



<p>(51) Internationale Patentklassifikation <sup>6</sup> : <b>G06F 9/40, 9/42, 9/445</b></p>	<b>A1</b>	<p>(11) Internationale Veröffentlichungsnummer: <b>WO 98/58311</b></p> <p>(43) Internationales Veröffentlichungsdatum: 23. Dezember 1998 (23.12.98)</p>
<p>(21) Internationales Aktenzeichen: PCT/DE98/01389</p> <p>(22) Internationales Anmeldedatum: 20. Mai 1998 (20.05.98)</p> <p>(30) Prioritätsdaten: 197 25 593.0 17. Juni 1997 (17.06.97) DE</p> <p>(71) Anmelder (für alle Bestimmungsstaaten ausser US): SIEMENS NIXDORF INFORMATIONSSYSTEME AG [DE/DE]; Heinz-Nixdorf-Ring 1, D-33106 Paderborn (DE).</p> <p>(72) Erfinder; und (75) Erfinder/Anmelder (nur für US): WEBER, Christian [DE/DE]; Lindacher Strasse 3a, D-85586 Poing (DE).</p> <p>(74) Anwälte: EPPING, Wilhelm usw.; Siemens Nixdorf Informationssysteme AG, Postfach 22 13 17, D-80503 München (DE).</p>	<p>(81) Bestimmungsstaaten: JP, US, europäisches Patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).</p> <p><b>Veröffentlicht</b> Mit internationalem Recherchenbericht. Vor Ablauf der für Änderungen der Ansprüche zugelassenen Frist; Veröffentlichung wird wiederholt falls Änderungen eintreffen.</p>	

(54) Title: METHOD FOR CONTROLLING A DATA PROCESSING SYSTEM

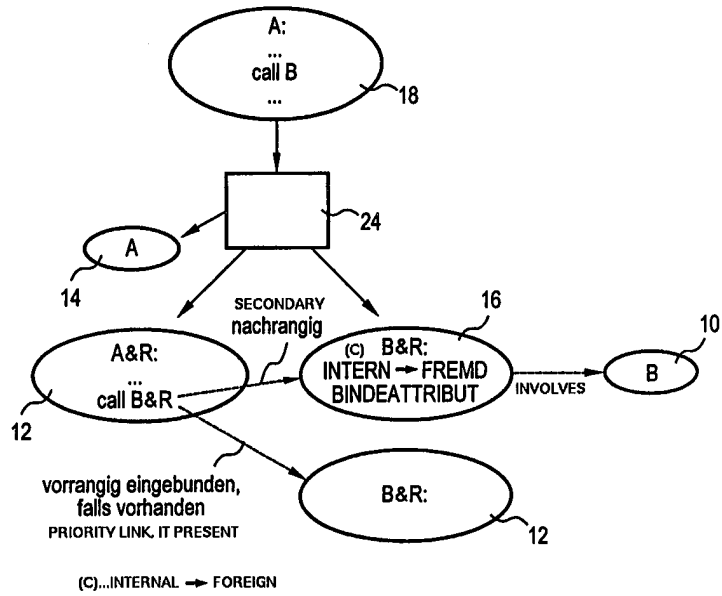
(54) Bezeichnung: VERFAHREN ZUM STEuern EINER DATENVERARBEITUNGSANLAGE

(57) Abstract

The invention relates to a method for controlling a data processing system for generating a target programme (26) by linking several programme modules. According to said method, internal programme modules (12) generated by a very high-level language translator (24) and foreign programme modules (10) are given names from mutually exclusive name categories. For each internal programme module (12) a matching routine (14) of a first type is generated and for each programme module (10) invoked by the internal programme modules (12) a matching routine (16) of a second type is generated. The matching routines bridge different logic declarations.

(57) Zusammenfassung

Bei einem Verfahren zum Steuern einer Datenverarbeitungsanlage zum Erzeugen eines Zielprogramms (26) durch Verknüpfen mehrerer Programm-Module werden interne Programm-Module (12), die durch einen Hochsprach-Übersetzer (24) erzeugt worden sind, und fremde Programm-Module (10) mit Namen aus einander ausschließenden Namensklassen versehen. Für die internen Programm-Module (12) wird jeweils eine Anpassungsroutine (14) erster Art, und für die von den internen Programm-Modulen (12) aufgerufenen Programm-Module (10) wird jeweils eine Anpassungsroutine (16) zweiter Art erzeugt. Die Anpassungsroutinen überbrücken unterschiedliche Verknüpfungsvereinbarungen.



### LEDIGLICH ZUR INFORMATION

Codes zur Identifizierung von PCT-Vertragsstaaten auf den Kopfbögen der Schriften, die internationale Anmeldungen gemäss dem PCT veröffentlichen.

AL	Albanien	ES	Spanien	LS	Lesotho	SI	Slowenien
AM	Armenien	FI	Finnland	LT	Litauen	SK	Slowakei
AT	Österreich	FR	Frankreich	LU	Luxemburg	SN	Senegal
AU	Australien	GA	Gabun	LV	Lettland	SZ	Swasiland
AZ	Aserbaidshjan	GB	Vereinigtes Königreich	MC	Monaco	TD	Tschad
BA	Bosnien-Herzegowina	GE	Georgien	MD	Republik Moldau	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagaskar	TJ	Tadschikistan
BE	Belgien	GN	Guinea	MK	Die ehemalige jugoslawische Republik Mazedonien	TM	Turkmenistan
BF	Burkina Faso	GR	Griechenland	ML	Mali	TR	Türkei
BG	Bulgarien	HU	Ungarn	MN	Mongolei	TT	Trinidad und Tobago
BJ	Benin	IE	Irland	MR	Mauretanien	UA	Ukraine
BR	Brasilien	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Island	MX	Mexiko	US	Vereinigte Staaten von Amerika
CA	Kanada	IT	Italien	NE	Niger	UZ	Usbekistan
CF	Zentralafrikanische Republik	JP	Japan	NL	Niederlande	VN	Vietnam
CG	Kongo	KE	Kenia	NO	Norwegen	YU	Jugoslawien
CH	Schweiz	KG	Kirgisistan	NZ	Neuseeland	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Demokratische Volksrepublik Korea	PL	Polen		
CM	Kamerun	KR	Republik Korea	PT	Portugal		
CN	China	KZ	Kasachstan	RO	Rumänien		
CU	Kuba	LC	St. Lucia	RU	Russische Föderation		
CZ	Tschechische Republik	LI	Liechtenstein	SD	Sudan		
DE	Deutschland	LK	Sri Lanka	SE	Schweden		
DK	Dänemark	LR	Liberia	SG	Singapur		
EE	Estland						

## Verfahren zum Steuern einer Datenverarbeitungsanlage

Die Erfindung betrifft ein Verfahren zum Steuern einer Daten-  
5 verarbeitungsanlage zum Erzeugen eines Zielprogramms durch  
Verknüpfen mehrerer Programm-Module.

Bei der Erstellung eines Zielprogramms, das in einer höheren  
Programmiersprache, beispielsweise COBOL, C, C++, Fortran  
10 etc. formuliert ist, werden in der Regel zunächst einzelne  
Quelltext-Module, beispielsweise Prozeduren und Funktionen  
erstellt, die unabhängig voneinander durch einen Hochsprach-  
Übersetzer, auch Compiler genannt, in die Maschinensprache,  
d.h. in für den Computer unmittelbar verständliche Befehle,  
15 übersetzt werden. Die auf diese Weise erzeugten Programm-  
Module werden dann durch einen Binder, auch Linker genannt,  
zu einem Zielprogramm verknüpft.

Bei einem Aufruf eines Programm-Moduls durch ein anderes muß  
20 sichergestellt sein, daß die beiden am Aufruf beteiligten  
Programm-Module einheitlichen Verknüpfungsvereinbarungen ge-  
nügen, um einen fehlerfreien Parametertransfer zwischen dem  
aufrufenden und dem aufgerufenen Programm-Modul zu gewährlei-  
sten. Solche Verknüpfungsvereinbarungen sind beispielsweise  
25 Konventionen über Registerbelegungen, Datenformate etc.

Hochsprach-Übersetzer für höhere Programmiersprachen erzeugen  
beim Übersetzen selbsttätig einen einheitlichen Vereinba-  
rungssatz von Verknüpfungsvereinbarungen, der für alle Pro-  
30 gramm-Module gültig ist, so daß sich der Programmentwickler  
nicht um die Verknüpfungsvereinbarung zu kümmern braucht.

Der Programmentwickler hat diese Verknüpfungsvereinbarungen  
jedoch zu beachten, sobald er Programm-Module, die nicht  
35 durch den vorgegebenen Hochsprach-Übersetzer erzeugt worden  
sind, in das Zielprogramm einbinden will, wie z.B. Programm-  
Module, die in einer Assembler-Sprache formuliert sind, im

folgenden kurz Assembler-Module genannt. Bei der Assembler-  
Programmierung werden die Verknüpfungsvereinbarungen durch  
den Programmentwickler selbst programmiert, so daß die Ver-  
knüpfungsvereinbarungen untrennbar in dem Assembler-Modul  
5 enthalten sind. Derartige Programm-Module, die nicht durch  
den vorgegebenen Hochsprach-Übersetzer erzeugt werden können  
und untrennbar an einen Vereinbarungssatz gekoppelt sind,  
werden im folgenden als fremde Programm-Module bezeichnet,  
während die durch den vorgegebenen Hochsprach-Übersetzer er-  
10 zeugten Module interne Programm-Module genannt werden.

Der Vereinbarungssatz für Modul-Verknüpfungen, der zum Zeit-  
punkt der Ersterstellung eines Zielprogramms sowohl durch den  
vorgegebenen Hochsprach-Übersetzer für interne Programm-Modu-  
15 le erzeugt als auch in fremden Programm-Modulen entsprechend  
berücksichtigt worden ist, wird im folgenden als Ausgangs-  
Vereinbarungssatz bezeichnet.

Soll das Zielprogramm auf ein anderes Computersystem übertra-  
20 gen werden, so ist es in der Regel zweckmäßig, einen neuen,  
an die Hardware des neuen Computersystems besonders gut ange-  
paßten Vereinbarungssatz für Modul-Verknüpfungen zu verwen-  
den, um gravierende Einbußen bei der Verarbeitungsgeschwin-  
digkeit auf dem neuen Computersystem zu vermeiden. Dieser  
25 neue Vereinbarungssatz wird im folgenden als effizienter Ver-  
einbarungssatz bezeichnet. Eine Verwendung dieses effizienten  
Vereinbarungssatzes ist dabei auch dann erforderlich, wenn  
kommerziell erhältliche Sammlungen von Programmroutinen des  
neuen Computersystems - im weiteren Standard-Module genannt -  
30 eingebunden werden sollen, die in der Regel nur für den be-  
sonders gut an die Hardware angepaßten effizienten Vereinba-  
rungssatz verfügbar sind. Aus diesen Gründen ist es wün-  
schenswert, die Verknüpfungsvereinbarungen des Zielprogramms  
möglichst umfassend auf den effizienten Vereinbarungssatz um-  
35 zustellen.

Der vorgegebene Hochsprach-Übersetzer könnte durch Neu-Übersetzung aller Quelltext-Module mit effizientem Vereinbarungssatz die erzeugten internen Programm-Module optimal an die neue Hardware anpassen. Die Verknüpfbarkeit der internen mit  
5 den fremden Programm-Modulen, die fest an den Ausgangs-Vereinbarungssatz gekoppelt sind, wäre dann aber nicht mehr gewährleistet.

Zur Lösung dieses Problems ist ein Verfahren bekannt, bei dem  
10 Anpassungsroutinen verwendet werden, um von einander abweichende Verknüpfungsvereinbarungen von Programm-Modulen auszugleichen. Um die Vereinbarungssätze von aufrufendem und aufgerufenem Programm-Modul aneinander anzupassen, muß der Programmentwickler bei dem bekannten Verfahren für alle betroffenen  
15 Programm-Module eine solche Anpassungsroutine erstellen. Er muß dabei nicht nur die Unterschiede in den Verknüpfungsvereinbarungen, sondern auch die Anzahl und den Typ der bei dem Aufruf jeweils übergebenen Parameter kennen, da diese für jedes Programm-Modul individuell festgelegt sind. Das bekannte  
20 Verfahren ist deshalb sehr aufwendig und erschwert die Übertragbarkeit des Zielprogramms auf ein neues Computersystem.

Es ist Aufgabe der Erfindung, ein Verfahren anzugeben, durch  
25 das ein Hochsprach-Übersetzer in die Lage versetzt wird, die eben genannten Anpassungsroutinen selbsttätig beim Übersetzen der Programm-Module zu erzeugen, und durch das die Anpassungsroutinen beim Verknüpfen der einzelnen Programm-Module durch einen Binder automatisch in das Zielprogramm eingebunden  
30 werden.

Die Erfindung löst diese Aufgabe durch das Verfahren mit den Merkmalen nach Anspruch 1.

35 Bei dem erfindungsgemäßen Verfahren werden durch einen Hochsprach-Übersetzer erzeugte, einem effizienten Vereinbarungssatz von Verknüpfungsvereinbarungen unterliegende interne

Programm-Module mit Namen aus einer internen Namensklasse  
versehen. Ferner werden einem Ausgangs-Vereinbarungssatz von  
Verknüpfungsvereinbarungen unterliegende fremde Programm-  
Module mit Namen aus einer Fremd-Namensklasse versehen, wobei  
5 sich die den jeweiligen Vereinbarungssätzen zugeordneten Na-  
mensklassen gegenseitig ausschließen. Es werden für die in-  
ternen Programm-Module jeweils eine Anpassungsroutine erster  
Art erzeugt, die mit einem dem Namen des zugehörigen internen  
Programm-Moduls entsprechenden Namen aus der Fremd-Namens-  
10 klasse versehen wird und die vor einem Aufruf des zugehörigen  
internen Programm-Moduls eine Anpassung des Ausgangs-Verein-  
barungssatzes des aufrufenden fremden Programm-Moduls an den  
effizienten Vereinbarungssatz des aufzurufenden internen Pro-  
gramm-Moduls vornimmt. Ferner werden Programm-Module von den  
15 internen Programm-Modulen aus mit Namen aus der internen Na-  
mensklasse aufgerufen. Für die von den internen Programm-  
Modulen aufzurufenden Programm-Module wird jeweils eine An-  
passungsroutine zweiter Art erzeugt, die mit einem dem Namen  
des zugehörigen Programm-Moduls entsprechenden Namen aus der  
20 internen Namensklasse versehen wird und die vor einem Ver-  
zweigen zum entsprechenden fremden Programm-Modul eine Anpas-  
sung des effizienten Vereinbarungssatzes an den Ausgangs-  
Vereinbarungssatz des aufzurufenden fremden Programm-Moduls  
vornimmt, wobei die Anpassungsroutine zweiter Art mit einem  
25 Bindeattribut versehen ist, so daß sie nur dann in ein Ziel-  
programm eingebunden wird, wenn neben ihr kein gleichnamiges  
internes Programm-Modul vorhanden ist. Weiterhin werden die  
internen Programm-Module, die fremden Programm-Module und die  
benötigten Anpassungsroutinen erster und zweiter Art mitein-  
30 ander zu dem Zielprogramm verknüpft.

Durch die Maßnahme, daß die internen und die fremden Pro-  
gramm-Module mit Namen aus einander ausschließenden Namens-  
klassen versehen werden, ist für den Binder erkennbar, wel-  
35 chem Vereinbarungssatz das jeweils einzubindende Programm-  
Modul zuzuordnen ist, so daß ein versehentliches Verknüpfen  
von Programm-Modulen, die unterschiedlichen Verknüpfungsver-

einbarungen unterliegen, sicher verhindert werden kann. Bei der Übersetzung der internen Programm-Module kann der Hochsprach-Übersetzer jeweils eine Anpassungsroutine erster Art automatisch erzeugen, da dem Hochsprach-Übersetzer die für diesen Zweck wesentlichen Informationen, nämlich die Anzahl und der Typ der in dem erzeugten internen Programm-Modul verwendeten Parameter, bekannt sind. Die fremden Programm-Module werden von den internen Programm-Modulen über die Anpassungsroutinen zweiter Art aufgerufen, die mit Namen aus der internen Namensklasse versehen sind und zu dem zugehörigen fremden Programm-Modul verzweigen. Auf diese Weise sind die für die Anpassung der unterschiedlichen Vereinbarungssätze notwendigen Anpassungsroutinen zweiter Art für die von den internen Programm-Modulen aufgerufenen fremden Programm-Module automatisch in das Zielprogramm eingebunden. Da die Anpassungsroutinen zweiter Art mit geeigneten Bindeattributen versehen sind, werden sie nur für solche Aufrufe eingebunden und durchlaufen, für die eine Anpassung des effizienten Vereinbarungssatzes an den Ausgangs-Vereinbarungssatz notwendig ist.

Bei dem vorgestellten Verfahren werden die Anpassungsroutinen beim Übersetzen durch den Hochsprach-Übersetzer automatisch erzeugt, durch den Binder in das Zielprogramm eingebunden und beim Abarbeiten des Zielprogramms aufgerufen. Insbesondere entfällt für den Programmentwickler die Notwendigkeit, die Anpassungsroutinen selbst zu erstellen. Die Erfindung erlaubt es, daß die internen und fremden Programm-Module korrekt über den Ausgangs-Vereinbarungssatz miteinander verknüpft werden, falls dies nötig ist. Die fremden Programm-Module können Assembler-Module oder durch einen anderen Hochsprach-Übersetzer erzeugte Programm-Module sein, die den Verknüpfungsvereinbarungen des Ausgangs-Vereinbarungssatzes unterliegen. Die durch den vorgegebenen Hochsprach-Übersetzer erzeugten internen Programm-Module können untereinander über den effizienten Vereinbarungssatz miteinander verknüpft werden, wodurch die Abarbeitungsgeschwindigkeit des Zielprogramms erhöht wird.

Eine vorteilhafte Weiterbildung der Erfindung besteht darin, daß die Namen der internen Programm-Module durch Anfügen einer in der verwendeten Hochsprache als Namensbestandteil nicht zugelassenen Zeichenfolge an die Namen der den internen Programm-Modulen zugrundeliegenden Quelltext-Module erzeugt werden. Auf diese Weise lassen sich die Namensklassen der internen und der fremden Programm-Module ohne besondere Anpassungen des Binders trennen. So kann beispielsweise der Namensbestandteil ein Zeichen enthalten, der in der vorgegebenen höheren Programmiersprache als Namensbestandteil nicht vorgesehen ist. Die Namensklassen werden so sicher voneinander getrennt, ohne daß die Namen der fremden Programm-Module verändert werden müssen.

15

Vorteilhaft wird für den Fall, daß das aufrufende Programm-Modul ein anderes Programm-Modul nicht über dessen Namen, sondern über dessen Adresse aufruft, vorausgesetzt, daß der Name des aufgerufenen Programm-Moduls der Fremd-Namensklasse zugeordnet ist. Für einen Aufruf eines Programm-Moduls über seine Adresse, im folgenden indirekter Aufruf genannt, wird also vorteilhaft angenommen, daß der Name des aufgerufenen Programm-Moduls in der Fremd-Namensklasse enthalten ist. Ferner wird durch die Anpassungsroutinen erster Art der aufgerufenen internen Programm-Module festgestellt, welchem Vereinbarungssatz das jeweilige aufrufende Programm-Modul unterliegt, und nur bei voneinander abweichenden Vereinbarungssätzen des aufrufenden und aufgerufenen Programm-Moduls die Anpassung der Vereinbarungssätze vorgenommen. Diese Maßnahme ermöglicht es unter anderem, daß die durch den Hochsprach-Übersetzer erzeugten internen Programm-Module auch von dem effizienten Vereinbarungssatz unterliegenden Standard-Modulen, beispielsweise von Bibliotheksroutinen über ihre Adresse aufgerufen werden können.

35

In einer weiteren vorteilhaften Weiterbildung der Erfindung werden die Anpassungsroutinen erster Art jeweils mit einem



Kennzeichnungscode versehen, durch den diese und die fremden Programm-Module voneinander zu unterscheiden sind. Bei einem indirekten Aufruf wird von dem aufrufenden internen Programm-Modul geprüft, ob unter der Adresse des aufgerufenen Programm-Moduls der Kennzeichnungscode abgelegt ist und eine Anpassung der voneinander abweichenden Vereinbarungssätze von aufrufendem und aufgerufenem Programm-Modul vorgenommen, falls kein Kennzeichnungscode unter dieser Adresse abgelegt ist. Dieser Verfahrensschritt erlaubt es, fremde Programm-Module, die keine Anpassungsroutine zweiter Art besitzen, aus den durch den Hochsprach-Übersetzer erzeugten internen Programm-Modulen heraus über ihre Adressen aufzurufen.

Vorteilhaft wird gemäß der Erfindung ein in Maschinensprache vorliegendes, dem effizienten Vereinbarungssatz unterliegendes Standard-Modul durch den Kennzeichnungscode ergänzt. Das Standard-Modul erhält neben dem ursprünglichen Namen aus der internen Namensklasse einen weiteren Namen aus der Fremd-Namensklasse. Auf diese Weise kann auch das Standard-Modul indirekt, d.h. über seine Adresse aufgerufen werden.

Durch die eben erläuterten Maßnahmen für indirekte Aufrufe werden die häufiger auftretenden direkten Aufrufe, d.h. die Aufrufe, welche die Namen der Programm-Module verwenden, kaum behindert.

Die Erfindung wird im folgenden an Hand der Figuren näher erläutert. Darin zeigen:

- 30   Figur 1    eine schematische Darstellung eines bekannten Verfahrens zum Anpassen der Vereinbarungssätze,
- Figur 2    eine schematische Darstellung, in der das Trennen der Namensklassen erläutert ist,

- Figur 3 eine schematische Darstellung, in der die Funktionsweise der Anpassungsroutine erster Art erläutert ist,
- 5 Figur 4 eine schematische Darstellung, in der die Funktionsweise der Anpassungsroutine zweiter Art erläutert ist,
- 10 Figur 5 eine schematische Darstellung der Aufrufbeziehungen des Zielprogramms,
- Figur 6 eine schematische Darstellung der erzeugten Programm-Module und Anpassungsroutinen des Zielprogramms nach Figur 5,
- 15 Figur 7 eine schematische Darstellung des von dem Binder erzeugten, lauffähigen Zielprogramms nach Figur 5,
- 20 Figur 8 ein Flußdiagramm der modifizierten Anpassungsroutine erster Art bei einem indirekten Modulaufruf,
- Figur 9 eine schematische Darstellung der Anpassung des Vereinbarungssatzes des aufrufenden Programm-Moduls beim indirekten Modulaufruf,
- 25 Figur 10 eine schematische Darstellung zur Erläuterung, wie ein Standard-Modul für indirekte Modulaufrufe zu modifizieren ist und
- 30 Figur 11 eine Übersicht über die Anpassungen beim indirekten Modulaufruf.

Zum besseren Verständnis werden in den folgenden Figuren fremde Programm-Module einheitlich mit der Bezugsziffer 10, 35 interne Programm-Module mit 12, Anpassungsroutinen erster bzw. zweiter Art mit 14 bzw. 16 und in einer Hochsprache ver-

faßte Quelltext-Module, die den lauffähigen Programm-Modulen zugrundeliegen, mit 18 bezeichnet.

In Figur 1 ist das bekannte Verfahren zum Anpassen von Vereinbarungssätzen erläutert. In Figur 1 soll ein fremdes Programm-Modul 10 mit dem Namen A, das einem Ausgangs-Vereinbarungssatz von Verknüpfungsvereinbarungen unterliegt, mit einem internen Programm-Modul 12 mit dem Namen B verknüpft werden, für das ein effizienter Vereinbarungssatz von Verknüpfungsvereinbarungen vorgesehen ist. Diese beiden Vereinbarungssätze weichen voneinander ab, falls beispielsweise das interne Programm-Modul 12 mit dem Namen B durch einen Hochsprach-Übersetzer erzeugt ist und das fremde Programm-Modul 10 mit dem Namen A ein Assembler-Modul ist oder durch einen mit dem vorgegebenen Hochsprach-Übersetzer nicht kompatiblen Hochsprach-Übersetzer einer anderen höheren Programmiersprache erzeugt worden ist. Als Hochsprach-Übersetzer kommen Übersetzer in Frage, die in höheren Programmiersprachen wie COBOL, C, C++, Fortran etc. verfaßte Quelltext-Module in eine Maschinensprache übersetzen und so lauffähige Programm-Module erzeugen. Um bei dem bekannten Verfahren die voneinander abweichenden Verknüpfungsvereinbarungen des fremden Programm-Moduls 10 mit dem Namen A und des internen Programm-Moduls 12 mit dem Namen B zu überbrücken, muß von einem Programmentwickler eine Anpassungsroutine 20 erstellt werden. Beim Verknüpfen der beiden Programm-Module 10, 12 durch einen Binder (nicht dargestellt) wird die Anpassungsroutine 20 mit eingebunden, um bei einem Aufruf des internen Programm-Moduls 12 mit dem Namen B durch das fremde Programm-Modul 10 mit dem Namen A die Anpassung der voneinander abweichenden Verknüpfungsvereinbarungen vorzunehmen. Bei dem bekannten Verfahren muß der Programmentwickler für jedes Modul, das von einem Programm-Modul mit abweichendem Vereinbarungssatz aufgerufen werden soll, eine solche Anpassungsroutine 20 erstellen.

35

In Figur 2 ist dargestellt, wie bei dem erfindungsgemäßen Verfahren den unterschiedlichen Vereinbarungssätzen voneinan-

der getrennte Namensklassen zugeordnet werden. In der vorgegebenen Hochsprache verfaßte Quelltext-Module 18 mit den Namen A und B werden sowohl durch einen Hochsprach-Übersetzer 22 gemäß dem Ausgangs-Vereinbarungssatz als auch durch einen  
5 Hochsprach-Übersetzer 24 gemäß dem effizienten Vereinbarungssatz übersetzt. Es wird nun festgelegt, daß die Namen der durch den Hochsprach-Übersetzer 22 erzeugten fremden Programm-Module 10 gegenüber den Namen der zugrundeliegenden Quelltext-Module 18 unverändert bleiben, nämlich A und B.  
10 Demgegenüber wird an die Namen der durch den Hochsprach-Übersetzer 24 erzeugten internen Programm-Module 12 die Zeichenfolge "&R" angehängt. Der Buchstabe "&" in der Zeichenfolge "&R" ist in keiner höheren Programmiersprache als Namensbestandteil eines Quelltext-Moduls zugelassen, so daß die Namen  
15 der internen Programm-Module 12 eine Namensklasse bilden, welche die Namen der fremden Programm-Module 10 mit Sicherheit nicht enthält. Den fremden Programm-Modulen 10 und den internen Programm-Modulen 12 sind also Namen aus sich gegenseitig ausschließenden Namensklassen zugeordnet, die eine  
20 Identifizierung des den jeweiligen Programm-Modulen zugeordneten Vereinbarungssatzes erlauben. In dem Beispiel nach Figur 2 sind die fremden Programm-Module 10 durch den Hochsprach-Übersetzer 22 erzeugt worden. Die fremden Programm-Module 10 können jedoch ebenso Assembler-Module oder in Maschinensprache vorliegende Programm-Module sein, die den Verknüpfungsvereinbarungen des Ausgangs-Vereinbarungssatzes unterliegen.

In Figur 3 ist die Funktionsweise einer Anpassungsroutine erster Art erläutert. Der Hochsprach-Übersetzer 24 erzeugt aus  
30 dem Quelltext-Modul 18 mit dem Namen A das interne Programm-Modul 12 mit dem Namen A&R. Das interne Programm-Modul 12 erfüllt dabei die Verknüpfungsvereinbarungen des effizienten Vereinbarungssatzes. Zusätzlich wird durch den Hochsprach-Übersetzer 24 eine Anpassungsroutine 14 erster Art erzeugt,  
35 die denselben Namen A wie das Quelltext-Modul 18 erhält. Der Name A der Anpassungsroutine 14 ist der dem Ausgangs-Verein-

barungssatz zugehörigen Namensklasse, d.h. der Fremd-Namensklasse zugeordnet. Die mit dem Namen A aufzurufende Anpassungsroutine 14 paßt den Ausgangs-Vereinbarungssatz an den effizienten Vereinbarungssatz an und ruft das mit dem Namen  
5 A&R versehene interne Programm-Modul 12 auf. Die Anpassungsroutine 14 erster Art kann von dem Hochsprach-Übersetzer 24 automatisch erzeugt werden, da diesem die Anzahl und der Typ der Parameter des erzeugten internen Programm-Moduls 12 bekannt sind.

10

In Figur 4 ist die Funktionsweise einer Anpassungsroutine 16 zweiter Art erläutert. In dem Quelltext-Modul 18 mit dem Namen A ist ein Programmbefehl "call B" vorgesehen, mit dem ein weiteres Quelltext-Modul mit dem Namen B aufgerufen wird.  
15 Durch den Hochsprach-Übersetzer 24 wird aus dem Quelltext-Modul 18 mit dem Namen A die in Figur 3 erläuterte Anpassungsroutine 14 erster Art ebenfalls mit dem Namen A und das interne Programm-Modul 12 mit dem Namen A&R erzeugt. Von dem internen Programm-Modul 12 aus wird nun jedes Programm-Modul  
20 grundsätzlich gemäß dem effizienten Vereinbarungssatz und mit dem für den effizienten Vereinbarungssatz festgelegten Namen aufgerufen. In diesem Beispiel ruft so das interne Programm-Modul A&R das Programm-Modul mit dem Namen B&R auf. Für jedes von dem internen Programm-Modul 12 aufgerufene Programm-Modul  
25 wird eine Anpassungsroutine 16 zweiter Art erzeugt, die in der Lage ist, den effizienten Vereinbarungssatz an den Ausgangs-Vereinbarungssatz anzupassen. Die Anpassungsroutine 16 zweiter Art erhält dabei einen Namen aus der dem effizienten Vereinbarungssatz zugeordneten Namensklasse, d.h. der internen  
30 Namensklasse. In dem Beispiel nach Figur 4 erstellt der Hochsprach-Übersetzer 24 somit die Anpassungsroutine 16 zweiter Art mit dem Namen B&R. Die Anpassungsroutine 16 mit dem Namen B&R ist zudem mit einem Bindeattribut versehen, welches den Binder veranlaßt, die Anpassungsroutine 16 nur dann ein-  
35 zubinden, wenn kein internes Programm-Modul mit dem Namen B&R vorhanden ist. Trifft der Binder ein solches internes Programm-Modul nicht an, so bindet er die Anpassungsroutine 16

ein, welche die notwendige Anpassung der Vereinbarungssätze vornimmt und daraufhin das fremde Programm-Modul 10 mit dem Namen B aufruft.

5 Durch die in den Figuren 2 bis 4 erläuterten Maßnahmen können einzelne Programm-Module, die unterschiedlichen Vereinbarungssätzen unterliegen, durch den Binder zu einem Zielprogramm verknüpft werden, ohne daß der Programmentwickler die Anpassungsroutinen selbst erstellen muß.

10

Das Zusammenwirken der eben erläuterten Maßnahmen wird im folgenden an Hand der Figuren 5 bis 8 an einem Beispiel erläutert.

15 In Figur 5 sind die zu einem Zielprogramm 26 zu verknüpfenden Programm-Module und die Aufrufbeziehungen zwischen diesen Programm-Modulen dargestellt. Das Zielprogramm 26 besteht aus Assembler-Modulen 28 mit den Namen A und F, die dem Ausgangsvereinbarungssatz unterliegen, den in der Hochsprache verfaß-

20 ten Quelltext-Modulen 18 mit den Namen B, D, E, die dem effizienten Vereinbarungssatz unterliegen, und einem ebenfalls dem effizienten Vereinbarungssatz unterliegenden Standard-Modul 30 mit dem Namen G. Das Zielprogramm 26 sieht vor, daß das Assembler-Modul A die Quelltext-Module B, D, E aufruft.

25 Von dem Quelltext-Modul B werden wiederum das Quelltext-Modul D und das Standard-Modul G aufgerufen. Das Quelltext-Modul D ruft schließlich das Quelltext-Modul E, das Assembler-Modul F und das Standard-Modul G auf.

30 In Figur 6 sind die durch den Hochsprach-Übersetzer 24 erzeugten Programm-Module und die Anpassungsroutinen des Zielprogramms nach Figur 5 dargestellt. Beim Übersetzen der Quelltext-Module 18 (vgl. Fig. 5) mit den Namen B, D, E werden die entsprechenden internen Programm-Module 12 mit den

35 Namen B&R, D&R, E&R erzeugt. Ferner erstellt der Hochsprach-Übersetzer 24 die Anpassungsroutinen 14 erster Art mit den Namen B, D, E. Von den internen Programm-Modulen 12 werden

weitere Programm-Module mit den Namen aus der internen Namensklasse aufgerufen. Das interne Programm-Modul B&R ruft so das Standard-Modul 30 mit dem Namen G&R sowie das interne Programm-Modul D&R auf. Ebenso ruft das interne Programm-Modul 12 mit dem Namen D&R das Standard-Modul 30 mit dem Namen G&R, das Assembler-Modul 28 mit dem Namen F&R sowie das interne Programm-Modul E&R auf. Es ist darauf hinzuweisen, daß das dem effizienten Vereinbarungssatz unterliegende Standard-Modul 30 mit dem Namen G&R zu versehen ist, falls es diesen Namen nicht schon hat. Dies läßt sich leicht über eine automatisch realisierbare Umbenennung erreichen.

Der Hochsprach-Übersetzer 24 erzeugt schließlich für die Programm-Module, die von den internen Programm-Modulen 12 aufgerufen werden, die entsprechenden, mit dem Bindeattribut versehenen Anpassungsroutinen 16 zweiter Art. Für das interne Programm-Modul B&R sind dies die Anpassungsroutinen 16 mit den Namen D&R und G&R. Entsprechend werden für das interne Programm-Modul 12 mit dem Namen D&R die Anpassungsroutinen 16 mit den Namen G&R, F&R und G&R erzeugt. Wie Figur 6 zu entnehmen ist, können die Anpassungsroutinen durch verschiedene Übersetzungsvorgänge mehrmals erzeugt werden. So wird beispielsweise die Anpassungsroutine 16 mit dem Namen G&R sowohl bei der Übersetzung des Quelltext-Moduls B als auch bei der Übersetzung des Quelltext-Moduls D erstellt.

In Figur 7 ist schließlich das von dem Binder erzeugte, lauffähige Zielprogramm 26 gezeigt, das durch Verknüpfen der einzelnen Programm-Module entstanden ist. Hier ist nochmals verdeutlicht, wie die internen Programm-Module B&R, D&R, E&R von dem Assembler-Modul 28 mit dem Namen A aus über die Namen B, D und E aufgerufen werden. Dementsprechend wird das Assembler-Modul 28 mit dem Namen F von dem internen Programm-Modul 12 mit dem Namen D&R über die Anpassungsroutine mit dem Namen F&R aufgerufen. Im Gegensatz dazu ruft das interne Programm-Modul 12 mit dem Namen B&R das interne Programm-Modul 12 mit dem Namen G&R ohne Zwischenschaltung einer Anpassungsroutine

auf, da bei diesem Aufruf keine Anpassung der Vereinbarungssätze nötig ist.

In manchen höheren Programmiersprachen, z.B. der Sprache C,  
5 ist es möglich, ein Programm-Modul nicht nur über seinen Namen, sondern auch über seine Adresse, im weiteren Ziel-Adresse genannt, d.h. über einen "Funktions-Zeiger" auf seine Ziel-Adresse aufzurufen, so daß beim aufrufenden Programm-Modul der Name des aufgerufenen Programm-Moduls nicht mehr  
10 bekannt ist, sondern nur noch dessen Ziel-Adresse. Für diese indirekten Modulaufrufe ist das oben beschriebene Verfahren, das auf dem Trennen der den verschiedenen Vereinbarungssätzen zugrundeliegenden Namensklassen beruht, etwas zu modifizieren, wie im folgenden an Hand der Figuren 8 bis 10 erläutert  
15 wird.

Figur 8 zeigt ein Flußdiagramm einer modifizierten Anpassungsroutine erster Art bei einem indirekten Modulaufruf. Zunächst wird vorausgesetzt, daß der Funktions-Zeiger stets auf  
20 eine Variante des Programm-Moduls zeigt, die dem Ausgangs-Vereinbarungssatz unterliegt. Der indirekte Aufruf erfolgt so in Figur 8 auf die Routine mit dem Namen A. Die Anpassungsroutine erster Art ist nun insofern modifiziert, als daß sie durch eine Abfrage feststellt, welchem Vereinbarungssatz das aufrufende Programm-Modul unterliegt. Bei der Anpassungsroutine nach Figur 8 wird abgefragt, ob der Aufruf gemäß dem  
25 Ausgangs-Vereinbarungssatz erfolgt ist. Ist dies nicht der Fall, d.h. erfolgte der indirekte Aufruf gemäß dem effizienten Vereinbarungssatz, so findet keine Anpassung der Vereinbarungssätze statt und es wird das Programm-Modul mit dem Namen A&R aufgerufen. Stellt die modifizierte Anpassungsroutine erster Art fest, daß das aufrufende Programm-Modul dem Ausgangs-Vereinbarungssatz unterliegt, so findet die Umsetzung dieses Vereinbarungssatzes auf den effizienten Vereinbarungssatz statt, bevor das Programm-Modul A&R aufgerufen  
30 wird.  
35



Wie ein Kriterium zum Erfassen des Vereinbarungssatzes, gemäß dem der indirekte Aufruf erfolgt, festgelegt werden kann, hängt von den konkreten Verknüpfungsvereinbarungen ab. So ist es beispielsweise möglich, ein Register, das bei dem Ausgangs-Vereinbarungssatz nicht verwendet wird, aber bei dem effizienten Vereinbarungssatz stets mit einer gültigen Befehlsadresse besetzt ist, als Kriterium festzulegen, indem es bei Routinen mit Ausgangs-Vereinbarungssatz definiert auf eine vorgegebene ungültige Befehls-Adresse gesetzt wird.

10

Die in Figur 8 erläuterte Modifizierung der Anpassungsroutine erster Art erlaubt es, die durch den Hochsprach-Übersetzer erzeugten internen Programm-Module auch von Standard-Modulen aus, die dem effizienten Vereinbarungssatz unterliegen, z.B. Bibliotheksroutinen, über Funktions-Zeiger aufzurufen.

15

Um ferner den indirekten Aufruf von fremden Programm-Modulen, die keine Anpassungsroutinen besitzen, aus internen Programm-Modulen heraus zu ermöglichen, ist eine weitere Modifizierung der Anpassungsroutinen erster Art nötig. So wird den Anpassungsroutinen erster Art ein Kennzeichnungscode vorangestellt, der die Anpassungsroutine als solche kennzeichnet. Als Kennzeichnungscode kann beispielsweise ein spezieller Leerbefehl verwendet werden, der ansonsten bei der Erzeugung eines Maschinencodes nicht verwendet wird. Bei einem indirekten Aufruf über die Ziel-Adresse des aufzurufenden Programm-Moduls wird geprüft, ob unter der Ziel-Adresse der Kennzeichnungscode abgelegt ist, wie in Figur 9 dargestellt ist. Ist dies der Fall, d.h. zeigt der Funktions-Zeiger auf eine Anpassungsroutine erster Art, so wird diese übersprungen, da auch die aufzurufende Programmroutine dem effizienten Vereinbarungssatz unterliegt und somit keine Anpassung der Vereinbarungssätze notwendig ist. Wird unter der Ziel-Adresse der Erkennungscode nicht vorgefunden, so bedeutet dies, daß über den Funktions-Zeiger ein fremdes Programm-Modul mit Ausgangs-Vereinbarungssatz aufgerufen werden soll. In diesem Fall wird der effiziente Vereinbarungssatz des aufrufenden Programm-

30

35

Moduls auf den Ausgangs-Vereinbarungssatz des aufzurufenden Programm-Moduls umgesetzt und die Ziel-Adresse angesprungen.

5 In Figur 10 ist gezeigt, wie das Standard-Modul 30 mit effizientem Vereinbarungssatz zu modifizieren ist, um indirekte Modulaufrufe mit Funktions-Zeigern zu ermöglichen. Das Standard-Modul 30 mit dem Namen A&R wird mittels eines Hilfsprogramms 40 mit dem in Figur 9 erläuterten Kennzeichnungscode versehen. Zudem erhält das Standard-Modul 30 neben dem Namen  
10 A&R den Namen A, der dem Ausgangs-Vereinbarungssatz zugeordnet ist. Bei einem indirekten Modulaufruf des Standard-Moduls 30 wird nun unter der Ziel-Adresse der Kennzeichnungscode vorgefunden, so daß wie in Figur 9 erläutert, ein indirekter Aufruf des Standard-Moduls 30 von dem internen Programm-Modul  
15 aus erfolgen kann, ohne daß eine Anpassung der Vereinbarungsätze vorgenommen werden muß.

Durch das Zusammenspiel der in den Figuren 8 bis 10 erläuterten Maßnahmen sind indirekte Modulaufrufe über Funktions-  
20 Zeiger möglich zwischen:

internen Programm-Modulen untereinander, die durch den Hochsprach-Übersetzer 24 erzeugt sind,

25 internen Programm-Modulen, die durch den Hochsprach-Übersetzer 24 erzeugt sind, und fremden Programm-Modulen, die dem Ausgangs-Vereinbarungssatz unterliegen, z.B. Assembler-Modulen,

30 internen Programm-Modulen, die durch den Hochsprach-Übersetzer erzeugt sind, und Standard-Modulen, die dem effizienten Vereinbarungssatz unterliegen, z.B. Bibliotheksroutinen.

35 Dies ist nochmals in der nachfolgenden Tabelle zusammenfassend dargestellt.

Typ des Aufrufers \ Typ des Aufgerufenen	I	II	III
I	+	+	-
II	+	+	+
III	-	+	+

I: Fremdes Programm-Modul Assembler-Modul mit Ausgangs-Vereinbarungssatz

5 II: Internes Programm-Modul mit effizientem Vereinbarungssatz

III: Standard-Modul mit effizientem Vereinbarungssatz

10 +: erlaubt

-: nicht erlaubt

15 Figur 11 gibt eine Übersicht über die durchgeführten Anpassungen der Vereinbarungssätze bei indirekten Modulaufrufen.

Dargestellt ist das Assembler-Modul 28 mit dem Namen A, das durch den Hochsprach-Übersetzer 24 erzeugte interne Programm-Modul 12 mit dem Namen B&R und die diesem zugeordnete Anpassungsroutine 14 erster Art mit dem Namen B, das durch den Hochsprach-Übersetzer 24 erzeugte interne Programm-Modul 12 mit dem Namen C&R und die diesem zugeordnete Anpassungsroutine 14 erster Art mit dem Namen C und das Standard-Modul 30, das sowohl mit dem Namen D als auch mit dem Namen D&R vor-  
25 liegt.

Bei dem indirekten Modulaufruf 1 findet die Anpassung der Vereinbarungssätze durch die Anpassungsroutine 14 mit dem Namen B statt. Bei den Aufrufen 2 findet keine Anpassung statt, da die aufrufenden Programm-Module 12 mit den Namen B&R, C&R  
5 den Kennzeichnungscode der Anpassungsroutinen 14 mit den Namen B bzw. C erkennen, und wie in Figur 9 gezeigt verzweigt wird. Bei Aufruf 3 wird bei dem Assembler-Modul 28 mit dem Namen A der Kennzeichnungscode nicht vorgefunden, so daß die Anpassung der Vereinbarungssätze bei dem aufrufenden Programm-Modul 12 mit dem Namen B&R vorgenommen wird. Bei Aufruf  
10 4 erkennt das aufrufende Programm-Modul 12 mit dem Namen C&R wie bei Aufruf 2 den Kennzeichnungscode und verzweigt, wie in Figur 9 dargestellt ist, ohne eine Anpassung der Vereinbarungssätze vorzunehmen. Auch bei Aufruf 5 wird keine Anpassung der Vereinbarungssätze vorgenommen, da die aufgerufene Anpassungsroutine 14 mit dem Namen C das interne Programm-Modul 12 mit dem Namen C&R aufruft, und für das Standard-Modul 30 mit dem Namen D&R und das interne Programm-Modul 12 mit dem Namen C&R keine Anpassung der Vereinbarungssätze notwendig ist. Mit 6 sind Aufrufe zwischen Programm-Modulen bezeichnet, die von vornherein demselben Vereinbarungssatz unterliegen, so daß auch in diesem Fall keine Anpassung der Vereinbarungssätze notwendig ist. Mit 7 sind schließlich indirekte Aufrufe zwischen dem Assembler-Modul 28 mit dem Namen  
20 A und dem Standard-Modul 30 mit dem Namen D bzw. D&R bezeichnet, die bei dem erfindungsgemäßen Verfahren verboten sind.

## Patentansprüche

1. Verfahren zum Steuern einer Datenverarbeitungsanlage zum Erzeugen eines Zielprogramms (26) durch Verknüpfen mehrerer Programm-Module,
- 5
- bei dem durch einen Hochsprach-Übersetzer (24) erzeugte, einem effizienten Vereinbarungssatz von Verknüpfungsvereinbarungen unterliegende interne Programm-Module mit Namen aus einer internen Namensklasse versehen werden (12),
- 10
- bei dem einem Ausgangs-Vereinbarungssatz von Verknüpfungsvereinbarungen unterliegende fremde Programm-Module (10) mit Namen aus einer Fremd-Namensklasse versehen werden, wobei sich die den jeweiligen Vereinbarungssätzen zugeordneten Namensklassen gegenseitig ausschließen,
- 15
- bei dem für die internen Programm-Module (12) jeweils eine Anpassungsroutine (14) erster Art erzeugt wird, die mit einem dem Namen des zugehörigen internen Programm-Moduls (12) entsprechenden Namen aus der Fremd-Namensklasse versehen wird und die vor einem Aufruf des zugehörigen internen Programm-Moduls (12) eine Anpassung des Ausgangs-Vereinbarungssatzes des aufrufenden fremden Programm-Moduls (10) an den effizienten Vereinbarungssatz des aufzurufenden internen Programm-Moduls (12) vornimmt,
- 20
- 25
- bei dem von den internen Programm-Modulen aus aufzurufende Programm-Module mit Namen aus der internen Namensklasse aufgerufen werden,
- 30
- bei dem für die von den internen Programm-Modulen (12) aufzurufenden Programm-Module jeweils eine Anpassungsroutine (16) zweiter Art erzeugt wird, die mit einem dem Namen des zugehörigen Programm-Moduls entsprechenden Namen aus der internen Namensklasse versehen wird und die vor einem Verzweigen zu einem fremden Programm-Modul eine An-
- 35

passung des effizienten Vereinbarungssatzes an den Ausgangs-Vereinbarungssatz des aufzurufenden externen Programm-Moduls (10) vornimmt, wobei die Anpassungsroutine (16) zweiter Art mit einem Bindeattribut versehen ist, so daß sie nur dann in ein Zielprogramm (26) eingebunden wird, wenn neben ihr kein gleichnamiges internes Programm-Modul (12) vorhanden ist,

und bei dem die internen Programm-Module (12), die fremden Programm-Module (10) und die benötigten Anpassungsroutinen erster (14) und zweiter Art (16) miteinander zu dem Zielprogramm (26) verknüpft werden.

2. Verfahren nach Anspruch 1, dadurch **gekennzeichnet**, daß die fremden Programm-Module (10) in einer Assembler-Sprache verfaßt sind.

3. Verfahren nach Anspruch 1, dadurch **gekennzeichnet**, daß die fremden Programm-Module (10) in einer Maschinensprache vorliegen.

4. Verfahren nach einem der vorhergehenden Ansprüche, dadurch **gekennzeichnet**, daß die Namen der internen Programm-Module (12) durch Anfügen einer in der verwendeten Hochsprache als Namensbestandteil nicht zugelassenen Zeichenfolge (&R) an die Namen der den internen Programm-Modulen zugrundeliegenden Quelltext-Module (18) erzeugt werden.

5. Verfahren nach einem der vorhergehenden Ansprüche, dadurch **gekennzeichnet**, daß für den Fall, daß das aufrufende Programm-Modul ein anderes Programm-Modul nicht über dessen Namen, sondern über dessen Adresse aufruft, vorausgesetzt wird, daß der Name des aufgerufenen Programm-Moduls der Fremd-Namensklasse zugeordnet ist,

- und daß durch die Anpassungsroutinen erster Art der aufgerufenen internen Programm-Module (12) festgestellt wird, welchem Vereinbarungssatz das jeweilige aufrufende Programm-Modul unterliegt, und nur bei voneinander abweichenden Vereinbarungssätzen des aufrufenden und aufgerufenen Programm-Moduls die Anpassung der Vereinbarungssätze vorgenommen wird.
- 5
6. Verfahren nach Anspruch 5, dadurch **gekennzeichnet**, daß die Anpassungsroutinen (16) erster Art jeweils mit einem Kennzeichnungscode versehen werden, durch den diese und die fremden Programm-Module (10) voneinander zu unterscheiden sind.
- 10
7. Verfahren nach Anspruch 6, dadurch **gekennzeichnet**, daß bei einem indirekten Aufruf geprüft wird, ob unter der Adresse des aufgerufenen Programm-Moduls der Kennzeichnungscode abgelegt ist und eine Anpassung der voneinander abweichenden Vereinbarungssätze von aufrufendem und aufgerufenem Programm-Modul genau dann vorgenommen wird, falls kein Kennzeichnungscode unter dieser Adresse abgelegt ist.
- 15
- 20
8. Verfahren nach Anspruch 7, dadurch **gekennzeichnet**, daß ein in Maschinensprache vorliegendes, dem effizienten Vereinbarungssatz unterliegendes Standard-Modul (30) durch den Kennzeichnungscode ergänzt wird und neben dem ursprünglichen Namen aus der internen Namensklasse einen weiteren Namen aus der Fremd-Namensklasse erhält.
- 25

FIG 1

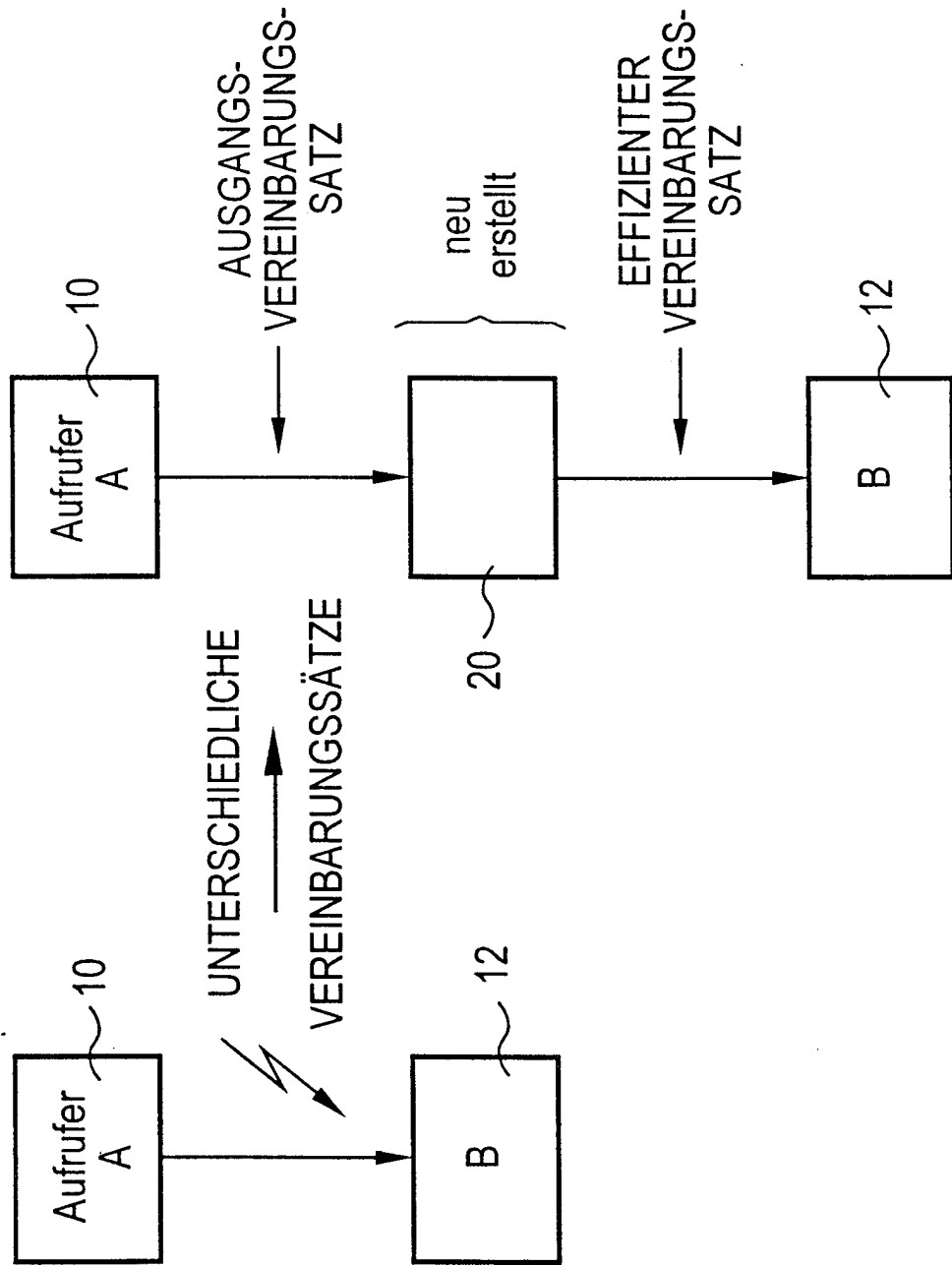
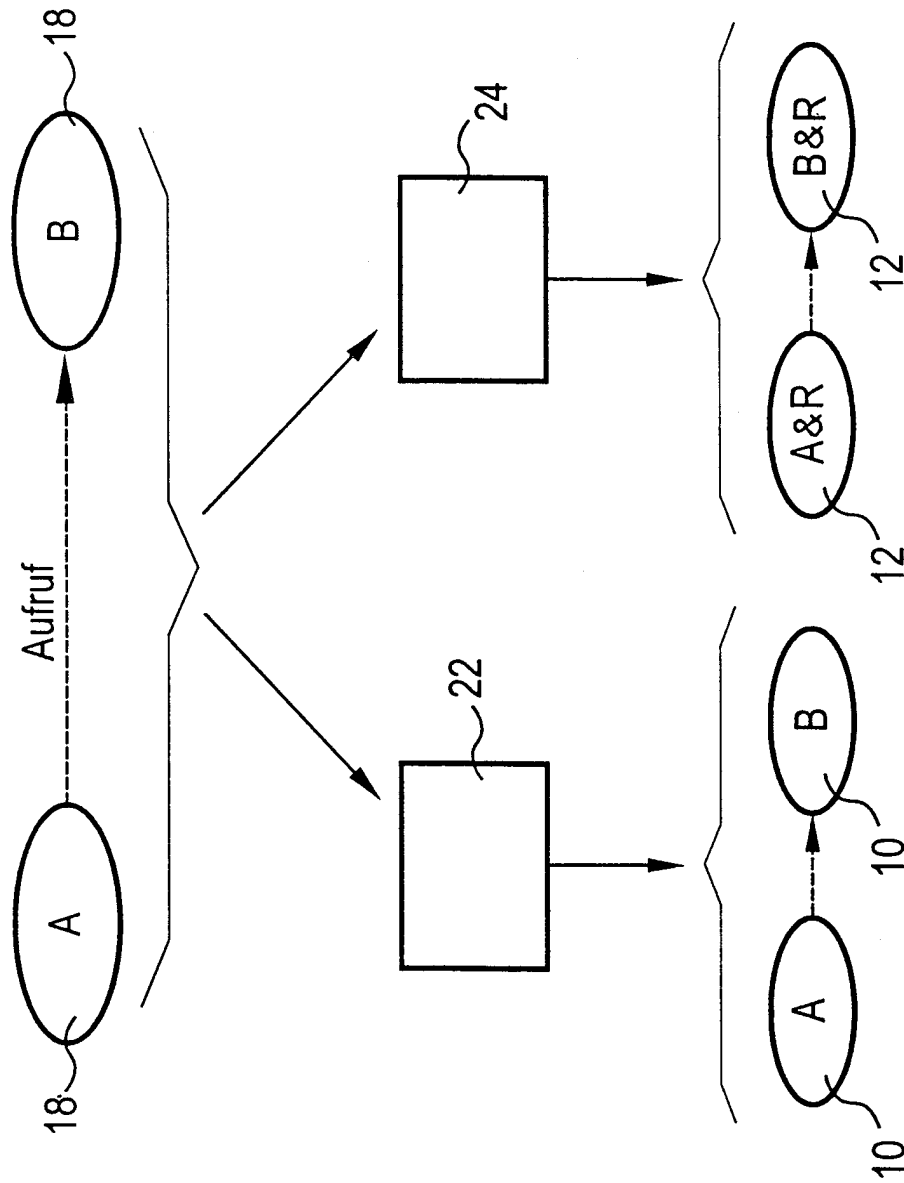




FIG 2



3/9

FIG 3

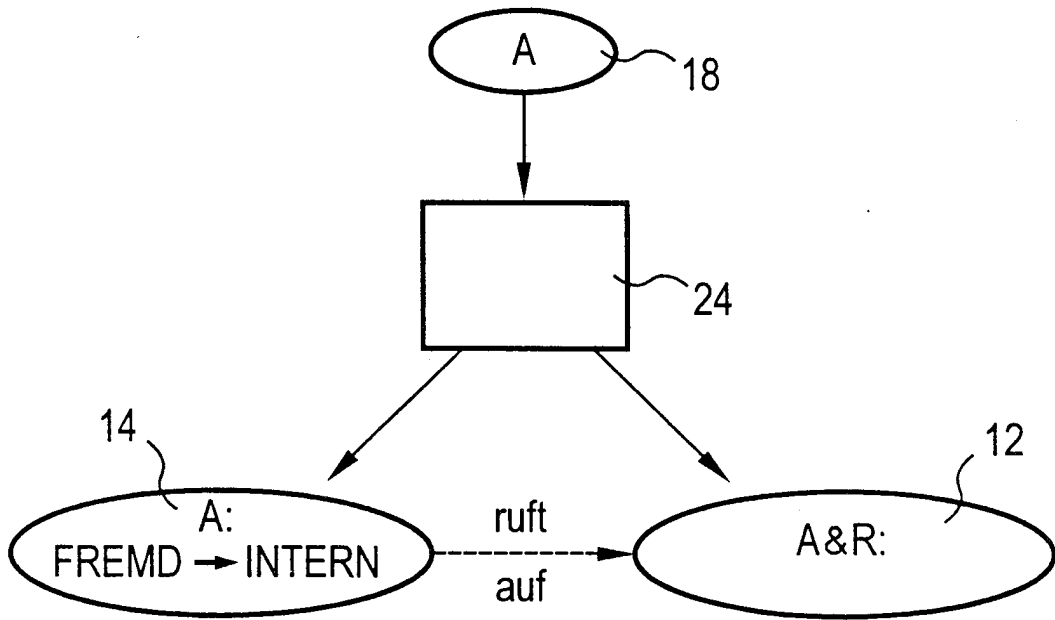


FIG 5

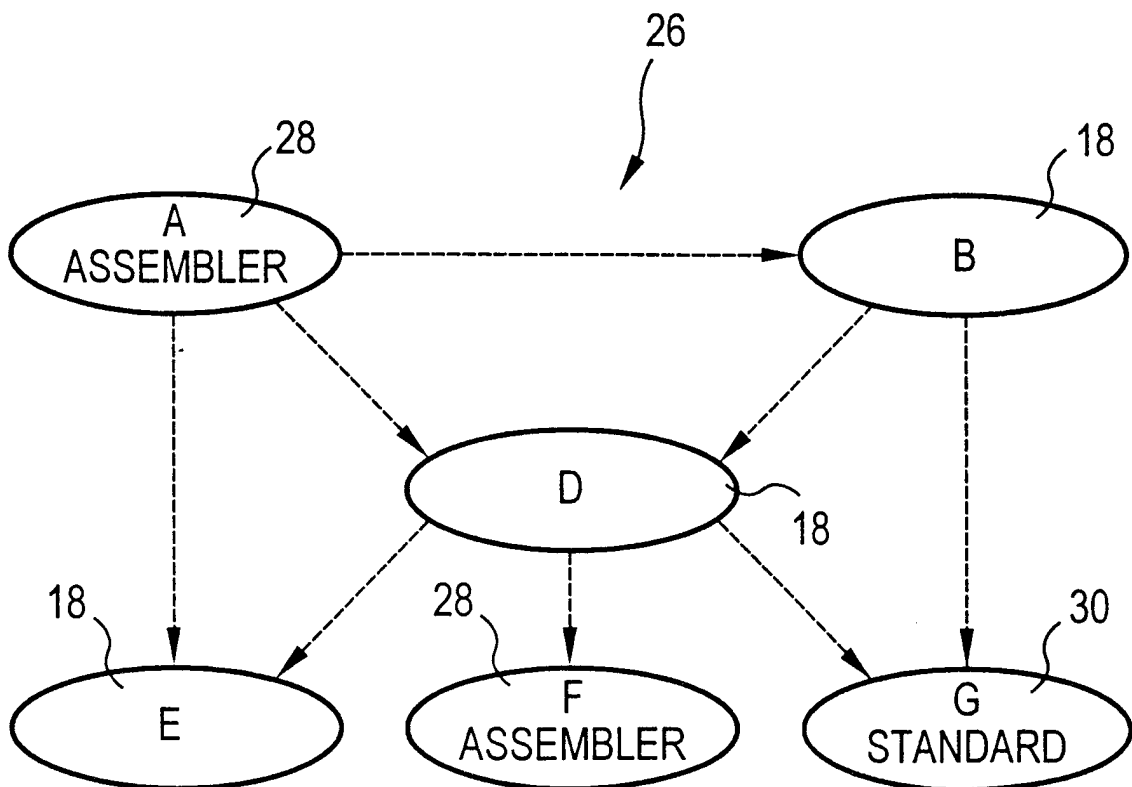


FIG 4

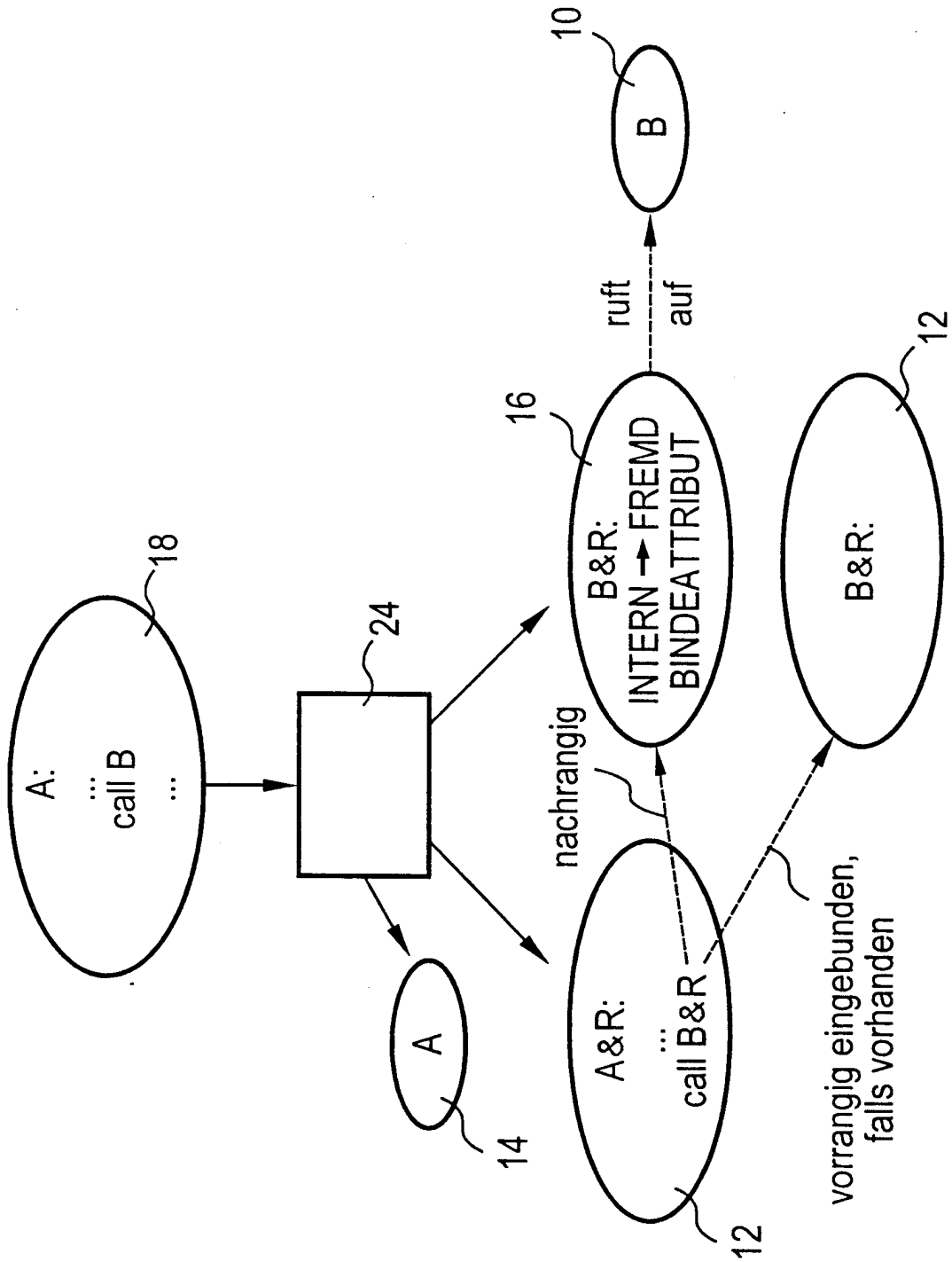


FIG 6

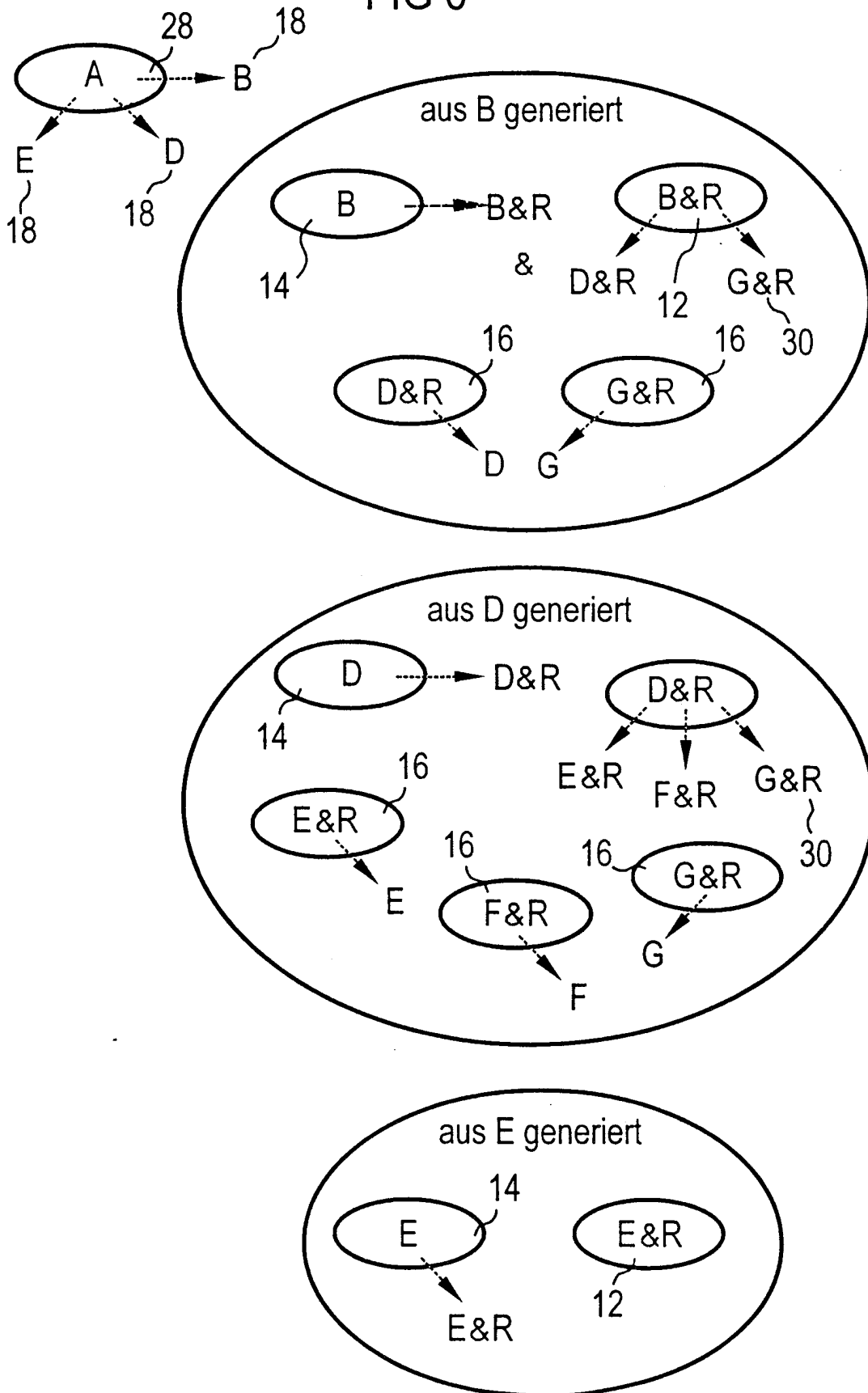


FIG 7

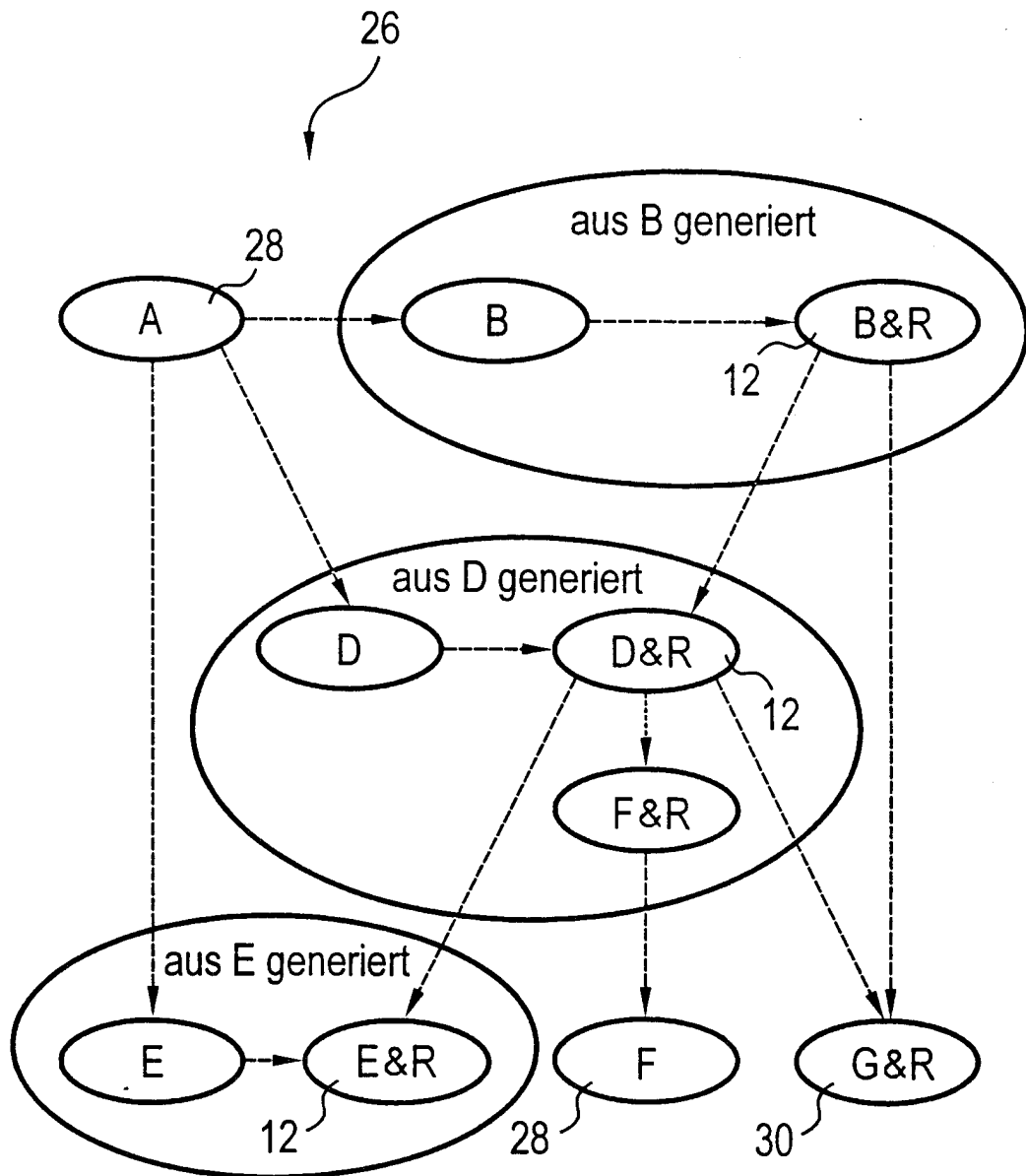


FIG 8

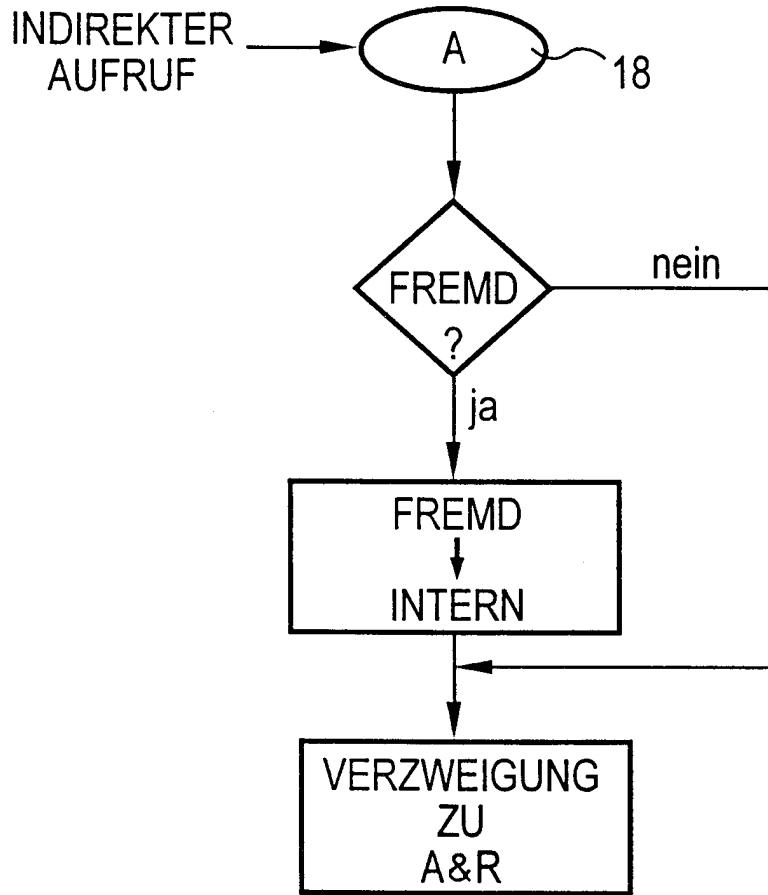
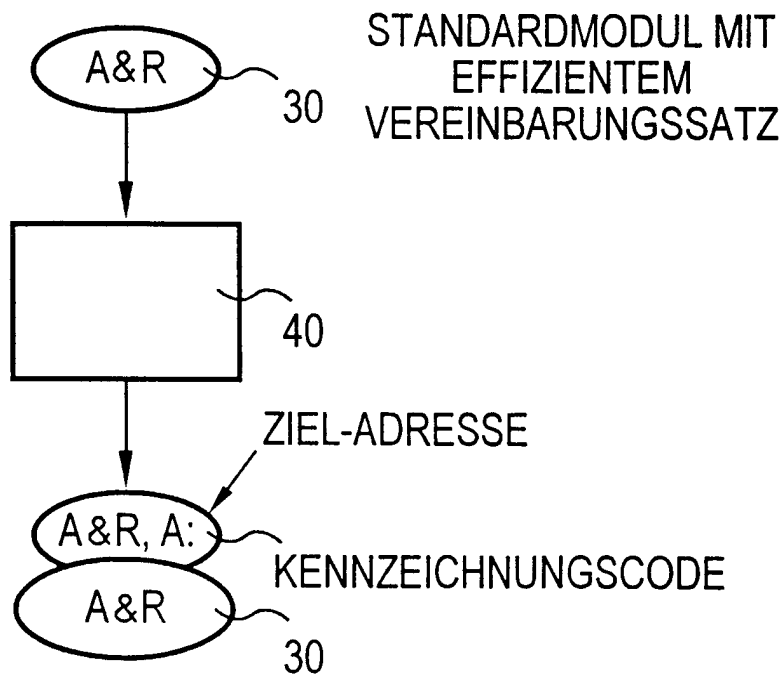


FIG 10



8/9

FIG 9

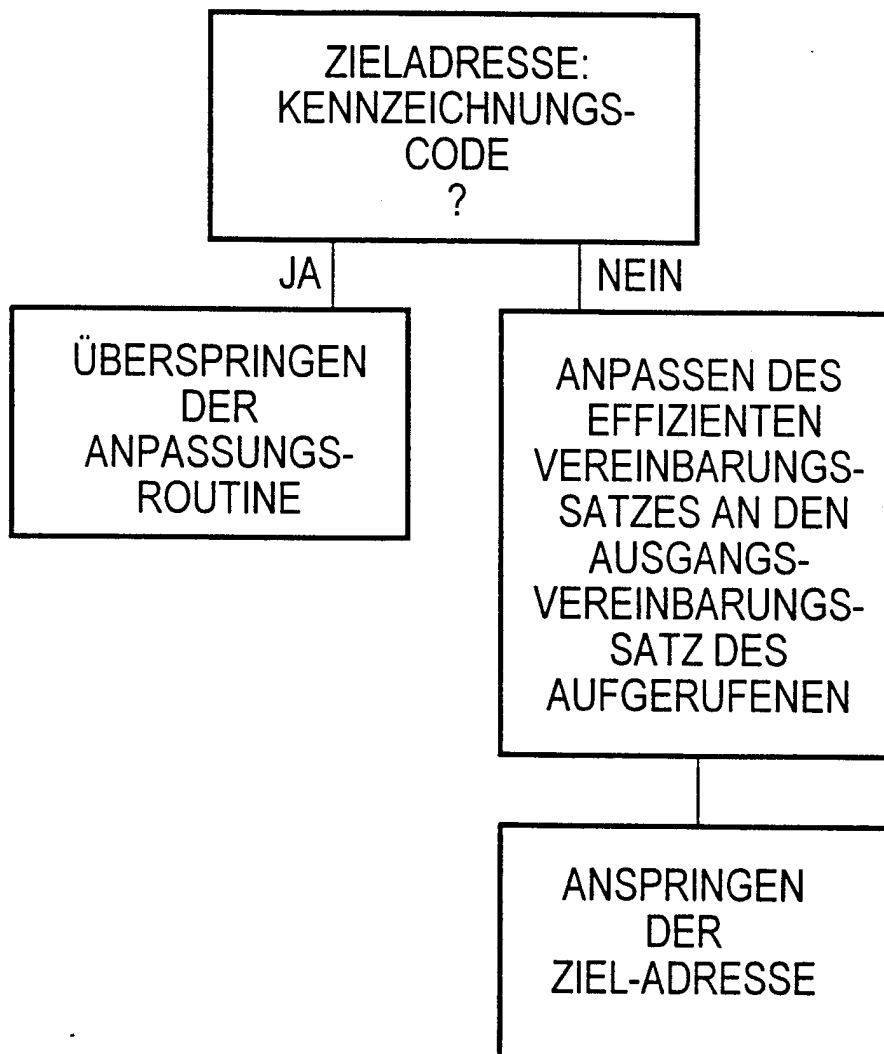
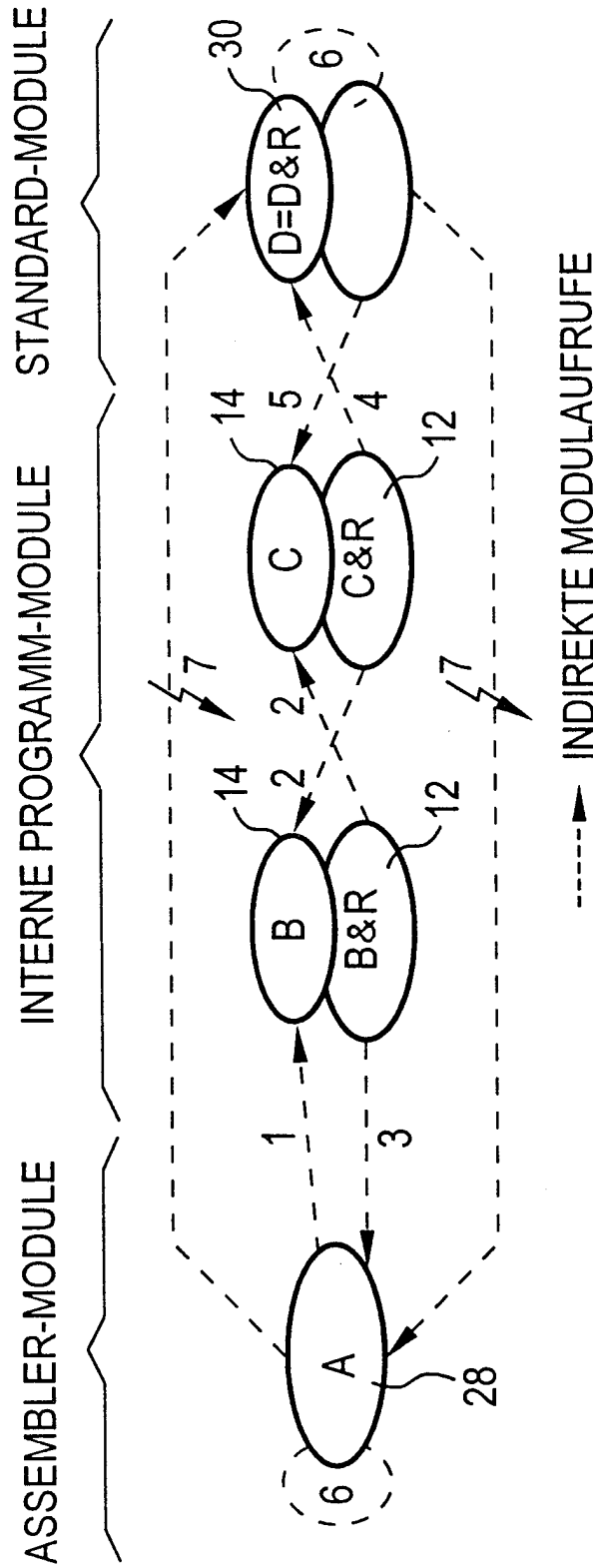


FIG 11





# INTERNATIONAL SEARCH REPORT

International Application No

PCT/DE 98/01389

**A. CLASSIFICATION OF SUBJECT MATTER**

IPC 6 G06F9/40 G06F9/42 G06F9/445

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)

IPC 6 G06F

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)

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 92 15962 A (DIGITAL EQUIPMENT CORP) 17 September 1992 see abstract see page 1, line 18 - page 2, line 23 see page 3, line 12 - line 26 see page 15, line 1 - page 18, line 30 see page 26, line 39 - page 28, line 12 see page 28, line 23 - page 31, line 15 see page 33, line 1 - page 42, line 37 see claims; figures ---	1-8
A	WO 92 15936 A (DIGITAL EQUIPMENT CORP) 17 September 1992 see the whole document ---	1-8

Further documents are listed in the continuation of box C.

Patent family members are listed in 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

5 November 1998

Date of mailing of the international search report

12/11/1998

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
 NL - 2280 HV Rijswijk  
 Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
 Fax: (+31-70) 340-3016

Authorized officer

Wiltink, J

1

INTERNATIONAL SEARCH REPORT

International Application No

PCT/DE 98/01389

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 581 769 A (WALLACE WILLIAM D ET AL) 3 December 1996 see abstract see column 4, line 64 - column 6, line 13 -----	1-8

## INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/DE 98/01389

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9215962 A	17-09-1992	US 5339422 A	16-08-1994
		AU 658413 B	13-04-1995
		AU 1644292 A	06-10-1992
		CA 2082409 A, C	08-09-1992
		DE 69226484 D	10-09-1998
		EP 0530350 A	10-03-1993
		IL 100996 A	19-01-1996
		JP 7069835 B	31-07-1995
		JP 6502736 T	24-03-1994
		KR 9603137 B	05-03-1996
		MX 9200940 A	01-03-1993
WO 9215936 A	17-09-1992	AU 655983 B	19-01-1995
		AU 1587992 A	06-10-1992
		CA 2082068 A	08-09-1992
		DE 69225543 D	25-06-1998
		EP 0528020 A	24-02-1993
		IL 100994 A	08-12-1995
		JP 5505274 T	05-08-1993
		KR 9601642 B	03-02-1996
		MX 9200937 A	01-03-1993
		US 5764947 A	09-06-1998
		US 5581769 A	03-12-1996

<b>A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES</b> IPK 6 G06F9/40 G06F9/42 G06F9/445		
Nach der Internationalen Patentklassifikation (IPK) oder nach der nationalen Klassifikation und der IPK		
<b>B. RECHERCHIERTE GEBIETE</b>		
Recherchierter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole ) IPK 6 G06F		
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)		
<b>C. ALS WESENTLICH ANGESEHENE UNTERLAGEN</b>		
Kategorie <sup>o</sup>	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
X	WO 92 15962 A (DIGITAL EQUIPMENT CORP) 17. September 1992 siehe Zusammenfassung siehe Seite 1, Zeile 18 - Seite 2, Zeile 23 siehe Seite 3, Zeile 12 - Zeile 26 siehe Seite 15, Zeile 1 - Seite 18, Zeile 30 siehe Seite 26, Zeile 39 - Seite 28, Zeile 12 siehe Seite 28, Zeile 23 - Seite 31, Zeile 15 siehe Seite 33, Zeile 1 - Seite 42, Zeile 37 siehe Ansprüche; Abbildungen --- -/--	1-8
<input checked="" type="checkbox"/> Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen		
<input checked="" type="checkbox"/> Siehe Anhang Patentfamilie		
<sup>o</sup> 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 5. November 1998		Absendedatum des internationalen Recherchenberichts 12/11/1998
Name und Postanschrift der Internationalen Recherchenbehörde Europäisches Patentamt, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016		Bevollmächtigter Bediensteter Wiltink, J

## C.(Fortsetzung) ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
A	WO 92 15936 A (DIGITAL EQUIPMENT CORP) 17. September 1992 siehe das ganze Dokument -----	1-8
A	US 5 581 769 A (WALLACE WILLIAM D ET AL) 3. Dezember 1996 siehe Zusammenfassung siehe Spalte 4, Zeile 64 - Spalte 6, Zeile 13 -----	1-8

# INTERNATIONALER RECHERCHENBERICHT

Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören

Internationales Aktenzeichen

PCT/DE 98/01389

Im Recherchenbericht angeführtes Patentdokument	Datum der Veröffentlichung	Mitglied(er) der Patentfamilie	Datum der Veröffentlichung
WO 9215962 A	17-09-1992	US 5339422 A	16-08-1994
		AU 658413 B	13-04-1995
		AU 1644292 A	06-10-1992
		CA 2082409 A,C	08-09-1992
		DE 69226484 D	10-09-1998
		EP 0530350 A	10-03-1993
		IL 100996 A	19-01-1996
		JP 7069835 B	31-07-1995
		JP 6502736 T	24-03-1994
		KR 9603137 B	05-03-1996
		MX 9200940 A	01-03-1993
		WO 9215936 A	17-09-1992
AU 1587992 A	06-10-1992		
CA 2082068 A	08-09-1992		
DE 69225543 D	25-06-1998		
EP 0528020 A	24-02-1993		
IL 100994 A	08-12-1995		
JP 5505274 T	05-08-1993		
KR 9601642 B	03-02-1996		
MX 9200937 A	01-03-1993		
US 5764947 A	09-06-1998		
US 5581769 A	03-12-1996	KEINE	