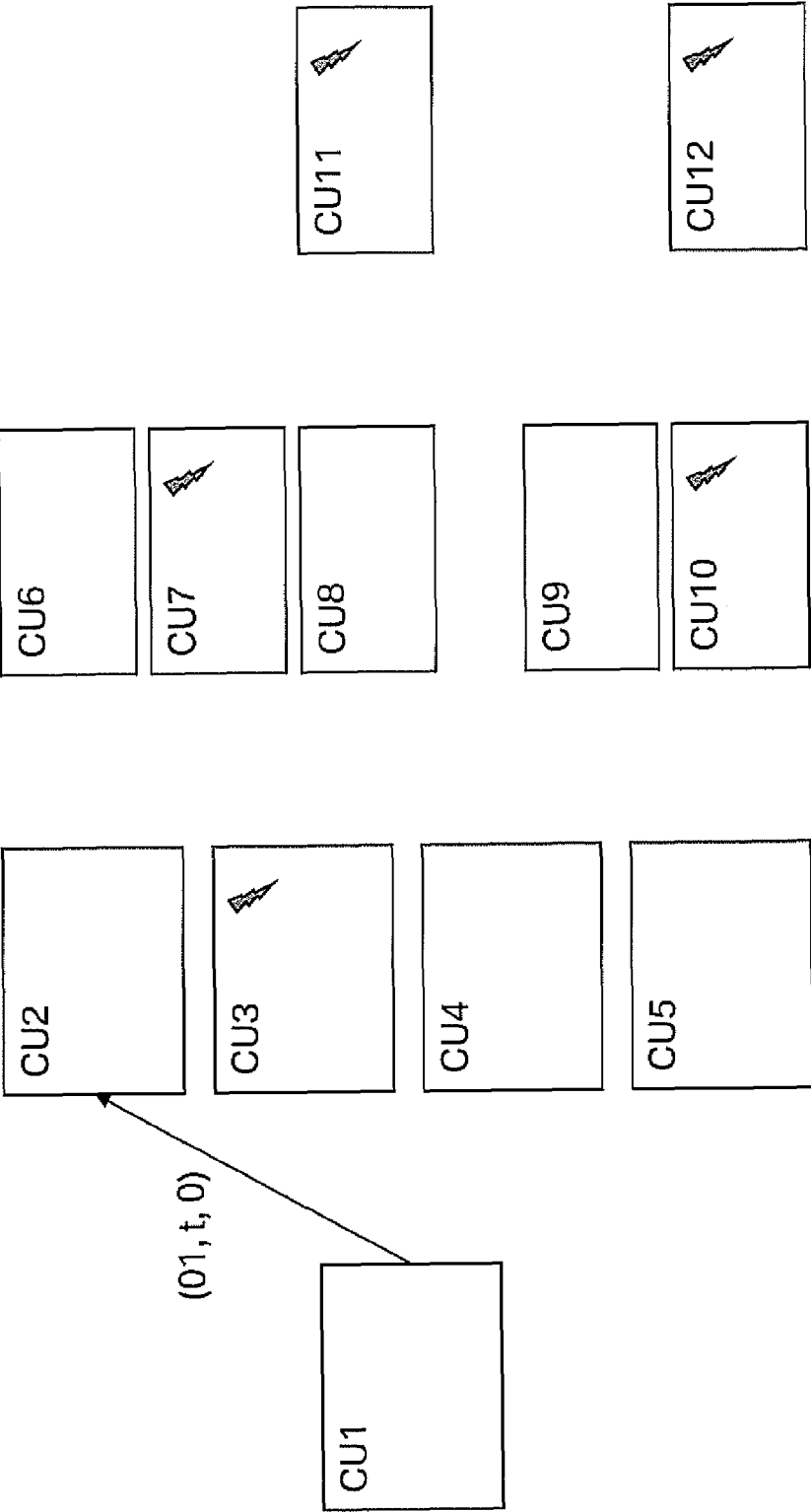


Fig. 3

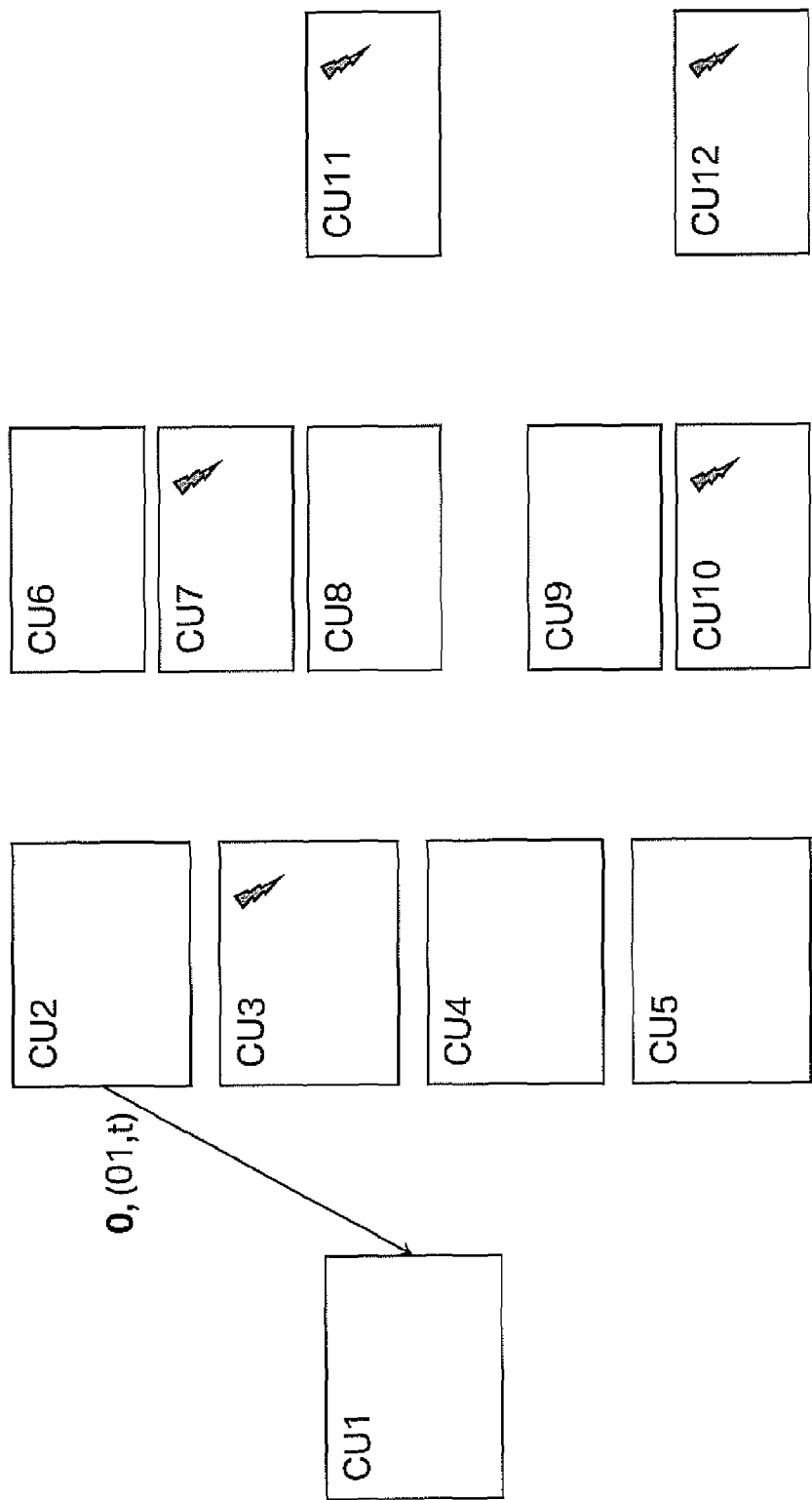


Fig. 4

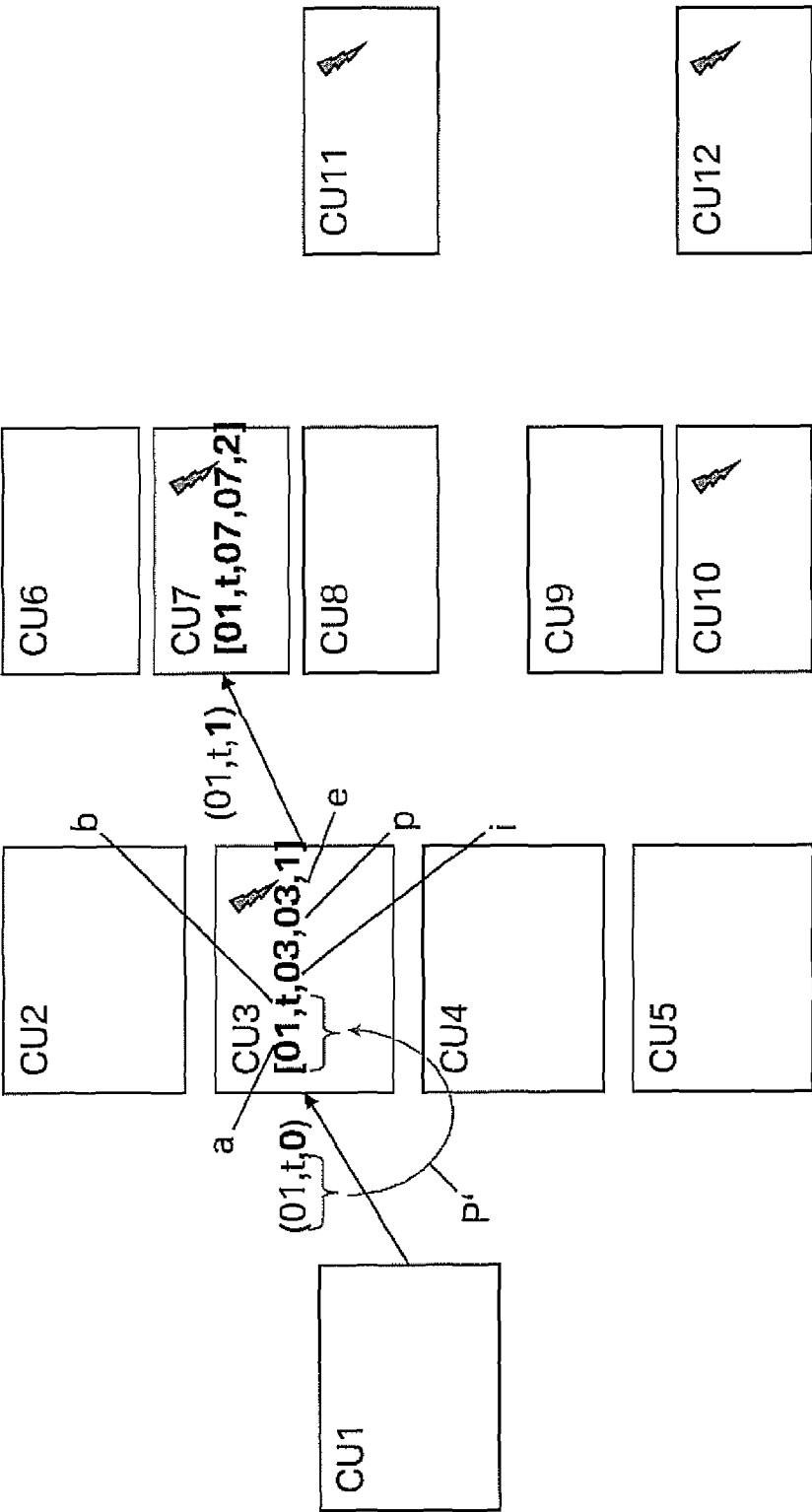


Fig. 5

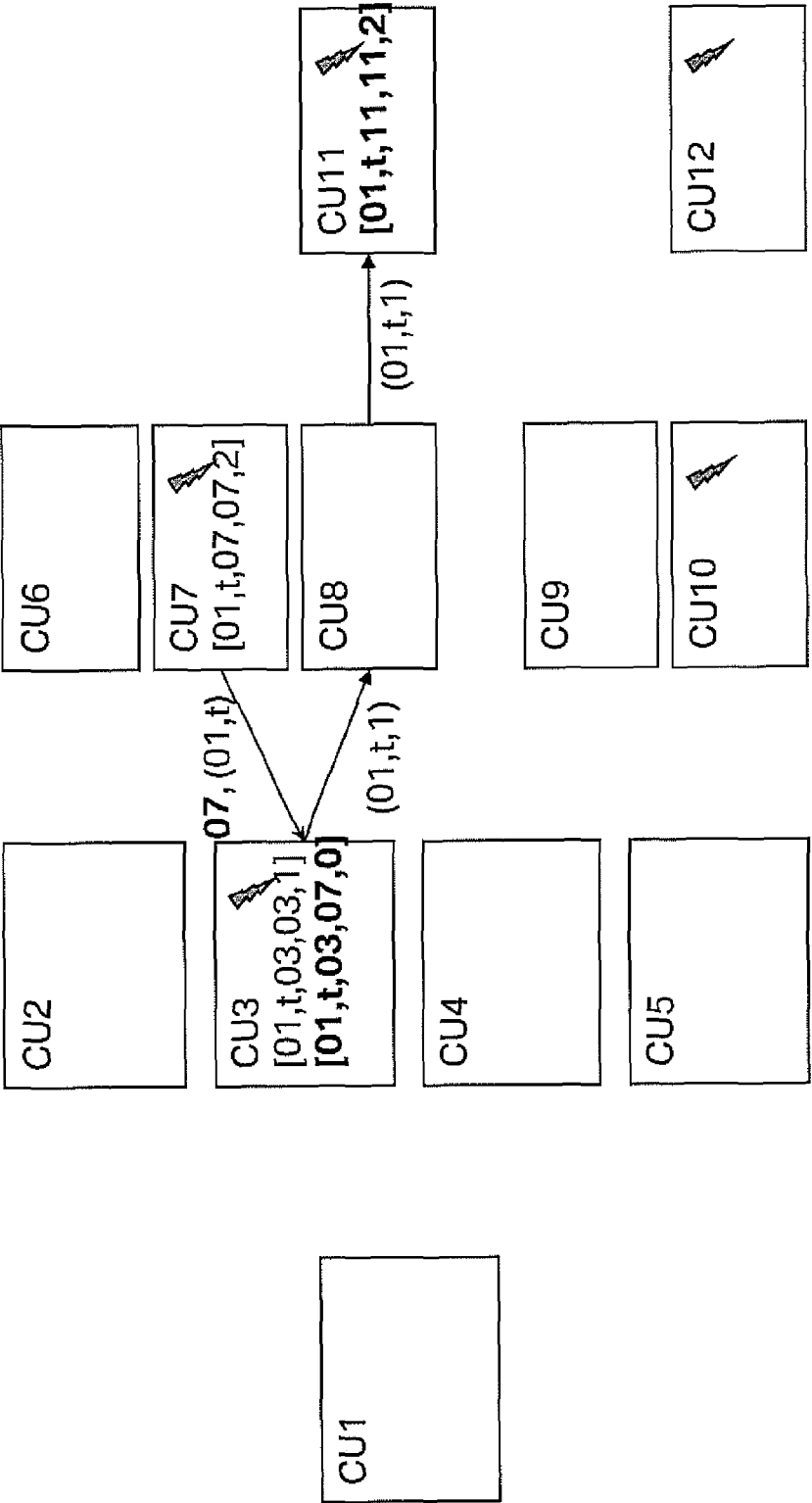


Fig. 6

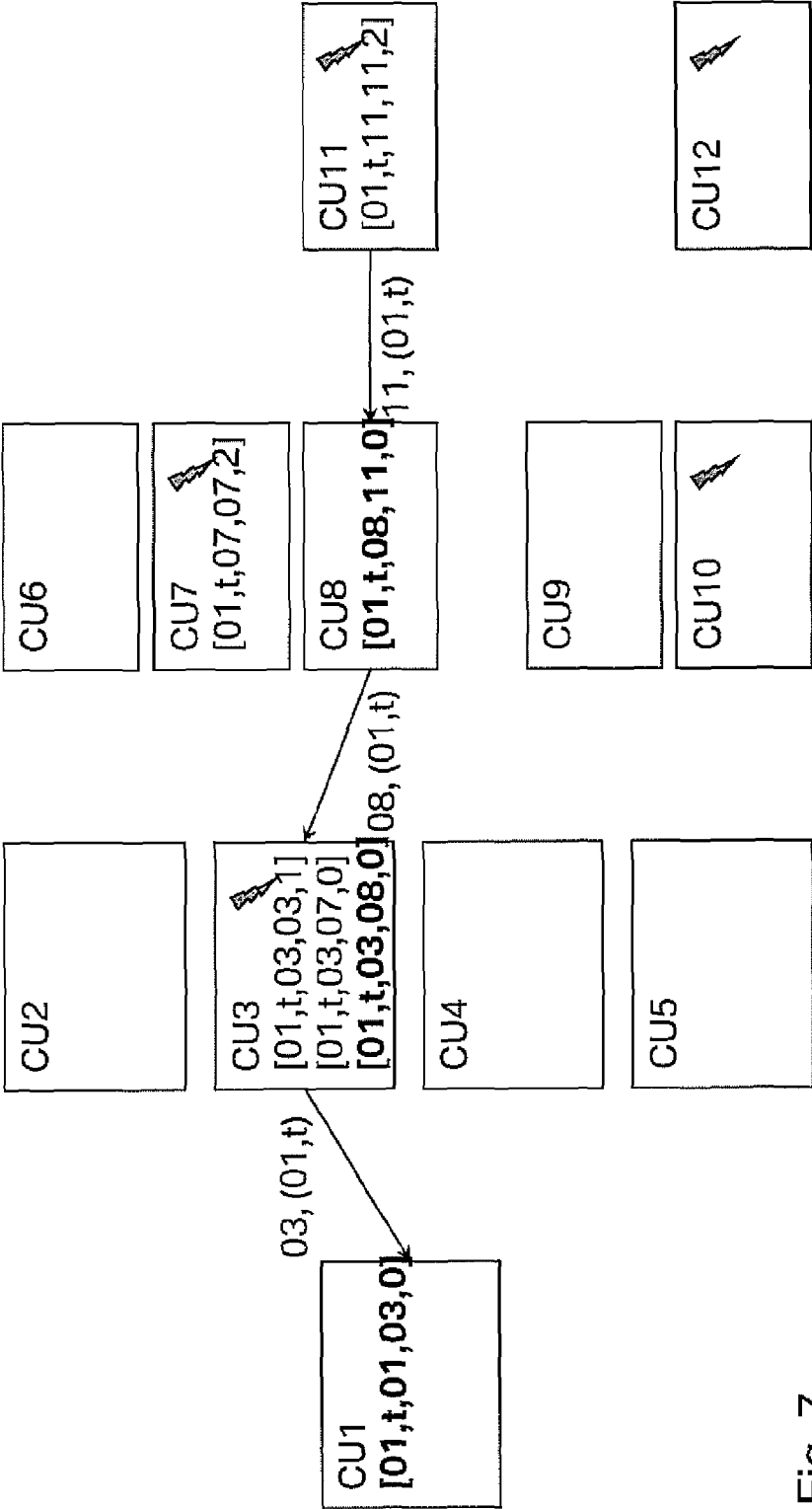


Fig. 7

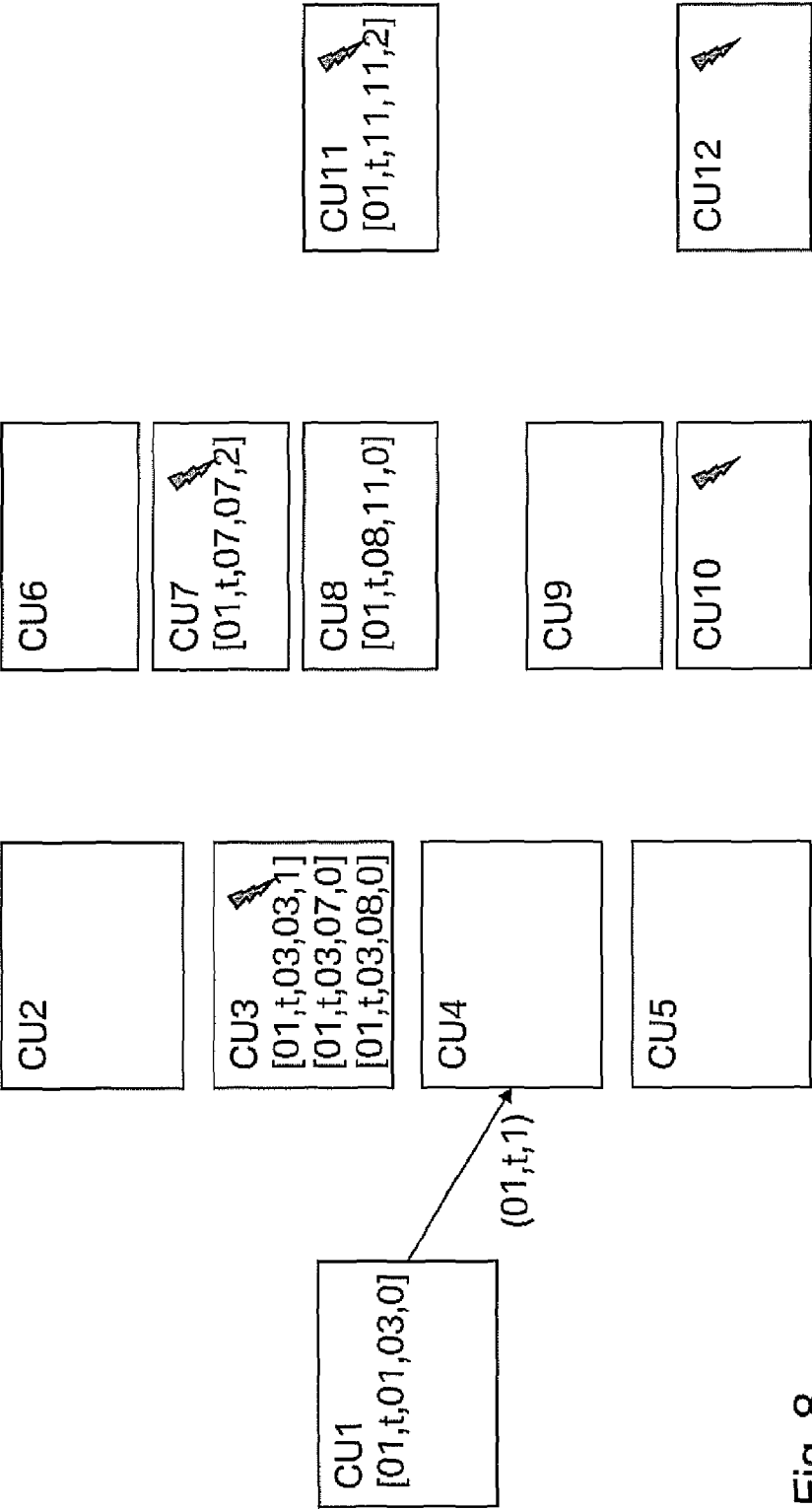


Fig. 8

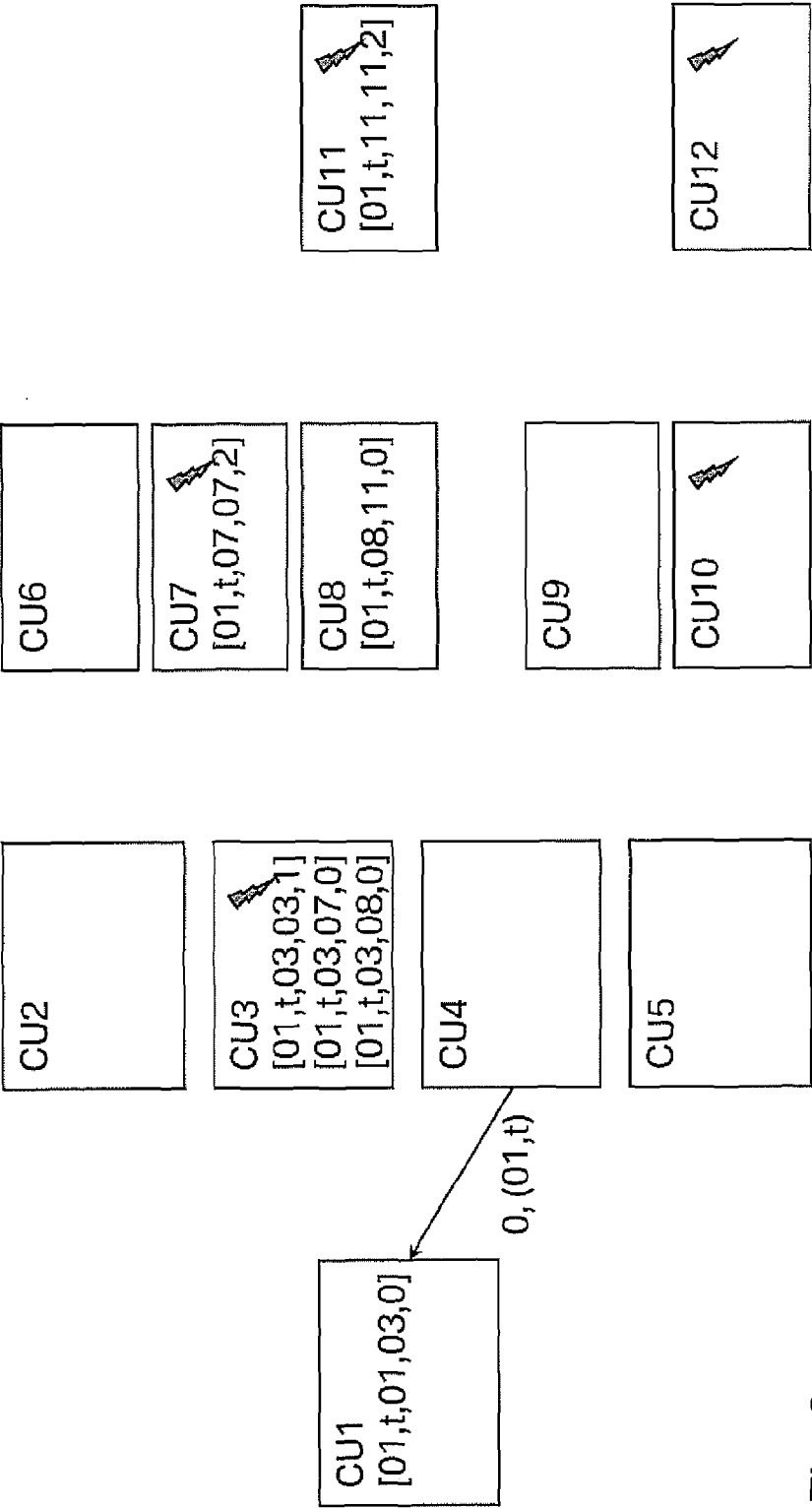


Fig. 9

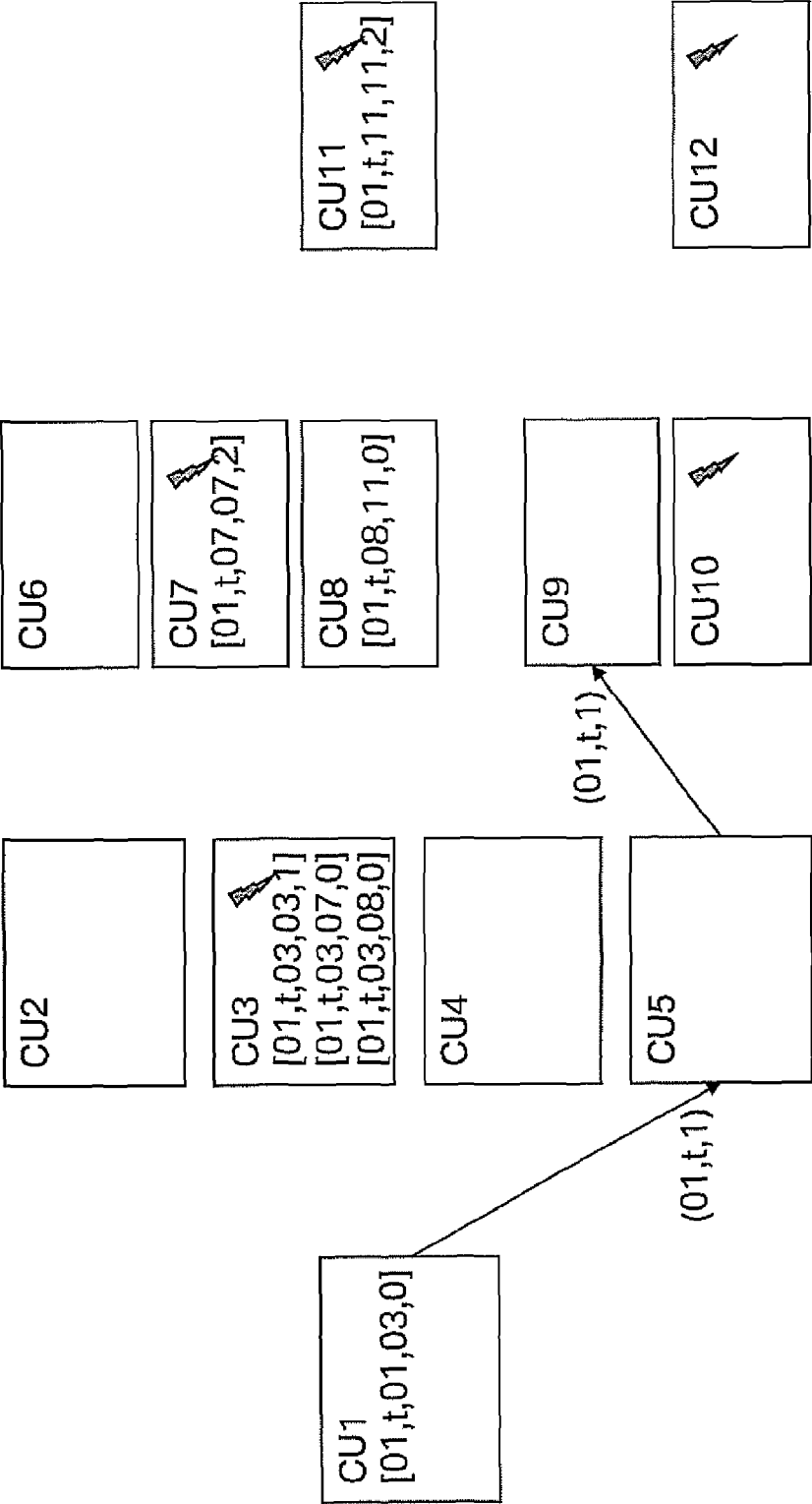


Fig. 10

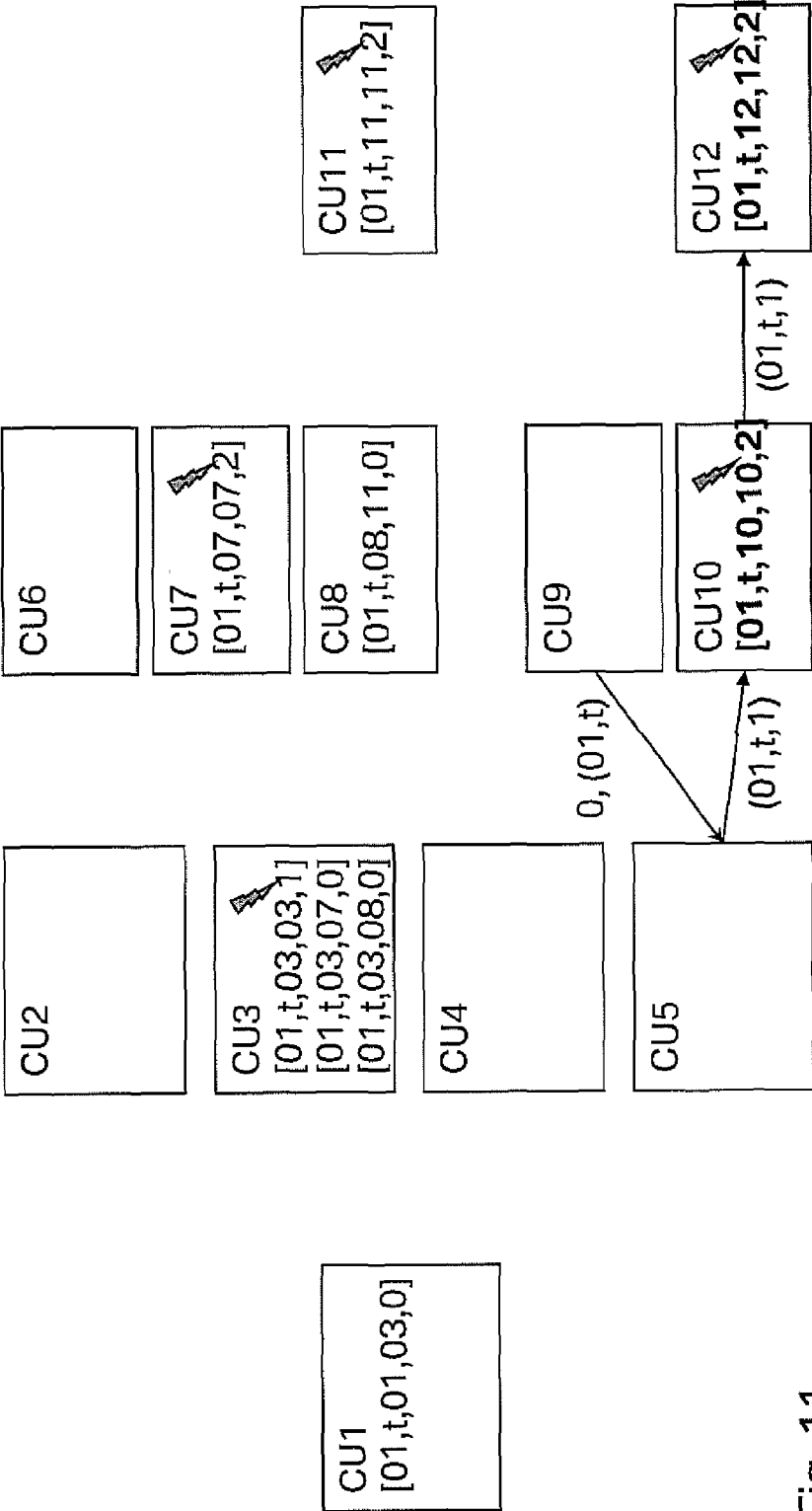


Fig. 11

12/13

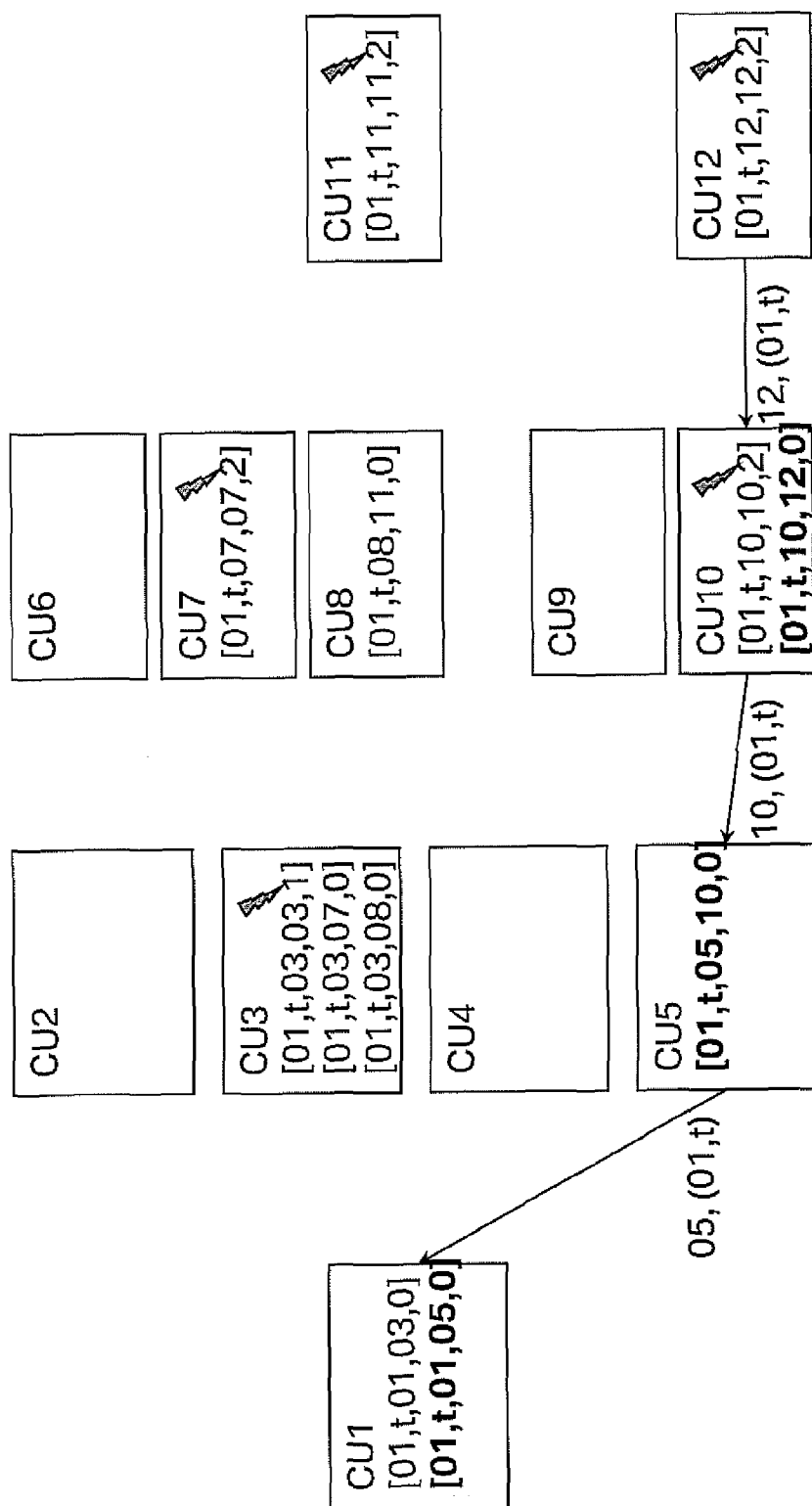


Fig. 12

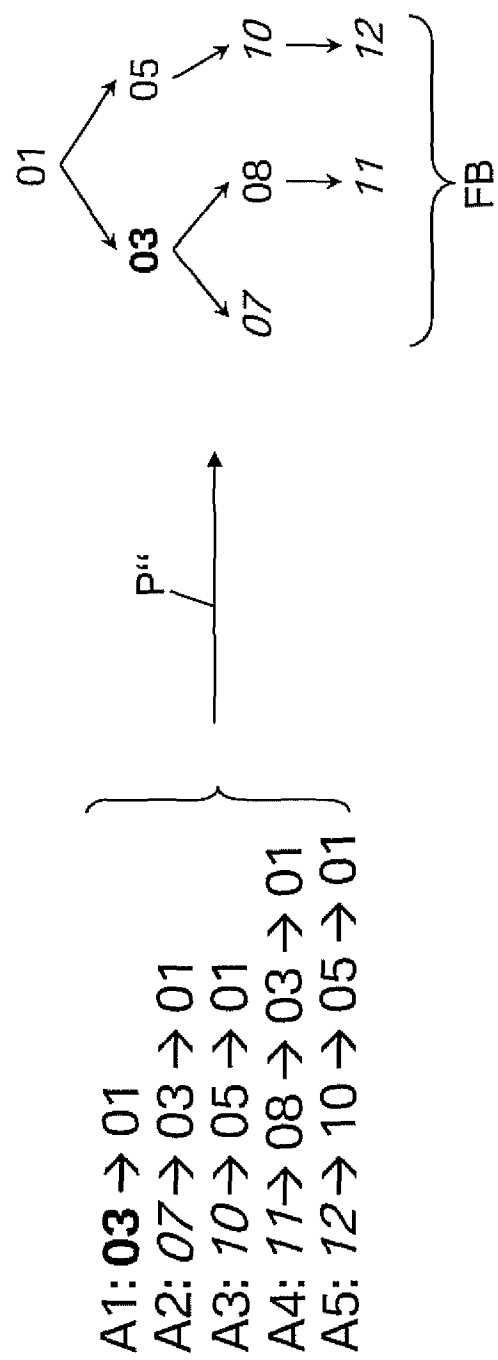


Fig. 13

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2010/057977

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F11/07 G05B23/02
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F G05B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, INSPEC, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2003/204804 A1 (PETRI ROBERT J [US] ET AL) 30 October 2003 (2003-10-30) the whole document	1-13
X	WO 97/19403 A1 (MCI COMMUNICATIONS CORP [US]) 29 May 1997 (1997-05-29) the whole document	1-13
A	US 5 948 107 A (RAMANATHAN RAMANATHAN [US]) 7 September 1999 (1999-09-07) the whole document	1-13
-/--		

☒ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

2 September 2010

Date of mailing of the international search report

09/09/2010

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Kielhöfer, Patrick

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2010/057977

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No. .
A	HEINECKE H ET AL: "AUTOSAR - Current results and preparations for exploitation" INTERNET CITATION May 2006 (2006-05), XP002466416 Retrieved from the Internet: URL: http://www.autosar.org/download/AUTOSAR_Euroforum_2006.pdf [retrieved on 2008-01-28] page 5, left-hand column, last paragraph - right-hand column, paragraph 5 -----	1,11-13

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2010/057977

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 2003204804	A1	30-10-2003	NONE	
WO 9719403	A1	29-05-1997	CA 2235838 A1 EP 0861466 A1 JP 2001511918 T US 5974568 A	29-05-1997 02-09-1998 14-08-2001 26-10-1999
US 5948107	A	07-09-1999	NONE	

INTERNATIONALER RECHERCHENBERICHT

Internationales Aktenzeichen

PCT/EP2010/057977

A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES

INV. G06F11/07 G05B23/02
ADD.

Nach der Internationalen Patentklassifikation (IPC) oder nach der nationalen Klassifikation und der IPC

B. RECHERCHIERTE GEBIETE

Recherchierter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole)

G06F G05B

Recherchierte, aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen

Während der internationalen Recherche konsultierte elektronische Datenbank (Name der Datenbank und evtl. verwendete Suchbegriffe)

EPO-Internal, INSPEC, WPI Data

C. ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
X	US 2003/204804 A1 (PETRI ROBERT J [US] ET AL) 30. Oktober 2003 (2003-10-30) das ganze Dokument	1-13
X	WO 97/19403 A1 (MCI COMMUNICATIONS CORP [US]) 29. Mai 1997 (1997-05-29) das ganze Dokument	1-13
A	US 5 948 107 A (RAMANATHAN RAMANATHAN [US]) 7. September 1999 (1999-09-07) das ganze Dokument	1-13
	----- -/--	

☒ Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen ☒ Siehe Anhang Patentfamilie

* Besondere Kategorien von angegebenen Veröffentlichungen :

"A" Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als besonders bedeutsam anzusehen ist

"E" älteres Dokument, das jedoch erst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist

"L" Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweifelhaft erscheinen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden soll oder die aus einem anderen besonderen Grund angegeben ist (wie ausgeführt)

"O" Veröffentlichung, die sich auf eine mündliche Offenbarung, eine Benutzung, eine Ausstellung oder andere Maßnahmen bezieht

"P" Veröffentlichung, die vor dem internationalen Anmeldedatum, aber nach dem beanspruchten Prioritätsdatum veröffentlicht worden ist

"T" Spätere Veröffentlichung, die nach dem internationalen Anmeldedatum oder dem Prioritätsdatum veröffentlicht worden ist und mit der Anmeldung nicht kollidiert, sondern nur zum Verständnis des der Erfindung zugrundeliegenden Prinzips oder der ihr zugrundeliegenden Theorie angegeben ist

"X" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann allein aufgrund dieser Veröffentlichung nicht als neu oder auf erfinderischer Tätigkeit beruhend betrachtet werden

"Y" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann nicht als auf erfinderischer Tätigkeit beruhend betrachtet werden, wenn die Veröffentlichung mit einer oder mehreren anderen Veröffentlichungen dieser Kategorie in Verbindung gebracht wird und diese Verbindung für einen Fachmann naheliegend ist

"&" Veröffentlichung, die Mitglied derselben Patentfamilie ist

Datum des Abschlusses der internationalen Recherche

2. September 2010

Absendedatum des internationalen Recherchenberichts

09/09/2010

Name und Postanschrift der Internationalen Recherchenbehörde
Europäisches Patentamt, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Bevollmächtigter Bediensteter

Kielhöfer, Patrick

INTERNATIONALER RECHERCHENBERICHT

Internationales Aktenzeichen

PCT/EP2010/057977

C. (Fortsetzung) ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
A	HEINECKE H ET AL: "AUTOSAR - Current results and preparations for exploitation" INTERNET CITATION Mai 2006 (2006-05), XP002466416 Gefunden im Internet: URL:http://www.autosar.org/download/AUTOSAR_Euroforum_2006.pdf [gefunden am 2008-01-28] Seite 5, linke Spalte, letzter Absatz - rechte Spalte, Absatz 5 -----	1,11-13

INTERNATIONALER RECHERCHENBERICHT

Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören

Internationales Aktenzeichen

PCT/EP2010/057977

Im Recherchenbericht angeführtes Patentdokument	Datum der Veröffentlichung	Mitglied(er) der Patentfamilie	Datum der Veröffentlichung
US 2003204804 A1	30-10-2003	KEINE	
WO 9719403 A1	29-05-1997	CA 2235838 A1	29-05-1997
		EP 0861466 A1	02-09-1998
		JP 2001511918 T	14-08-2001
		US 5974568 A	26-10-1999
US 5948107 A	07-09-1999	KEINE	