

364	468.09	Subclass
		Class
ISSUE CLASSIFICATION		



UTILITY SERIAL NUMBER	08/707187	PATENT DATE	OCT 20 1998	PATENT NUMBER	
-----------------------	-----------	-------------	-------------	---------------	--

SERIAL NUMBER	08/707,187	FILING DATE	09/03/96	CLASS	364	SUBCLASS	468.09	GROUP ART UNIT	2787	EXAMINER	LAO
---------------	------------	-------------	----------	-------	-----	----------	--------	----------------	------	----------	-----

APPLICANTS NEERAJ GUPTA, AUSTIN, TX; VENKY VEERARAGHAVAN, AUSTIN, TX; AJAY AGARWAL, AUSTIN, TX.

CONTINUING DATA***
VERIFIED

none / kll

FOREIGN/PCT APPLICATIONS***
VERIFIED

none / kll

FOREIGN FILING LICENSE GRANTED 11/30/96

Foreign priority claimed 35 USC 119 conditions met	<input type="checkbox"/> yes <input checked="" type="checkbox"/> no	AS FILED	STATE OR COUNTRY	SHEETS DRWGS.	TOTAL CLAIMS	INDEP. CLAIMS	FILING FEE RECEIVED	ATTORNEY'S DOCKET NO.
Verified and Acknowledged	Examiner's initials	→	TX	24	59	5	\$1,894.00	85160.915

ADDRESS HECKER & HARRIMAN
SUITE 1600
2029 CENTURY PARK EAST
LOS ANGELES CA 90067

TITLE METHOD AND APPARATUS FOR MAINTAINING AND CONFIGURING SYSTEMS
U.S. DEPT. OF COMM./ PAT. & TM—PTO-436L (Rev.12-94)

PARTS OF APPLICATION FILED SEPARATELY		Applications Examiner	
NOTICE OF ALLOWANCE MAILED	Assistant Examiner	CLAIMS ALLOWED	
5-22-98 ✓		Total Claims	Print Claim
		85	1
ISSUE FEE		DRAWING	
Amount Due	Date Paid	Sheets Drwg.	Figs. Drwg.
1320.00	1-10-98	24	24
Label Area		Print Fig.	6
		ISSUE BATCH NUMBER	
		E36	
PREPARED FOR ISSUE			
WARNING: The information disclosed herein may be restricted. Unauthorized disclosure may be prohibited by the United States Code Title 35, Sections 122, 181 and 368. Possession outside the U.S. Patent & Trademark Office is restricted to authorized employees and contractors only.			

08/707187

APPROVED FOR LICENSE ☐

PATENT APPLICATION

INITIALS DCI 109634



08707187

CONTENTS

Date
Entered
or
Counted

Date
Received
or
Mailed

1. Application 24 SHEETS papers.

2. LR RE: DEC.

10-18

3. Declaration Seuchy

11-4-96

12/22 4. Ref 3 mos

1/6/98

5. IDS

4/17/98

5-15-98. 6. Req. for Ext. of Time (1) / Amrt. A

5-11-98. (of 5-6-98.)

5-21-98= 7. NOTICE of ALLOW

5-22-98

8/17/98

8. Formal Drawings 24 (shs) sub 1

8/3/98

9. PTO GRANT OCT 20 1998

10.

11.

12.

13.

14.

15.

16.

17.

18.

19.

20.

21.

22.

23.

24.

25.

26.

27.

28.

29.

30.

31.

32.

Subclass
Class
707
MBER
187
GUP
TX
INVI
TED
SIGN/
TED
Luk
IN FI
ity claims
condition:
Acknowle
ER &
E 160
CENT
ANGEL
ID AP
APPLIC.
RATEL
ALLOV
ISSUE I
Labe
Area

PATENT NUMBER		ORIGINAL CLASSIFICATION	
		CLASS	SUBCLASS
		364	468.09
APPLICATION/SERIAL NUMBER		CROSS REFERENCE(S)	
08/707,187			
APPLICANT'S NAME (PLEASE PRINT)		CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)
GUPTA et al.		364	468
IF REISSUE, ORIGINAL PATENT NUMBER			
INTERNATIONAL CLASSIFICATION		GROUP ART UNIT	
G06F		2787	
01/00		ASSISTANT EXAMINER (PLEASE STAMP OR PRINT FULL NAME)	
		PRIMARY EXAMINER (PLEASE STAMP OR PRINT FULL NAME)	
		KEVIN A. KRIESS	

PTO 270 (REV. 5-91) ISSUE CLASSIFICATION SLIP PRIMARY EXAMINER DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

INDEX OF CLAIMS

Claim	Date
Final	Original
1	12-31-97
2	11-5-97
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	
48	
49	
50	

- SYMBOLS
- ✓ Rejected
 - = Allowed
 - (Through numeral) Canceled
 - + Restricted
 - N Non-elected
 - I Interference
 - A Appeal
 - O Objected

Claim	Date
Final	Original
51	12-31-97
52	11-5-97
53	
54	
55	
56	
57	
58	
59	
60	
61	
62	
63	
64	
65	
66	
67	
68	
69	
70	
71	
72	
73	
74	
75	
76	
77	
78	
79	
80	
81	
82	
83	
84	
85	
86	
87	
88	
89	
90	
91	
92	
93	
94	
95	
96	
97	
98	
99	
100	

SEARCHED

Class	Sub.	Date	Exmr.

SEARCH NOTES

	Date	Exmr.
--	------	-------

INTERFERENCE SEARCHED

Class	Sub.	Date	Exmr.
-------	------	------	-------



US005825651A

United States Patent [19]

Gupta et al.

[11] Patent Number: **5,825,651**
[45] Date of Patent: **Oct. 20, 1998**

[54] **METHOD AND APPARATUS FOR
MAINTAINING AND CONFIGURING
SYSTEMS**

5,355,317 10/1994 Talbott et al. 364/468
5,357,440 10/1994 Talbott et al. 364/467
5,586,052 12/1996 Iannuzzi et al. 364/512
5,659,478 8/1997 Pennisi et al. 364/468.01

[75] Inventors: **Neeraj Gupta; Venky Veeraraghavan;
Ajay Agarwal**, all of Austin, Tex.

Primary Examiner—Kevin A. Kriess
Attorney, Agent, or Firm—Hecker & Harriman

[73] Assignee: **Trilogy Development Group, Inc.**,
Austin, Tex.

[57] ABSTRACT

The invention provides the ability to interactively select and configure a product among a set of related products based on availability and compatibility of features and options. It does not impose an order in the selection of products, features or options; only valid selections can be made at any time. To create an electronic representation of the product information to achieve the above goal, the invention provides a framework for defining a systems by defining the components of the system using elements contained in a parts catalog and defining relationships between the components of a system. A configuration system validates a configuration using the system definition, the current state of the configuration and user input.

[21] Appl. No.: **707,187**

[22] Filed: **Sep. 3, 1996**

[51] Int. Cl.⁶ **G06F 1/00**

[52] U.S. Cl. **364/468.09**; 364/488

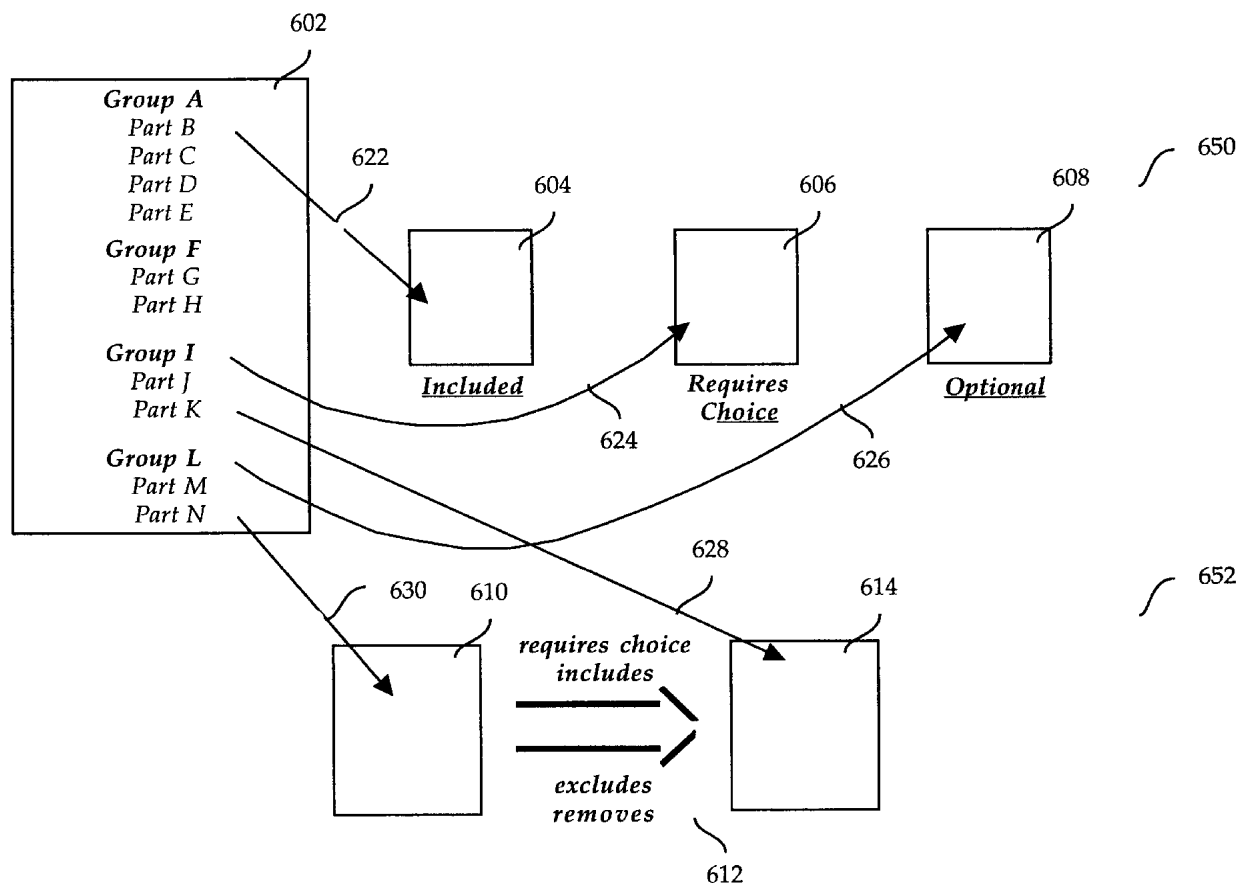
[58] Field of Search 364/468.09, 468.02,
364/468.03, 468.1, 488; 395/651

[56] References Cited

U.S. PATENT DOCUMENTS

4,796,194 1/1989 Atherton .
5,019,961 5/1991 Addesso et al. 364/192
5,019,992 5/1991 Brown et al. 364/468

85 Claims, 24 Drawing Sheets



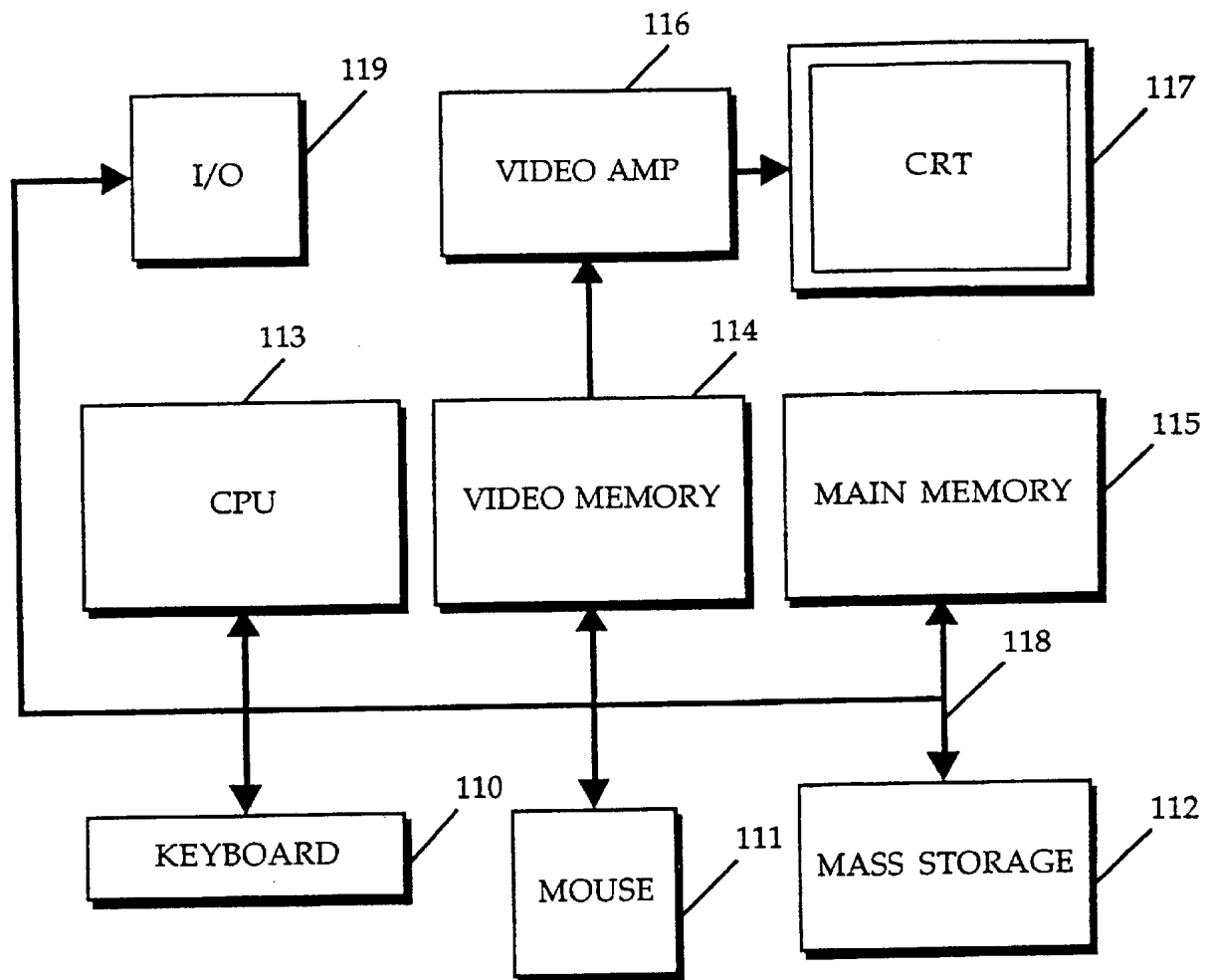


FIG. 1

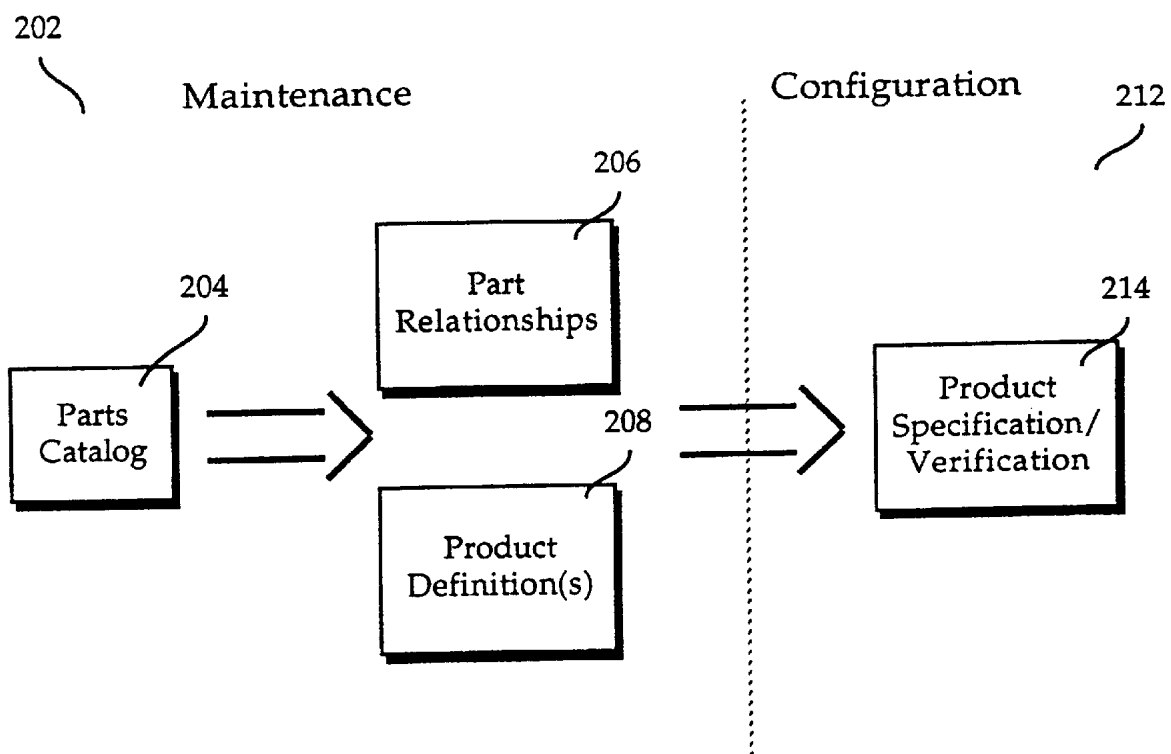


FIG. 2

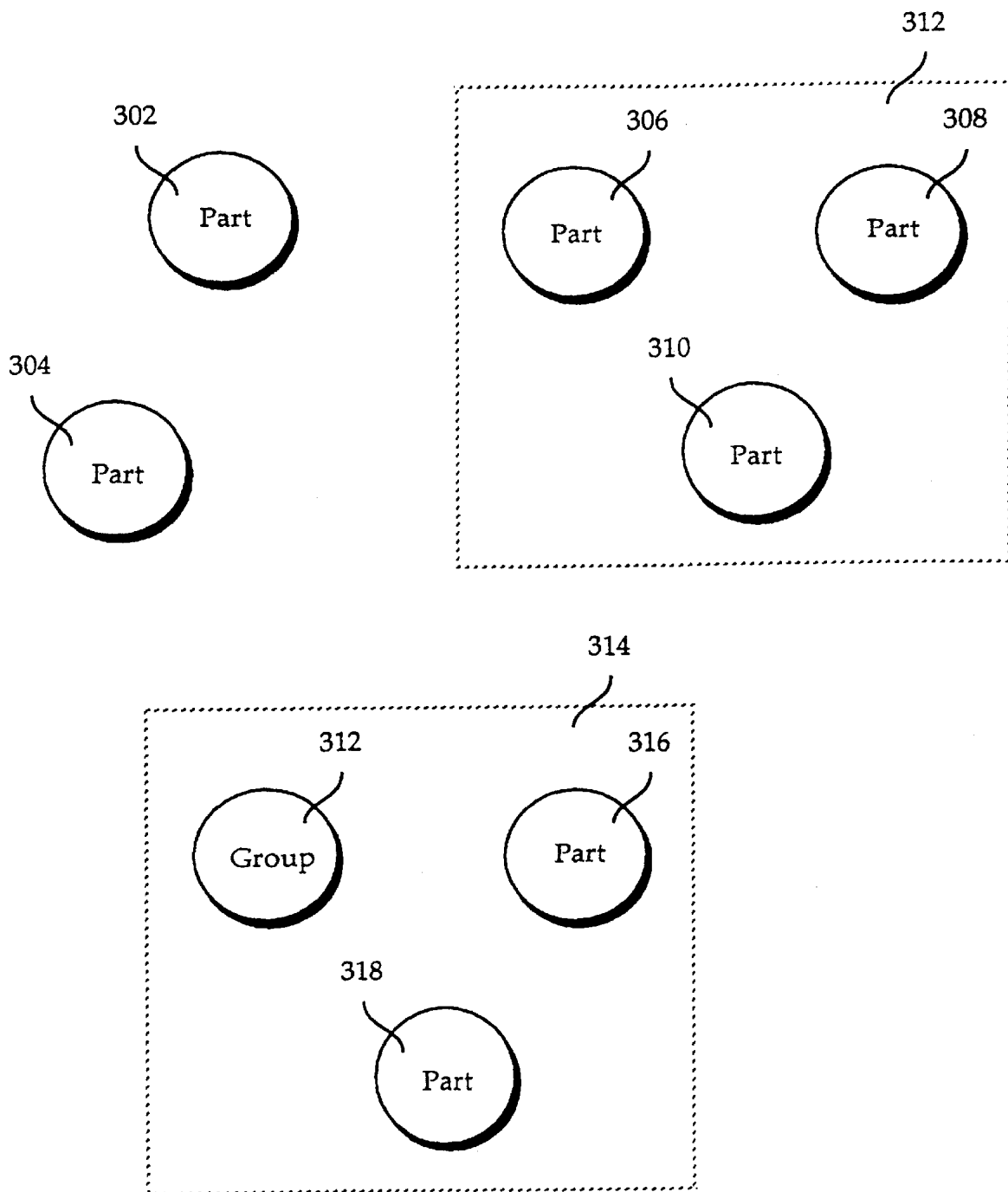


FIG. 3

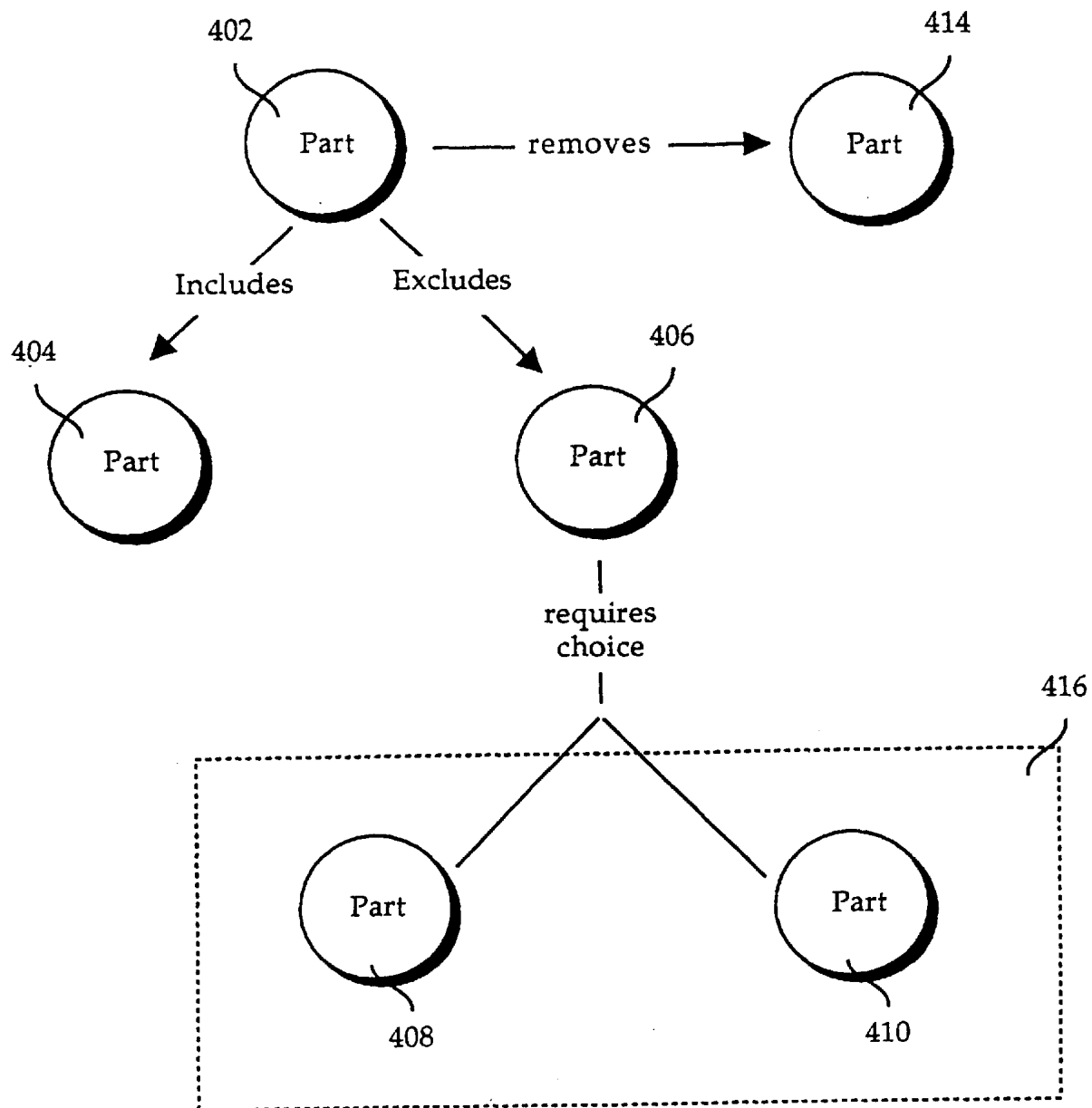


FIG. 4

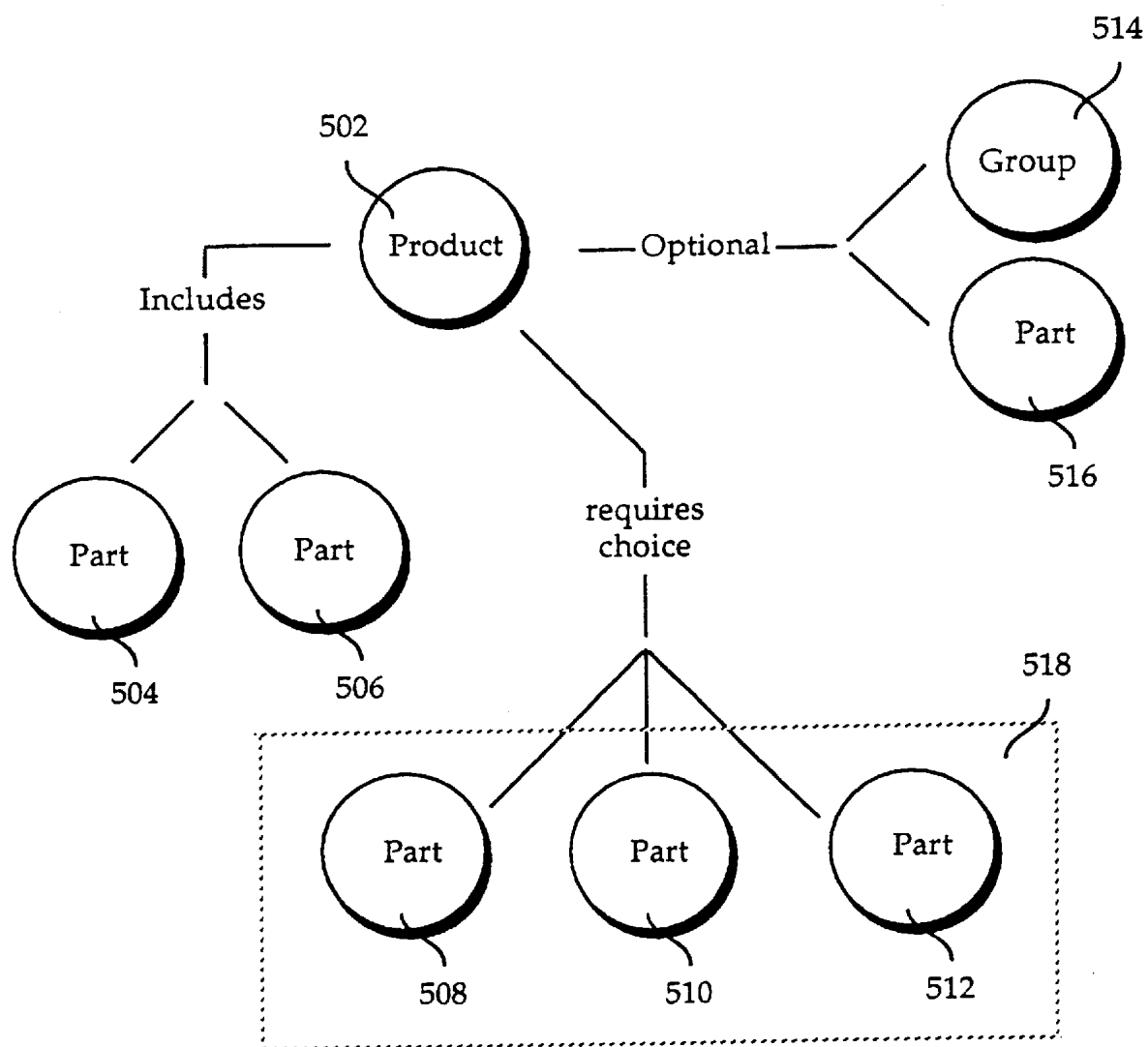


FIG. 5

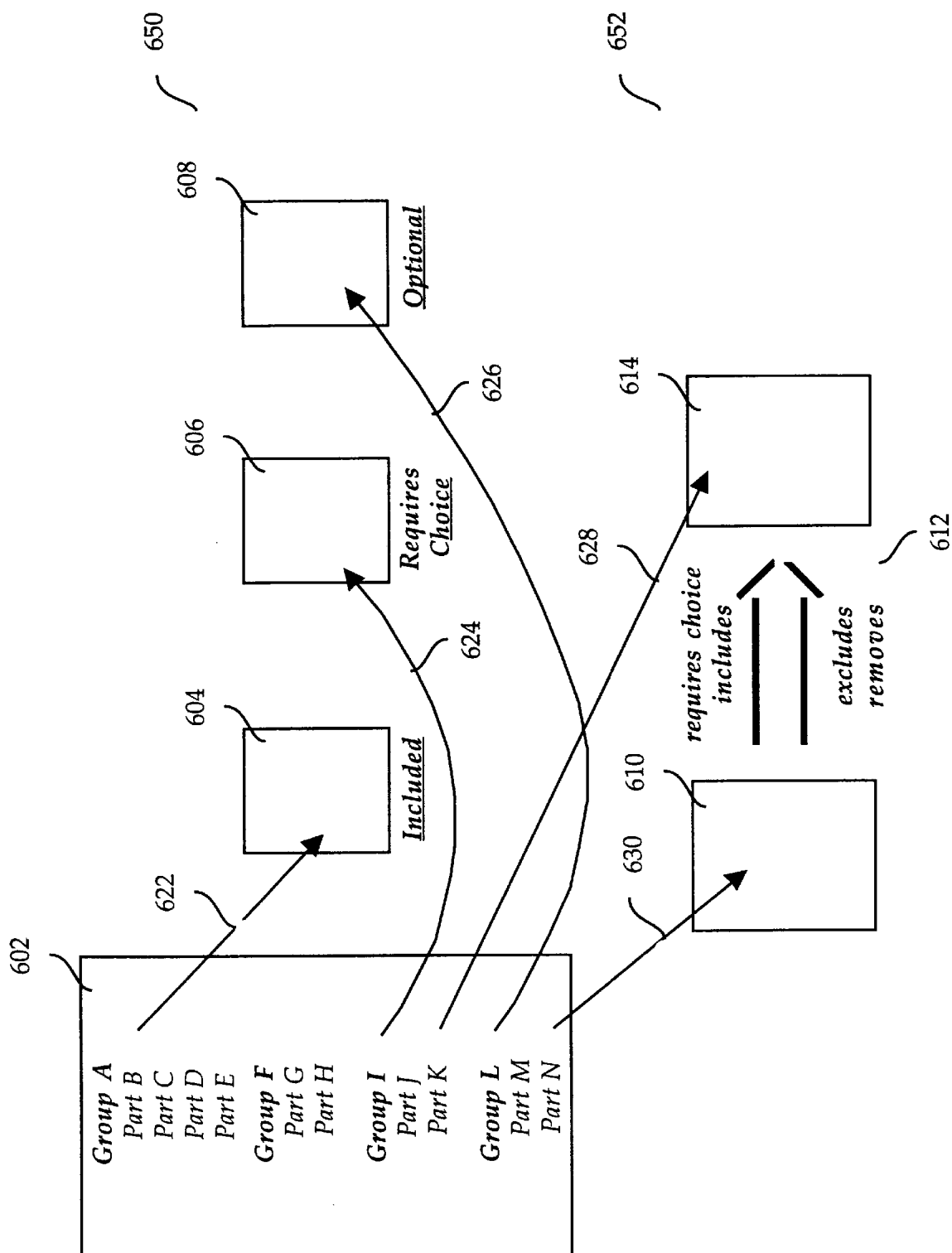


FIG. 6

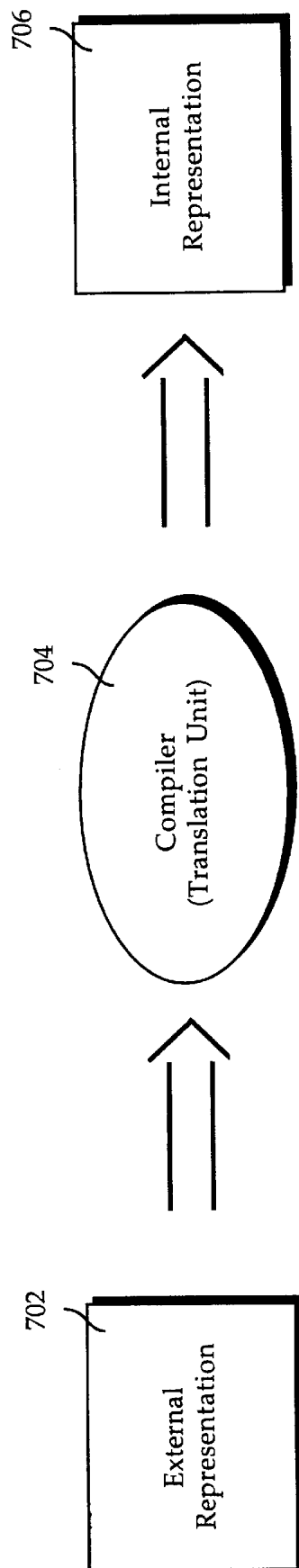


Figure 7

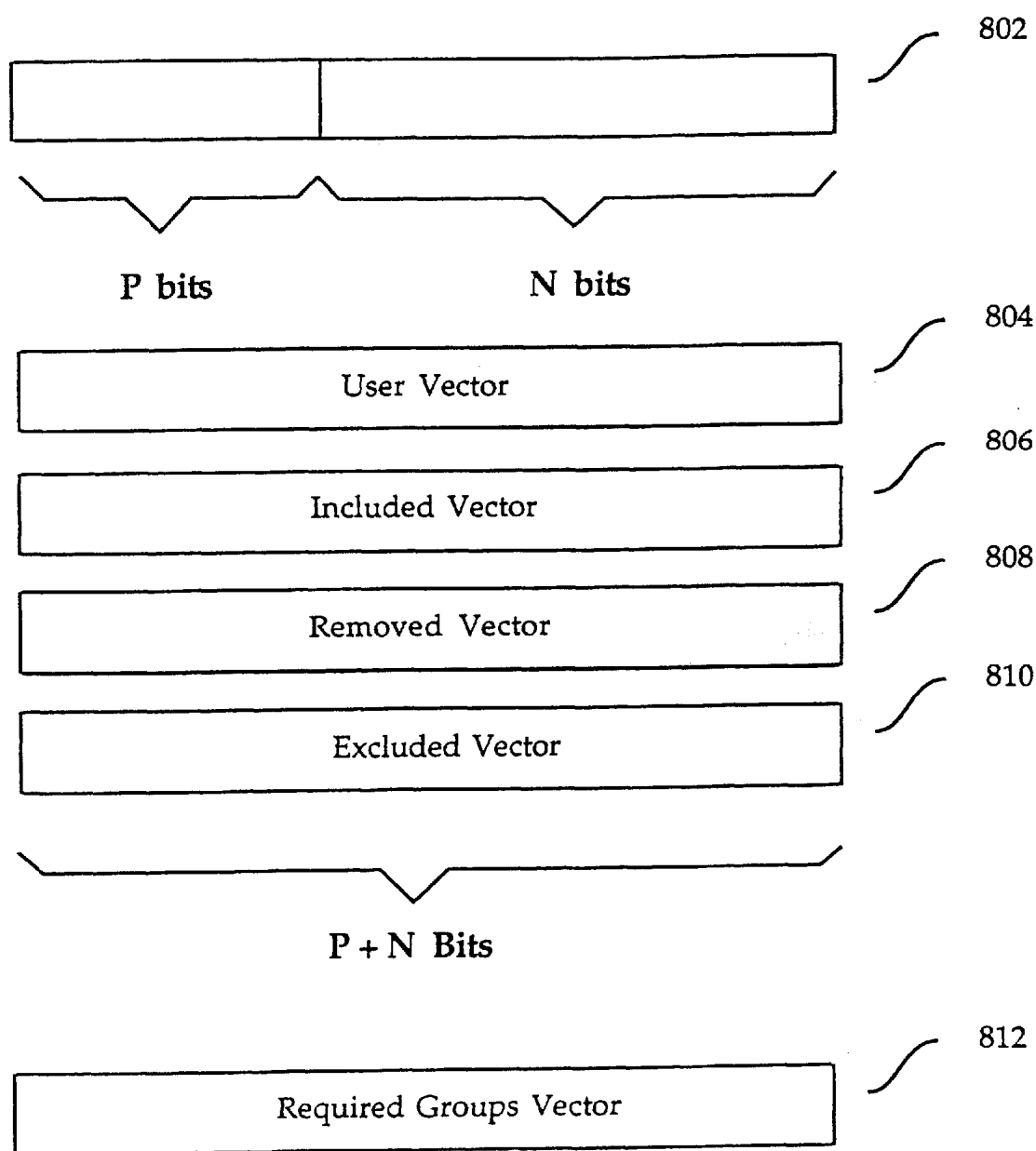


FIG. 8A

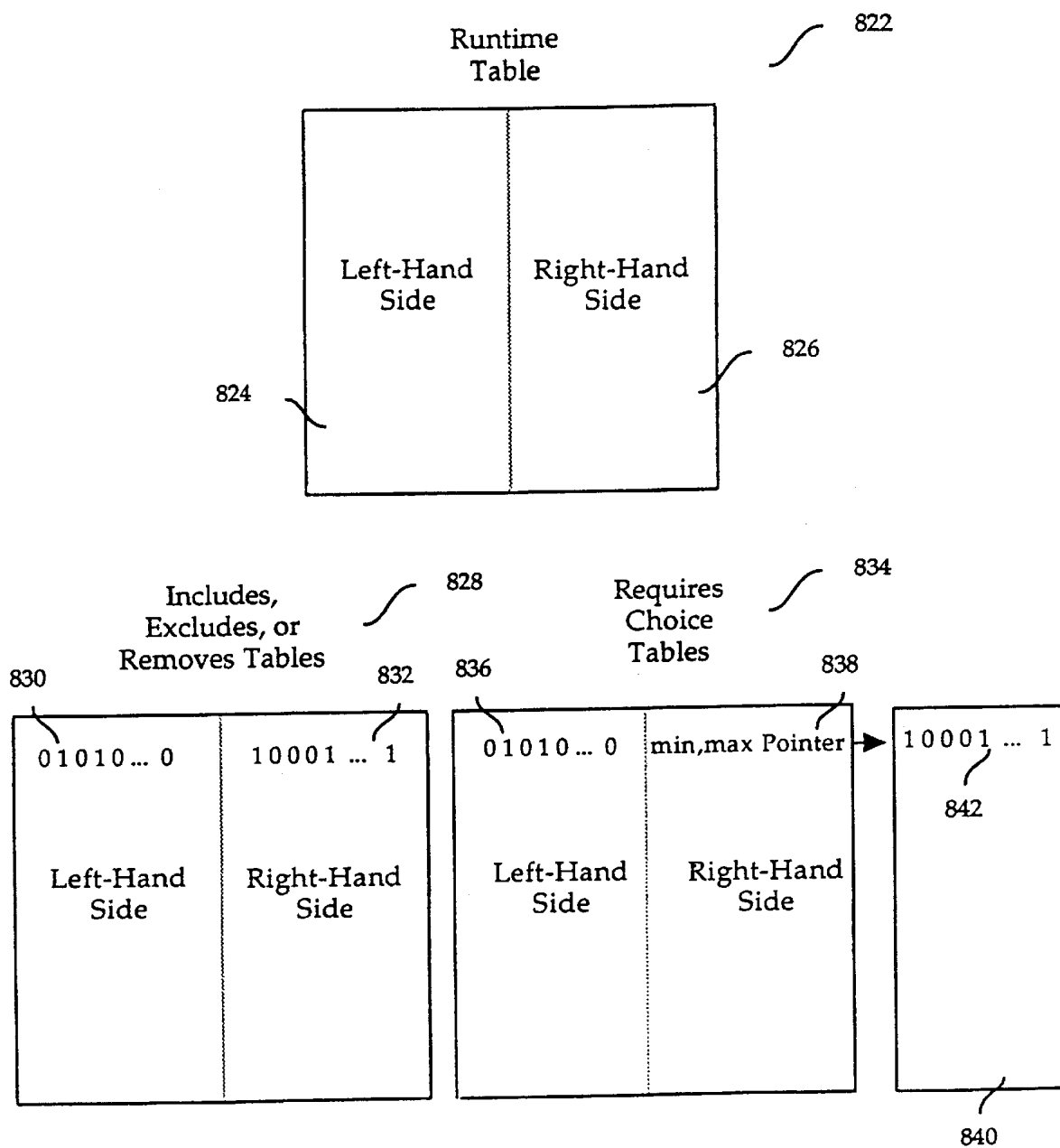


FIG. 8B

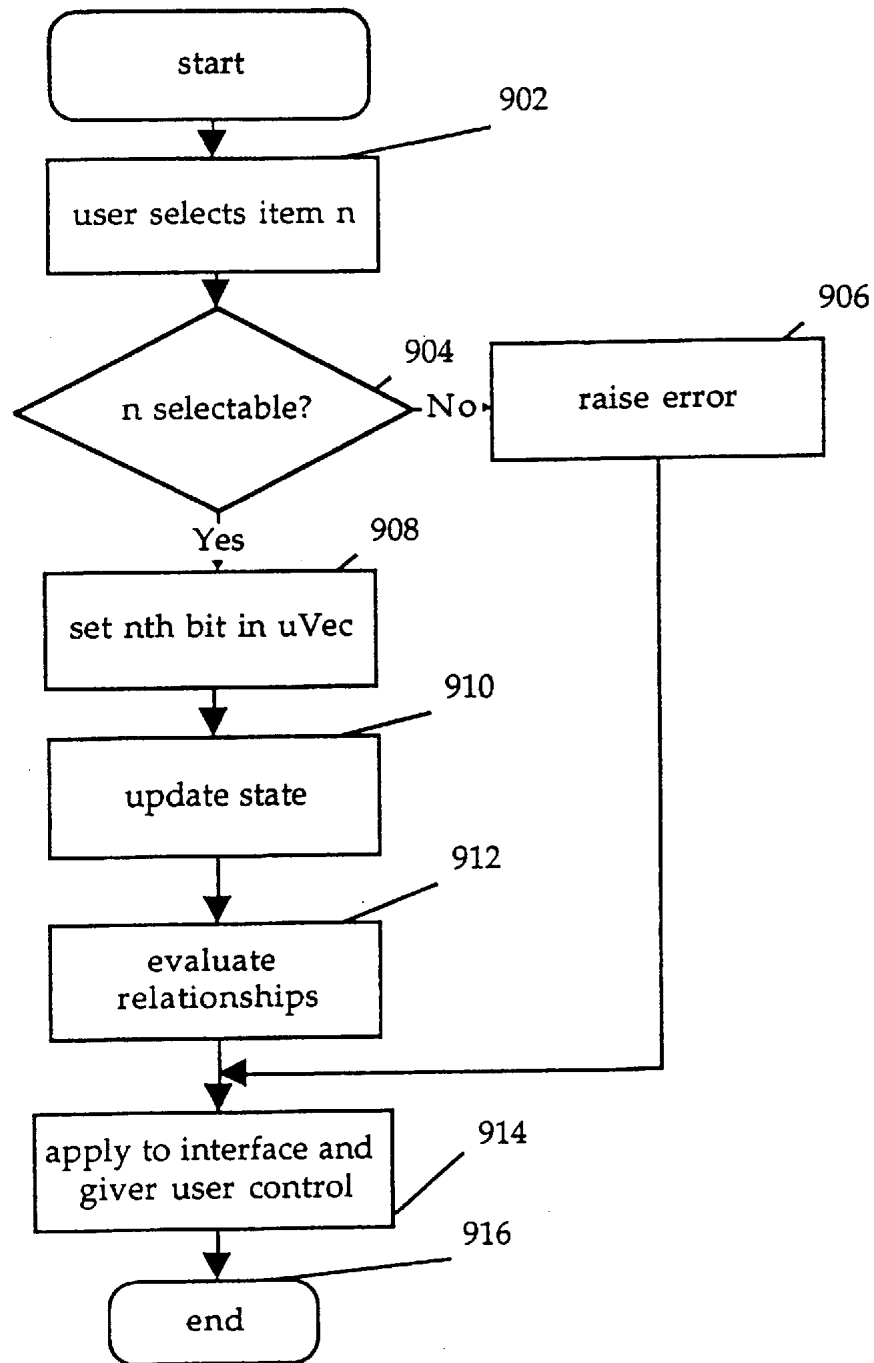


FIG. 9

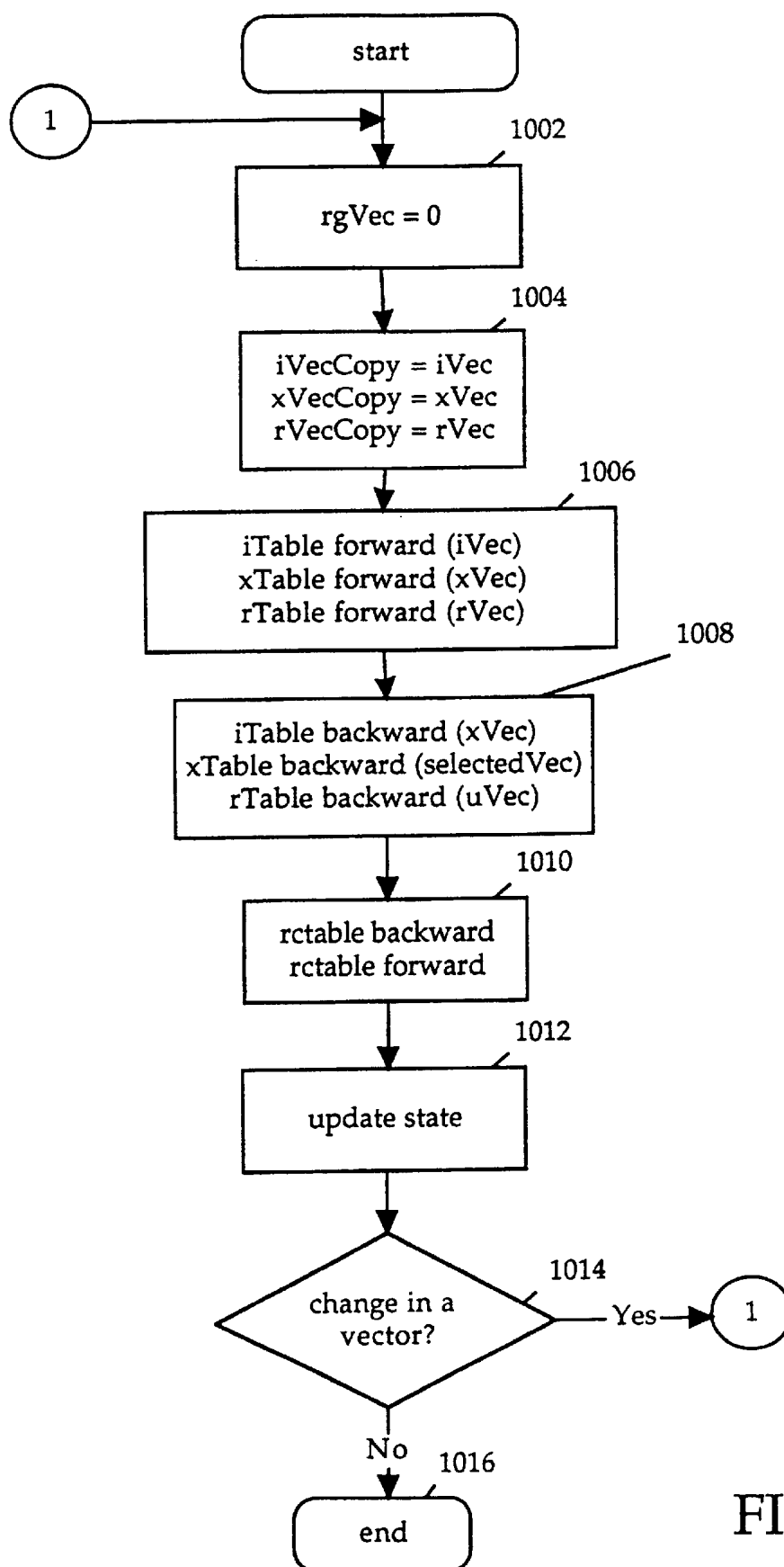


FIG. 10

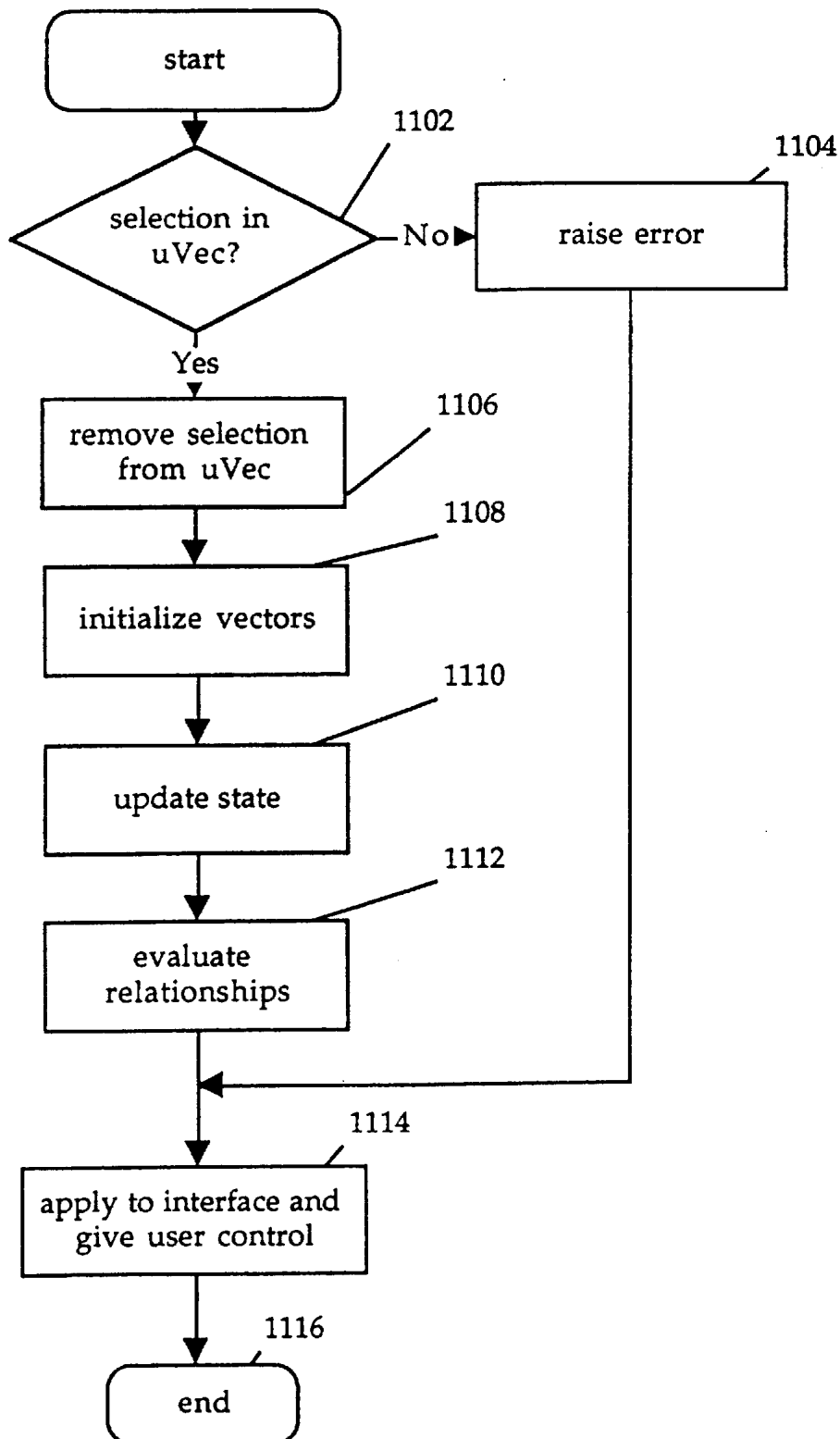


FIG. 11

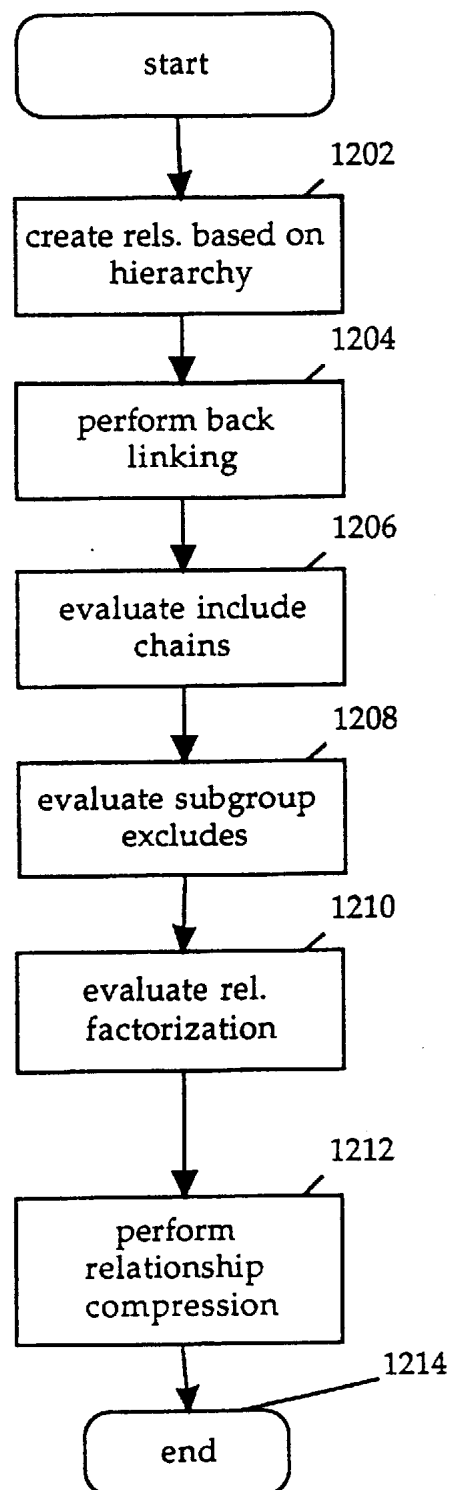


FIG. 12

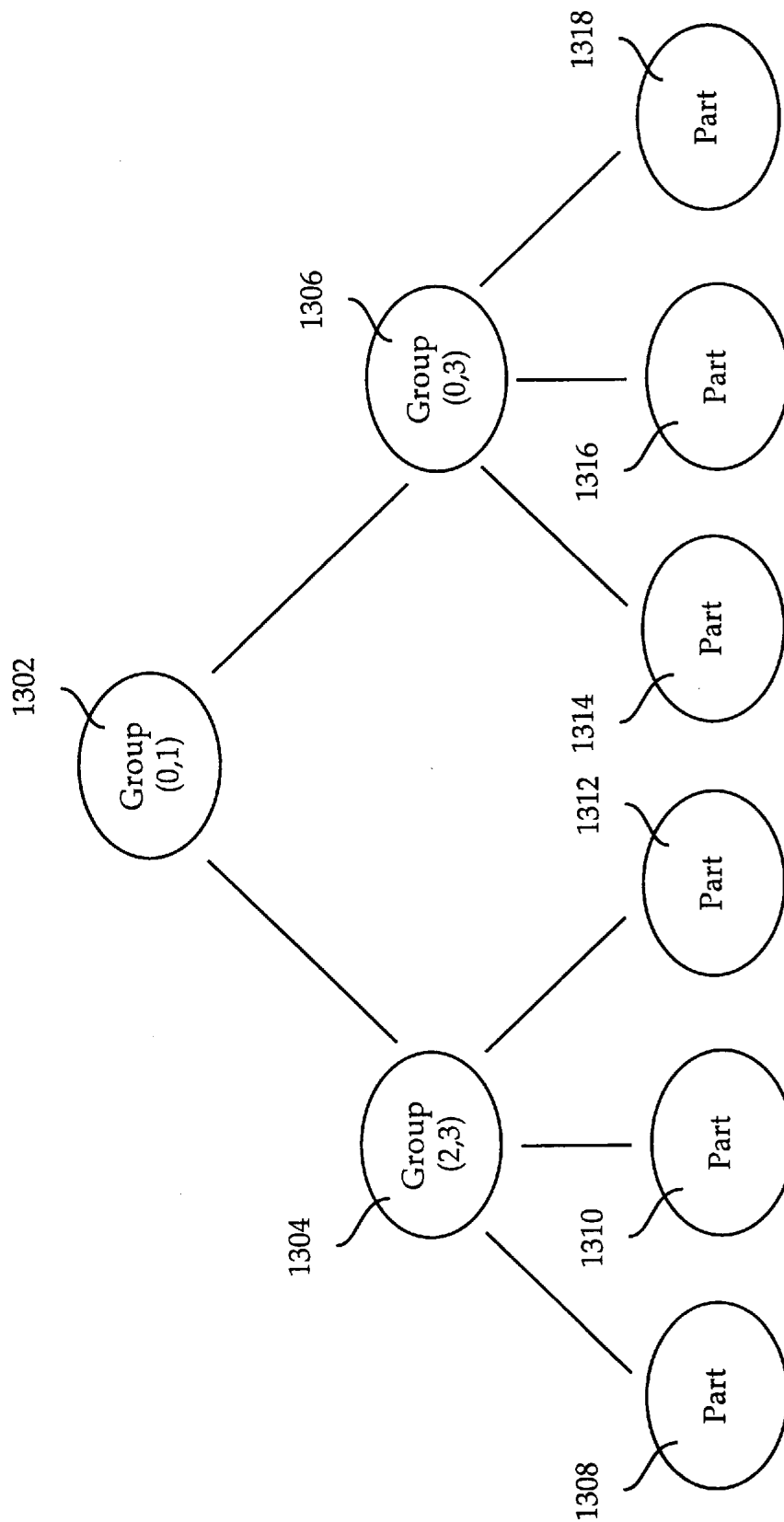


FIG. 13A

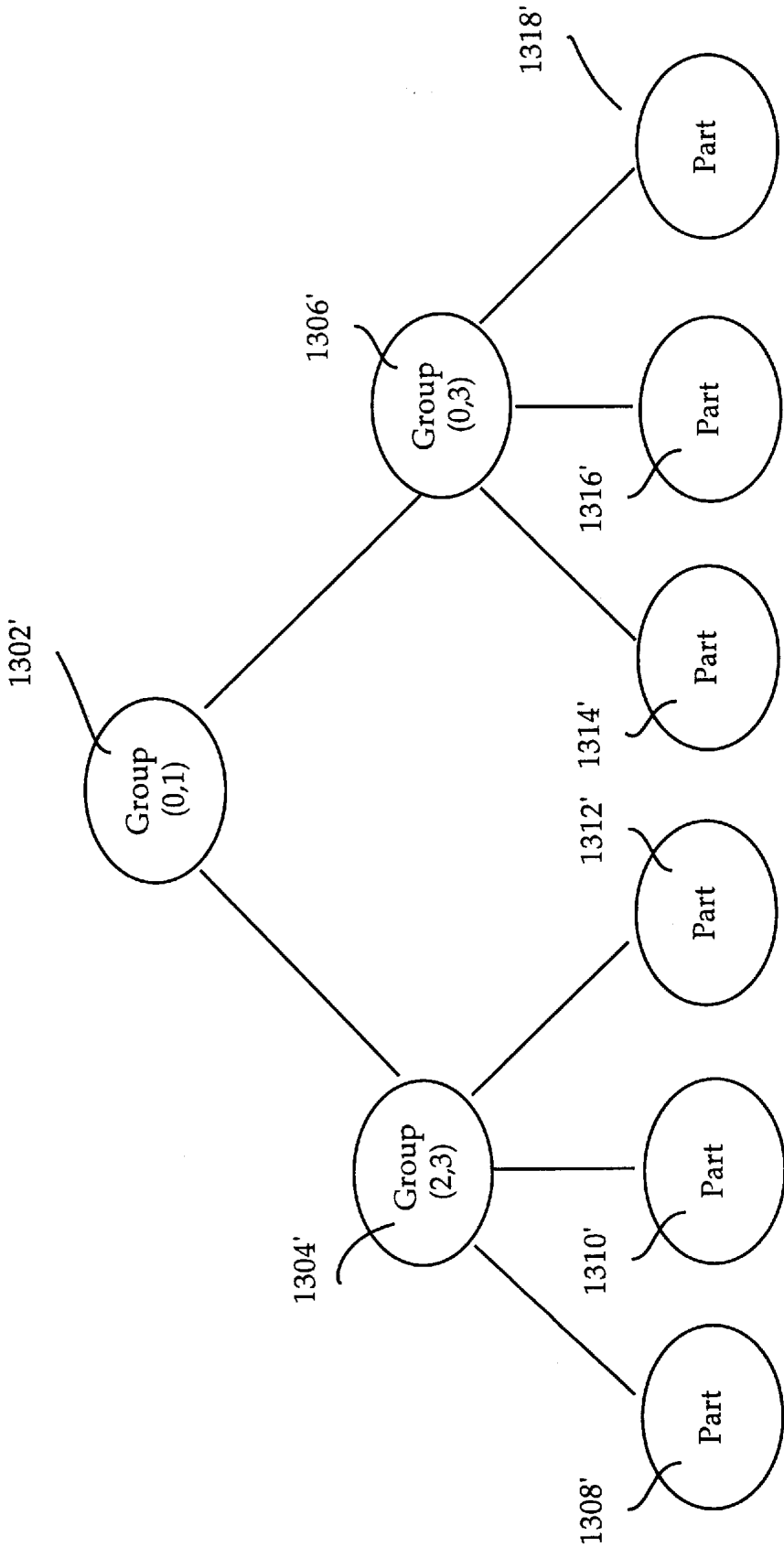


FIG. 13B

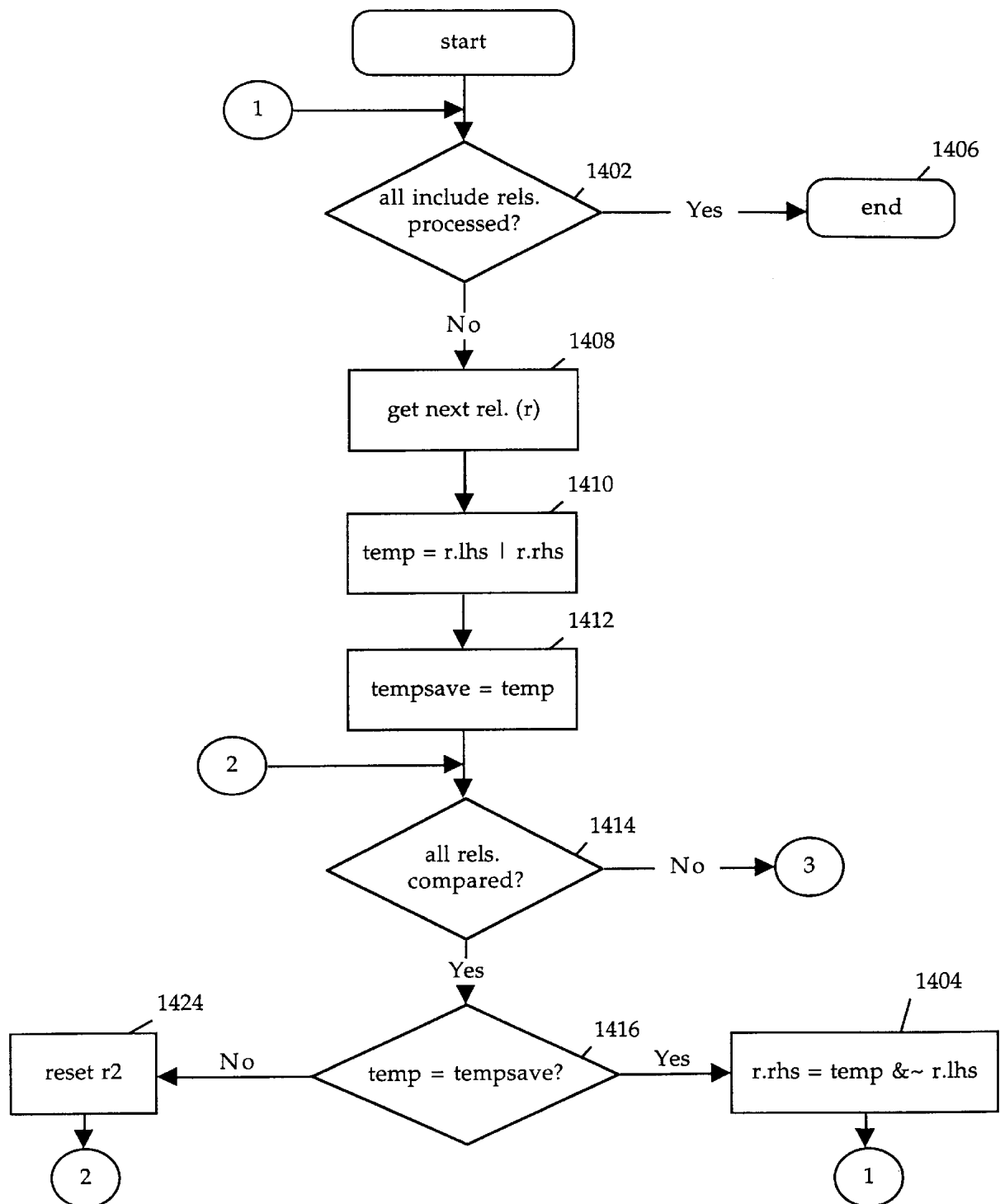


Fig. 14A

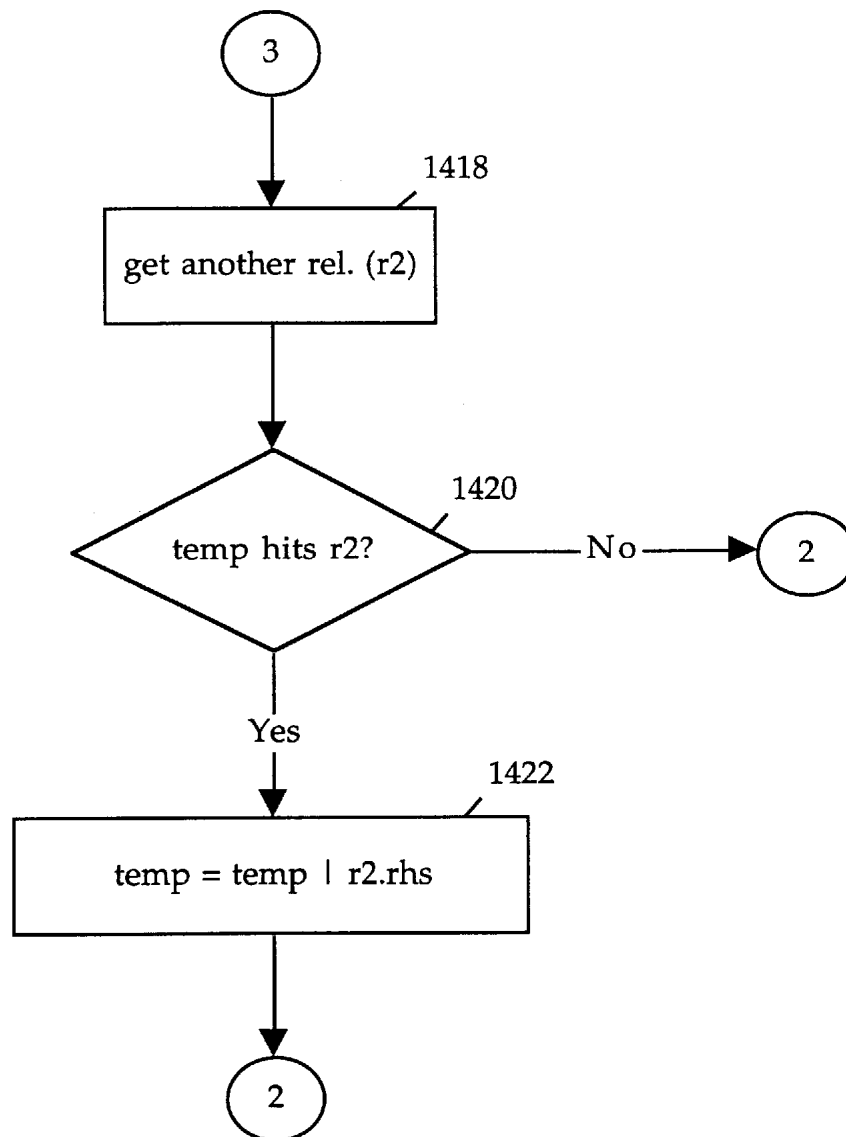


FIG. 14B

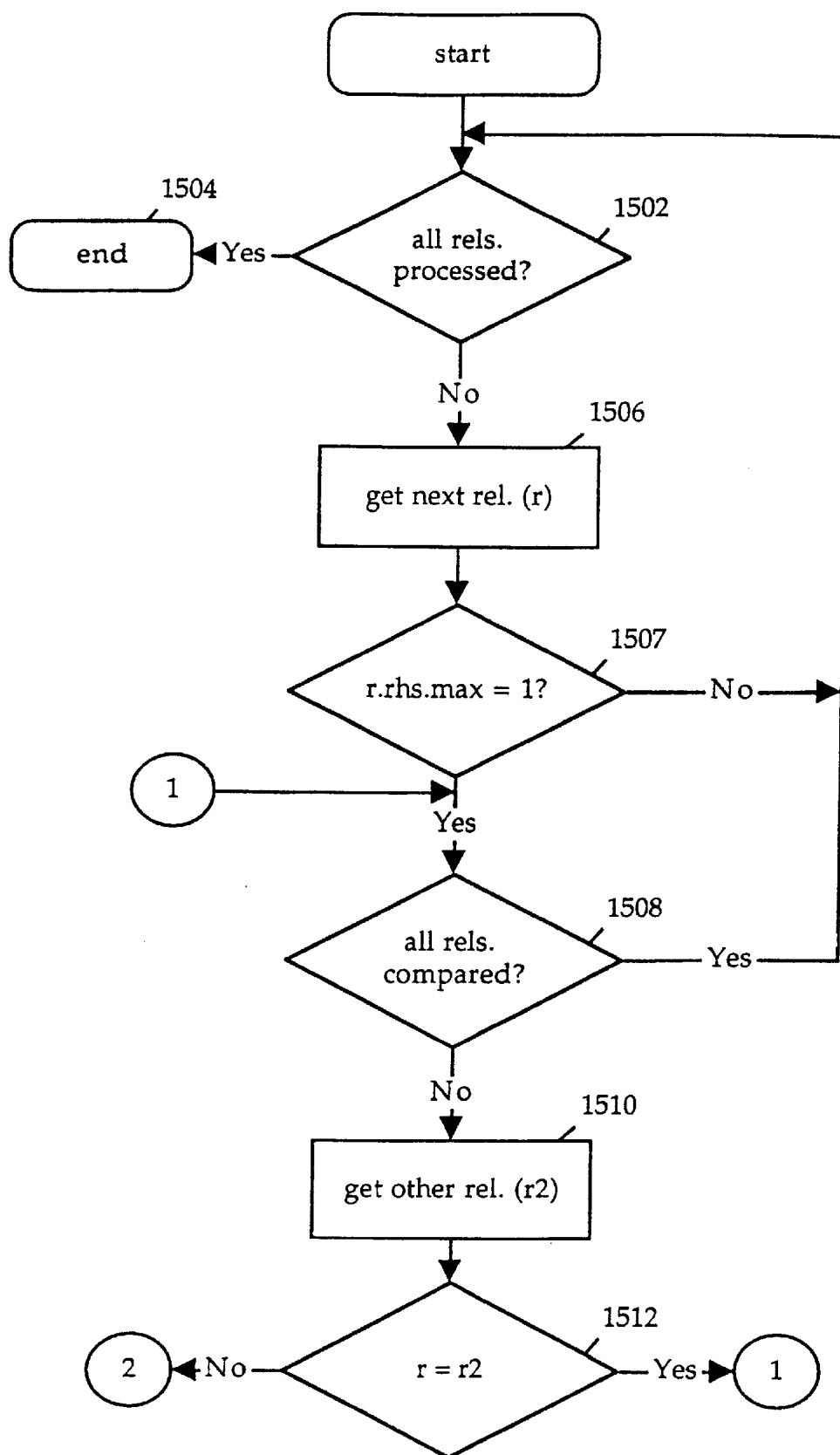


FIG. 15A

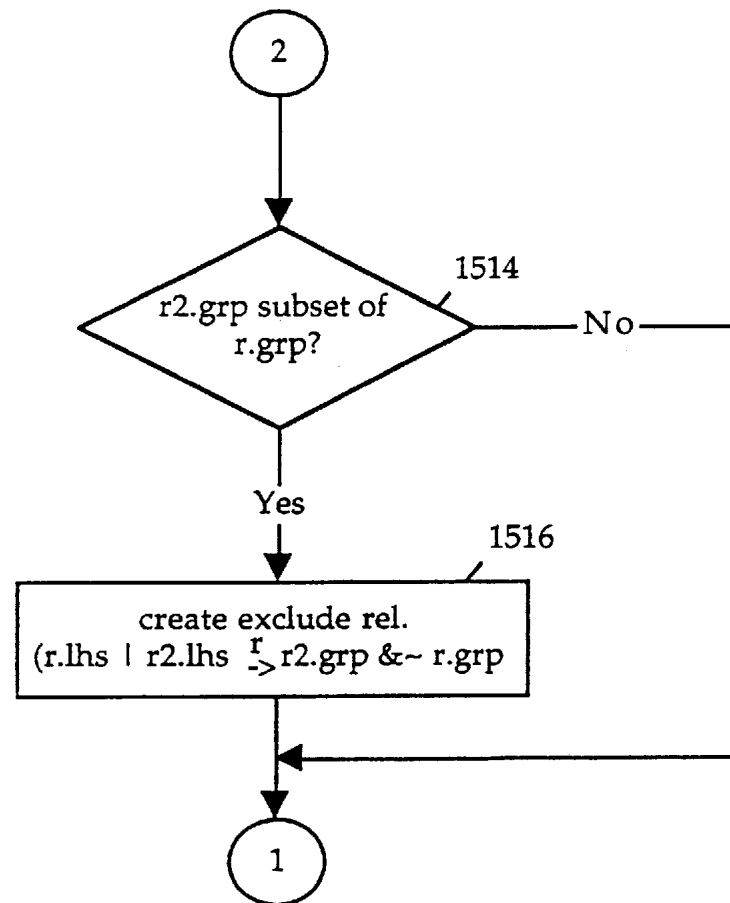


FIG. 15B

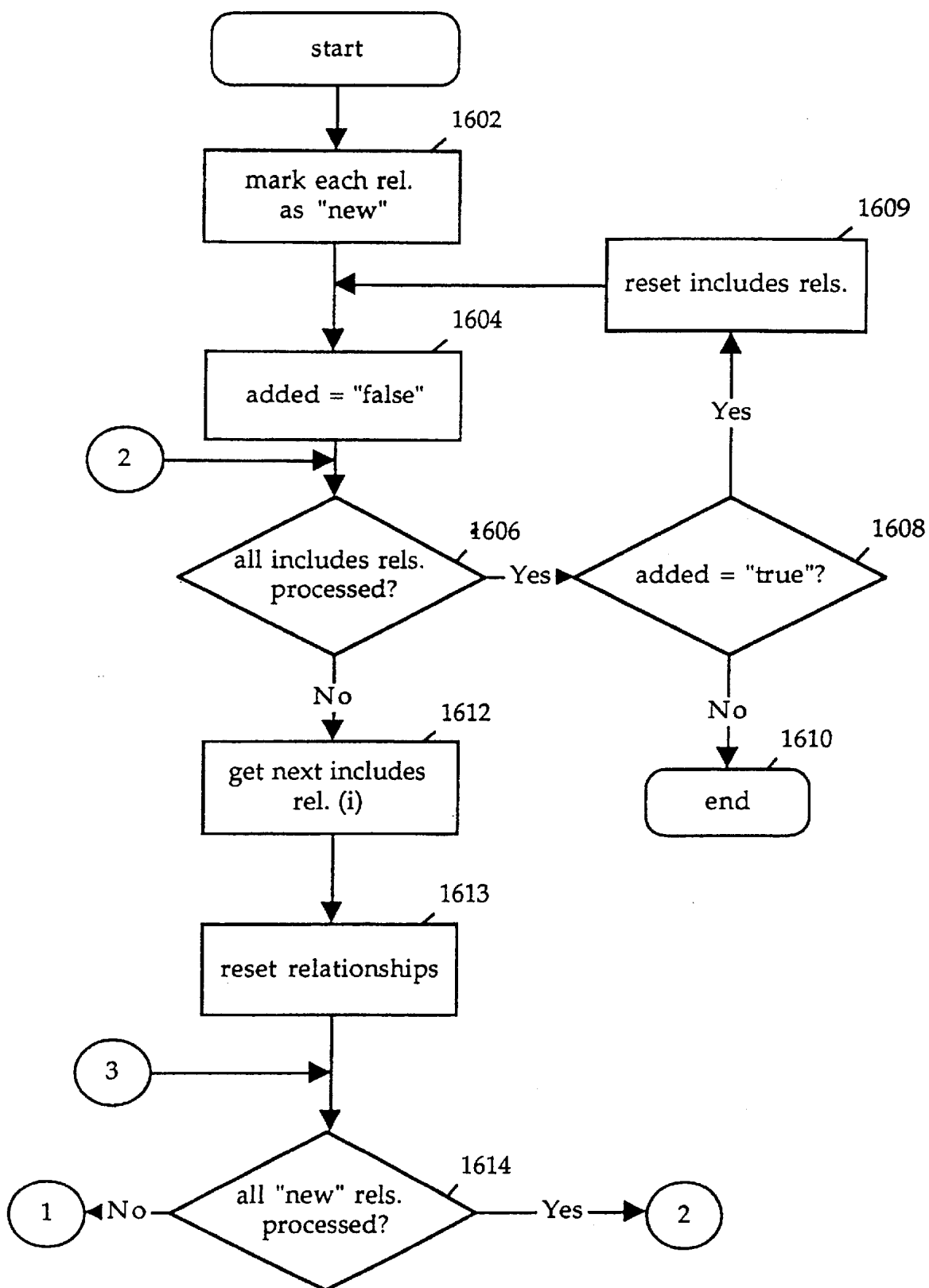


FIG. 16A

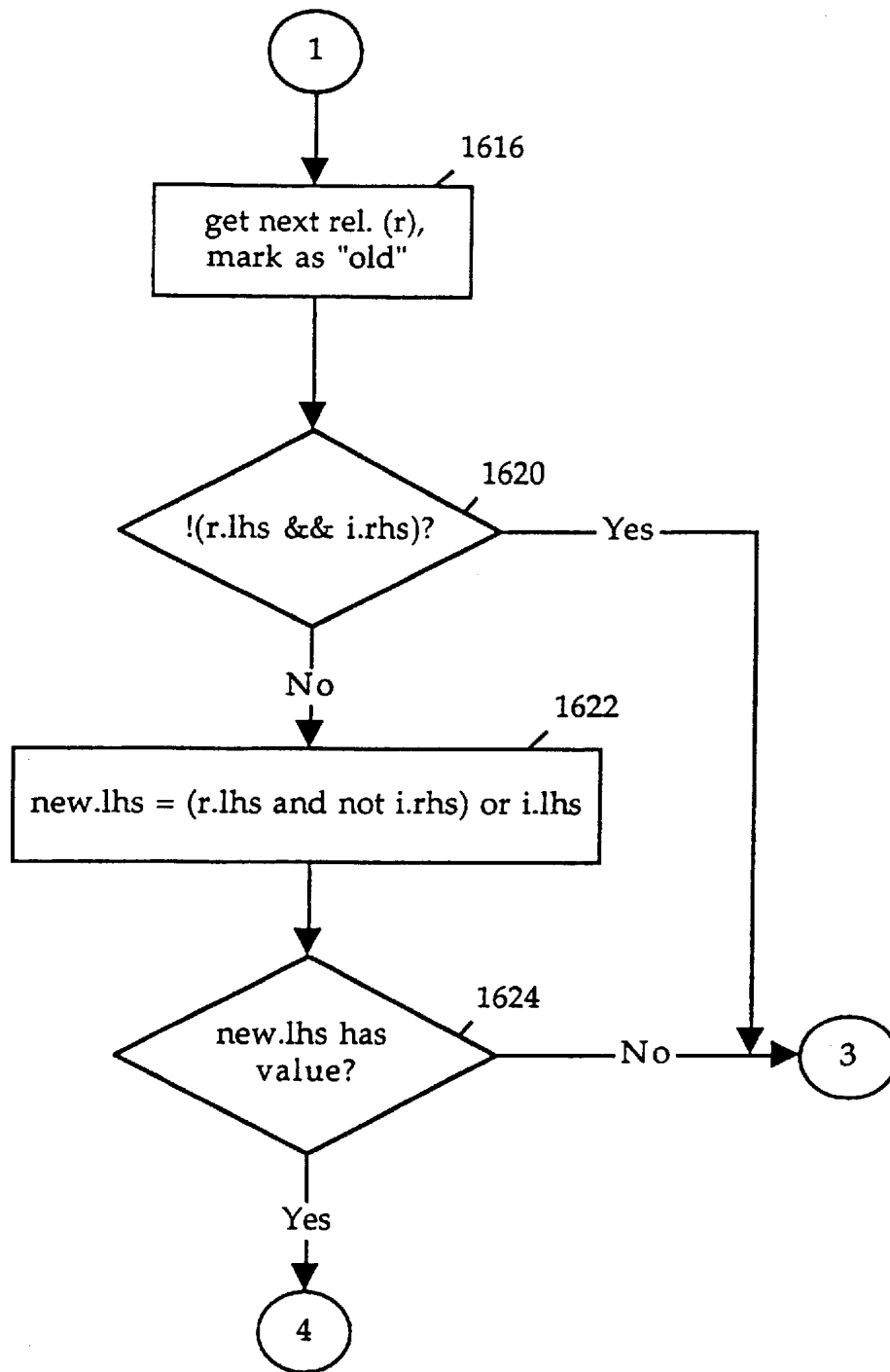


FIG. 16B

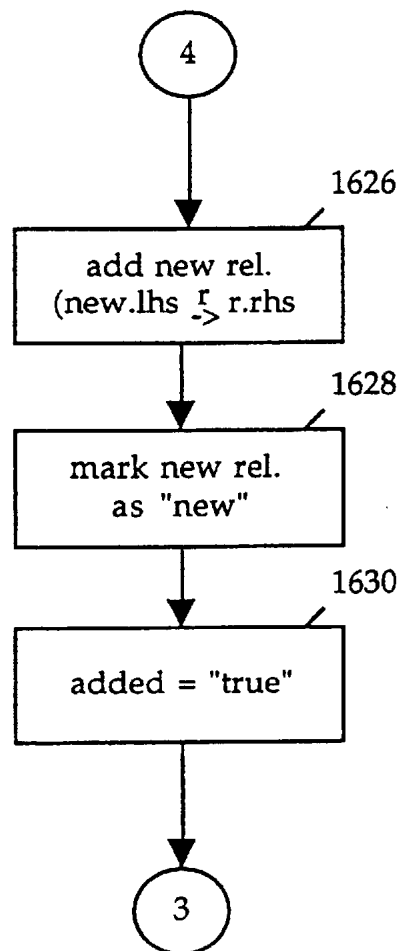


FIG. 16C

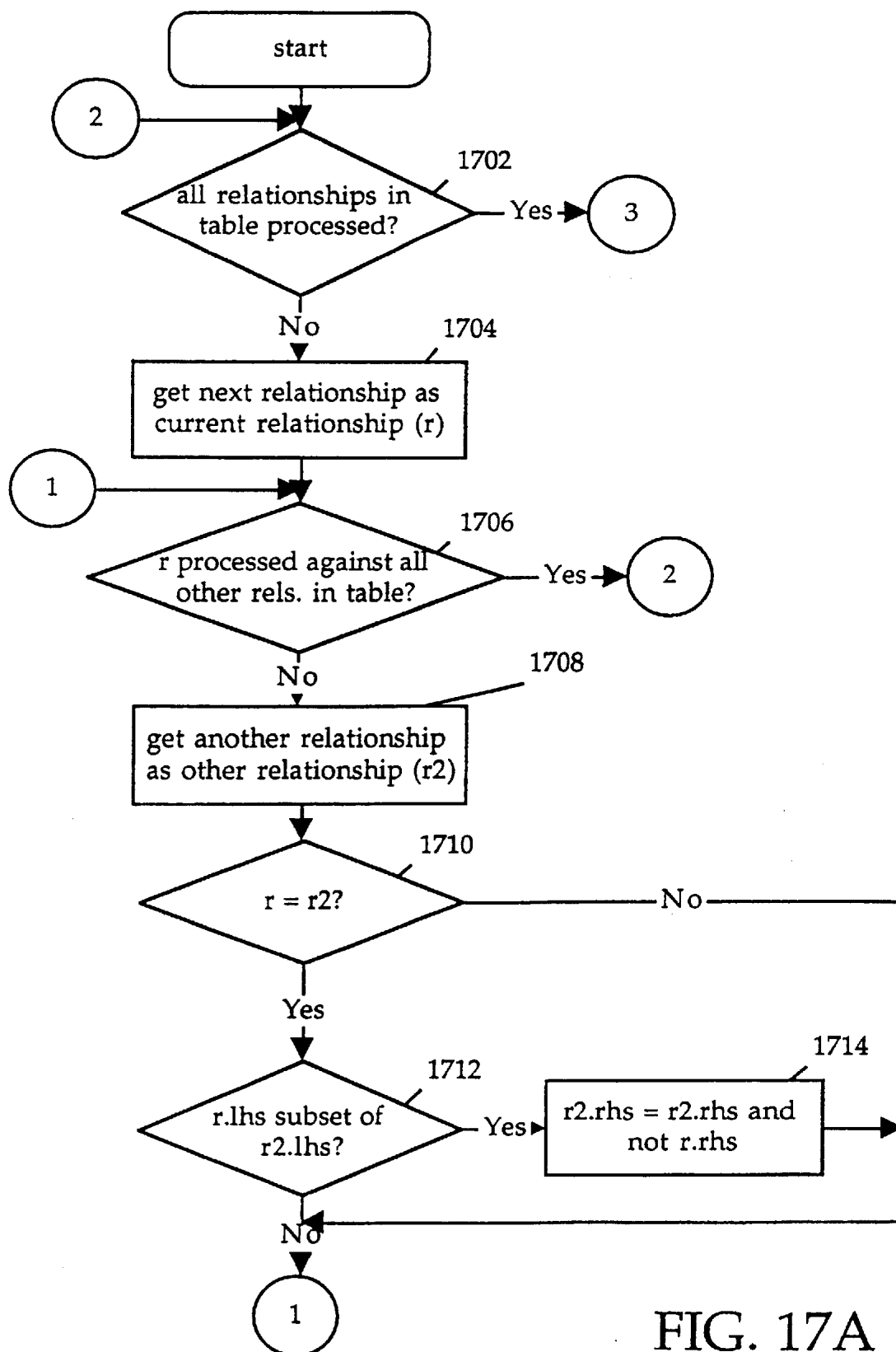


FIG. 17A

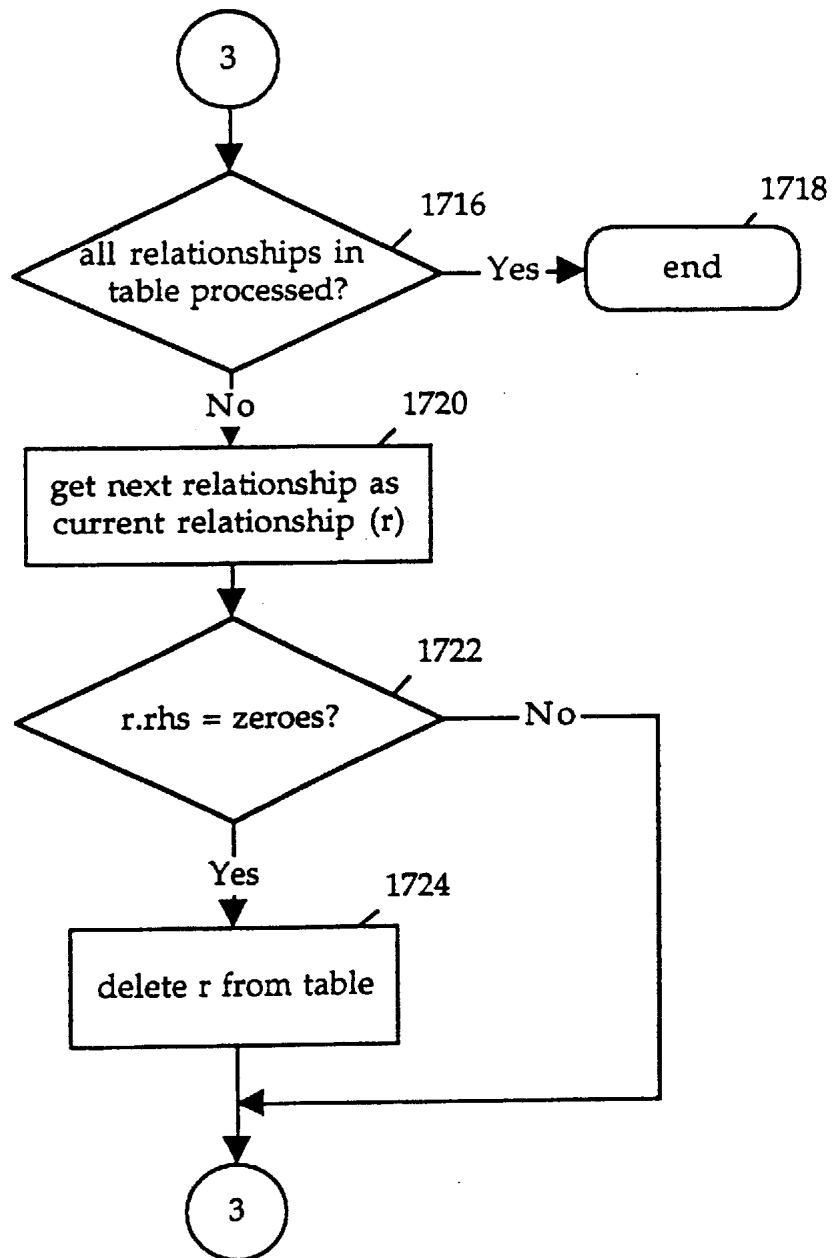


FIG. 17B

METHOD AND APPARATUS FOR MAINTAINING AND CONFIGURING SYSTEMS

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to maintaining and configuring systems.

2. Background Art

A system is comprised of components. Before a system can be built the components of the system must be identified. To configure a system, a user must select the parts to include in the system. Typically, one who is knowledgeable about a system and its components defines the system. Thus, for example, an automobile salesperson assists an automobile buyer in determining the type and features of the automobile. The salesperson understands the features and options that are available to create a valid configuration. Some features and options cannot be combined. The selection of some features caused other features to be unavailable, etc. It would otherwise be difficult for the buyer to identify all of the features and options available on the automobile that can be combined to create a valid configuration.

Computer systems have been developed to assist one in configuring a system. However, these systems use a configuration language to define a system. Like a programming language, a configuration language uses a syntax that must be understood by a user who is maintaining the data (i.e., a data maintainer). To use one of these configuration systems, it is necessary for a data maintainer to understand the configuration language. This limits the number of users who are able to use the configuration systems. That is, the level of sophistication needed to communicate with the configuration system (through a configuration language) results in less sophisticated users being unable to use the system.

In addition, configuration systems impose a flow or ordering to the user operations. For example, a user is required to remove components from the system in reverse of the order in which they were chosen. Thus, a user may be forced to remove components that the user wants to keep in the configuration to remove an unwanted component. A novice user may have perform many removal operations before achieving an acceptable configuration. If the novice user is required to remove components in a preset order, the user can become frustrated or confused and abort the configuration process.

These systems are designed for a more sophisticated user that has knowledge of the system that is being configured as well as the configuration system used to configure the system. A end user such as an automobile shopper would have difficulty using these systems.

Further, to use these systems a user must be trained to understand the configuration language. Thus, a user who otherwise has knowledge of the systems that are being configured must undergo training to be able to use these configuration systems to configure systems. This leads to increased expenditures such as for training.

SUMMARY OF THE INVENTION

The invention provides the ability to interactively select and configure a product among a set of related products based on availability and compatibility of features and options. It does not impose an order in the selection of products, features or options; only valid selections can be

made at any time. To create an electronic representation of the product information to achieve the above goal, the invention provides a framework for defining a product line.

A product line is defined as a set of related products. A product line has a set of products that contain parts, or components. Parts used to define a product are selected from a parts catalog. Parts in a product definition are related or classified as: included (parts that are included by default), required choices (a choice among a group of parts that must be made to achieve a valid configuration), optional (parts that can be optionally included in the configuration).

Relationships can be defined between the parts in a product definition. A relationship relates a first set of parts with a second set of parts. A set can include multiple parts. The incorporation of parts in a set can be arbitrary. That is, a multi-part set can contain parts that are otherwise unrelated. For example, a set can contain parts such as an engine, sun roof and a color. These parts seem to be unrelated, however, it is possible to combine them into a relationship set for purposes of forming a relationship using the present invention.

Preferably, the part relationships are: included, excluded, removed, and requires choice. An included part is included automatically. A part is excluded from the configuration when its inclusion would result in an invalid configuration. A part may be removed when another part is added. Thus, when a first part exists in the configuration and a second part is added, the first part is removed from the configuration. The requires choice relationship is used to allow a set of choices to be made from a group of parts. The number of parts chosen is limited to a valid bounds specification. The relations that are created between parts within a product are enforced only on that particular product. However, if some part-to-part relationships are to be enforced on all products within a product line, then the relations are generated once and enforced for all products.

A maintenance system is used to define a product. Using the maintenance system, a product can be defined using the product classifications and the part relationships. A graphical user interface (GUI) is used to allow the user to interactively generate a definition. Instead of configuration languages, GUI operations such as drag and drop and selection operations can be used to specify a definition. The notions of included, optional and required choice are easily comprehensible to a user. Further, the idea that parts have interrelationships is also easily understood. Thus, a product can be defined without having to learn a complicated configuration language.

A configuration system is used to configure a system using a definition created by the maintenance system. The configuration system ensures that the current configuration state is always valid. The user can select and unselect parts in any order. When user input is received, the configuration system validates the input based on the current state of the configuration. In addition, the configuration system identifies selections that could cause a valid configuration to become invalid. The configuration removes these selections from the set of possible selections so that the user does not make an invalid selection.

The configuration system evaluates the current state of a configuration based on the product definition, part relationships and state information. After receipt of input from a user, the configuration system evaluates relationships in both the forward and backward direction. Forward and backward evaluations can result in the addition or deletion of elements from the configuration.

The invention uses both an external and internal representation of a definition or definitions. A translation mechanism is used to translate an external representation into an internal representation. The external representation uses a conceptually understandable set of relationships for defining a system and the relationships between the components of the system. The invention takes the definition created by a user and supplements and compresses the definition when necessary to create an internal representation. The internal representation is used during configuration to initialize and validate a configuration based on user input.

During configuration, the invention maintains runtime information that is stored in tables and vectors. To achieve greater processing efficiency, the system represents elements in a configuration (e.g., product, part, and group) as a bit in a bit vector. Thus, for example, a vector has a length that is equal to the total number of elements. An element's bit can be set or reset to specify the state of the element in the current configuration. For example, a user vector can be used that specifies for each element whether the element has been selected by the user during the configuration. In addition, excluded and removed vectors identify whether an element is excluded or removed (respectively) from a configuration. Vectors can be used to identify whether an element 1) has been selected (by the user or the configuration system), 2) is selectable, and 3) not selectable.

Tables contain element relationships. A table is used to represent the includes, excludes, removes, and requires choice relationships, for example. Each table has a left-hand side and a right-hand side that corresponds to the left-hand and right-hand sides of a relationship. In each case, the left-hand side is a bit vector that contains bits that correspond to elements. The includes, excludes and removes tables contain a bit vector in the right-hand side that represents configuration elements. The right-hand side of the requires choice table is a pointer that points to an entry in a group table. The group table entry is a bit vector that identifies the elements that are contained in the group from which a choice is to be made. The right-hand side of a requires choice table entry further includes minimum and maximum designations. Minimum and maximum values identify the minimum and maximum number of group members that are to be selected to satisfy a requires group relationship.

A bit vector implementation of relationships and internal runtime state allows for fast and efficient computation of relationship based configuration. A comparison of bits can be performed in one machine instruction in most cases.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 provides an illustration of a computer system that can be used with the invention according to an embodiment of the invention.

FIG. 2 provides an overview of the maintenance and configuration systems according to an embodiment of the invention.

FIG. 3 illustrates examples of elements in a parts catalog according to an embodiment of the invention.

FIG. 4 illustrates relationships between parts according to an embodiment of the invention.

FIG. 5 provides an example of product classifications according to an embodiment of the invention.

FIG. 6 provides an example of a GUI screen used in maintenance system 202 according to an embodiment of the present invention.

FIG. 7 provides a block diagram illustrating the conversion process according to one embodiment of the invention.

FIGS. 8A–8B illustrate components of internal representation 706 according to an embodiment of the invention.

FIG. 9 provides a process flow for processing a user selection according to an embodiment of the invention.

FIG. 10 provides an example of a relationship evaluation process flow according to an embodiment of the invention.

FIG. 11 provides an example of an unselect item process flow according to an embodiment of the invention.

FIG. 12 provides a flow for translation processing according to an embodiment of the invention.

FIGS. 13A–13B provide an illustration of groups and nested groups according to an embodiment of the invention.

FIGS. 14A–14B provide an includes chain process flow according to an embodiment of the invention.

FIGS. 15A–15B provide an example of a subgroup excludes process flow according to an embodiment of the invention.

FIGS. 16A–16C provide an example of a relationship factorization process flow according to an embodiment of the invention.

FIGS. 17A–17B provide an illustration of a relationship compression process flow according to an embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

A method and apparatus for maintaining and configuring systems is described. In the following description, numerous specific details are set forth in order to provide a more thorough description of the present invention. It will be apparent, however, to one skilled in the art, that the present invention may be practiced without these specific details. In other instances, well-known features have not been described in detail so as not to obscure the invention.

The present invention can be implemented on a general purpose computer such as illustrated in FIG. 1. A keyboard 110 and mouse 111 are coupled to a bidirectional system bus 118. The keyboard and mouse are for introducing user input to the computer system and communicating that user input to CPU 113. The computer system of FIG. 1 also includes a video memory 114, main memory 115 and mass storage 112, all coupled to bi-directional system bus 118 along with keyboard 110, mouse 111 and CPU 113. The mass storage 112 may include both fixed and removable media, such as magnetic, optical or magnetic optical storage systems or any other available mass storage technology. Bus 118 may contain, for example, 32 address lines for addressing video memory 114 or main memory 115. The system bus 118 also includes, for example, a 32-bit DATA bus for transferring DATA between and among the components, such as CPU 113, main memory 115, video memory 114 and mass storage 112. Alternatively, multiplex DATA/address lines may be used instead of separate DATA and address lines.

In the preferred embodiment of this invention, the CPU 113 is a 32-bit microprocessor manufactured by Motorola, such as the 680X0 processor or a microprocessor manufactured by Intel, such as the 80X86, or Pentium processor. However, any other suitable microprocessor or microcomputer may be utilized. Main memory 115 is comprised of dynamic random access memory (DRAM). Video memory 114 is a dual-ported video random access memory. One port of the video memory 114 is coupled to video amplifier 116. The video amplifier 116 is used to drive the cathode ray tube

(CRT) raster monitor **117**. Video amplifier **116** is well known in the art and may be implemented by any suitable means. This circuitry converts pixel DATA stored in video memory **114** to a raster signal suitable for use by monitor **117**. Monitor **117** is a type of monitor suitable for displaying graphic images.

The computer system described above is for purposes of example only. The present invention may be implemented in any type of computer system or programming or processing environment.

The invention maintains and configures systems. The invention eliminates the need for a user to learn a configuration language or write code to maintain and/or configure a system. A user interface uses various operations such as drag and drop and item selection to define a product, for example. Elements that comprise a definition (e.g., of a product) can be added or removed in any order. No order is imposed on the user. There is no requirement, for example, that the user remove parts of a product in the order in which they were added. No fixed flow or order is required to edit a given product.

A definition includes an identification of the components that comprise the definition and the interrelationships between the components. The interrelationships are conceptually easy for the user to understand. The same relationships are used to define and configure any system. An external representation that includes these relationships allows a user to view and maintain the definition. The external representation is translated into an internal representation that is designed to decrease processing time and increase response time.

In addition, during a configuration session, the invention is capable of allowing a user to select or unselect configuration items without imposing a flow on the user. There is no order imposed on the user in terms of the sequence in which items are selected for the configuration or unselected from the configuration. For example, it is not necessary to select a product before choosing options. The invention can identify the products that are still available based on the options that have already been selected. Further, a user can unselect an item (e.g., delete the item from the configuration) without regard to the order in which it was selected (e.g., added to the configuration).

Examples of systems that can be maintained or configured using the invention include automobiles, computers, time clock machines, and shoes. Terms such as part, product line, parts catalog, and product are used herein for illustration purposes. It should be apparent that this invention can be used to configure systems that are not limited to products and product lines, etc.

FIG. 2 provides an overview of the maintenance and configuration systems according to an embodiment of the invention. Maintenance system **202** maintains a parts catalog **204**, parts relationships **206** and product definitions **208**. Maintenance system **202** uses a user interface that includes the ability to add items to parts catalog **204** and specify part relationships **206** and product definitions **208**. The user interface displays a set of hierarchies that provide a conceptually easier way of viewing a definition. The invention maps the set of hierarchies (an external representation) to an internal representation.

The internal representation is used by configuration system **212** to maintain and configure systems based on user input. Configuration system **212** provides the ability to specify a product. Configuration system **212** verifies the product specification. Using configuration system **212**, a

user can interactively select and configure a product among a set of related products based on availability and compatibility of features and options. It does not impose an order in the selection of products, features or options. The configuration system **212** allows the user to only make valid selections. However, selections can be made in any order.

Parts catalog **204** consists of parts that are components of products. Similar parts are grouped together to form a part hierarchy. Easy maintenance of relationships is achieved by the hierarchy. For example, when a group of parts is assigned a behavior, all the members inherit that behavior automatically. FIG. 3 illustrates examples of elements in a parts catalog according to an embodiment of the invention. Referring to FIG. 3, a parts catalog (e.g., parts catalog **204**) contains parts **302–310**, **316**, and **318**. A parts catalog can also contain a group of parts such as group **312**. Group **312** contains parts **306–310**. A group can contain other groups. For example, group **314** contains group **312** and parts **316–318**. Behavior assigned to elements of group **314** is inherited by members of group **312**.

Part-to-part relationships can be created between parts within a product. Relationships are defined between a first set of parts and a second set of parts. During configuration, when all of the members of the first set of parts are selected, the relationship between the two sets is enforced on the parts in the second set. In the preferred embodiment, there are four kinds of relationships between parts: requires choice, includes, can't work with (or excluded), and removes. FIG. 4 illustrates relationships between parts according to an embodiment of the invention.

Part **402** includes part **404**. The includes relation causes a set of parts in a second set (e.g., part **404**) to be included in the configuration when a first set of parts (e.g., part **402**) is selected. For example, a luxury package includes a CD player; when the luxury package is selected, a CD player is included in the configuration. The can't work with (or excluded) relation ensures that a set of parts from a second set are never in the same configuration as parts in the first set. For example, part **402** (e.g., sun roof) can't work with part **406** (e.g., a roof-top antenna). When part **402** is selected, part **406** cannot be selected. Part **406** is excluded such that it cannot be selected.

When part **402** is selected, part **414** is removed from the configuration. The removes relation causes items that are included in a second set of to be removed from the configuration when the left side is selected. For example, when a high end stereo is selected the standard stereo is removed from the configuration.

The requires choice relation recognizes that a choice (between a minimum and maximum number) has to be made between a second set of parts (or members of a group) to ensure a valid configuration when the parts in a first set are selected. Part **406** (e.g., climate control feature) requires that a choice be made among parts **408–410** (e.g., 1 zone A/C or 2 zone A/C). A requires choice relationship requires that a number of items be selected based on minimum and maximum values from the right-hand side of the relationship to satisfy the relationship.

Parts **408** and **410** can be combined to form group **416**. Group **416** can be defined by the user. If the user does not define group **416**, the invention preferably creates group **416** to contain parts **408** and **410**. Thus, when two or more parts are defined on the right-hand side of a requires choice relationship, the invention preferably creates a group that contains these parts. The requires choice then becomes a requires choice on the group (e.g., group **416**).

Parts are used to illustrate relationships in FIG. 4. However, a relationship can contain groups. To illustrate, a group of parts can be substituted wherever a part is used in the illustration. Thus, for example, parts **402**, **404**, **406**, **408**, **410**, and **414** can each be replaced by a group of parts. Thus, for example, if parts **402** and **404** are replaced by groups (e.g., groups **402** and **404**, respectively), when the members of group **402** are selected, the members in group **406** are included in the configuration. To further illustrate, part **408** can be replaced by group **408**. In that instance, group **408** is a member of group **416**. A hierarchy of groups (e.g., group **408** contained within group **416**) can be used in a relationship.

Parts can be combined to form an arbitrary grouping. Thus, it is not necessary to combine parts in a group based on a logical or intuitive relationship between the parts in the group. For example, a group can contain an engine, a color, and a sun roof.

The relations that are created between parts within a product are enforced only on that particular product. However, if some part-to-part relationships are to be enforced on all products within a product line, then the relations are created once and are enforced for all products. These relationships are referred to as global relationships.

A product includes zero or more elements of parts catalog **204**. Product definition **208** (see FIG. 2) is generated by populating it with its component parts. Product definition **208** is generated by population of a product with its component parts. The parts within a product are classified as one of three different types: included parts, required choices, or optional parts. A part that is not classified as one of these types is assumed to be unavailable in that product. FIG. 5 provides an example of product classifications according to an embodiment of the invention.

An included part is a part that is included in a product by default. For example, parts **504** and **506** are automatically included in product **502**. For example, when a configuration user chooses the product definition product **502**, the parts **504** and **506** are automatically included in the configuration. A required choices classification specifies that a choice among a group of parts has to be made to create a valid product configuration. For example, product **502** (e.g., automobile) can include a color group **518** containing red (part **508**), green (part **510**) and blue (part **512**) that is a required choice. In configuring a product, the user must choose a color in this group to create a valid product configuration. Parts **514** and **516** are optional parts. An optional part is not required for a valid configuration.

A product line is defined as a set of related products. The invention provides the framework for defining a product line. To define a product line: all parts (components) in the product line are entered into parts catalog **204**. An instance in products definition **208** is created that identifies parts from parts catalog **204** that are assigned to a product. Part relationships **206** can also be defined for the product. Maintenance System

Parts catalog **204**, part relationships **206**, and product definition **208** are created and maintained using maintenance system **202**. While any method can be used for creation and maintenance, a graphical user interface (GUI) is preferably used. A text editor could also be used for data entry, for example. FIG. 6 provides an example of a GUI screen used in maintenance system **202** according to an embodiment of the present invention. One who uses the GUI screen (a maintainer) can define a product (e.g., an automobile). A product definition is used by a configuration user to config-

ure a system defined in maintenance system **202**. Parts catalog **204** uses a database or other type of storage and retrieval capability, for example, to store information about its parts (e.g., part number, description, price, etc.).

The GUI screen of FIG. 6 is divided into a product definition section **650** and a part relationship definition section **652**. Pane **602** displays elements from parts catalog **204**. Panes **604**, **606**, **608** are used to define a product. Panes **604**, **606**, and **608** represent classifications or product relationships. Panes **610** and **614** and relationship **612** are used to define part relationships.

A user can drag elements from pane **602** to panes **604**–**608** to define a product. For example, to include Part B in the product definition, Part B is dragged from pane **602** to pane **604**. Alternatively, to drag parts B, C, D, and E, group A can be dragged from pane **602** to pane **604**. Group A and its component parts (parts B, C, D, and E) are thereby included in the product definition. Similarly, a user can specify that a configuration user must choose a part from a group, e.g., Group I, by dragging one or more parts or a group into pane **606**. An optional part or group can be identified by dragging an element, e.g., Group L, into pane **608**. If an element from pane **602** is not moved to one of panes **604**–**608** it is assumed that the maintainer wants to exclude that element from the product that is being defined. The product-level relationships or classifications (or types) illustrated in FIG. 5 can be defined using panes **604**–**608**.

A subset of the product-level relationships map to a subset of the part relationships. For example, the include and requires choice product-level relationships map to the same named part relationships (as defined in FIG. 4 and shown in relationships **612**). Minimum and maximum values set for a requires choice product-level relationship map to the minimum and maximum values of a requires choice part relationship. Elements that are not dragged into one of panes **604**, **606** or **608** are considered to be excluded.

In addition to defining the elements (and their types) that comprise the product, the maintainer can specify relationships between parts of the product. Panes **610** and **614** and relationship **612** are used to define relationships between parts. The maintainer can drag an element (or elements) from pane **602** into pane **610**. For example, the user can drag Part N from pane **602** into pane **610**. The element(s) dragged into pane **610** are referred to as the left-hand side of the relationship. The element (or elements) from pane **602** that is related to the left-hand element(s) is dragged into pane **614**. The element(s) dragged into pane **614** are referred to as the right-hand side of the relationship. For example, the user can drag Part K from pane **602** to pane **614**.

To identify the type of relationship that exists between the left-hand side and right-hand side elements, the maintainer chooses one of the four part-to-part relationships from relationship **612**. For example, if the maintainer chooses the includes relationship, the right-hand side elements are included in the product when a configuration user selects the left-hand side element(s). Thus, for example, Part N includes Part K. When Part N is chosen by a configuration user, Part K is automatically included in the configuration.

When the maintainer completes the product definition using the GUI screen, the product definition is saved (e.g., in parts definition **208**). The product definition can then be used by a configuration user in configuration system **212**. External and Internal Representations

The GUI screen of FIG. 6 illustrates a view of a product as seen by a user (e.g., maintainer and configuration user). A view of the product that is seen by a user is referred to as the external representation. A user sees a product or product

definition from a conceptual level that is easy to understand. The user does not need to express a definition or configuration using a language that has a syntax that is foreign to the user. A GUI having graphical selections and operations is preferably used to allow the user to see the external representation. The external representation uses terminology that is familiar or intuitive to the maintainer. For example, an include product relationship defines elements that can be included in a product.

An external representation seen by the user is translated into an internal representation that can be used by the invention to process user input. FIG. 7 provides a block diagram illustrating the conversion process according to one embodiment of the invention. Compiler 704 is used to perform the conversion. Compiler 704 generates internal representation 706 from external representation 702.

External representation 702 is the representation of a product definition that is created using the maintenance system of the invention. External representation 702 provides a graphical representation of a system definition, its components and their interrelationships. FIG. 6 can be used to display external representation 702. External representation 702 can include a multi-level hierarchy of parts that help to define the product at a conceptual level. The hierarchies are not required to drive a configuration session, however. Therefore, compiler 704 flattens out the hierarchies. A set of relationships is generated using external representation 702 during a translation process.

The set of relationships generated for a system definition by compiler 704 comprise internal representation 706. Internal representation 706 is generated from external representation 702 and can be used to control a configuration session. In addition to internal representation 706, a configuration state is used during the configuration session. FIGS. 8A–8B illustrate components of internal representation 706 and the configuration state according to an embodiment of the invention.

Referring to FIG. 8A, vector 802 provides an example of a vector used to store state information. Vector 802 is comprised of bits. A set of p bits is used to represent the products that are known to maintenance system 202 and configuration system 212. Each bit of the P bits corresponds to a product. A set of N bits is used to represent the parts. Each bit in the N-bit set corresponds to a part in parts catalog 204, for example. Vector 802 is therefore P+N bits in length.

Vectors 804–810 provides implementation examples of vector 802. Vectors 804–810 are maintained by configuration system 212 in a configuration session. Vector 804 is a user vector that contains the set of elements that are selected by the user. Vector 804 is referred to as the user vector, or uVec. The set of elements that are included in the configuration by the configuration system (e.g., an item that is automatically included when the user selects the left-hand in an includes relationship) is contained in included vector 806, or iVec. Removed vector 808, or rVec, contains the elements that are removed from the configuration by configuration system 212. Elements excluded from the configuration by configuration system 212 are identified in excluded vector 810, or xVec.

Each element that can be used to configure a system has a bit in these vectors. When an element is included, excluded, removed (or deleted), or selected, the corresponding bit in the appropriate vector (iVec, xVec, rVec, or uVec) is set. The use of vectors to represent elements results in increased efficiency in processing. Bit-wise operations can then be used during processing. For example, it is possible to determine whether a group of elements have been selected

by the user by performing a bit comparison using the uVec. The increased efficiency results in optimal response times that are necessary for an interactive system such as is preferred for configuration system 212. The bit vector implementation used by the invention allows for fast and efficient computation of runtime algorithms. A bit vector implementation of relationships and internal runtime state allows for fast and efficient computation of relationship based configuration.

Another vector that contains state information is required groups vector 812, or rgVec. The rgVec contains a bit for each of the required groups specified for a product. As previously discussed, the maintainer can include a required group in a product definition by dragging a group from pane 602 to pane 606, for example (see FIG. 6). The number of bits in rgVec can differ from the number of bits in vectors 802–810. The number of bits contained in rgVec corresponds to the total number of required groups defined for a product. Each bit in rgVec is set to indicate that it has not yet been satisfied. It is unset (set to zero) when a required group has been satisfied. Thus, when a user attempts to accept a configuration, configuration system 212 can determine whether all of the required choices have been satisfied.

In addition to these vectors, additional vectors are used to determine a current state. The current state is defined by three types of state information that track the state of parts in the configuration. The current state is visible to the user. The three state types are: selected, selectable, and notSelectable. The selected state identifies all of the parts that are currently included in the configuration. The selectable state identifies the parts that are not in the configuration but that can be selected by the user for the configuration. The notSelectable state identifies the parts that are not in the configuration and cannot be selected by the user. Configuration system 212 ensures that only parts that can form a valid configuration are selectable; those that cannot are notSelectable.

The uVec, iVec, xVec, and rVec vectors can be used to generate the three visible states. The states are dynamic and can change based on user input. The following provides definitions using the formulae that can be used to generate the three visible states (+is union and–is difference):

$$\text{Selected} = (\text{uVec} + \text{iVec}) - \text{rVec}$$

$$\text{NotSelectable} = \text{xVec} + \text{rVec}$$

$$\text{Selectable} = [\text{AllItems}] - \text{Selected} - \text{NotSelectable}$$

Computation of the visible state using C++, for example, can be accomplished using the following C++ code (the syntax assumes appropriate bit-wise operator overloads for the given operations):

$$\text{SelectedVec} = (\text{uVec} | \text{iVec}) \& \sim \text{rVec}$$

$$\text{NotSelectableVec} = \text{xVec} | \text{rVec}$$

$$\text{SelectableVec} = \sim (\text{SelectedVec} | \text{NotSelectableVec})$$

The selected state is the union of the elements selected by the user or automatically included in the configuration minus the elements removed by configuration system 212. The notSelectable state is the union of the elements that are excluded or removed. The selectable state is the set of all elements minus the elements already selected and the elements that cannot be selected. The selected, selectable, and notSelectable states can be generated from these four vectors when needed. They can be stored as vectors that have the structure of vector 802.

11

Runtime tables are preferably used to store internal representation **706** for a configuration session. FIG. **8B** provides an example of runtime tables. Runtime table **822** provides an example of a runtime table. Runtime table **822** can be used to store relationship information. Runtime table **822** has left-hand side **824** and right-hand side **826**. Left-hand side **824** contains the left-hand side of a relationship (e.g., includes, removes, can't work with or excludes). The right-hand side of a relationship is stored in right-hand side **826**.

Table **828** provides an example of runtime table **822**. Table **828** can be an includes, excludes, or removes table, for example. Left-hand side **830** contains a bit vector (i.e., "01010 . . . 0"). Right-hand side **832** contains a bit vector (i.e., "10001 . . . 1"). Each bit in bit vectors **830** and **832** corresponds to an element in parts catalog **204** or a product. For example, if the product definition is for an automobile, each "1" bit in bit vector "01010 . . . 0" corresponds to a part in parts catalog **204** that is contained in an automobile's product definition. Zero bits indicate that the part is not included in the product definition. With reference to bit vectors, parts can be identified by the position of their corresponding bit in the bit vector. For example, the part (that corresponds with the "1" bit in the bit vector "0010" is part number three.

If table **830** is an instance of an includes table, when parts designated on left-hand side **830** (e.g., part numbers two and four) are selected, configuration system **212** includes the parts designated in right-hand side **832** (e.g., part numbers one and five). Similarly, if table **830** is excludes (or can't work with) table, parts one and five are excluded from the selectable parts when parts two and four are selected. Parts one and five are removed from the configuration when parts two and four are selected.

Like the includes, excludes, and removes tables, the requires choice table contains a left-hand side and a right-hand side. In one embodiment, both sides of the requires choice table contains a bit vector that corresponds to elements in parts catalog **204**. Thus, for example, when selected, the part(s) identified on left-hand side **836** require a choice of a minimum and maximum number of parts from the parts contained on the right-hand side **838**.

Table **822** can be altered to optimize the storage and use of information for a required choice table. Table **834** provides an example of an altered table **822**. Preferably, however, right-hand side **838** contains a pointer to a group table (e.g., group table **840**) that contains a bit vector (bit vector **842**) that identifies the parts contained in the group. An advantage that the latter (altered table **822**) format has over the former (unaltered table **822**) is that it is only necessary to define the members of a group once. Each product definition that includes the group can point to the entry in the group table that contains the group definition.

One relationship table can be used for all products per relationship type. That is, a single includes table is used for includes relationships for all of the products. Alternatively, a table can be partitioned for each product. That is, the relationships associated with a product are segregated. For example, there would be a includes relationship table for each product. This has the advantage of only having to process only those relationships that exist for the chosen product. When a product is unselected or not selectable, there is no need to process the tables associated with that product. The latter alternative is preferable because it lessens processing requirements.

Configuration System

Using configuration system **212**, a configuration user can configure a system given a product definition (e.g., product

12

definition **208**) and part relationships associated with the product definition. Configuration system **212** accepts a configuration user's input and processes it to update the configuration based on the user's input or notify the user that the input is invalid given the current state of the configuration and the system definition. Configuration system **212** ensures that the current configuration state (determined by the product definition and order-independent user selection) is always valid.

User Selection

When user input is received by configuration system **212**, it is processed based on the current configuration state, the product definition and the parts relationships. FIG. **9** provides a process flow for processing a user selection according to an embodiment of the invention.

At step **902**, the user selects item *n* (e.g., part) preferably using a GUI screen. At step **904** (i.e., "*n* selectable?"), a determination is made whether the item is selectable. Thus, for example, configuration system **212** determines whether the *n*th bit in the selectable state information is set. If it is not, configuration system **212** determines that the item is not selectable by the user and raises an error at step **922**. Processing then continues at step **914** to update the interface (if necessary) and return control of the GUI screen to the user. Processing of the current input ends at step **916**.

If, however, it is determined at step **904** that the item selected by the user is selectable, processing continues at step **908** to set the *n*th bit in *uVec* **804**. At step **910**, the visible state information is updated as discussed above. At step **912**, the relationships associated with the product definition are evaluated as discussed below. Processing continues at step **914** to update the interface (if necessary) and return control of the GUI screen to the user. Processing of the current input ends at step **916**.

Relationship Evaluation

Relationships associated with items contained in the product definition are evaluated when user input is received. Configuration system **212** determines what relationships are active and inactive given the user input (at step **912**, for example). A relationship is active when all the items on the left-hand side of the relationship are selected. A relationship is inactive until all of the parts on the left-hand side of the relationship are selected.

As previously discussed, relationship tables can be used to store the left-hand and right-hand sides of a relationship. That is, a relationship can be defined as a pair of bit vectors, the *lhsVec* (left-hand side) and the *rhsVec* (right-hand side). A relationship is made active if the following expression is true:

$$\text{SelectedVec} \ \& \ \text{lhsVec} == \text{lhsVec}$$

When a relationship is made active, updating the appropriate state vector (*iVec*, *xVec*, or *rVec*) becomes:

$$\text{stateVec} |= \text{rhsVec}$$

An active relationship causes the configuration state to be updated. When an include relationship is activated, for example, parts identified on the right-hand side of the include relationship are added to the configuration, if they are not already included. The state is changed by modifying *iVec* **806**. A change in *iVec* **806** can change other state information such as the selected and selectable state information. Parts that are included as a result of the activation of the includes relation can result in a ripple effect that causes other relationships to become activated. Thus, additional state changes can occur as a result of the include relationship's activation.

To further illustrate, when a requires choice relation becomes active, configuration system **212** includes and excludes items of the given group where applicable: if exactly max (maximum number of) items of the given group are selected, the engine excludes all other items in the group (added into xVec); otherwise, if less than min (minimum number of) items of the group are selected and only one valid selection path exists to make exactly min items selected, the engine includes (into iVec) the items that are needed (note that these must have been selectable items).

Configuration system **212** also ensures that no relationship that will put the configuration into an invalid state can become active (for example: if b is Selected then the relationship [a] can't work with [b] should not be active). If a relationship will make a configuration invalid, it is made notActivateable. For example, a relation should be made notActivateable, if:

- 1) an includes relationship has excluded (xVec) parts on the right-hand side;
- 2) a excludes (can't work with) relationship has Selected parts on the right-hand side;
- 3) a removes relation has user-selected (uVec) parts on the right side; or
- 4) a requires choice relation has a group for which between [min, max] parts can't be Selected.

In addition to the relationships created by a maintainer, there are some relationships that are dynamically created or deduced at runtime. If a relation becomes notActivateable and all but one item from the left-hand side is Selected, then the unSelected item is excluded by the engine. This ensures that the relation will not become active.

Configuration system **212** evaluates each relationship forward and backward. In a forward evaluation, the left-hand side of a relationship is examined to determine whether all of its parts are included in the configuration. If they are, the relationship is considered to be active. An active relationship causes the right-hand side of the relationship to then be evaluated. The evaluation is dependent on the type of relationship. An includes relationship causes the parts specified in the right-hand side to be included in the configuration specification. A removes relationship causes the right-hand side parts to be removed from the configuration. An excludes relationship cause the left-hand side parts to become notSelectable. A requires choice relationship recognizes that a choice has to be made by the user between the set of right-hand parts.

In a backward evaluation, configuration system **212** determines whether a relationship if activated would result in an invalid state. If so, configuration system **212** takes steps to ensure that the relationship cannot be activated. Instead of evaluating the left-hand side first as in the case of a forward evaluation, a backward evaluation evaluates the relationship by first examining the right-hand side. If a relationship fails the backward evaluation (i.e., its activation would result in an invalid state), configuration engine **212** takes steps to ensure that the relationship cannot become active. For example, if a relationship states that part A can't work with part B and part B is selected, configuration system **212** excludes part A. This ensures that part A cannot be selected and the excludes relationship cannot become active. Otherwise, a user can select part A which would result in an invalid configuration state.

The invention performs a relaxation over the set of relationships. A relaxation is an iterative process that is repeated until a state that has been defined no longer changes. That is, relationship evaluation is completed when none of iVec, xVec, or rVec are modified during an iteration.

The invention ensures that the relaxation process terminates by only adding bits to the state. That is, because iVec, xVec, and rVec all grow through the iteration, the relaxation process is guaranteed to terminate.

FIG. **10** provides an example of a relationship evaluation process flow according to an embodiment of the invention. At step **1002**, rgVec is initialized to zero. During the relationship evaluation, rgVec is regenerated. At step **1004**, copies of iVec, xVec, and rVec are made. The copies are used to determine whether relaxation is finished. At step **1006**, the includes table associated with the product definition is evaluated in the forward direction. Similarly, forward evaluation is performed on the excluded and removed tables. The forward evaluation performs an "or" operation with the selectedVec and a specified configuration state vector (e.g., a forward evaluation of the includes table performs an "or" operation with the selectedVec and the iVec) when the left-hand side of the relationship is true. For example, when the items that correspond with the "1" bits in left-hand side **830** are all selected (i.e., in the selected vector), right-hand side **832** is "or'ed" with the iVec to add the items that correspond with the "1" bits in right-hand side **832** to the configuration where a forward evaluation of an includes relationship is being performed.

At step **1008**, a backward evaluation is performed on the relationships contained in the includes, excludes, and removed tables. In a backward evaluation of an include relationship, the xVec is examined to determine whether any right-hand side items of an include relationship are excluded. If so, selection of all of the left-hand side items would result in an invalid configuration state. Therefore, the last remaining left-hand side item (if only one remains) is excluded to cause the relationship to never become active. Thus, the relationship is rendered notActivateable.

Backward evaluation of an excluded relationship examines the selectedVec. If a left-hand side item (an item that is supposed to be excluded when the relationship becomes active) is selected, selection of all of the left-hand side items would result in an invalid configuration state. To ensure against an invalid configuration state, a left-hand side item is excluded so that the relationship never becomes active (the relationship is rendered notActivateable).

In the preferred embodiment, configuration system **212** cannot delete, or remove, a user-selected item. The uVec is examined in a backward evaluation of a removes relationship. If a right-hand side item is a user selected item, the selection of all of the left-hand side items would result in an invalid configuration state. To ensure against an invalid configuration state, a left-hand side item is excluded so that the relationship becomes notActivateable.

At step **1010**, backward and forward evaluation is performed on the requires choice table **834**. In the backward evaluation, a determination is made whether a requires choice relationship should become notActivateable. For example, if it is impossible to satisfy a group, then a left-hand side item should be excluded to cause the relationship to become notActivateable.

In the forward evaluation, rgVec is set and items are included or excluded where appropriate. For example, if the right-hand side items are within range (within min and max values for the required choice relationship), then the remaining right-hand side items are excluded. If the only way to make a requires choice valid is to select the items remaining on the right-hand side item, configuration system **212** includes these items. In addition, a requires choice relationship is evaluated to determine whether the right-hand side items selected are within a valid range. If not, the corre-

sponding bit in the rgVec is set. If so, the corresponding bit in the rgVec is zeroed.

The selected, selectable, and notSelectable states are updated at step 1012. At step 1014 (i.e., “change in a vector?”), a determination is made whether any of iVec, xVec, or rVec were modified during the current iteration of the relaxation process. If not, a stable state has been achieved, and processing ends at step 1016. If so, additional iterations are needed to complete the relaxation process. Processing therefore continues at step 1002 to initiate a new iteration.

In evaluating relationships, it is not necessary to evaluate a relationship in either the forward or backward direction once it has become active. Also, when a relationship is notActivateable, there is no need to evaluate the relationship in the forward direction. When a relationship is notActivateable and has been excluded, there is no need to evaluate in the backward, or reverse, direction.

Unselection

FIG. 9 described processing where the user input was an item selection. Configuration system 212 also processes user input to unselect an item from the configuration. As previously discussed, an item can be unselected in any order. There is no requirement to unselect items the reverse order in which they were selected. FIG. 11 provides an example of an unselect item process flow according to an embodiment of the invention.

At step 1102 (i.e., “selection in uVec?”), a determination is made whether the item that is selected for deletion was selected for inclusion in the configuration by the user. If it is determined that the item is not a user-selected item, processing continues at step 1104 to raise an error. Processing continues at step 1114 to update the GUI (if necessary) and return control to the user. Processing of the current input ends at step 1116.

If it is determined at step 1102 that the item selected for unselection was selected by the user, processing continues at step 1106 to reset the bit in the uVec that corresponds to the unselected item (i.e., set the bit to zero). This unselects the item from the current configuration. At step 1108, the iVec, xVec, and rVec vectors are initialized (set to zero). Steps 1110, 1112 and 1114 correspond to steps 910, 912, and 914, respectively, of FIG. 9. Processing ends at step 1116.

Translation Between Representations

To ensure correct runtime behavior of the engine, a potentially incomplete user-defined set of relationships is converted into a set of complete runtime relationship tables. The external representation is translated into an internal representation. The invention maps the hierarchies contained in parts catalog 204 to relationships and maps relationships to the runtime tables, for example.

As discussed above, compiler 704 performs a mapping between product-level relationships and part-level relationships. The translation process further provides the ability to translate a multi-level hierarchy to a single-level hierarchy. That is, the translation process can be used to flatten out the intuitive hierarchy created by a maintenance user using the user interface of FIG. 6, for example. FIG. 12 provides a flow for translation processing according to an embodiment of the invention.

At step 1202, a set of relationships is created based on the hierarchies. Thus, for example, group 312 is translated into an requires choice relationship. That is, group 312 requires a choice between parts 306, 308, and 310. Relationships can be defined in terms of groups. An identifier (item ID) is used to identify the group. When a group appears on the left-hand side of a relationship, the group is assigned an item ID and

the relationship is defined in terms of the new item ID. This process, referred to as back-linking, is performed at step 1204.

At step 1206, an include chain evaluation is performed. In an include chain evaluation, all include relationship chains are fully evaluated so that relationship contains all of the parts that are included in part regardless of whether the included relationship is direct (e.g., mentioned in the includes relationship) or indirect (mentioned in another includes relationship that contains an included part on the right-hand side).

At step 1208, subgroup excludes are evaluated. In this process, active requires choice relationships are evaluated where one relationship’s right-hand side is a subset of another relationship’s right-hand side. In this case, if both relationships are active, the right-hand side parts that are not within the subset are excluded. Relationship factorization is performed at step 1210. Relationship factorizing can create new relationships that ensure proper runtime behavior when more than one item is selected during some iteration.

Removing redundancy in relations may cause some relations to become unnecessary. A relation becomes unnecessary when it contains no items on its right side. Redundant relationships are removed at step 1212. Processing ends at step 1214.

Back-Linking

Relationships can be defined in terms of groups. When a group appears on the left-hand side of a relationship, the group is assigned an item ID and behaves as a normal item (e.g., part) and the relationship is defined in terms of this new item ID. A relationship is defined for each item in the group:

item includes item ID (of the group)

It is only necessary to create this relationship once per group, regardless of how many times that group appears on the left side of a relationship. Back-Linking can be used to generate relationships in nested groups. FIGS. 13A–13B provide an illustration of groups and nested groups according to an embodiment of the invention. Group 1302 contains two items (groups 1304 and 1306) that are nested within group 1302. Notation is provided for a group to specify the minimum and maximum bounds for a group. For example, the notation “(0,1)” indicates that between zero and one items in group 1302 are to be chosen.

The items contained in group 1302 are also groups that contain multiple items. Groups 1304 and 1306 are nested within group 1302. Two or three of parts 1308–1312 can be chosen from Group 1304. Zero to three of parts 1314–1318 can be chosen from Group 1306.

The invention creates a metapart for these groups (groups 1302–1306). Relationships are then created between the groups and their members such that items are back-linked to their parents. FIG. 13B provides an illustration of metaparts for the groups and nested group of FIG. 13A according to an embodiment of the invention. Groups 1302–1306 are metaparts for groups 1302–1306, respectively. Relationships can be generated using these metaparts as follows:

Group 1302’ requires choice [Group 1304’, Group 1306’] 0,1

Group 1304’ requires choice [part 1308, part 1310, part 1312] 2,3

Group 1306’ requires choice [part 1314, part 1316, part 1318] 0,3

The metapart is used internally; it is unnecessary for a user to be aware of them. Using back-linking, relationships can be generated such that an item (e.g., part 1308) links back to its predecessor(s) (e.g., nested group, group 1302).

The following new relationships can be created for the examples of FIGS. 13A–13B:

Group 1304' includes Group 1302'

Group 1306' includes Group 1302'

Part 1308 includes Group 1304'

Part 1310 includes Group 1304'

Part 1312 includes Group 1304'

Part 1314 includes Group 1306'

Part 1316 includes Group 1306'

Part 1318 includes Group 1306'

When a user selects part 1308, part 1308 includes group 1304'. Group 1304' is added to the configuration. Further, group 1304' includes group 1302' thereby including group 1302'. When group 1302' is included in the configuration, it activates the relationship that requires a choice between groups 1304' and 1306'. Since group 1304' is already selected, group 1306' can be excluded. When group 1306' is excluded, parts 1314–1318 are excluded when the newly generated relationships are evaluated in the reverse (backward) direction. Further, the bounds specified by group 1304 are also enforced using these relationships.

As discussed above, a requires choice relationship can be defined at the product level. A product can require a choice between items in a group. The product-level requires choice relationship is treated similarly to that of a requires choice part-level relationship.

Include Chains

In an include chain evaluation, all include relationship chains are fully evaluated so that a relationship can be generated that contains all of the parts that are included regardless of whether the relationship is direct (e.g., the included parts mentioned in the includes relationship) or indirect (a part is mentioned in another includes relationship that contain an included part on the right-hand side). For example, the following relationships exist in the external representation:

a includes b and c

b includes d

d and c includes e

An include chain exists that starts with a. The first relationship indicates that a includes b and c. Based on the second and third relationships, b and c include other parts. Based on all of the relationships, a includes b, c, d, and e. Include chain evaluation expands these relationships into the following relationships:

a includes b and c and d and e

b includes d

d and c includes e

It is not necessary to generate additional relationships to perform includes chain evaluation. Preferably, includes chain evaluation can modify an existing relationship to include both direct and indirect includes relationships. Each includes relationship can be evaluated against all other includes relationships to determine whether its right-hand side should be modified to include both direct and indirect includes relationships. Once all of the includes relationships have been evaluated, a new right-hand side value can be generated based on the evaluation. FIGS. 14A–14B provide an includes chain process flow according to an embodiment of the invention.

At step 1402 (i.e., “all include rels processed?”), a determination is made whether all of the includes relationships have been processed. If other includes relationships remain to be processed, processing continues at step 1408 to get the next includes relationship, r. At step 1410, a temporary

vector, temp, is created that is the result of an “or” operation between the left-hand and right-hand sides of r. Another temporary vector, tempsave, is created that is set to the value of temp at step 1412. At step 1414 (i.e., “all rels. compared?”), a determination is made whether all of the other includes relationships have been evaluated against r. If not, processing continues at step 1418 to get another relationship, r₂. At step 1420 (i.e., “temp hits r₂?”), a determination is made whether any of the items contained in temp are also contained in r₂. If not, processing continues at step 1414 to process any remaining relationships against r. If so, processing continues at step 1422 to perform a “or” operation between temp and the right-hand side of r₂ to add the right-hand side of r₂ to the temp vector.

If it is determined, at step 1414, that all of the relationships r₂ have been evaluated for r, processing continues at step 1416. At step 1416 (i.e., “temp=tempsave?”), a determination is made whether any items were added to the temp vector. If not, there are no indirect includes and processing continues at step 1424 to reset r₂ and processing continues at step 1414 to process any remaining relationships against this relationship. If so, processing continues at step 1404 to modify the right-hand side of r to include the items contained in the temp vector minus the items contained in the left-hand side of r. Processing continues at step 1402 to process any remaining relationships.

If it is determined at step 1402 (i.e., all include rels. processed?) that all of the includes relationships have been evaluated for include chains, processing ends at step 1404.

Subgroup Excludes

The subgroup excludes evaluation is performed on requires choice relationships. Assume a requires choice relationship points to a group (A) that is a complete subgroup of a group (B) of another requires choice relationship. If B has a max of 1 and both relationships are made active, the items in B that are not in A, Q, cannot be chosen. If a pairwise comparison of requires choice relationships yields such cases, a relationship such as the following is generated:

[union of left sides of both relationships] can't work with Q

For example, assume that the following relationships have been defined:

a b requires choice [x, y, z] (0,1)

d requires choice [x, y] (0,2)

The second relationship contains a group that is a subgroup of the first relationship. When both relationships are active, the non-subset is excluded. This avoids the case where a user selects a, b, and d and the relationship evaluations do not exclude the items not included in both groups (the non-subset). The invention combines these relationships and creates an excludes relationship. The excludes relationship for the above example is as follows:

a b d excludes z

FIGS. 15A–15B provides an example of a subgroup excludes process flow according to an embodiment of the invention. At step 1502 (i.e., “all rels. processed?”), a determination is made whether all of the requires choice relationships have been processed. If so, processing ends at step 1504. If not, processing continues at step 1506 to get the next relationship, r. At step 1507 (i.e., “r.rhs.max=1?”), a determination is made whether the maximum value for the relationship is equal to one. If not, processing continues at step 1502 to process any remaining relationships. If so, processing continues at step 1508. At step 1508 (i.e., “all rels. compared?”), a determination is made whether all of the other requires choice relationships have been processed against r. If so, processing continues at step 1502 to process another requires choice relationship.

If not, processing continues at step 1510 to get another relationship, r_2 . At step 1512 (i.e., " $r=r_2$?"), a determination is made whether the two relationships are equal. If so, processing continues at step 1508 to process any remaining requires choice relationships against the current r . If r and r_2 are not equal, processing continues at 1514. At step 1514 (i.e., " $r_2.grp$ subset of $r.grp$?"), a determination is made whether the right-hand side of r_2 is a subset of the right-hand side of r . If not, processing continues at step 1508 to process any remaining requires choice relationships against the current r . If so, processing continues at step 1516 to create an exclude relationship where the left-hand side is the union of the left-hand side of r and r_2 and the right-hand side is the non-subset of the right-hand side of the two relationships. Processing continues at step 1508 to process any remaining requires choice relationships against the current r .

Relationship Factorization

The invention assumes that one item on the left side of a relationship is preferably selected during any iteration. However, the includes and requires choice relationships may cause additional items to become selected (e.g., selected by the system). Factorizing of relationships can create new relationships to ensure proper runtime behavior when more than one item is selected during some iteration.

Factorization causes a relationship to be created where a relationship has items on the left-hand side that also appear on the right-hand side of an includes relationship. The new relationship represents the common factorization of both relationships. For example, where the following two relationships exist:

[a] includes [b and c]

[b and c and d and e] can't work with [x and y]
the following relationship is created:

[a and d and e] can't work with [x and y]

The invention uses a technique in factorization that avoids combinatorial explosion. Relationships are created during factorization when they add value. The invention identifies value where the new relationship:

1. replaces more than one item from the left-hand side;
2. has at least two different includes relationships replace at least one unique item from the left-hand side of the relationship being factorized; and
3. the two includes relationships have at least one common item in their left-hand side.

FIGS. 16A–16C provide an example of a relationship factorization process flow according to an embodiment of the invention. At step 1602, each relationship is marked as "new". At step 1604, a state variable, added, is initialized to "false". At step 1606 (i.e., "all includes rels. processed?"), a determination is made whether all of the includes relationships have been processed. If so, processing continues at step 1608 (i.e., "added='true'?") to determine whether any new relationships were added. If not, processing ends at step 1610. If there are new relationships, processing continues at step 1609 to reset the includes relationships and processing continues at step 1604 to initialize added to "false".

If it is determined at step 1606 that all of the includes relationships have not been processed, processing continues at step 1612 to get the next includes relationship, i . At step 1613, the relationships are reset. At step 1614 (i.e., "all 'new' rels. processed?"), a determination is made whether all of the relationships have been marked as "new" have been processed against i . If so, processing continues at step 1606 to process any remaining includes relationships. If not, processing continues at step 1616.

At step 1616, the next relationship, r , is marked as "old". At step 1620 (i.e., " $!(r.lhs \&\& i.rhs)$?"), a determination is

made whether the right-hand side of the i relationship and the left-hand side of the r relationship have any bits in common. If they do not have bits in common, processing continues at step 1614 to process any remaining relationships against the current i . If they do have bits in common, processing continues at step 1622 to generate new.lhs from the includes relationship and the left-hand side of r . Step 1622 takes out items in the right-hand side of i from left-hand side of r and adds in items that are in the left-hand side of i . At step 1624 (i.e., "new.lhs has value?") a determination is made whether new.lhs has value. This determination can use the guidelines as specified above.

If new.lhs does not have value, processing continues at step 1614 to processing any remaining relationships against i . If it does have value, processing continues at step 1626 to add a new relationship (of the type r) that uses new.lhs as the left-hand side and the right-hand side of r . The new relationship is marked as "new" at step 1628. The value of added is set to "true" at step 1630. Processing continues at step 1614 to process any remaining relationships.

Relationship Compression

Removing redundancy in relations may cause some relations to become unnecessary. A relation becomes unnecessary when it contains no items on its right side. If a relationship's left-hand side is a proper subset of the left-hand side of another relationship, then the first relationship is guaranteed to be active whenever the second relationship is active. If both relationships are also the same type of relationship, the common right-hand side elements can be removed from the second relationship. This can be repeated to remove all unnecessary relationships. The following relationships provide an example:

[a] removes [c]

[a and b] removes [c and d]

The left-hand side of the first relationship is a proper subset of the left-hand side of the second relationship (e.g., all elements "a" in the left-hand side of the first relationship are contained in the left-hand side of the second relationship). The relationships are the same type. Therefore, the common right-hand side elements can be removed from the right-hand side of the second relationship. The following is the result after modifying the relationship:

[a] removes [c]

[a and b] removes [d]

FIGS. 17A–17B provide an illustration of a relationship compression process flow according to an embodiment of the invention. At step 1702 (i.e., "all relationships in table processed?"), a determination is made whether all of the relationships in the table have been processed. If not, processing continues at step 1704 to get the next relationship, r . At step 1706 (i.e., " r processed against all other rels. in table?"), a determination is made whether the other relationships in the table have been processed against r . If so, processing continues at step 1702 to process any remaining relationships in the table as r .

If it is determined at step 1706 that all of the other relationships have not been processed against r , processing continues at step 1708. Another relationship is retrieved as r_2 at step 1708. At step 1710 (i.e., " $r=r_2$?"), a determination is made whether the relationships are the same. If not, processing continues at step 1706 to process any remaining relationships against r . If so, processing continues at step 1712. At step 1712 (i.e., " $r.lhs$ subset of $r_2.lhs$?"), a determination is made whether the left-hand side of r is a subset of the left-hand side of r_2 . If the subset condition is true, processing continues at step 1714 to remove the common items out of the right-hand side of r (i.e., perform the

operation $r_2.rhs=r_2.rhs$ and not $r.rhs$). Processing continues at step 1706 to process any remaining relationships against r .

If it is determined at step 1702 that all of the relationships have been processed as r , processing continues at step 1716. At step 1716 (i.e., "all relationships in table processed?") a determination is made whether all of the relationships have been processed. If so, processing ends at step 1718. If not, processing continues at step 1720 to get another relationship as r . At step 1722 (i.e., " $r.rhs=zeros?$ "), a determination is made whether the right-hand side of r has been set to zero has a result of compression. If not, processing continues at step 1716 to process any remaining relationships. If so, processing continues at step 1724 to delete r from the table. Processing continues at step 1716 to process any remaining relationships.

Global Relationships

Global relationships can be defined that are processed for all products. Thus, it is unnecessary to redefining a relationship for all of the products. The relationship can be defined as a global relationship that is applied for every product. If, however, partitioned relationship tables are used for each product, it is necessary to evaluate the relationships in a product's partitioned relationship tables as well as the global relationships. For example, the process flows described herein that operate on relationship tables can be performed once for the product's partitioned relationship tables and once for the global relationships.

Thus, a method and apparatus for maintaining and configuring systems has been provided.

We claim:

1. A method of configuring a system comprising the steps of:

generating a definition for said system, said definition containing one or more elements and being conveyed graphically using a set of product relationships, said product relationships identifying classifications for said one or more elements, said product relationships comprising an includes classification;

generating a set of part relationships between said one or more elements, said set of part relationships being conveyed graphically; and

configuring said system using said definition and said set of product relationships and said part relationships.

2. The method of claim 1 wherein said step of configuring further comprises the steps of:

receiving input from a configuration user;

validating said input based on said definition, said set of relationships, and a current configuration state; and

identifying a set of valid configuration options using said definition, said set of relationships and said current configuration state.

3. The method of claim 2 wherein said step of identifying further comprises the steps of:

determining whether a relationship in said set of relationships is active;

modifying said configuration to satisfy said active relationship when it is determined that a modification is needed;

making said relationship notActivateable when it is determined that said relationship is not active and the activation of said relationship would render the configuration invalid.

4. The method of claim 3 wherein said step of determining whether a relationship is active further comprises the step of determining whether the elements specified in the left-hand side of said relationship are present in said configuration.

5. The method of claim 4 wherein said step of determining whether the elements specified in the left-hand side of said relationship are present in said configuration further comprises the step of performing a bit-level comparison of the bits corresponding to the elements in said left-hand side of the relationship with a "selected" bit vector.

6. The method of claim 3 wherein said relationship is an includes relationship, said step of modifying further comprises the step of including in said configuration elements identified in a right-hand side of said relationship.

7. The method of claim 3 wherein said relationship is an excludes relationship, said step of modifying further comprises the step of marking said elements identified in the right-hand side of said relationship as excluded and notSelectable.

8. The method of claim 3 wherein said relationship is a removes relationship, said step of modifying further comprises the step of removing said elements identified in the right-hand side of said relationship from said configuration.

9. The method of claim 3 wherein said relationship is a requires choice relationship, said step of modifying further comprises the steps of:

determining whether a plurality of elements identified in a right-hand side of said relationship are present in said configuration such that said relationship is satisfied;

modifying said configuration to satisfy said requires choice relationship when it is determined that only one path exists to satisfy said relationship and said relationship is not already satisfied.

10. The method of claim 3 wherein said step of making said relationship notActivateable further comprises the step of marking one or more elements in said left-hand side of said relationship as notSelectable.

11. The method of claim 2 wherein said definition, each of said set of relationships, and said current configuration state are expressed as bit vectors having bits that correspond to elements.

12. The method of claim 2 wherein said step of validating further comprises the step of determining whether the relationships other than those marked notActivateable in said set of relationships can be satisfied.

13. The method of claim 2 wherein said step of generating a definition selects said one or more elements for inclusion in said definition, said input is made without regard for the order in which said one or more elements are selected for said definition.

14. The method of claim 2 wherein said configuration state identifies a set of selected elements and a set of selectable elements for said system, said input capable of selecting any member of said set of selectable elements and being made despite the order in which members of said set of selected elements were selected.

15. The method of claim 1 wherein a parts catalog is used to store information associated with said one or more elements, said information including an identifier and description.

16. A method for defining a system that can be configured comprising the steps of:

identifying an element of said system;

specifying said element as an included element when said element is automatically included in the configuration;

specifying said element as an optional element when said element is not a necessary component of said system;

specifying said element as a required choice when said element is a group that contains one or more members from which to choose.

23

17. The method of claim 16 wherein said element is a member of a first element set said first element set containing one or more elements, said method further comprising the steps of:

- identifying a second element set of said system, said second element set containing one or more elements; and
- specifying a relationship between said first element set and said second element set.

18. The method of claim 17 wherein said step of specifying further comprises the steps of:

- specifying that said second element set is to be included in said system when said first element set is present in said system;
- specifying that said second element set is to be excluded from said system when said first element set is present in said system; and
- specifying that said second element set is to be removed from said system when said first element set is present in said system.

19. The method of claim 16 further comprising the steps of:

- specifying a group of elements;
- specifying a relationship between said first element set and said group of elements such that one or more elements in said group of elements must be chosen when said first element set is present in said system.

20. An article of manufacture comprising:

- a computer usable medium having computer readable program code embodied therein for configuring a system comprising:

- computer readable program code configured to cause a computer to generate a definition for said system, said definition containing one or more elements and being conveyed graphically using a set of product relationships, said product relationships identifying classifications for said one or more elements, said product relationships comprising an includes classification;

- computer readable program code configured to cause a computer to generate a set of part relationships between said one or more elements, said set of part relationships being conveyed graphically; and

- computer readable program code configured to cause a computer to configure said system using said definition and said set of product relationships and said part relationships.

21. The article of manufacture of claim 20 wherein said computer readable program code configured to cause a computer to configure further comprises:

- computer readable program code configured to cause a computer to receive input from a configuration user;
- computer readable program code configured to cause a computer to validate said input based on said definition, said set of relationships, and a current configuration state; and

- computer readable program code configured to cause a computer to identify a set of valid configuration options using said definition, said set of relationships and a current configuration state.

22. The article of manufacture of claim 21 wherein said computer readable program code configured to cause a computer to identify further comprises:

- computer readable program code configured to cause a computer to determine whether a relationship in said set of relationships is active;

24

computer readable program code configured to cause a computer to modify said configuration to satisfy said active relationship when it is determined that a modification is needed;

computer readable program code configured to cause a computer to make said relationship notActivatable when it is determined that said relationship is not active and the activation of said relationship would render the configuration invalid.

23. The article of manufacture of claim 22 wherein said computer readable program code configured to cause a computer to determine whether a relationship is active further comprises computer readable program code configured to cause a computer to determine whether the elements specified in the left-hand side of said relationship are present in said configuration.

24. The article of manufacture of claim 23 wherein said computer readable program code configured to cause a computer to determine whether the elements specified in the left-hand side of said relationship are present in said configuration further comprises computer readable program code configured to cause a computer to perform a bit-level comparison of the bits corresponding to the elements in said left-hand side of the relationship with a "selected" bit-vector.

25. The article of manufacture of claim 22 wherein said relationship is an includes relationship, said computer readable program code configured to cause a computer to modify further comprises computer readable program code configured to cause a computer to include in said configuration elements identified in a right-hand side of said relationship.

26. The article of manufacture of claim 22 wherein said relationship is an excludes relationship, said computer readable program code configured to cause a computer to modify further comprises computer readable program code configured to cause a computer to mark said elements identified in the right-hand side of said relationship as excluded and notSelectable.

27. The article of manufacture of claim 22 wherein said relationship is a removes relationship, said computer readable program code configured to cause a computer to modify further comprises computer readable program code configured to cause a computer to remove said elements identified in the right-hand side of said relationship from said configuration.

28. The article of manufacture of claim 22 wherein said relationship is a requires choice relationship, said computer readable program code configured to cause a computer to modify further comprises:

- computer readable program code configured to cause a computer to determine whether a plurality of elements identified in a right-hand side of said relationship are present in said configuration such that said relationship is satisfied;

- computer readable program code configured to cause a computer to modify said configuration to satisfy said requires choice relationship when it is determined that only one path exists to satisfy said relationship and said relationship is not already satisfied.

29. The article of manufacture of claim 22 wherein said computer readable program code configured to cause a computer to make said relationship notActivatable further comprises computer readable program code configured to cause a computer to mark one or more elements in said left-hand side of said relationship as notSelectable.

30. The article of manufacture of claim 21 wherein said definition, each of said set of relationships, and said current configuration state are expressed as bit vectors having bits that correspond to elements.

25

31. The article of manufacture of claim 21 wherein said code configured to cause a computer to validate further comprises program code configured to cause a computer to determine whether the relationships other than those marked notActivatable in said set of relationships can be satisfied.

32. The article of manufacture of claim 21 wherein said code configured to cause a computer to generate a definition selects said one or more elements for inclusion in said definition, said input is made without regard for the order in which said one or more elements are selected for said definition.

33. The article of manufacture of claim 21 wherein said configuration state identifies a set of selected elements and a set of selectable elements for said system, said input capable of selecting any member of said set of selectable elements and being made despite the order in which members of said set of selected elements were selected.

34. The article of manufacture of claim 17 wherein a parts catalog is used to store information associated with said one or more elements, said information including an identifier and description.

35. An article of manufacture comprising

a computer usable medium having computer readable program code embodied therein for defining a system that can be configured comprising:

computer readable program code configured to cause a computer to identify an element of said system;

computer readable program code configured to cause a computer to specify said element as an included element when said element is automatically included in the configuration;

computer readable program code configured to cause a computer to specify said element as an optional element when said element is not a necessary component of said system;

computer readable program code configured to cause a computer to specify said element as a required choice when said element is a group that contains one or more members from which to choose.

36. The article of manufacture of claim 35 wherein said element is a member of a first element set said first element set containing one or more elements, said article of manufacture further comprising:

computer readable program code configured to cause a computer to identify a second element set of said system, said second element set containing one or more elements; and

computer readable program code configured to cause a computer to specify a relationship between said first element set and said second element set.

37. The article of manufacture of claim 36 wherein said computer readable program code configured to cause a computer to specify further comprises:

computer readable program code configured to cause a computer to specify that said second element set is to be included in said system when said first element set is present in said system;

computer readable program code configured to cause a computer to specify that said second element set is to be excluded from said system when said first element set is present in said system; and

computer readable program code configured to cause a computer to specify that said second element set is to be removed from said system when said first element set is present in said system.

38. The article of manufacture of claim 35 further comprising:

26

computer readable program code configured to cause a computer to specify a group of elements;

computer readable program code configured to cause a computer to specify a relationship between said first element set and said group of elements such that one or more elements in said group of elements must be chosen when said first element set is present in said system.

39. A method of configuring a system comprising the steps of:

translating a system definition from an external representation comprising a set of intuitive relationships between components of said system definition into an internal representation comprising a set of internal relationships;

validating configuration input using said internal representation; and

modifying a configuration state based on said user input.

40. The method of claim 39 wherein said step of translating further comprises the step of:

removing a plurality of hierarchical levels from said external representation such that said internal representation is a single level.

41. The method of claim 40 wherein said step of translating further comprises the step of:

creating a set of relationships based on said hierarchical levels, each one in said set of relationships having a first set of components and a second set of components and a relationship operator.

42. The method of claim 41 wherein said relationship operator is chosen from the operators include, exclude, remove, and requires choice.

43. The method of claim 39 wherein each of said internal relationships has a left-hand side, a relationship operator and a right-hand side, said step of translating further comprises the steps of:

determining whether a group of components exists on said left-hand side of one of said internal relationships;

performing the following steps when a group of components exist on said left-hand side:

creating a new component for said group; and
creating a new relationship for each component of said group of components such that each component will include said new component.

44. The method of claim 39 wherein a type of said internal relationships is an include relationship, said include relationship identifies a second set of components that are included in said system when all of a first set of components exist in said system, said step of translating further comprises the steps of:

determining whether a subset of the union of said first and second sets of components in a first includes relationship comprises a first set of components in a second includes relationship; and

generating a combined includes relationship, said combined relationship comprises said first set of components of said first includes relationship and said second set of components from said first and second includes relationships when said subset of the union of said first and second sets of components in said first includes relationship comprises a first set of components in a second includes relationship.

45. The method of claim 39 wherein a type of said internal relationships is a requires choice relationship, said requires choice relationship identifies a second set of components

from which a number of components must be chosen when a first set of components exist in said system definition, said step of translating further comprises the steps of:

determining whether a second set of components of a second requires choice relationship is a proper subset of a second set of components of a first requires choice relationship;

performing the following steps when said condition exists:

creating a new relationship such that said new relationship excludes a second set of components of said new relationship when a first set of components of said new relationship is selected, said new relationship's first set of components is the union of the first set of components of said first and second requires choice relationships and said new relationship's second set of components is the difference of the second set of components of said first and second requires choice relationship.

46. The method of claim 39 wherein a type of said internal relationships is an includes relationship, said includes relationship identifies a second set of components that are included in said system when all of a first set of components exist in said system, said step of translating further comprises the steps of:

determining whether the second set of components of an includes relationship are contained in a first set of components of another internal relationship;

replacing said second set of components of said includes relationship in said another internal relationship with said first set of components of said includes relationship when said second set of components of an includes relationship are contained in a first set of components of another internal relationship.

47. The method of claim 39 said internal representation comprises multiple types of relationships, each type of relationship having a first and second set of components and an operation, said operation is performed on said second set of components when said first set of components exist in said system.

48. The method of claim 47 wherein said step of translating further comprises the steps of:

determining whether a first and second relationship are the same type of relationship;

performing the following steps when they are the same type:

determining whether the first relationship's first set of components is a proper subset of the second relationship's first set of components;

removing the first relationship's second set of components that exist from the second relationship's second set of components when said proper subset exists;

removing said second relationship when said second relationship's second set of components is an empty set.

49. The method of claim 47 wherein said step of validating further comprises the steps of:

determining whether a component selected by said configuration input is selectable; and

invalidating said configuration input when said component is not selectable.

50. The method of claim 49 wherein said one of said internal relationships is an includes relationship, said includes relationship causing a second set of components to be included in said system when all members of a first set of

components exist in said system, said second step of determining further comprises the step of:

determining whether any member of the second set of components cannot be included in said system.

51. The method of claim 49 wherein said one of said internal relationships is an excludes relationship, said excludes relationship causing a second set of components to be excluded from said system when all members of a first set of components exist in said system, said second step of determining further comprises the step of:

determining whether any member of the second set of components already exists in said system.

52. The method of claim 49 wherein said one of said internal relationships is an removes relationship, said removes relationship causing a second set of components to be removed from said system when all members of a first set of components exist in said system, said second step of determining further comprises the step of:

determining whether any member of the second set of components has been selected for inclusion in said system by a configuration user.

53. The method of claim 49 wherein said one of said internal relationships is a requires choice relationship, said requires choice relationship requiring a selection of a number of members of a second set of components when all members of a first set of components exist in said system, said second step of determining further comprises the step of:

determining whether said number of members of said second set of components cannot be selected for addition to said system.

54. The method of claim 47 wherein said step of modifying further comprises the steps of:

determining whether the performance of said operation of one of said internal relationships would result in an invalid configuration;

causing said one of said internal relationships to become inactivateable.

55. The method of claim 54 wherein said step of causing further comprises the step of excluding the last remaining member of said first set of components such that said last remaining member cannot be added to said system, said last remaining member being the only member of said first set of components that has not already been selected.

56. The method of claim 39 wherein said step of translating further comprises the step of:

mapping said intuitive relationships of said external representation into a set of relationships in said internal representation.

57. The method of claim 39 wherein said internal representation is stored as a set of bit vectors wherein each bit vector includes a bit that corresponds to one of said components of said system definition, the impact of changing the number of said components of said system definition thereby being minimal.

58. The method of claim 57 wherein said step of validating comprising a plurality of bit-level operations.

59. The method of claim 39 wherein said step of modifying further comprises the step of updating said configuration state to represent which of said components of said system definition are selectable, selected, included, and excluded.

60. A method of configuring a system comprising the steps of:

generating a definition for said system, said definition containing one or more elements and being conveyed graphically using a set of product relationships;

generating a set of part relationships between said one or more elements, said set of part relationships being conveyed graphically using a set of part relationships; and

receiving input from a configuration user;

validating said input based on said definition, said set of product relationships, said set of part relationships, and a current configuration state; and

identifying a set of valid configuration options using said definition, said set of product relationships, said set of part relationships and said current configuration state.

61. The method of claim 60 wherein said step of identifying further comprises the steps of:

determining whether a relationship in said set of relationships is active;

modifying said configuration to satisfy said active relationship when it is determined that a modification is needed;

making said relationship notActivateable when it is determined that said relationship is not active and the activation of said relationship would render the configuration invalid.

62. The method of claim 61 wherein said step of determining whether a relationship is active further comprises the step of determining whether the elements specified in the left-hand side of said relationship are present in said configuration.

63. The method of claim 62 wherein said step of determining whether the elements specified in the left-hand side of said relationship are present in said configuration further comprises the step of performing a bit-level comparison of the bits corresponding to the elements in said left-hand side of the relationship with a "selected" bit vector.

64. The method of claim 61 wherein said relationship is an includes relationship, said step of modifying further comprises the step of including in said configuration elements identified in a right-hand side of said relationship.

65. The method of claim 61 wherein said relationship is an excludes relationship, said step of modifying further comprises the step of marking said elements identified in the right-hand side of said relationship as excluded and notSelectable.

66. The method of claim 61 wherein said relationship is a removes relationship, said step of modifying further comprises the step of removing said elements identified in the right-hand side of said relationship from said configuration.

67. The method of claim 61 wherein said relationship is a requires choice relationship, said step of modifying further comprises the steps of:

determining whether a plurality of elements identified in a right-hand side of said relationship are present in said configuration such that said relationship is satisfied;

modifying said configuration to satisfy said requires choice relationship when it is determined that only one path exists to satisfy said relationship and said relationship is not already satisfied.

68. The method of claim 61 wherein said step of making said relationship notActivateable further comprises the step of marking one or more elements in said left-hand side of said relationship as notSelectable.

69. The method of claim 60 wherein said definition, each of said set of relationships, and said current configuration state are expressed as bit vectors having bits that correspond to elements.

70. The method of claim 60 wherein said step of validating further comprises the step of determining whether the

relationships other than those marked notActivateable in said set of relationships can be satisfied.

71. The method of claim 60 wherein said step of generating a definition selects said one or more elements for inclusion in said definition, said input is made without regard for the order in which said one or more elements are selected for said definition.

72. The method of claim 60 wherein said configuration state identifies a set of selected elements and a set of selectable elements for said system, said input capable of selecting any member of said set of selectable elements and being made despite the order in which members of said set of selected elements were selected.

73. An article of manufacture comprising:

a computer usable medium having computer readable program code embodied therein for configuring a system comprising:

computer readable program code configured to cause a computer to generate a definition for said system, said definition containing one or more elements and being conveyed graphically using a set of product relationships;

computer readable program code configured to cause a computer to generate a set of part relationships between said one or more elements, said set of part relationships being conveyed graphically using a set of part relationships; and

computer readable program code configured to cause a computer to receive input from a configuration user; computer readable program code configured to cause a computer to validate said input based on said definition, said set of product relationships, said set of part relationships, and a current configuration state; and

computer readable program code configured to cause a computer to identify a set of valid configuration options using said definition, said set of product relationships, said set of part relationships and said current configuration state.

74. The article of manufacture of claim 73 wherein said computer readable program code configured to cause a computer to identify further comprises:

computer readable program code configured to cause a computer to determine whether a relationship in said set of relationships is active;

computer readable program code configured to cause a computer to modify said configuration to satisfy said active relationship when it is determined that a modification is needed;

computer readable program code configured to cause a computer to make said relationship notActivateable when it is determined that said relationship is not active and the activation of said relationship would render the configuration invalid.

75. The article of manufacture of claim 74 wherein said computer readable program code configured to cause a computer to determine whether a relationship is active further comprises computer readable program code configured to cause a computer to determine whether the elements specified in the left-hand side of said relationship are present in said configuration.

76. The article of manufacture of claim 75 wherein said computer readable program code configured to cause a computer to determine whether the elements specified in the left-hand side of said relationship are present in said configuration further comprises computer readable program code configured to cause a computer to perform a bit-level

31

comparison of the bits corresponding to the elements in said left-hand side of the relationship with a "selected" bit-vector.

77. The article of manufacture of claim 74 wherein said relationship is an includes relationship, said computer readable program code configured to cause a computer to modify further comprises computer readable program code configured to cause a computer to include in said configuration elements identified in a right-hand side of said relationship.

78. The article of manufacture of claim 74 wherein said relationship is an excludes relationship, said computer readable program code configured to cause a computer to modify further comprises computer readable program code configured to cause a computer to mark said elements identified in the right-hand side of said relationship as excluded and notSelectable.

79. The article of manufacture of claim 74 wherein said relationship is a removes relationship, said computer readable program code configured to cause a computer to modify further comprises computer readable program code configured to cause a computer to remove said elements identified in the right-hand side of said relationship from said configuration.

80. The article of manufacture of claim 74 wherein said relationship is a requires choice relationship, said computer readable program code configured to cause a computer to modify further comprises:

computer readable program code configured to cause a computer to determine whether a plurality of elements identified in a right-hand side of said relationship are present in said configuration such that said relationship is satisfied;

computer readable program code configured to cause a computer to modify said configuration to satisfy said

32

requires choice relationship when it is determined that only one path exists to satisfy said relationship and said relationship is not already satisfied.

81. The article of manufacture of claim 74 wherein said computer readable program code configured to cause a computer to make said relationship notActivateable further comprises computer readable program code configured to cause a computer to mark one or more elements in said left-hand side of said relationship as notSelectable.

82. The article of manufacture of claim 73 wherein said definition, each of said set of relationships, and said current configuration state are expressed as bit vectors having bits that correspond to elements.

83. The article of manufacture of claim 73 wherein said code configured to cause a computer to validate further comprises program code configured to cause a computer to determine whether the relationships other than those marked notActivateable in said set of relationships can be satisfied.

84. The article of manufacture of claim 73 wherein said code configured to cause a computer to generate a definition selects said one or more elements for inclusion in said definition, said input is made without regard for the order in which said one ore more elements are selected for said definition.


85. The article of manufacture of claim 73 wherein said configuration state identifies a set of selected elements and a set of selectable elements for said system, said input capable of selecting any member of said set of selectable elements and being made despite the order in which members of said set of selected elements were selected.

* * * * *

PATENT APPLICATION SERIAL NO. **08/707187**

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

210 TL 10/10/96 08707187
1 101 1,744.00 CK

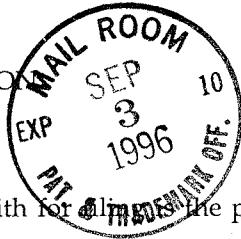
BAR CODE LABEL					
		U.S. PATENT APPLICATION			
SERIAL NUMBER		FILING DATE	CLASS	GROUP ART UNIT	
08/707,187		09/03/96	395	2316	
APPLICANT	NEERAJ GUPTA, AUSTIN, TX; VENKY VEERARAGHAVAN, AUSTIN, TX; AJAY AGARWAL, AUSTIN, TX.				
	CONTINUING DATA*** VERIFIED _____				
	FOREIGN/PCT APPLICATIONS*** VERIFIED _____				
	FOREIGN FILING LICENSE GRANTED 11/30/96				
STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS	FILING FEE RECEIVED	ATTORNEY DOCKET NO.
TX	24	59	5	\$1,894.00	85160.915
ADDRESS	HECKER & HARRIMAN SUITE 1600 2029 CENTURY PARK EAST LOS ANGELES CA 90067				
TITLE	METHOD AND APPARATUS FOR MAINTAINING AND CONFIGURING SYSTEMS				
This is to certify that annexed hereto is a true copy from the records of the United States Patent and Trademark Office of the application which is identified above.					
By authority of the COMMISSIONER OF PATENTS AND TRADEMARKS					
Date		Certifying Officer			

08/707187

Our Ref.: 85160.915

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

The Hon. Commissioner of
Patents and Trademarks
BOX PATENT APPLICATION
Washington, D.C. 20231



Sir:

Transmitted herewith for ~~mailing~~ the patent application of:

INVENTOR(S): Neeraj Gupta, Venky Veeraraghavan and Ajay Agarwal

FOR: METHOD AND APPARATUS FOR MAINTAINING AND CONFIGURING SYSTEMS

Enclosed are:

XX 24 sheet(s) of Informal Drawing(s).

 An Assignment of the Invention to: _____

 A Declaration and Power of Attorney

The Filing Fee has been calculated as shown below:

	(Col. 1)	(Col. 2)	SMALL ENTITY		OTHER THAN A SMALL ENTITY	
For:	No. Filed	No. Extra	Rate	Fee	Rate	Fee
Basic Fee:				\$375		\$750.00
Total Claims:	59-20	39	x 11	\$0	x 22	\$858.00
Indep. Claims:	5-3	2	x 39	\$0	x 78	\$156.00
<input type="checkbox"/> Multiple Dependent Claim(s) Presented			+125	\$0	+250	\$0
* If the difference in Col. 1 is less than zero, enter "0" in Col. 2			TOTAL	\$0	TOTAL	\$1,764.00

XX The Commissioner is hereby authorized to charge payment of any additional fees associated with this communication, or credit any overpayment, to Deposit Account Number 08-1520. A duplicate of this authorization is enclosed.

XX A check in the amount of \$ 1,764.00 to cover the Filing Fee is enclosed.

 A check in the amount of \$ to cover the fee for Recordation of the Assignment is enclosed.

Respectfully submitted,

HECKER & HARRIMAN

Dated: September 3, 1996

J. D. Harriman, II
Reg. No. 31,967

2029 Century Park East, Suite 1600
Los Angeles, California 90067 CA
(310) 286-0377

CERTIFICATE OF MAILING

This is to certify that this correspondence is being deposited with the United States Postal Service as Express Mail Label No. EM300601754US in an envelope addressed to: Commissioner of Patents and Trademarks, Box Patent Application, Washington, D.C. 20231 on: September 3, 1996

Carole A. Quinn September 3, 1996
Signature : Carole A. Quinn Date

08/707187



85160.915

UNITED STATES PATENT APPLICATION

FOR

METHOD AND APPARATUS FOR
MAINTAINING AND
CONFIGURING SYSTEMS

INVENTORS:

NEERAJ GUPTA
VENKY VEERARAGHAVAN
AJAY AGARWAL

PREPARED BY:

HECKER & HARRIMAN
2029 Century Park East
Suite 1600
Los Angeles, CA 90067

(310) 286-0377

CERTIFICATE OF MAILING
This is to certify that this correspondence is being
deposited with the United States Postal Service as Express
Mail Label No. EM300601754 US in
an envelope addressed to: Commissioner of Patents and
Trademarks, Washington, D.C. 20231 on:
September 3, 1996
Carol A. Quinn 9/3/96
Signature Date



BACKGROUND OF THE INVENTION

1. FIELD OF THE INVENTION

This invention relates to maintaining and configuring systems.

2. BACKGROUND ART

5 A system is comprised of components. Before a system can be built the components of the system must be identified. To configure a system, a user must select the parts to include in the system. Typically, one who is knowledgeable about a system and its components defines the system. Thus, for example, an automobile salesperson assists an automobile buyer in
10 determining the type and features of the automobile. The salesperson understands the features and options that are available to create a valid configuration. Some features and options cannot be combined. The selection of some features caused other features to be unavailable, etc. It would otherwise be difficult for the buyer to identify all of the features and
15 options available on the automobile that can be combined to create a valid configuration.

Computer systems have been developed to assist one in configuring a system. However, these systems use a configuration language to define a system. Like a programming language, a configuration language uses a
20 syntax that must be understood by a user who is maintaining the data (i.e., a data maintainer). To use one of these configuration systems, it is necessary

for a data maintainer to understand the configuration language. This limits the number of users who are able to use the configuration systems. That is, the level of sophistication needed to communicate with the configuration system (through a configuration language) results in less sophisticated users
5 being unable to use the system.

In addition, configuration systems impose a flow or ordering to the user operations. For example, a user is required to remove components from the system in reverse of the order in which they were chosen. Thus, a user may be forced to remove components that the user wants to keep in the
10 configuration to remove an unwanted component. A novice user may have perform many removal operations before achieving an acceptable configuration. If the novice user is required to remove components in a preset order, the user can become frustrated or confused and abort the configuration process.

15 These systems are designed for a more sophisticated user that has knowledge of the system that is being configured as well as the configuration system used to configure the system. A end user such as an automobile shopper would have difficulty using these systems.

Further, to use these systems a user must be trained to understand the
20 configuration language. Thus, a user who otherwise has knowledge of the systems that are being configured must undergo training to be able to use

these configuration systems to configure systems. This leads to increased expenditures such as for training.

SUMMARY OF THE INVENTION

The invention provides the ability to interactively select and configure a product among a set of related products based on availability and compatibility of features and options. It does not impose an order in the
5 selection of products, features or options; only valid selections can be made at any time. To create an electronic representation of the product information to achieve the above goal, the invention provides a framework for defining a product line.

A product line is defined as a set of related products. A product line
10 has a set of products that contain parts, or components. Parts used to define a product are selected from a parts catalog. Parts in a product definition are related or classified as: included (parts that are included by default), required choices (a choice among a group of parts that must be made to achieve a valid configuration), optional (parts that can be optionally included in the
15 configuration).

Relationships can be defined between the parts in a product definition. A relationship relates a first set of parts with a second set of parts. A set can include multiple parts. The incorporation of parts in a set can be arbitrary. That is, a multi-part set can contain parts that are otherwise
20 unrelated. For example, a set can contain parts such as an engine, sun roof and a color. These parts seem to be unrelated, however, it is possible to

combine them into a relationship set for purposes of forming a relationship using the present invention.

Preferably, the part relationships are: included, excluded, removed, and requires choice. An included part is included automatically. A part is
5 excluded from the configuration when its inclusion would result in an invalid configuration. A part may be removed when another part is added. Thus, when a first part exists in the configuration and a second part is added, the first part is removed from the configuration. The requires choice relationship is used to allow a set of choices to be made from a group of
10 parts. The number of parts chosen is limited to a valid bounds specification. The relations that are created between parts within a product are enforced only on that particular product. However, if some part-to-part relationships are to be enforced on all products within a product line, then the relations are generated once and enforced for all products.

15 A maintenance system is used to define a product. Using the maintenance system, a product can be defined using the product classifications and the part relationships. A graphical user interface (GUI) is used to allow the user to interactively generate a definition. Instead of configuration languages, GUI operations such as drag and drop and selection
20 operations can be used to specify a definition. The notions of included, optional and required choice are easily comprehensible to a user. Further, the idea that parts have interrelationships is also easily understood. Thus, a

product can be defined without having to learn a complicated configuration language.

A configuration system is used to configure a system using a definition created by the maintenance system. The configuration system ensures that the current configuration state is always valid. The user can select and unselect parts in any order. When user input is received, the configuration system validates the input based on the current state of the configuration. In addition, the configuration system identifies selections that could cause a valid configuration to become invalid. The configuration removes these selections from the set of possible selections so that the user does not make an invalid selection.

The configuration system evaluates the current state of a configuration based on the product definition, part relationships and state information. After receipt of input from a user, the configuration system evaluates relationships in both the forward and backward direction. Forward and backward evaluations can result in the addition or deletion of elements from the configuration.

The invention uses both an external and internal representation of a definition or definitions. A translation mechanism is used to translate an external representation into an internal representation. The external representation uses a conceptually understandable set of relationships for defining a system and the relationships between the components of the

85160.915 7

system. The invention takes the definition created by a user and supplements and compresses the definition when necessary to create an internal representation. The internal representation is used during configuration to initialize and validate a configuration based on user input.

5 During configuration, the invention maintains runtime information that is stored in tables and vectors. To achieve greater processing efficiency, the systems represents elements in a configuration (e.g., product, part, and group) as a bit in a bit vector. Thus, for example, a vector has a length that is equal to the total number of elements. An element's bit can be set or reset to
10 specify the state of the element in the current configuration. For example, a user vector can be used that specifies for each element whether the element has been selected by the user during the configuration. In addition, excluded and removed vectors identify whether an element is excluded or removed (respectively) from a configuration. Vectors can be used to identify whether
15 an element 1) has been selected (by the user or the configuration system), 2) is selectable, and 3) notSelectable.

Tables contain element relationships. A table is used to represent the includes, excludes, removes, and requires choice relationships, for example. Each table has a left-hand side and a right-hand side that corresponds to the
20 left-hand and right-hand sides of a relationship. In each case, the left-hand side is a bit vector that contains bits that correspond to elements. The includes, excludes and removes tables contain a bit vector in the right-hand

side that represents configuration elements. The right-hand side of the requires choice table is a pointer that points to an entry in a group table. The group table entry is a bit vector that identifies the elements that are contained in the group from which a choice is to be made. The right-hand
5 side of a requires choice table entry further includes minimum and maximum designations. Minimum and maximum values identify the minimum and maximum number of group members that are to be selected to satisfy a requires group relationship.

A bit vector implementation of relationships and internal runtime
10 state allows for fast and efficient computation of relationship based configuration. A comparison of bits can be performed in one machine instruction in most cases.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 provides an illustration of a computer system that can be used with the invention according to an embodiment of the invention.

Figure 2 provides an overview of the maintenance and configuration systems according to an embodiment of the invention.

Figure 3 illustrates examples of elements in a parts catalog according to an embodiment of the invention.

Figure 4 illustrates relationships between parts according to an embodiment of the invention.

Figure 5 provides an example of product classifications according to an embodiment of the invention.

Figure 6 provides an example of a GUI screen used in maintenance system 202 according to an embodiment of the present invention.

Figure 7 provides an block diagram illustrating the conversion process according to one embodiment of the invention.

Figures 8A-8B illustrate components of internal representation 706 according to an embodiment of the invention.

Figure 9 provides a process flow for processing a user selection according to an embodiment of the invention.

Figure 10 provides an example of a relationship evaluation process flow according to an embodiment of the invention.

Figure 11 provides an example of an unselect item process flow according to an embodiment of the invention.

5 Figure 12 provides a flow for translation processing according to an embodiment of the invention.

Figures 13A-13B provide an illustration of groups and nested groups according to an embodiment of the invention.

10 Figures 14A-14B provide an includes chain process flow according to an embodiment of the invention.

Figures 15A-15B provides an example of a subgroup excludes process flow according to an embodiment of the invention.

Figures 16A-16C provide an example of a relationship factorization process flow according to an embodiment of the invention.

15 Figures 17A-17B provide an illustration of a relationship compression process flow according to an embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

A method and apparatus for maintaining and configuring systems is described. In the following description, numerous specific details are set forth in order to provide a more thorough description of the present invention. It will be apparent, however, to one skilled in the art, that the present invention may be practiced without these specific details. In other instances, well-known features have not been described in detail so as not to obscure the invention.

The present invention can be implemented on a general purpose computer such as illustrated in Figure 1. A keyboard 110 and mouse 111 are coupled to a bi-directional system bus 118. The keyboard and mouse are for introducing user input to the computer system and communicating that user input to CPU 113. The computer system of Figure 1 also includes a video memory 114, main memory 115 and mass storage 112, all coupled to bi-directional system bus 118 along with keyboard 110, mouse 111 and CPU 113. The mass storage 112 may include both fixed and removable media, such as magnetic, optical or magnetic optical storage systems or any other available mass storage technology. Bus 118 may contain, for example, 32 address lines for addressing video memory 114 or main memory 115. The system bus 118 also includes, for example, a 32-bit DATA bus for transferring DATA between and among the components, such as CPU 113, main memory 115, video memory 114 and mass storage 112. Alternatively,

multiplex DATA/address lines may be used instead of separate DATA and address lines.

In the preferred embodiment of this invention, the CPU 113 is a 32-bit microprocessor manufactured by Motorola, such as the 680X0 processor or a
5 microprocessor manufactured by Intel, such as the 80X86, or Pentium processor. However, any other suitable microprocessor or microcomputer may be utilized. Main memory 115 is comprised of dynamic random access memory (DRAM). Video memory 114 is a dual-ported video random access memory. One port of the video memory 114 is coupled to video amplifier
10 116. The video amplifier 116 is used to drive the cathode ray tube (CRT) raster monitor 117. Video amplifier 116 is well known in the art and may be implemented by any suitable means. This circuitry converts pixel DATA stored in video memory 114 to a raster signal suitable for use by monitor 117. Monitor 117 is a type of monitor suitable for displaying graphic images.

15 The computer system described above is for purposes of example only. The present invention may be implemented in any type of computer system or programming or processing environment.

The invention maintains and configures systems. The invention eliminates the need for a user to learn a configuration language or write
20 code to maintain and/or configure a system. A user interface uses various operations such as drag and drop and item selection to define a product, for example. Elements that comprise a definition (e.g., of a product) can be

added or removed in any order. No order is imposed on the user. There is no requirement, for example, that the user remove parts of a product in the order in which they were added. No fixed flow or order is required to edit a given product.

5 A definition includes an identification of the components that comprise the definition and the interrelationships between the components. The interrelationships are conceptually easy for the user to understand. The same relationships are used to define and configure any system. An external representation that includes these relationships allows a user to view and
10 maintain the definition. The external representation is translated into an internal representation that is designed to decrease processing time and increase response time.

In addition, during a configuration session, the invention is capable of allowing a user to select or unselect configuration items without imposing a
15 flow on the user. There is no order imposed on the user in terms of the sequence in which items are selected for the configuration or unselected from the configuration. For example, it is not necessary to select a product before choosing options. The invention can identify the products that are still available based on the options that have already been selected. Further,
20 a user can unselect an item (e.g., delete the item from the configuration) without regard to the order in which it was selected (e.g., added to the configuration).

Examples of systems that can be maintained or configured using the invention include automobiles, computers, time clock machines, and shoes. Terms such as part, product line, parts catalog, and product are used herein for illustration purposes. It should be apparent that this invention can be
5 used to configure systems that are not limited to products and product lines, etc.

Figure 2 provides an overview of the maintenance and configuration systems according to an embodiment of the invention. Maintenance system 202 maintains a parts catalog 204, parts relationships 206 and product
10 definitions 208. Maintenance system 202 uses a user interface that includes the ability to add items to parts catalog 204 and specify part relationships 206 and product definitions 208. The user interface displays a set of hierarchies that provide a conceptually easier way of viewing a definition. The invention maps the set of hierarchies (an external representation) to an
15 internal representation.

The internal representation is used by configuration system 212 to maintain and configure systems based on user input. Configuration system 212 provides the ability to specify a product. Configuration system 212 verifies the product specification. Using configuration system 212, a user can
20 interactively select and configure a product among a set of related products based on availability and compatibility of features and options. It does not impose an order in the selection of products, features or options. The

configuration system 212 allows the user to only make valid selections. However, selections can be made in any order.

Parts catalog 204 consists of parts that are components of products. Similar parts are grouped together to form a part hierarchy. Easy
5 maintenance of relationships is achieved by the hierarchy. For example, when a group of parts is assigned a behavior, all the members inherit that behavior automatically. Figure 3 illustrates examples of elements in a parts catalog according to an embodiment of the invention. Referring to Figure 3, a parts catalog (e.g., parts catalog 204) contains parts 302-310, 316, and 318. A
10 parts catalog can also contain a group of parts such as group 312. Group 312 contains parts 306-310. A group can contain other groups. For example, group 314 contains group 312 and parts 316-318. Behavior assigned to elements of group 314 is inherited by members of group 312.

Part-to-part relationships can be created between parts within a
15 product. Relationships are defined between a first set of parts and a second set of parts. During configuration, when all of the members of the first set of parts are selected, the relationship between the two sets is enforced on the parts in the second set. In the preferred embodiment, there are four kinds of relationships between parts: requires choice, includes, can't work with (or
20 excluded), and removes. Figure 4 illustrates relationships between parts according to an embodiment of the invention.

Part 402 includes part 404. The includes relation causes a set of parts in a second set (e.g., part 404) to be included in the configuration when a first set of parts (e.g., part 402) is selected. For example, a luxury package includes a CD player; when the luxury package is selected, a CD player is included in the configuration. The can't work with (or excluded) relation ensures that a set of parts from a second set are never in the same configuration as parts in the first set. For example, part 402 (e.g., sun roof) can't work with part 406 (e.g., a roof-top antenna). When part 402 is selected, part 406 cannot be selected. Part 406 is excluded such that it cannot be selected.

When part 402 is selected, part 414 is removed from the configuration. The removes relation causes items that are included in a second set of to be removed from the configuration when the left side is selected. For example, when a high end stereo is selected the standard stereo is removed from the configuration.

The requires choice relation recognizes that a choice (between a minimum and maximum number) has to be made between a second set of parts (or members of a group) to ensure a valid configuration when the parts in a first set are selected. Part 406 (e.g., climate control feature) requires that a choice be made among parts 408-410 (e.g., 1 zone A/C or 2 zone A/C). A requires choice relationship requires that a number of items be selected based on minimum and maximum values from the right-hand side of the relationship to satisfy the relationship.

Parts 408 and 410 can be combined to form group 416. Group 416 can be defined by the user. If the user does not define group 416, the invention preferably creates group 416 to contain parts 408 and 410. Thus, when two or more parts are defined on the right-hand side of a requires choice relationship, the invention preferably creates a group that contains these parts. The requires choice then becomes a requires choice on the group (e.g., group 416).

Parts are used to illustrate relationships in Figure 4. However, a relationship can contain groups. To illustrate, a group of parts can be substituted wherever a part is used in the illustration. Thus, for example, parts 402, 404, 406, 408, 410, and 414 can each be replaced by a group of parts. Thus, for example, if parts 402 and 404 are replaced by groups (e.g., groups 402 and 404, respectively), when the members of group 402 are selected, the members in group 406 are included in the configuration. To further illustrate, part 408 can be replaced by group 408. In that instance, group 408 is a member of group 416. A hierarchy of groups (e.g., group 408 contained within group 416) can be used in a relationship.

Parts can be combined to form an arbitrary grouping. Thus, it is not necessary to combine parts in a group based on a logical or intuitive relationship between the parts in the group. For example, a group can contain an engine, a color, and a sun roof.

The relations that are created between parts within a product are enforced only on that particular product. However, if some part-to-part relationships are to be enforced on all products within a product line, then the relations are created once and are enforced for all products. These
5 relationships are referred to as global relationships.

A product includes zero or more elements of parts catalog 204. Product definition 208 (see Figure 2) is generate by populating it with its component parts. Product definition 208 is generated by population of a product with its component parts. The parts within a product are classified
10 as one of three different types: included parts, required choices, or optional parts. A part that is not classified as one of these types is assumed to be unavailable in that product. Figure 5 provides an example of product classifications according to an embodiment of the invention.

An included part is a part that is included in a product by default. For
15 example, parts 504 and 506 are automatically included in product 502. for example, when a configuration user chooses the product definition product 502, the parts 504 and 506 are automatically included in the configuration. A required choices classification specifies that a choice among a group of parts has to be made to create a valid product configuration. For example, product
20 502 (e.g., automobile) can include a color group 518 containing red (part 508), green (part 510) and blue (part 512) that is a required choice. In configuring a product, the user must choose a color in this group to create a valid product

configuration. Parts 514 and 516 are optional parts. An optional part is not required for a valid configuration.

A product line is defined as a set of related products. The invention provides the framework for defining a product line. To define a product
5 line: all parts (components) in the product line are entered into parts catalog 204. An instance in products definition 208 is created that identifies parts from parts catalog 204 that are assigned to a product. Part relationships 206 can also be defined for the product.

Maintenance System

10 Parts catalog 204, part relationships 206, and product definition 208 are created and maintained using maintenance system 202. While any method can be used for creation and maintenance, a graphical user interface (GUI) is preferably used. A text editor could also be used for data entry, for example. Figure 6 provides an example of a GUI screen used in maintenance system
15 202 according to an embodiment of the present invention. One who uses the GUI screen (a maintainer) can define a product (e.g., an automobile). A product definition is used by a configuration user to configure a system defined in maintenance system 202. Parts catalog 204 uses a database or other type of storage and retrieval capability, for example, to store information
20 about its parts (e.g., part number, description, price, etc.).

The GUI screen of Figure 6 is divided into a product definition section 650 and a part relationship definition section 652. Pane 602 displays elements from parts catalog 204. Panes 604, 606, 608 are used to define a product. Panes 604, 606, and 608 represent classifications or product relationships. Panes 610 and 614 and relationship 612 are used to define part relationships.

A user can drag elements from pane 602 to panes 604-608 to define a product. For example, to include Part B in the product definition, Part B is dragged from pane 602 to pane 604. Alternatively, to drag parts B, C, D, and E, group A can be dragged from pane 602 to pane 604. Group A and its component parts (parts B, C, D, and E) are thereby included in the product definition. Similarly, a user can specify that a configuration user must choose a part from a group, e.g., Group I, by dragging one or more parts or a group into pane 606. An optional part or group can be identified by dragging an element, e.g., Group L, into pane 608. If an element from pane 602 is not moved to one of panes 604-608 it is assumed that the maintainer wants to exclude that element from the product that is being defined. The product-level relationships or classifications (or types) illustrated in Figure 5 can be defined using panes 604-608.

A subset of the product-level relationships map to a subset of the part relationships. For example, the include and requires choice product-level relationships map to the same named part relationships (as defined in

Figure 4 and shown in relationships 612). Minimum and maximum values set for a requires choice product-level relationship map to the minimum and maximum values of a requires choice part relationship. Elements that are not dragged into one of panes 604, 606, or 608 are considered to be
5 excluded.

In addition to defining the elements (and their types) that comprise the product, the maintainer can specify relationships between parts of the product. Panes 610 and 614 and relationship 612 are used to define relationships between parts. The maintainer can drag an element (or
10 elements) from pane 602 into pane 610. For example, the user can drag Part N from pane 602 into pane 610. The element(s) dragged into pane 610 are referred to as the left-hand side of the relationship. The element (or elements) from pane 602 that is related to the left-hand element(s) is dragged into pane 614. The element(s) dragged into pane 614 are referred to as the
15 right-hand side of the relationship. For example, the user can drag Part K from pane 602 to pane 614.

To identify the type of relationship that exists between the left-hand side and right-hand side elements, the maintainer chooses one of the four part-to-part relationships from relationship 612. For example, if the
20 maintainer chooses the includes relationship, the right-hand side elements are included in the product when a configuration user selects the left-hand side element(s). Thus, for example, Part N includes Part K. When Part N is

chosen by a configuration user, Part K is automatically included in the configuration.

When the maintainer completes the product definition using the GUI screen, the product definition is saved (e.g., in parts definition 208). The
5 product definition can then be used by a configuration user in configuration system 212.

External and Internal Representations

The GUI screen of Figure 6 illustrates a view of a product as seen by a user (e.g., maintainer and configuration user). A view of the product that is
10 seen by a user is referred to as the external representation. A user sees a product or product definition from a conceptual level that is easy to understand. The user does not need to express a definition or configuration using a language that has a syntax that is foreign to the user. A GUI having graphical selections and operations is preferably used to allow the user to see
15 the external representation. The external representation uses terminology that is familiar or intuitive to the maintainer. For example, an include product relationship defines elements that can be included in a product.

An external representation seen by the user is translated into an internal representation that can be used by the invention to process user
20 input. Figure 7 provides a block diagram illustrating the conversion process according to one embodiment of the invention. Compiler 704 is used to

perform the conversion. Compiler 704 generates internal representation 706 from external representation 702.

External representation 702 is the representation of a product definition that is created using the maintenance system of the invention.

5 External representation 702 provides a graphical representation of a system definition, its components and their interrelationships. Figure 6 can be used to display external representation 702. External representation 702 can include a multi-level hierarchy of parts that help to define the product at a conceptual level. The hierarchies are not required to drive a configuration
10 session, however. Therefore, compiler 704 flattens out the hierarchies. A set of relationships is generated using external representation 702 during a translation process.

The set of relationships generated for a system definition by compiler 704 comprise internal representation 706. Internal representation 706 is
15 generated from external representation 702 and can be used to control a configuration session. In addition to internal representation 706, a configuration state is used during the configuration session. Figures 8A-8B illustrate components of internal representation 706 and the configuration state according to an embodiment of the invention.

20 Referring to Figure 8A, vector 802 provides an example of a vector used to store state information. Vector 802 is comprised of bits. A set of p bits is used to represent the products that are known to maintenance system
85160.915

202 and configuration system 212. Each bit of the P bits corresponds to a product. A set of N bits is used to represent the parts. Each bit in the N -bit set corresponds to a part in parts catalog 204, for example. Vector 802 is therefore $P + N$ bits in length.

5 Vectors 804-810 provides implementation examples of vector 802. Vectors 804-810 are maintained by configuration system 212 in a configuration session. Vector 804 is a user vector that contains the set of elements that are selected by the user. Vector 804 is referred to as the user vector, or uVec. The set of elements that are included in the configuration
10 by the configuration system (e.g., an item that is automatically included when the user selects the left-hand in an includes relationship) is contained in included vector 806, or iVec. Removed vector 808, or rVec, contains the elements that are removed from the configuration by configuration system 212. Elements excluded from the configuration by configuration system 212
15 are identified in excluded vector 810, or xVec.

Each element that can be used to configure a system has a bit in these vectors. When an element is included, excluded, removed (or deleted), or selected, the corresponding bit in the appropriate vector (iVec, xVec, rVec, or uVec) is set. The use of vectors to represent elements results in increased
20 efficiency in processing. Bit-wise operations can then be used during processing. For example, it is possible to determine whether a group of elements have been selected by the user by performing a bit comparison

using the uVec. The increased efficiency results in optimal response times that are necessary for an interactive system such as is preferred for configuration system 212. The bit vector implementation used by the invention allows for fast and efficient computation of runtime algorithms.

- 5 A bit vector implementation of relationships and internal runtime state allows for fast and efficient computation of relationship based configuration.

Another vector that contains state information is required groups vector 812, or rgVec. The rgVec contains a bit for each of the required groups specified for a product. As previously discussed, the maintainer can include
10 a required group in a product definition by dragging a group from pane 602 to pane 606, for example (see Figure 6). The number of bits in rgVec can differ from the number of bits in vectors 802-810. The number of bits contained in rgVec corresponds to the total number of required groups defined for a product. Each bit in rgVec is set to indicate that it has not yet
15 been satisfied. It is unset (set to zero) when a required group has been satisfied. Thus, when a user attempts to accept a configuration, configuration system 212 can determine whether all of the required choices have been satisfied.

In addition to these vectors, additional vectors are used to determine a
20 current state. The current state is defined by three types of state information that track the state of parts in the configuration. The current state is visible to the user. The three state types are: selected, selectable, and notSelectable.

The selected state identifies all of the parts that are currently included in the configuration. The selectable state identifies the parts that are not in the configuration but that can be selected by the user for the configuration. The notSelectable state identifies the parts that are not in the configuration and cannot be selected by the user. Configuration system 212 ensures that only parts that can form a valid configuration are selectable; those that cannot are notSelectable.

The uVec, iVec, xVec, and rVec vectors can be used to generate the three visible states. The states are dynamic and can change based on user input. The following provides definitions using the formulae that can be used to generate the three visible states (+ is union and - is difference):

$$\begin{aligned} \text{Selected} &= (uVec + iVec) - rVec \\ \text{NotSelectable} &= xVec + rVec \\ \text{Selectable} &= [\text{AllItems}] - \text{Selected} - \text{NotSelectable} \end{aligned}$$

Computation of the visible state using C++, for example, can be accomplished using the following C++ code (the syntax assumes appropriate bit-wise operator overloads for the given operations):

$$\begin{aligned} \text{SelectedVec} &= (uVec \mid iVec) \& \sim rVec \\ \text{NotSelectableVec} &= xVec \mid rVec \\ \text{SelectableVec} &= \sim(\text{SelectedVec} \mid \text{NotSelectableVec}) \end{aligned}$$

The selected state is the union of the elements selected by the user or automatically included in the configuration minus the elements removed

by configuration system 212. The notSelectable state is the union of the elements that are excluded or removed. The selectable state is the set of all elements minus the elements already selected and the elements that cannot be selected. The selected, selectable, and notSelectable states can be generated
5 from these four vectors when needed. They can be stored as vectors that have the structure of vector 802.

Runtime tables are preferably used to store internal representation 706 for a configuration session. Figure 8B provides an example of runtime tables. Runtime table 822 provides an example of a runtime table. Runtime
10 table 822 can be used to store relationship information. Runtime table 822 has left-hand side 824 and right-hand side 826. Left-hand side 824 contains the left-hand side of a relationship (e.g., includes, removes, can't work with or excludes). The right-hand side of a relationship is stored in right-hand side 826.

15 Table 828 provides an example of runtime table 822. Table 828 can be an includes, excludes, or removes table, for example. Left-hand side 830 contains a bit vector (i.e., "01010...0"). Right-hand side 832 contains a bit vector (i.e., "10001...1"). Each bit in bit vectors 830 and 832 corresponds to an element in parts catalog 204 or a product. For example, if the product
20 definition is for an automobile, each "1" bit in bit vector "01010...0" corresponds to a part in parts catalog 204 that is contained in an automobile's product definition. Zero bits indicate that the part is not included in the

that identifies the parts contained in the group. An advantage that the latter (altered table 822) format has over the former (unaltered table 822) is that it is only necessary to define the members of a group once. Each product definition that includes the group can point to the entry in the group table
5 that contains the group definition.

One relationship table can be used for all products per relationship type. That is, a single includes table is used for includes relationships for all of the products. Alternatively, a table can be partitioned for each product. That is, the relationships associated with a product are segregated. For
10 example, there would be a includes relationship table for each product. This has the advantage of only having to process only those relationships that exist for the chosen product. When a product is unselected or not selectable, there is no need to process the tables associated with that product. The latter alternative is preferable because it lessens processing requirements.

15 Configuration System

Using configuration system 212, a configuration user can configure a system given a product definition (e.g., product definition 208) and part relationships associated with the product definition. Configuration system 212 accepts a configuration user's input and processes it to update the
20 configuration based on the user's input or notify the user that the input is invalid given the current state of the configuration and the system definition. Configuration system 212 ensures that the current configuration

state (determined by the product definition and order-independent user selection) is always valid.

User Selection

When user input is received by configuration system 212, it is
5 processed based on the current configuration state, the product definition and the parts relationships. Figure 9 provides a process flow for processing a user selection according to an embodiment of the invention.

At step 902, the user selects item n (e.g., part) preferably using a GUI screen. At step 904 (i.e., "n selectable?"), a determination is made whether
10 the item is selectable. Thus, for example, configuration system 212 determines whether the nth bit in the selectable state information is set. If it is not, configuration system 212 determines that the item is not selectable by the user and raises an error at step 922. Processing then continues at step 914 to update the interface (if necessary) and return control of the GUI screen to
15 the user. Processing of the current input ends at step 916.

If, however, it is determined at step 904 that the item selected by the user is selectable, processing continues at step 908 to set the nth bit in uVec 804. At step 910, the visible state information is updated as discussed above. At step 912, the relationships associated with the product definition are
20 evaluated as discussed below. Processing continues at step 914 to update the

interface (if necessary) and return control of the GUI screen to the user.

Processing of the current input ends at step 916.

Relationship Evaluation

Relationships associated with items contained in the product
5 definition are evaluated when user input is received. Configuration system
212 determines what relationships are active and inactive given the user
input (at step 912, for example). A relationship is active when all the items
on the left-hand side of the relationship are selected. A relationship is
inactive until all of the parts on the left-hand side of the relationship are
10 selected.

As previously discussed, relationship tables can be used to store the
left-hand and right-hand sides of a relationship. That is, a relationship can
be defined as a pair of bit vectors, the lhsVec (left-hand side) and the rhsVec
(right-hand side). A relationship is made active if the following expression
15 is true:

$$\text{SelectedVec} \ \& \ \text{lhsVec} == \text{lhsVec}$$

When a relationship is made active, updating the appropriate state
vector (iVec, xVec, or rVec) becomes:

$$\text{stateVec} \ |= \ \text{rhsVec}$$

An active relationship causes the configuration state to be updated. When an include relationship is activated, for example, parts identified on the right-hand side of the include relationship are added to the configuration, if they are not already included. The state is changed by
5 modifying iVec 806. A change in iVec 806 can change other state information such as the selected and selectable state information. Parts that are included as a result of the activation of the includes relation can result in a ripple effect that causes other relationships to become activated. Thus, additional state changes can occur as a result of the include relationship's
10 activation.

To further illustrate, when a requires choice relation becomes active, configuration system 212 includes and excludes items of the given group where applicable: if exactly max (maximum number of) items of the given group are selected, the engine excludes all other items in the group (added
15 into xVec); otherwise, if less than min (minimum number of) items of the group are selected and only one valid selection path exists to make exactly min items selected, the engine includes (into iVec) the items that are needed (note that these must have been selectable items).

Configuration system 212 also ensures that no relationship that will
20 put the configuration into an invalid state can become active (for example: if b is Selected then the relationship [a] can't work with [b] should not be

active). If a relationship will make a configuration invalid, it is made notActivateable. For example, a relation should be made notActivateable, if:

- 1) an includes relationship has excluded (xVEc) parts on the right-hand side;
- 5 2) a excludes (can't work with) relationship has Selected parts on the right-hand side;
- 3) a removes relation has user-selected (uVec) parts on the right side; or
- 4) a requires choice relation has a group for which between [min, max]
- 10 parts can't be Selected.

In addition to the relationships created by a maintainer, there are some relationships that are dynamically created or deduced at runtime. If a relation becomes notActivateable and all but one item from the left-hand side is Selected, then the unSelected item is excluded by the engine. This

15 ensures that the relation will not become active.

Configuration system 212 evaluates each relationship forward and backward. In a forward evaluation, the left-hand side of a relationship is examined to determine whether all of its parts are included in the configuration. If they are, the relationship is considered to be active. An

20 active relationship causes the right-hand side of the relationship to then be evaluated. The evaluation is dependent on the type of relationship. An includes relationship causes the parts specified in the right-hand side to be included in the configuration specification. A removes relationship causes the right-hand side parts to be removed from the configuration. An

excludes relationship cause the left-hand side parts to become notSelectable. A requires choice relationship recognizes that a choice has to be made by the user between the set of right-hand parts.

In a backward evaluation, configuration system 212 determines
5 whether a relationship if activated would result in an invalid state. If so, configuration system 212 takes steps to ensure that the relationship cannot be activated. Instead of evaluating the left-hand side first as in the case of a forward evaluation, a backward evaluation evaluates the relationship by first examining the right-hand side. If a relationship fails the backward
10 evaluation (i.e., its activation would result in an invalid state), configuration engine 212 takes steps to ensure that the relationship cannot become active. For example, if a relationship states that part A can't work with part B and part B is selected, configuration system 212 excludes part A. This ensures that part A cannot be selected and the excludes relationship cannot become
15 active. Otherwise, a user can select part A which would result in an invalid configuration state.

The invention performs a relaxation over the set of relationships. A relaxation is an iterative process that is repeated until a state that has been defined no longer changes. That is, relationship evaluation is completed
20 when none of iVec, xVec, or rVec are modified during an iteration. The invention ensures that the relaxation process terminates by only adding bits

to the state. That is, because iVec, xVec, and rVec all grow through the iteration, the relaxation process is guaranteed to terminate.

Figure 10 provides an example of a relationship evaluation process flow according to an embodiment of the invention. At step 1002, rgVec is
5 initialized to zero. During the relationship evaluation, rgVec is regenerated. At step 1004, copies of iVec, xVec, and rVec are made. The copies are used to determine whether relaxation is finished. At step 1006, the includes table associated with the product definition is evaluated in the forward direction. Similarly, forward evaluation is performed on the excluded and removed
10 tables. The forward evaluation performs an "or" operation with the selectedVec and a specified configuration state vector (e.g., a forward evaluation of the includes table performs an "or" operation with the selectedVec and the iVec) when the left-hand side of the relationship is true. For example, when the items that correspond with the "1" bits in
15 left-hand side 830 are all selected (i.e., in the selected vector), right-hand side 832 is "or'ed" with the iVec to add the items that correspond with the "1" bits in right-hand side 832 to the configuration where a forward evaluation of an includes relationship is being performed.

At step 1008, a backward evaluation is performed on the relationships
20 contained in the includes, excludes, and removed tables. In a backward evaluation of an include relationship, the xVec is examined to determine whether any right-hand side items of an include relationship are excluded.

left-hand side item should be excluded to cause the relationship to become notActivateable.

In the forward evaluation, rgVec is set and items are included or excluded where appropriate. For example, if the right-hand side items are
5 within range (within min and max values for the required choice relationship), then the remaining right-hand side items are excluded. If the only way to make a requires choice valid is to select the items remaining on the right-hand side item, configuration system 212 includes these items. In addition, a requires choice relationship is evaluated to determine whether
10 the right-hand side items selected are within a valid range. If not, the corresponding bit in the rgVec is set. If so, the corresponding bit in the rgVec is zeroed.

The selected, selectable, and notSelectable states are updated at step 1012. At step 1014 (i.e., "change in a vector?"), a determination is made
15 whether any of iVec, xVec, or rVec were modified during the current iteration of the relaxation process. If not, a stable state has been achieved, and processing ends at step 1016. If so, additional iterations are needed to complete the relaxation process. Processing therefore continues at step 1002 to initiate a new iteration.

20 In evaluating relationships, it is not necessary to evaluate a relationship in either the forward or backward direction once it has become active. Also, when a relationship is notActivateable, there is no need to

evaluate the relationship in the forward direction. When a relationship is notActivatable and has been excluded, there is no need to evaluate in the backward, or reverse, direction.

Unseletion

5 Figure 9 described processing where the user input was an item selection. Configuration system 212 also processes user input to unselect an item from the configuration. As previously discussed, an item can be unselected in any order. There is no requirement to unselect items the reverse order in which they were selected. Figure 11 provides an example of
10 an unselect item process flow according to an embodiment of the invention.

At step 1102 (i.e., "selection in uVec?"), a determination is made whether the item that is selected for deletion was selected for inclusion in the configuration by the user. If it is determined that the item is not a user-selected item, processing continues at step 1104 to raise an error.

15 Processing continues at step 1114 to update the GUI (if necessary) and return control to the user. Processing of the current input ends at step 1116.

If it is determined at step 1102 that the item selected for unselection was selected by the user, processing continues at step 1106 to reset the bit in the uVec that corresponds to the unselected item (i.e., set the bit to zero).

20 This unselects the item from the current configuration. At step 1108, the iVec, xVec, and rVec vectors are initialized (set to zero). Steps 1110, 1112 and

1114 correspond to steps 910, 912, and 914, respectively, of Figure 9.

Processing ends at step 1116.

Translation Between Representations

To ensure correct runtime behavior of the engine, a potentially
5 incomplete user-defined set of relationships is converted into a set of
complete runtime relationship tables. The external representation is
translated into an internal representation. The invention maps the
hierarchies contained in parts catalog 204 to relationships and maps
relationships to the runtime tables, for example.

10 As discussed above, compiler 704 performs a mapping between
product-level relationships and part-level relationships. The translation
process further provides the ability to translate a multi-level hierarchy to a
single-level hierarchy. That is, the translation process can be used to flatten
out the intuitive hierarchy created by a maintenance user using the user
15 interface of Figure 6, for example. Figure 12 provides a flow for translation
processing according to an embodiment of the invention.

At step 1202, a set of relationships is created based on the hierarchies.
Thus, for example, group 312 is translated into an requires choice
relationship. That is, group 312 requires a choice between parts 306, 308, and
20 310. Relationships can be defined in terms of groups. An identifier (item ID)
is used to identify the group. When a group appears on the left-hand side of

a relationship, the group is assigned an item ID and the relationship is defined in terms of the new item ID. This process, referred to as back-linking, is performed at step 1204.

At step 1206, an include chain evaluation is performed. In an include chain evaluation, all include relationship chains are fully evaluated so that a relationship contains all of the parts that are included in that part regardless of whether the included relationship is direct (e.g., mentioned in the includes relationship) or indirect (mentioned in another includes relationship that contains an included part on the right-hand side).

At step 1208, subgroup excludes are evaluated. In this process, active requires choice relationships are evaluated where one relationship's right-hand side is a subset of another relationship's right-hand side. In this case, if both relationships are active, the right-hand side parts that are not within the subset are excluded. Relationship factorization is performed at step 1210. Relationship factorizing can create new relationships that ensure proper runtime behavior when more than one item is selected during some iteration.

Removing redundancy in relations may cause some relations to become unnecessary. A relation becomes unnecessary when it contains no items on its right side. Redundant relationships are removed at step 1212. Processing ends at step 1214.

Back-Linking

Relationships can be defined in terms of groups. When a group appears on the left-hand side of a relationship, the group is assigned an item ID and behaves as a normal item (e.g., part) and the relationship is defined
5 in terms of this new item ID. A relationship is defined for each item in the group:

item includes item ID (of the group)

It is only necessary to create this relationship once per group, regardless of how many times that group appears on the left side of a
10 relationship. Back-Linking can be used to generate relationships in nested groups. Figures 13A-13B provide an illustration of groups and nested groups according to an embodiment of the invention. Group 1302 contains two items (groups 1304 and 1306) that are nested within group 1302. Notation is provided for a group to specify the minimum and maximum bounds for a
15 group. For example, the notation "(0,1)" indicates that between zero and one items in group 1302 are to be chosen.

The items contained in group 1302 are also groups that contain multiple items. Groups 1304 and 1306 are nested within group 1302. Two or three of parts 1308-1312 can be chosen from Group 1304. Zero to three of
20 parts 1314-1318 can be chosen from Group 1306.

The invention creates a metapart for these groups (groups 1302-1306). Relationships are then created between the groups and their members such that items are back-linked to their parents. Figure 13B provides an illustration of metaparts for the groups and nested group of Figure 13A according to an embodiment of the invention. Groups 1302'-1306' are metaparts for groups 1302-1306, respectively. Relationships can be generated using these metaparts as follows:

Group 1302' requires choice [Group 1304', Group 1306'] 0,1
Group 1304' requires choice [part 1308, part 1310, part 1312] 2,3
Group 1306' requires choice [part 1314, part 1316, part 1318] 0,3

The metapart is used internally; it is unnecessary for a user to be aware of them. Using back-linking, relationships can be generated such that an item (e.g., part 1308) links back to its predecessor(s) (e.g., nested group, group 1302). The following new relationships can be created for the examples of Figures 13A-13B:

Group 1304' includes Group 1302'
Group 1306' includes Group 1302'
Part 1308 includes Group 1304'
Part 1310 includes Group 1304'
Part 1312 includes Group 1304'
Part 1314 includes Group 1306'
Part 1316 includes Group 1306'
Part 1318 includes Group 1306'

When a user selects part 1308, part 1308 includes group 1304'. Group 1304' is added to the configuration. Further, group 1304' includes group 1302' thereby including group 1302'. When group 1302' is included in the configuration, it activates the relationship that requires a choice between groups 1304' and 1306'. Since group 1304' is already selected, group 1306' can be excluded. When group 1306' is excluded, parts 1314-1318 are excluded when the newly generated relationships are evaluated in the reverse (backward) direction. Further, the bounds specified by group 1304 are also enforced using these relationships.

As discussed above, a requires choice relationship can be defined at the product level. A product can require a choice between items in a group. The product-level requires choice relationship is treated similarly to that of a requires choice part-level relationship.

Include Chains

In an include chain evaluation, all include relationship chains are fully evaluated so that a relationship can be generated that contains all of the parts that are included regardless of whether the relationship is direct (e.g., the included parts mentioned in the includes relationship) or indirect (a part is mentioned in another includes relationship that contain an included part on the right-hand side). For example, the following relationships exists in the external representation:

a includes b and c

85160.915

44

b includes d
d and c includes e

An include chain exists that starts with *a*. The first relationship indicates that *a* includes *b* and *c*. Based on the second and third
5 relationships, *b* and *c* include other parts. Based on all of the relationships, *a* includes *b, c, d*, and *e*. Include chain evaluation expands these relationships into the following relationships:

a includes b and c and d and e
b includes d
10 *d and c includes e*

It is not necessary to generate additional relationships to perform includes chain evaluation. Preferably, includes chain evaluation can modify an existing relationship to include both direct and indirect includes relationships. Each includes relationship can be evaluated against all other
15 includes relationships to determine whether its right-hand side should be modified to include both direct and indirect includes relationships. Once all of the includes relationships have been evaluated, a new right-hand side value can be generated based on the evaluation. Figures 14A-14B provide an includes chain process flow according to an embodiment of the invention.

20 At step 1402 (i.e., "all include rels. processed?"), a determination is made whether all of the includes relationships have been processed. If other includes relationships remain to be processed, processing continues at step

1408 to get the next includes relationship, r . At step 1410, a temporary vector, temp, is created that is the result of an "or" operation between the left-hand and right-hand sides of r . Another temporary vector, tempsave, is created that is set to the value of temp at step 1412. At step 1414 (i.e., "all rels.
5 compared?"), a determination is made whether all of the other includes relationships have been evaluated against r . If not, processing continues at step 1418 to get another relationship, r_2 . At step 1420 (i.e., "temp hits r_2 ?"), a determination is made whether any of the items contained in temp are also contained in r_2 . If not, processing continues at step 1414 to process any
10 remaining relationships against r . If so, processing continues at step 1422 to perform a "or" operation between temp and the right-hand side of r_2 to add the right-hand side of r_2 to the temp vector.

If it is determined, at step 1414, that all of the relationships r_2 have been evaluated for r , processing continues at step 1416. At step 1416 (i.e.,
15 "temp=tempsave?"), a determination is made whether any items were added to the temp vector. If not, there are no indirect includes and processing continues at step 1424 to reset r_2 and processing continues at step 1414 to process any remaining relationships against this relationship. If so, processing continues at step 1404 to modify the right-hand side of r to
20 include the items contained in the temp vector minus the items contained in the left-hand side of r . Processing continues at step 1402 to process any remaining relationships.

If it is determined at step 1402 (i.e., all include rels. processed?") that all of the includes relationships have been evaluated for include chains, processing ends at step 1404.

Subgroup Excludes

5 The subgroup excludes evaluation is performed on requires choice relationships. Assume a requires choice relationship points to a group (A) that is a complete subgroup of a group (B) of another requires choice relationship. If B has a max of 1 and both relationships are made active, the items in B that are not in A, Q, cannot be chosen. If a pairwise comparison
10 of requires choice relationships yields such cases, a relationship such as the following is generated:

[union of left sides of both relationships] can't work with Q

For example, assume that the following relationships have been defined:

15 *a b requires choice [x, y, z] (0,1)*
 d requires choice [x, y] (0,2)

The second relationship contains a group that is a subgroup of the first relationship. When both relationships are active, the non-subset is excluded. This avoids the case where a user selects a, b, and d and the
20 relationship evaluations do not exclude the items not included in both groups (the non-subset). The invention combines these relationships and

creates an excludes relationship. The excludes relationship for the above example is as follows:

a b d excludes z

Figures 15A-15B provides an example of a subgroup excludes process flow according to an embodiment of the invention. At step 1502 (i.e., "all
5 rels. processed?"), a determination is made whether all of the requires choice relationships have been processed. If so, processing ends at step 1504. If not, processing continues at step 1506 to get the next relationship, *r*. At step 1507 (i.e., "*r*.rhs.max = 1?"), a determination is made whether the maximum
10 value for the relationship is equal to one. If not, processing continues at step 1502 to process any remaining relationships. If so, processing continues at step 1508. At step 1508 (i.e., "all rels. compared?"), a determination is made whether all of the other requires choice relationships have been processed against *r*. If so, processing continues at step 1502 to process another requires
15 choice relationship.

If not, processing continues at step 1510 to get another relationship, *r*₂. At step 1512 (i.e., "*r*=*r*₂?"), a determination is made whether the two relationships are equal. If so, processing continues at step 1508 to process any remaining requires choice relationships against the current *r*. If *r* and *r*₂ are
20 not equal, processing continues at 1514. At step 1514 (i.e., "*r*₂.grp subset of *r*.grp?"), a determination is made whether the right-hand side of *r*₂ is a subset of the right-hand side of *r*. If not, processing continues at step 1508 to process any remaining requires choice relationships against the current *r*. If

so, processing continues at step 1516 to create an exclude relationship where the left-hand side is the union of the left-hand side of r and r_2 and the right-hand side is the non-subset of the right-hand side of the two relationships. Processing continues at step 1508 to process any remaining
5 requires choice relationships against the current r .

Relationship Factorization

The invention assumes that one item on the left side of a relationship is preferably selected during any iteration. However, the includes and requires choice relationships may cause additional items to become selected
10 (e.g., selected by the system). Factorizing of relationships can create new relationships to ensure proper runtime behavior when more than one item is selected during some iteration.

Factorization causes a relationship to be created where a relationship has items on the left-hand side that also appear on the right-hand side of an
15 includes relationship. The new relationship represents the common factorization of both relationships. For example, where the following two relationships exist:

[a] includes [b and c]
[b and c and d and e] can't work with [x and y]

20 the following relationship is created:

[a and d and e] can't work with [x and y]

The invention uses a technique in factorization that avoids combinatorial explosion. Relationships are created during factorization when they add value. The invention identifies value where the new relationship:

- 5 1. replaces more than one item from the left-hand side;
2. has at least two different includes relationships replace at least one unique item from the left-hand side of the relationship being factorized; and
3. the two includes relationships have at least one common item
- 10 in their left-hand side.

Figures 16A-16C provide an example of a relationship factorization process flow according to an embodiment of the invention. At step 1602, each relationship is marked as "new". At step 1604, a state variable, added, is initialized to "false". At step 1606 (i.e., "all includes rels. processed?"), a

15 determination is made whether all of the includes relationships have been processed. If so, processing continues at step 1608 (i.e., "added = 'true'?") to determine whether any new relationships were added. If not, processing ends at step 1610. If there are new relationships, processing continues at step 1609 to reset the includes relationships and processing continues at step 1604

20 to initialize added to "false".

If it is determined at step 1606 that all of the includes relationships have not been processed, processing continues at step 1612 to get the next includes relationship, *i*. At step 1613, the relationships are reset. At step

1614 (i.e., "all 'new' rels. processed?"), a determination is made whether all of the relationships have been marked as "new" have been processed against *i*. If so, processing continues at step 1606 to process any remaining includes relationships. If not, processing continues at step 1616.

- 5 At step 1616, the next relationship, *r*, is marked as "old". At step 1620 (i.e., " $!(r.lhs \ \&\& \ i.rhs)?$ ", a determination is made whether the right-hand side of the *i* relationship and the left-hand side of the *r* relationship have any bits in common. If they do not have bits in common, processing continues at step 1614 to process any remaining relationships against the
- 10 current *i*. If they do have bits in common, processing continues at step 1622 to generate new.lhs from the includes relationship and the left-hand side of *r*. Step 1622 takes out items in the right-hand side of *i* from left-hand side of *r* and adds in items that are in the left-hand side of *i*. At step 1624 (i.e., "new.lhs has value?") a determination is made whether new.lhs has value.
- 15 This determination can use the guidelines as specified above.

- If new.lhs does not have value, processing continues at step 1614 to processing any remaining relationships against *i*. If it does have value, processing continues at step 1626 to add a new relationship (of the type *r*) that uses new.lhs as the left-hand side and the right-hand side of *r*. The new
- 20 relationship is marked as "new" at step 1628. The value of added is set to "true" at step 1630. Processing continues at step 1614 to process any remaining relationships.

Relationship Compression

Removing redundancy in relations may cause some relations to become unnecessary. A relation becomes unnecessary when it contains no items on its right side. If a relationship's left-hand side is a proper subset of the left-hand side of another relationship, then the first relationship is guaranteed to be active whenever the second relationship is active. If both relationships are also the same type of relationship, the common right-hand side elements can be removed from the second relationship. This can be repeated to remove all unnecessary relationships. The following relationships provide an example:

[a] removes [c]
[a and b] removes [c and d]

The left-hand side of the first relationship is a proper subset of the left-hand side of the second relationship (e.g., all elements "a" in the left-hand side of the first relationship are contained in the left-hand side of the second relationship). The relationships are the same type. Therefore, the common right-hand side elements can be removed from the right-hand side of the second relationship. The following is the result after modifying the relationship:

[a] removes [c]
[a and b] removes [d]

Figures 17A-17B provide an illustration of a relationship compression process flow according to an embodiment of the invention. At step 1702 (i.e., "all relationships in table processed?"), a determination is made whether all of the relationships in the table have been processed. If not, processing
5 continues at step 1704 to get the next relationship, r . At step 1706 (i.e., "r processed against all other rels. in table?"), a determination is made whether the other relationships in the table have been processed against r . If so, processing continues at step 1702 to process any remaining relationships in the table as r .

10 If it is determined at step 1706 that all of the other relationships have not been processed against r , processing continues at step 1708. Another relationship is retrieved as r_2 at step 1708. At step 1710 (i.e., " $r = r_2$?"), a determination is made whether the relationships are the same. If not, processing continues at step 1706 to process any remaining relationships
15 against r . If so, processing continues at step 1712. At step 1712 (i.e., " $r.lhs$ subset of $r_2.lhs$?"), a determination is made whether the left-hand side of r is a subset of the left-hand side of r_2 . If the subset condition is true, processing continues at step 1714 to remove the common items out of the right-hand side of r (i.e., perform the operation $r_2.rhs = r_2.rhs \text{ and not } r.rhs$). Processing
20 continues at step 1706 to process any remaining relationships against r .

If it is determined at step 1702 that all of the relationships have been processed as r , processing continues at step 1716. At step 1716 (i.e., "all

relationships in table processed?") a determination is made whether all of the relationships have been processed. If so, processing ends at step 1718. If not, processing continues at step 1720 to get another relationship as r . At step 1722 (i.e., " $r.rhs = zeroes?$ "), a determination is made whether the
5 right-hand side of r has been set to zero has a result of compression. If not, processing continues at step 1716 to process any remaining relationships. If so, processing continues at step 1724 to delete r from the table. Processing continues at step 1716 to process any remaining relationships.

Global Relationships

10 Global relationships can be defined that are processed for all products. Thus, it is unnecessary to redefining a relationship for all of the products. The relationship can be defined as a global relationship that is applied for every product. If, however, partitioned relationship tables are used for each product, it is necessary to evaluate the relationships in a product's
15 partitioned relationship tables as well as the global relationships. For example, the process flows described herein that operate on relationship tables can be performed once for the product's partitioned relationship tables and once for the global relationships.

Thus, a method and apparatus for maintaining and configuring
20 systems has been provided.

CLAIMS

- sub. a1
1. A method of configuring a system comprising the steps of:
- generating a definition for said system, said definition containing one or more elements and being conveyed graphically using a set of product relationships;
- 5 generating a set of relationships between said one or more elements, said set of relationships being conveyed graphically using a set of part relationships; and
- 10 configuring said system using said definition and said set of relationships.
2. The method of claim 1 wherein said step of configuring further comprises the steps of:
- receiving input from a configuration user;
- validating said input based on said definition, said set of relationships,
- 15 and a current configuration state; and
- identifying a set of valid configuration options using said definition, said set of relationships and said current configuration state.

3. The method of claim 2 wherein said step of identifying further comprises the steps of:

determining whether a relationship in said set of relationships is active;

5 modifying said configuration to satisfy said active relationship when it is determined that a modification is needed;

making said relationship notActivatable when it is determined that said relationship is not active and the activation of said relationship would render the configuration invalid.

10 4. The method of claim 3 wherein said step of determining whether a relationship is active further comprises the step of determining whether the elements specified in the left-hand side of said relationship are present in said configuration.

5. The method of claim 4 wherein said step of determining
15 whether the elements specified in the left-hand side of said relationship are present in said configuration further comprises the step of performing a bit-level comparison of the bits corresponding to the elements in said left-hand side of the relationship with a "selected" bit vector.

6. The method of claim 3 wherein said relationship is an includes
20 relationship, said step of modifying further comprises the step of including

in said configuration elements identified in a right-hand side of said relationship.

7. The method of claim 3 wherein said relationship is an excludes relationship, said step of modifying further comprises the step of marking
5 said elements identified in the right-hand side of said relationship as excluded and notSelectable.

8. The method of claim 3 wherein said relationship is a removes relationship, said step of modifying further comprises the step of removing
10 said elements identified in the right-hand side of said relationship from said configuration.

9. The method of claim 3 wherein said relationship is a requires choice relationship, said step of modifying further comprises the steps of:

determining whether a plurality of elements identified in a right-hand side of said relationship are present in said configuration such
15 that said relationship is satisfied;

modifying said configuration to satisfy said requires choice relationship when it is determined that only one path exists to satisfy said relationship and said relationship is not already satisfied.

10. The method of claim 3 wherein said step of making said
20 relationship notActivateable further comprises the step of marking one or more elements in said left-hand side of said relationship as notSelectable.

11. The method of claim 2 wherein said definition, each of said set of relationships, and said current configuration state are expressed as bit vectors having bits that correspond to elements.

12. The method of claim 2 wherein said step of validating further
5 comprises the step of determining whether the relationships other than those marked notActivateable in said set of relationships can be satisfied.

13. The method of claim 2 wherein said step of generating a definition selects said one or more elements for inclusion in said definition, said input is made without regard for the order in which said one ore more
10 elements are selected for said definition.

14. The method of claim 2 wherein said configuration state identifies a set of selected elements and a set of selectable elements for said system, said input capable of selecting any member of said set of selectable elements and being made despite the order in which members of said set of
15 selected elements were selected.

15. The method of claim 1 wherein a parts catalog is used to store information associated with said one or more elements, said information including an identifier and description.

16. A method for defining a system that can be configured comprising the steps of:

- identifying an element of said system;
- specifying said element as an included element when said element is
- 5 automatically included in the configuration;
- specifying said element as an optional element when said element is not a necessary component of said system;
- specifying said element as a required choice when said element is a group that contains one or more members from which to choose.

10 17. The method of claim 16 wherein said element is a member of a first element set said first element set containing one or more elements, said method further comprising the steps of:

- identifying a second element set of said system, said second element set containing one or more elements; and
- 15 specifying a relationship between said first element set and said second element set.

18. The method of claim 17 wherein said step of specifying further comprises the steps of:

- specifying that said second element set is to be included in said system
- 20 when said first element set is present in said system;

specifying that said second element set is to be excluded from said system when said first element set is present in said system; and

specifying that said second element set is to be removed from said system when said first element set is present in said system;

5 19. The method of claim 16 further comprising the steps of:

specifying a group of elements;

specifying a relationship between said first element set and said group of elements such that one or more elements in said group of elements must be chosen when said first element set is present in said system.

10 sub. a2 20. An article of manufacture comprising:

a computer usable medium having computer readable program code embodied therein for configuring a system comprising:

15 computer readable program code configured to cause a computer to generate a definition for said system, said definition containing one or more elements and being conveyed graphically using a set of product relationships;

20 computer readable program code configured to cause a computer to generate a set of relationships between said one or more elements, said set of relationships being conveyed graphically using a set of part relationships; and

~~computer readable program code configured to cause a computer to
configure said system using said definition and said set of relationships.~~

21. The article of manufacture of claim 20 wherein said computer
readable program code configured to cause a computer to configure further
5 comprises:

computer readable program code configured to cause a computer to
receive input from a configuration user;

computer readable program code configured to cause a computer to
validate said input based on said definition, said set of relationships, and a
10 current configuration state; and

computer readable program code configured to cause a computer to
identify a set of valid configuration options using said definition, said set of
relationships and a current configuration state.

22. The article of manufacture of claim 21 wherein said computer
15 readable program code configured to cause a computer to identify further
comprises:

computer readable program code configured to cause a computer to
determine whether a relationship in said set of relationships is active;

computer readable program code configured to cause a computer to
20 modify said configuration to satisfy said active relationship when it is
determined that a modification is needed;

computer readable program code configured to cause a computer to make said relationship notActivatable when it is determined that said relationship is not active and the activation of said relationship would render the configuration invalid.

5 23. The article of manufacture of claim 22 wherein said computer readable program code configured to cause a computer to determine whether a relationship is active further comprises computer readable program code configured to cause a computer to determine whether the elements specified in the left-hand side of said relationship are present in said configuration.

10 24. The article of manufacture of claim 23 wherein said computer readable program code configured to cause a computer to determine whether the elements specified in the left-hand side of said relationship are present in said configuration further comprises computer readable program code configured to cause a computer to perform a bit-level comparison of the bits
15 corresponding to the elements in said left-hand side of the relationship with a "selected" bit-vector.

 25. The article of manufacture of claim 22 wherein said relationship is an includes relationship, said computer readable program code configured to cause a computer to modify further comprises computer
20 readable program code configured to cause a computer to include in said configuration elements identified in a right-hand side of said relationship.

26. The article of manufacture of claim 22 wherein said relationship is an excludes relationship, said computer readable program code configured to cause a computer to modify further comprises computer readable program code configured to cause a computer to mark said
5 elements identified in the right-hand side of said relationship as excluded and notSelectable.

27. The article of manufacture of claim 22 wherein said relationship is a removes relationship, said computer readable program code configured to cause a computer to modify further comprises computer
10 readable program code configured to cause a computer to remove said elements identified in the right-hand side of said relationship from said configuration.

28. The article of manufacture of claim 22 wherein said relationship is a requires choice relationship, said computer readable
15 program code configured to cause a computer to modify further comprises:

computer readable program code configured to cause a computer to determine whether a plurality of elements identified in a right-hand side of said relationship are present in said configuration such that said relationship is satisfied;

20 computer readable program code configured to cause a computer to modify said configuration to satisfy said requires choice relationship when it

is determined that only one path exists to satisfy said relationship and said relationship is not already satisfied.

29. The article of manufacture of claim 22 wherein said computer readable program code configured to cause a computer to make said
5 relationship notActivateable further comprises computer readable program code configured to cause a computer to mark one or more elements in said left-hand side of said relationship as notSelectable.

30. The article of manufacture of claim 21 wherein said definition, each of said set of relationships, and said current configuration state are
10 expressed as bit vectors having bits that correspond to elements.

31. The article of manufacture of claim 21 wherein said code configured to cause a computer to validate further comprises program code configured to cause a computer to determine whether the relationships other than those marked notActivateable in said set of relationships can be
15 satisfied.

32. The article of manufacture of claim 21 wherein said code configured to cause a computer to generate a definition selects said one or more elements for inclusion in said definition, said input is made without regard for the order in which said one ore more elements are selected for
20 said definition.

33. The article of manufacture of claim 21 wherein said configuration state identifies a set of selected elements and a set of selectable elements for said system, said input capable of selecting any member of said set of selectable elements and being made despite the order in which
5 members of said set of selected elements were selected.

34. The article of manufacture of claim 17 wherein a parts catalog is used to store information associated with said one or more elements, said information including an identifier and description.

35. An article of manufacture comprising
10 a computer usable medium having computer readable program code embodied therein for defining a system that can be configured comprising:

computer readable program code configured to cause a computer to identify an element of said system;

computer readable program code configured to cause a computer to
15 specify said element as an included element when said element is automatically included in the configuration;

computer readable program code configured to cause a computer to specify said element as an optional element when said element is not a necessary component of said system;

computer readable program code configured to cause a computer to specify said element as a required choice when said element is a group that contains one or more members from which to choose.

36. The article of manufacture of claim 35 wherein said element is
5 a member of a first element set said first element set containing one or more elements, said article of manufacture further comprising:

computer readable program code configured to cause a computer to identify a second element set of said system, said second element set containing one or more elements; and

10 computer readable program code configured to cause a computer to specify a relationship between said first element set and said second element set.

37. The article of manufacture of claim 36 wherein said computer
readable program code configured to cause a computer to specify further
15 comprises:

computer readable program code configured to cause a computer to specify that said second element set is to be included in said system when said first element set is present in said system;

computer readable program code configured to cause a computer to
20 specify that said second element set is to be excluded from said system when said first element set is present in said system; and

computer readable program code configured to cause a computer to specify that said second element set is to be removed from said system when said first element set is present in said system/.

38. The article of manufacture of claim 35 further comprising:
5 computer readable program code configured to cause a computer to specify a group of elements;

computer readable program code configured to cause a computer to specify a relationship between said first element set and said group of elements such that one or more elements in said group of elements must be
10 chosen when said first element set is present in said system.

39. A method of configuring a system comprising the steps of:

translating a system definition from an external representation comprising a set of intuitive relationships between components of said system definition into an internal representation comprising a set of
15 internal relationships;

validating configuration input using said internal representation; and

modifying a configuration state based on said user input.

40. The method of claim 39 wherein said step of translating further comprises the step of:

20 removing a plurality of hierarchical levels from said external representation such that said internal representation is a single level.

41. The method of claim 40 wherein said step of translating further comprises the step of:

creating a set of relationships based on said hierarchical levels, each one in said set of relationships having a first set of components and a second
5 set of components and a relationship operator.

42. The method of claim 41 wherein said relationship operator is chosen from the operators include, exclude, remove, and requires choice.

43. The method of claim 39 wherein each of said internal relationships has a left-hand side, a relationship operator and a right-hand
10 side, said step of translating further comprises the steps of:

determining whether a group of components exists on said left-hand side of one of said internal relationships;

performing the following steps when a group of components exist on said left-hand side:

15 creating a new component for said group; and
creating a new relationship for each component of said group of components such that each component will include said new component.

44. The method of claim 39 wherein a type of said internal relationships is an include relationship, said include relationship identifies a
20 second set of components that are included in said system when all of a first

set of components exist in said system, said step of translating further comprises the steps of:

determining whether a subset of the union of said first and second sets of components in a first includes relationship comprises a first set of components in a second includes relationship; and

generating a combined includes relationship, said combined relationship comprises said first set of components of said first includes relationship and said second set of components from said first and second includes relationships when said subset of the union of said first and second sets of components in said first includes relationship comprises a first set of components in a second includes relationship.

45. The method of claim 39 wherein a type of said internal relationships is a requires choice relationship, said requires choice relationship identifies a second set of components from which a number of components must be chosen when a first set of components exist in said system definition, said step of translating further comprises the steps of:

determining whether a second set of components of a second requires choice relationship is a proper subset of a second set of components of a first requires choice relationship:

performing the following steps when said condition exists:

creating a new relationship such that said new relationship excludes a second set of components of said new relationship when a

first set of components of said new relationship is selected, said new relationship's first set of components is the union of the first set of components of said first and second requires choice relationships and said new relationship's second set of components is the difference of the second set of components of said first and second requires choice relationship.

46. The method of claim 39 wherein a type of said internal relationships is an includes relationship, said includes relationship identifies a second set of components that are included in said system when all of a first set of components exist in said system, said step of translating further comprises the steps of:

determining whether the second set of components of an includes relationship are contained in a first set of components of another internal relationship;

replacing said second set of components of said includes relationship in said another internal relationship with said first set of components of said includes relationship when said second set of components of an includes relationship are contained in a first set of components of another internal relationship.

47. The method of claim 39 said internal representation comprises multiple types of relationships, each type of relationship having a first and second set of components and an operation, said operation is performed on

said second set of components when said first set of components exist in said system.

48. The method of claim 47 wherein said step of translating further comprises the steps of:

5 determining whether a first and second relationship are the same type of relationship;

performing the following steps when they are the same type:

determining whether the first relationship's first set of components is a proper subset of the second relationship's first set of components;

10 removing the first relationship's second set of components that exist from the second relationship's second set of components when said proper subset exists;

removing said second relationship when said second relationship's second set of components is an empty set.

49. The method of claim 47 wherein said step of validating further comprises the steps of:

determining whether a component selected by said configuration input is selectable; and

20 invalidating said configuration input when said component is not selectable.

50. The method of claim 49 wherein said one of said internal relationships is an includes relationship, said includes relationship causing a second set of components to be included in said system when all members of a first set of components exist in said system, said second step of determining
5 further comprises the step of:

determining whether any member of the second set of components cannot be included in said system.

51. The method of claim 49 wherein said one of said internal relationships is an excludes relationship, said excludes relationship causing a
10 second set of components to be excluded from said system when all members of a first set of components exist in said system, said second step of determining further comprises the step of:

determining whether any member of the second set of components already exists in said system.

15 52. The method of claim 49 wherein said one of said internal relationships is an removes relationship, said removes relationship causing a second set of components to be removed from said system when all members of a first set of components exist in said system, said second step of determining further comprises the step of:

20 determining whether any member of the second set of components has been selected for inclusion in said system by a configuration user.

53. The method of claim 49 wherein said one of said internal relationships is a requires choice relationship, said requires choice relationship requiring a selection of a number of members of a second set of components when all members of a first set of components exist in said
5 system, said second step of determining further comprises the step of:

determining whether said number of members of said second set of components cannot be selected for addition to said system.

54. The method of claim 47 wherein said step of modifying further comprises the steps of:

10 determining whether the performance of said operation of one of said internal relationships would result in an invalid configuration;
causing said one of said internal relationships to become inactivateable.

55. The method of claim 54 wherein said step of causing further
15 comprises the step of excluding the last remaining member of said first set of components such that said last remaining member cannot be added to said system, said last remaining member being the only member of said first set of components that has not already been selected.

56. The method of claim 39 wherein said step of translating further comprises the step of:

mapping said intuitive relationships of said external representation into a set of relationships in said internal representation.

5 57. The method of claim 39 wherein said internal representation is stored as a set of bit vectors wherein each bit vector includes a bit that corresponds to one of said components of said system definition, the impact of changing the number of said components of said system definition thereby being minimal.

10 58. The method of claim 57 wherein said step of validating comprising a plurality of bit-level operations.

59. The method of claim 39 wherein said step of modifying further comprises the step of updating said configuration state to represent which of said components of said system definition are selectable, selected, included,
15 and excluded.

add a3 >

08/707,18 7

ABSTRACT OF THE INVENTION

The invention provides the ability to interactively select and configure a product among a set of related products based on availability and compatibility of features and options. It does not impose an order in the
5 selection of products, features or options; only valid selections can be made at any time. To create an electronic representation of the product information to achieve the above goal, the invention provides a framework for defining a systems by defining the components of the system using elements contained in a parts catalog and defining relationships between the
10 components of a system. A configuration system validates a configuration using the system definition, the current state of the configuration and user input.

Handwritten signature

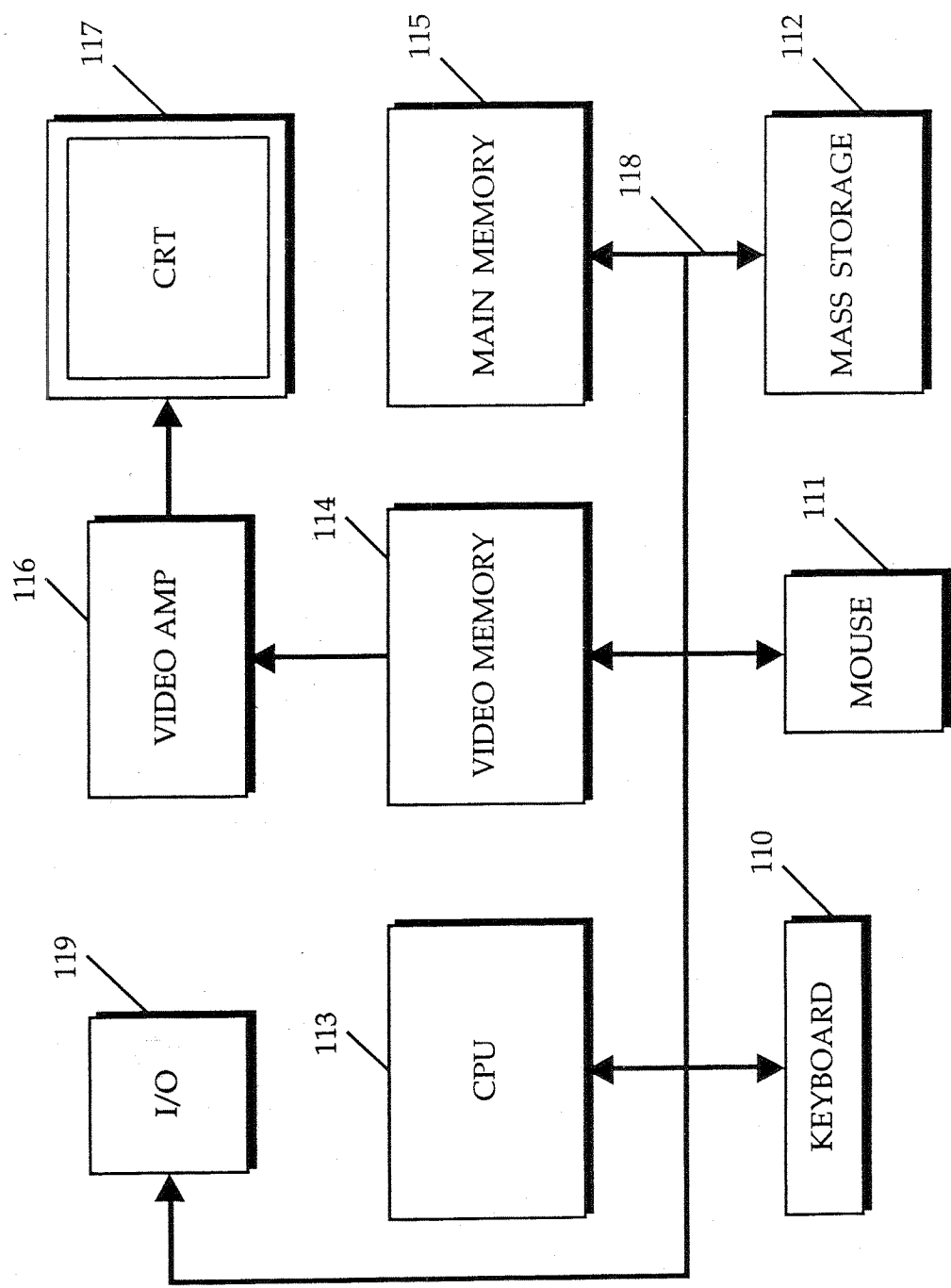
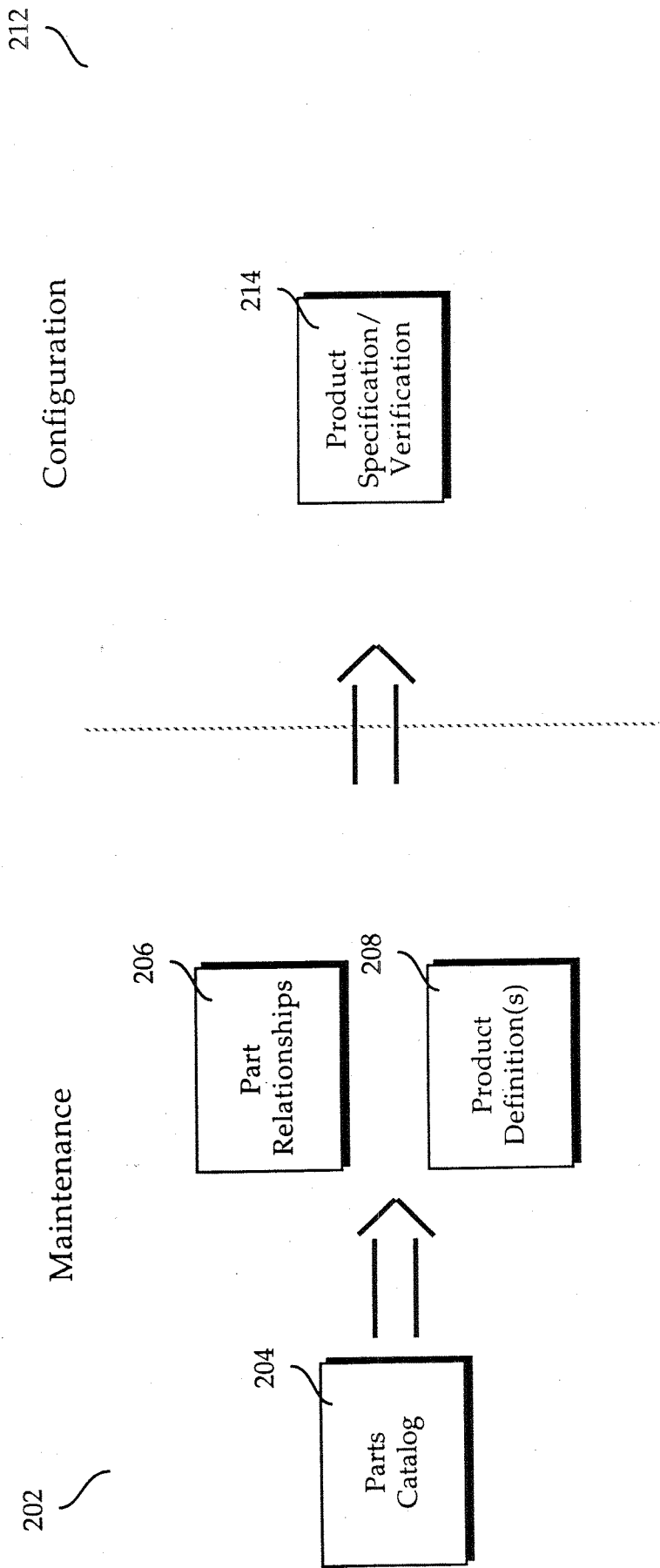


FIGURE 1



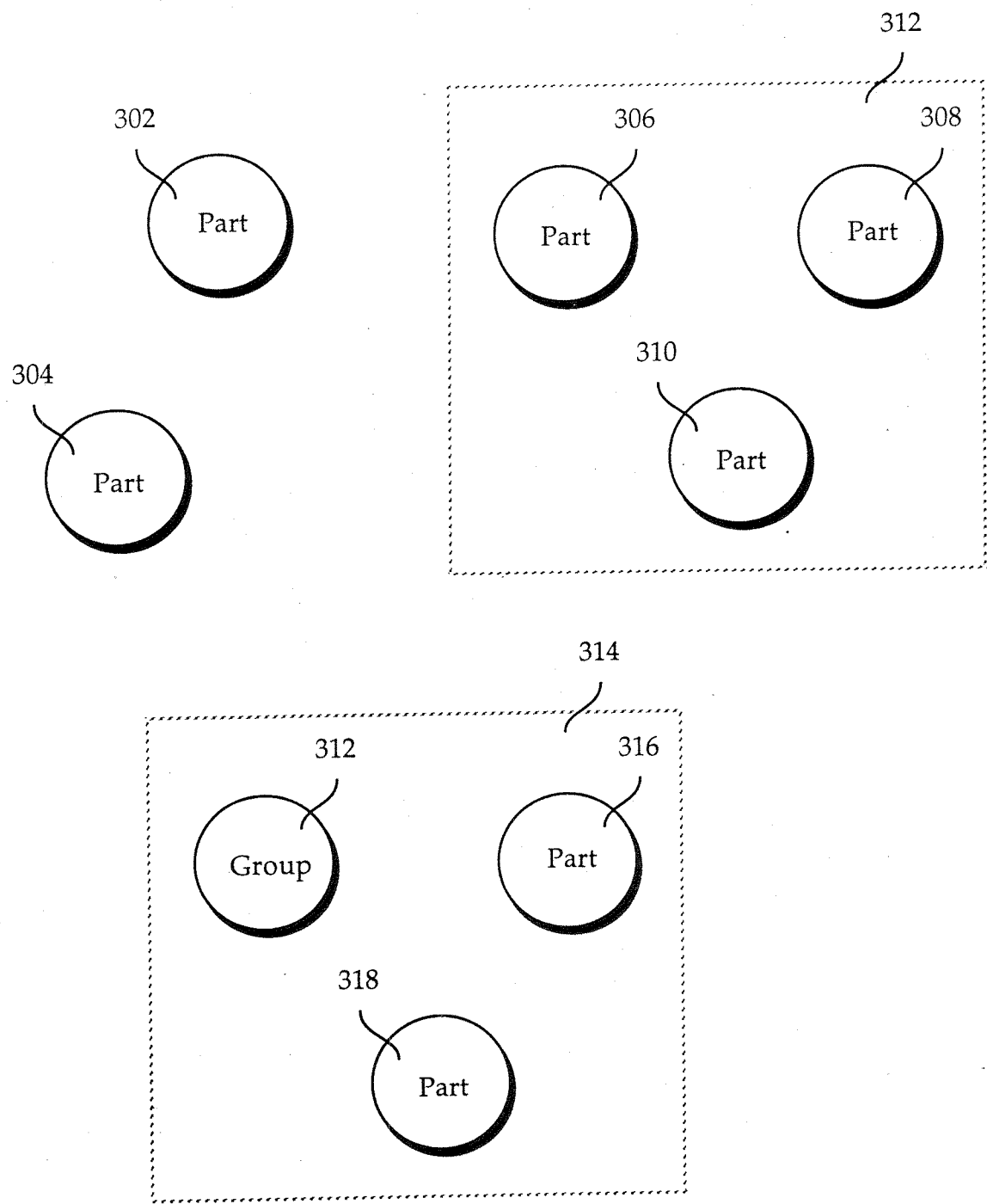


Figure 3

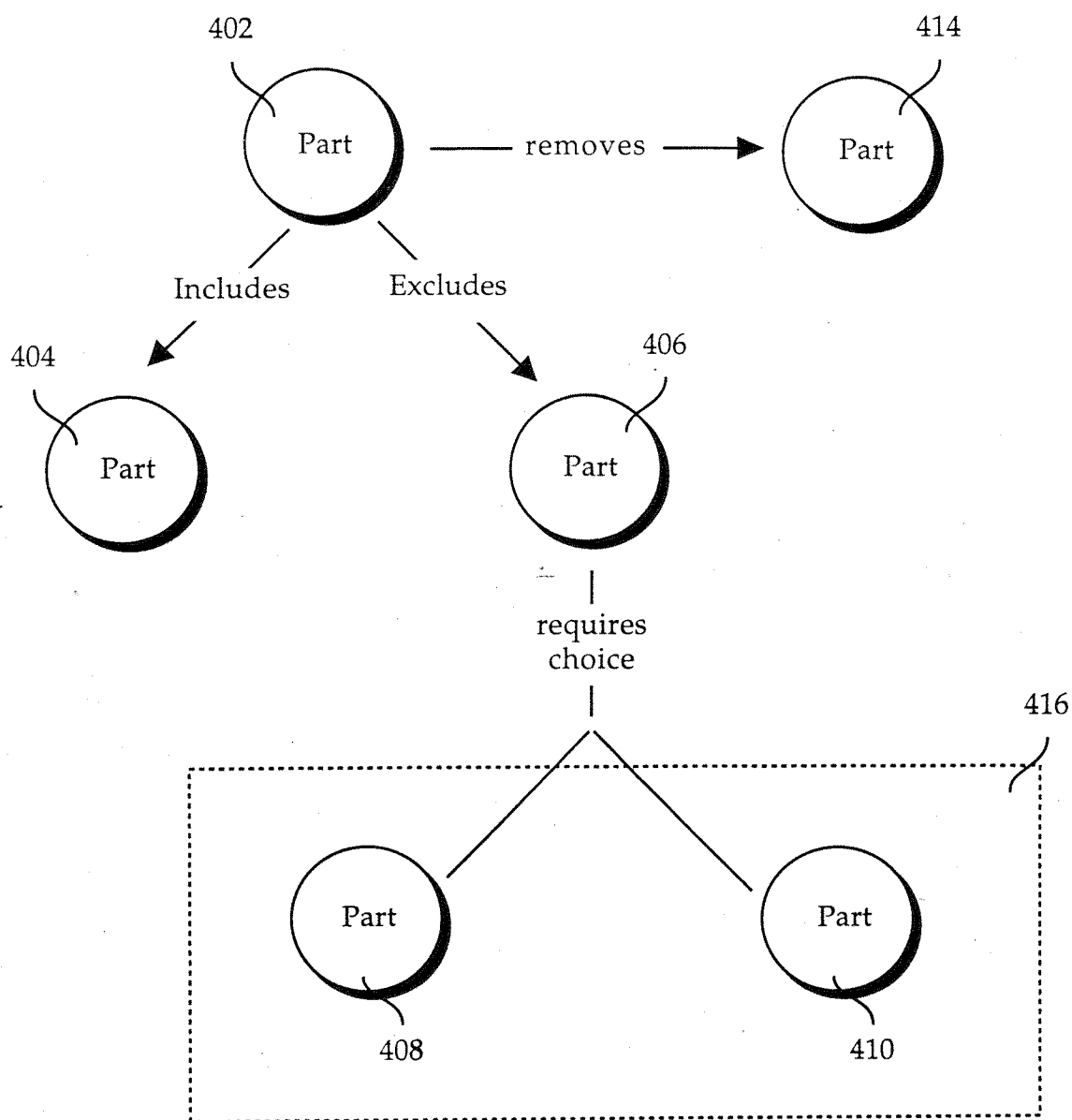


Figure 4

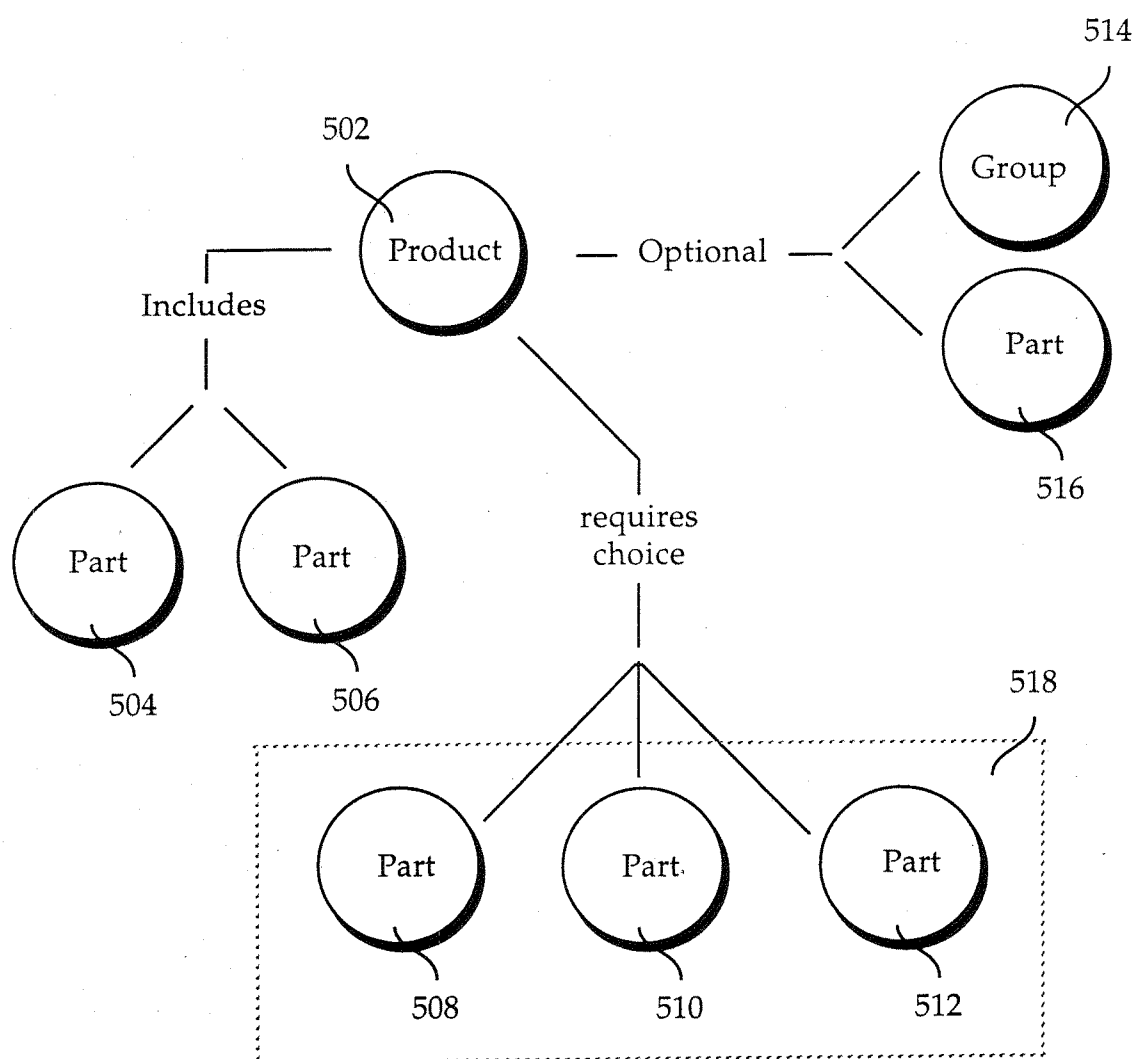


Figure 5

08/707187

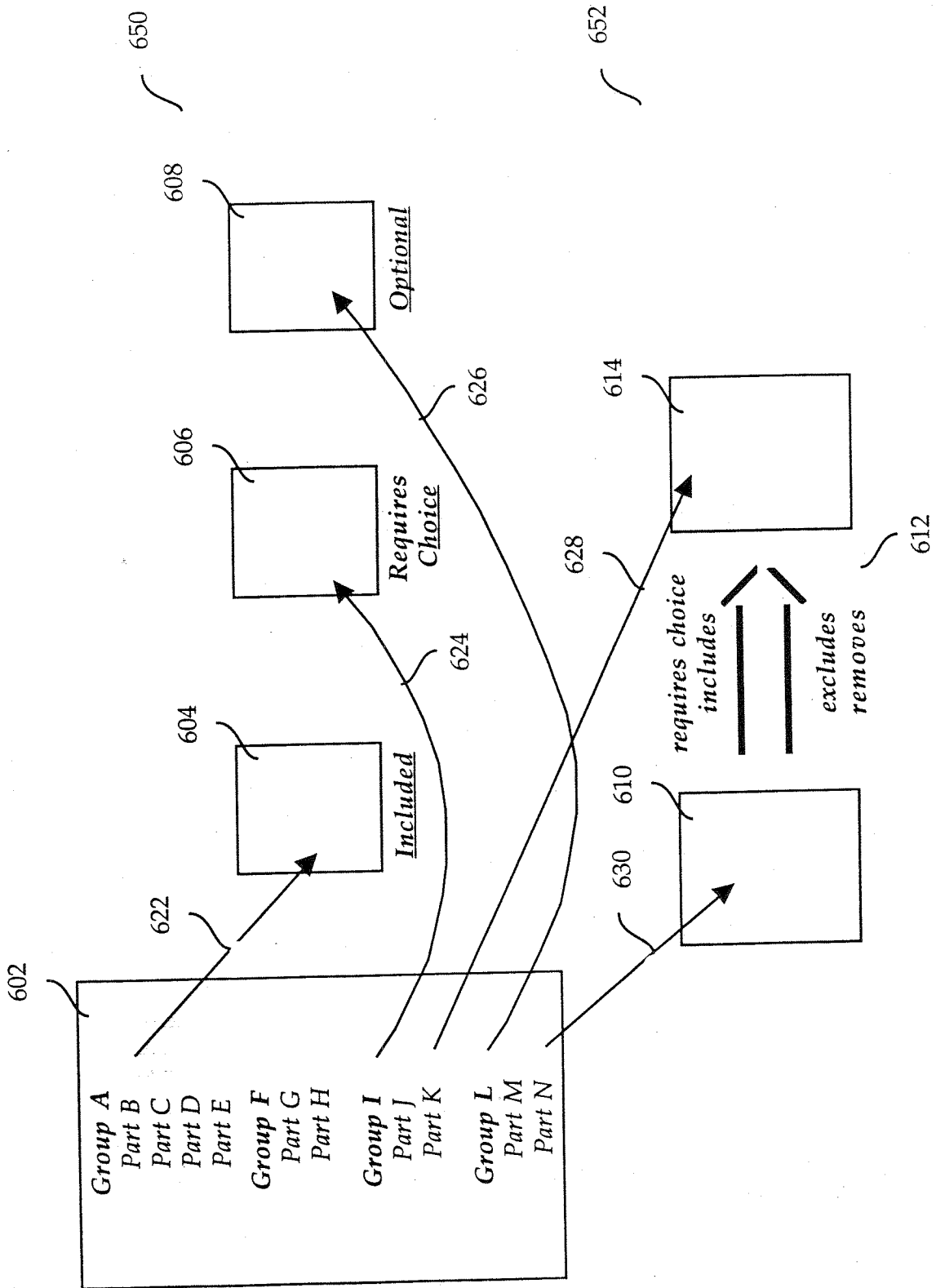


Figure 6

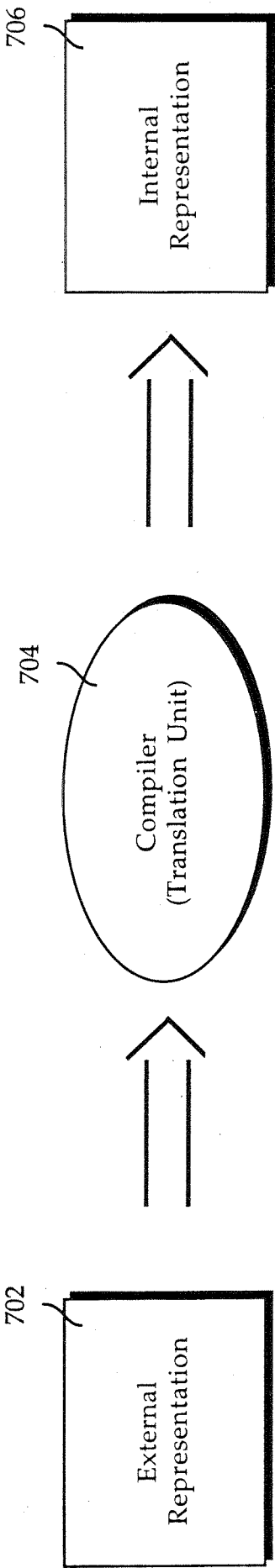


Figure 7

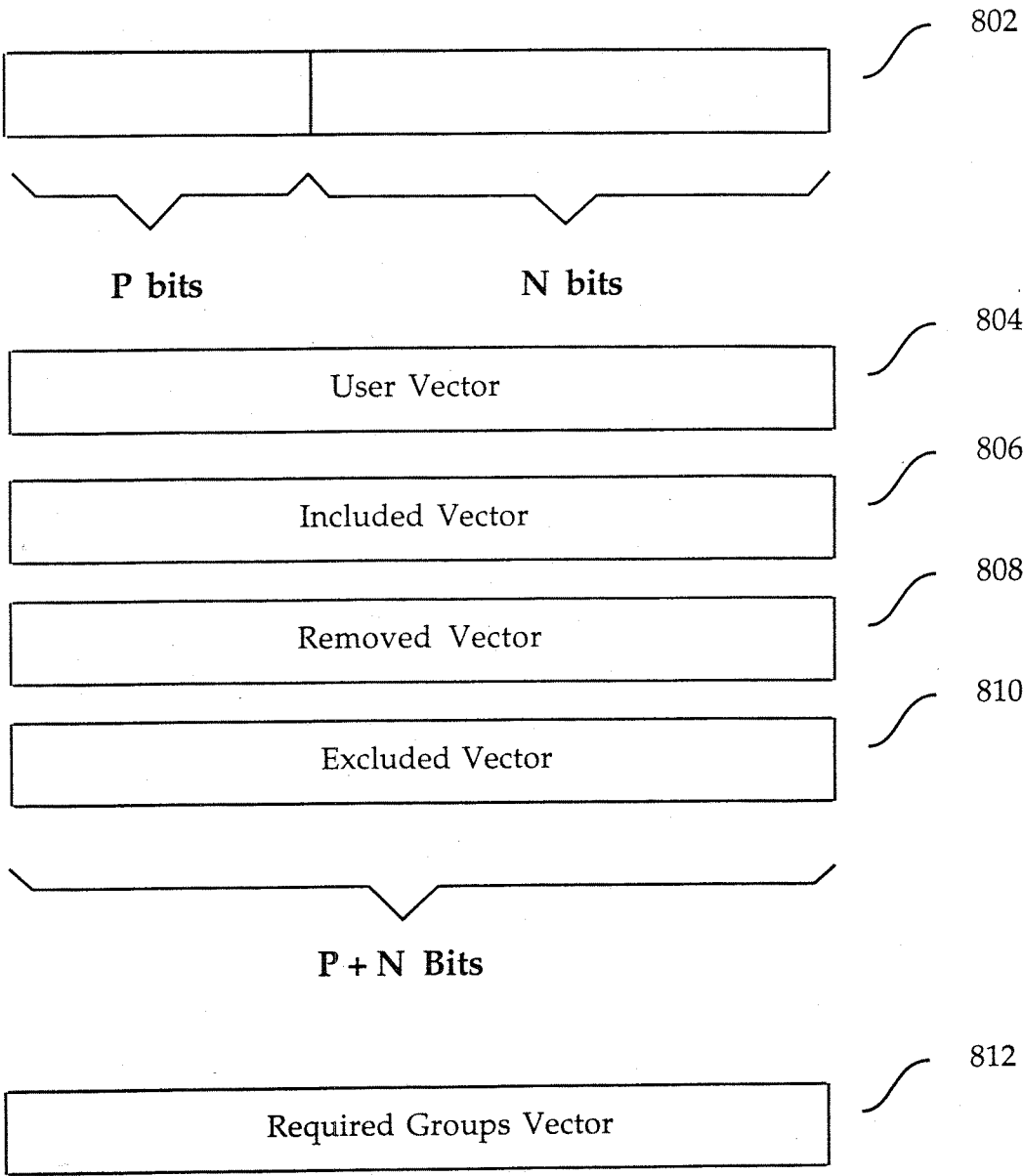


Figure 8A

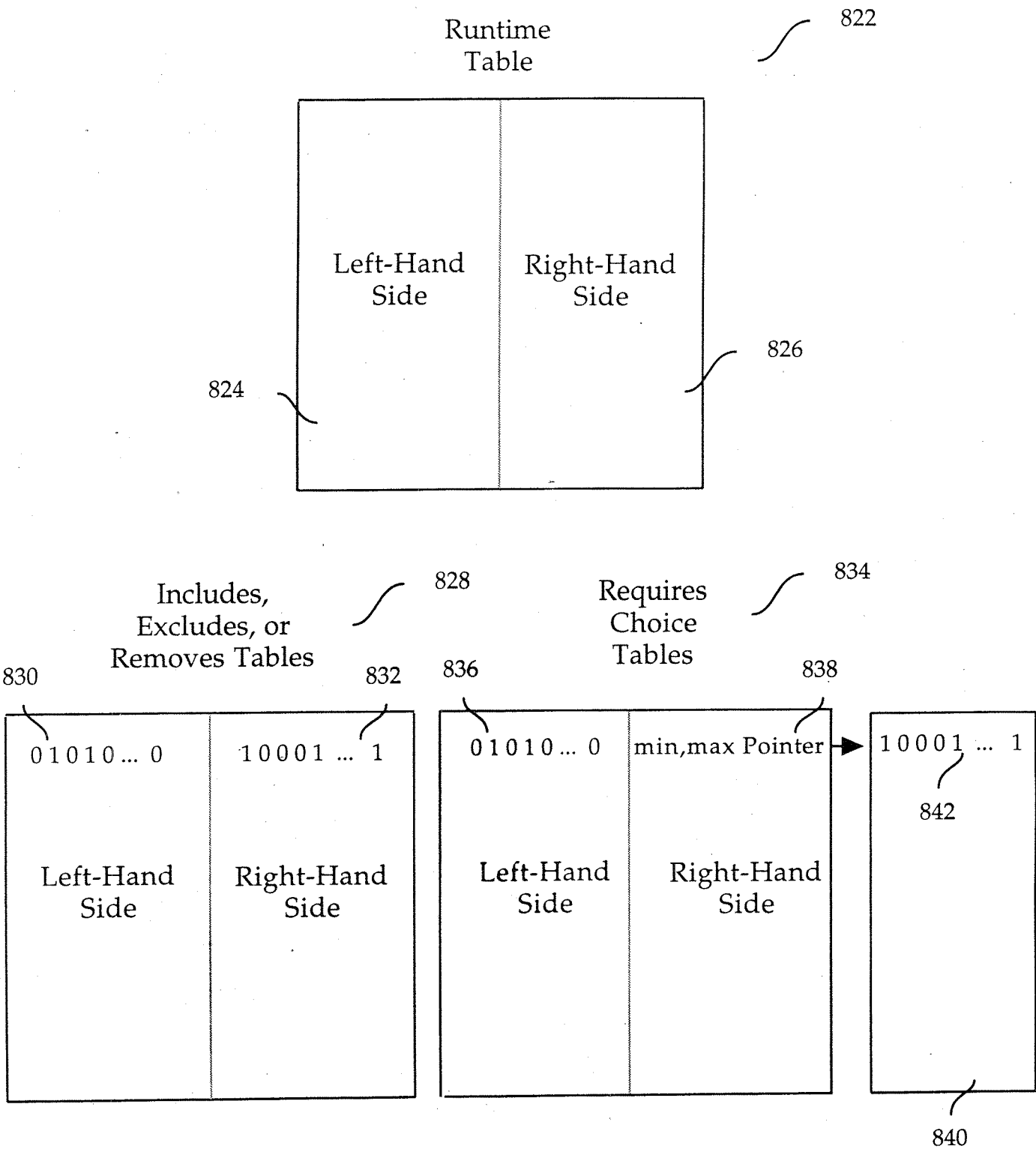


Figure 8B

08/707187

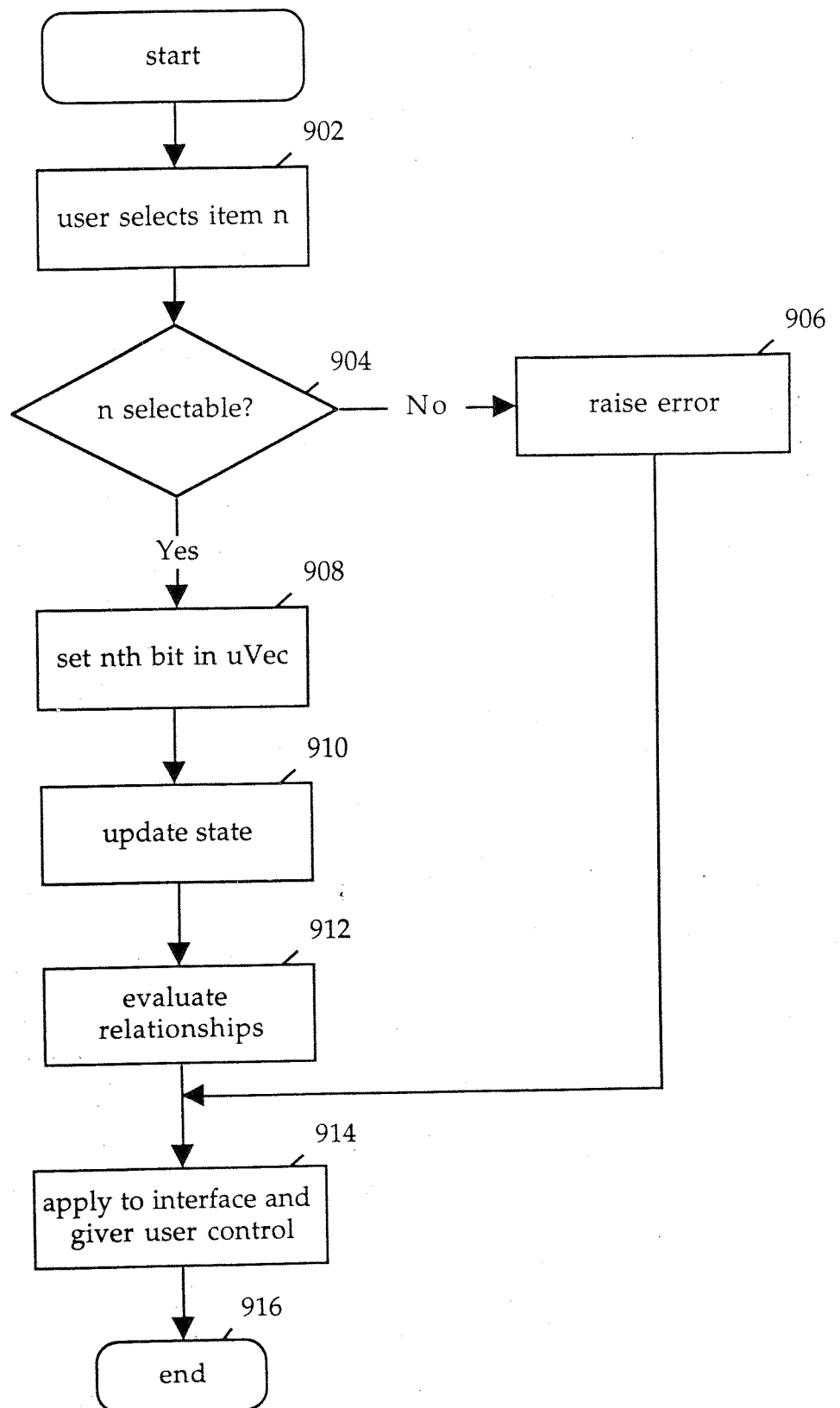


Figure 9

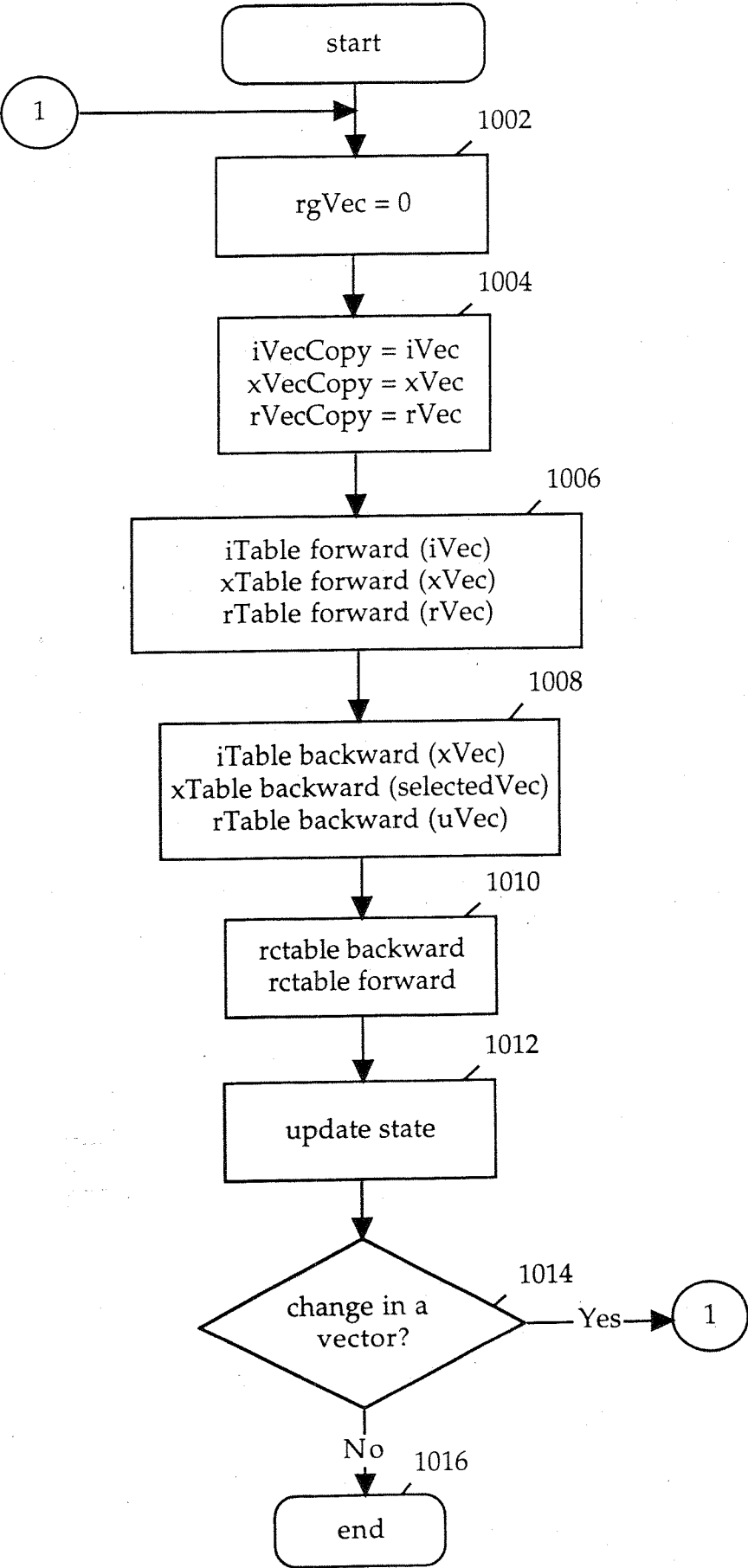


Figure 10

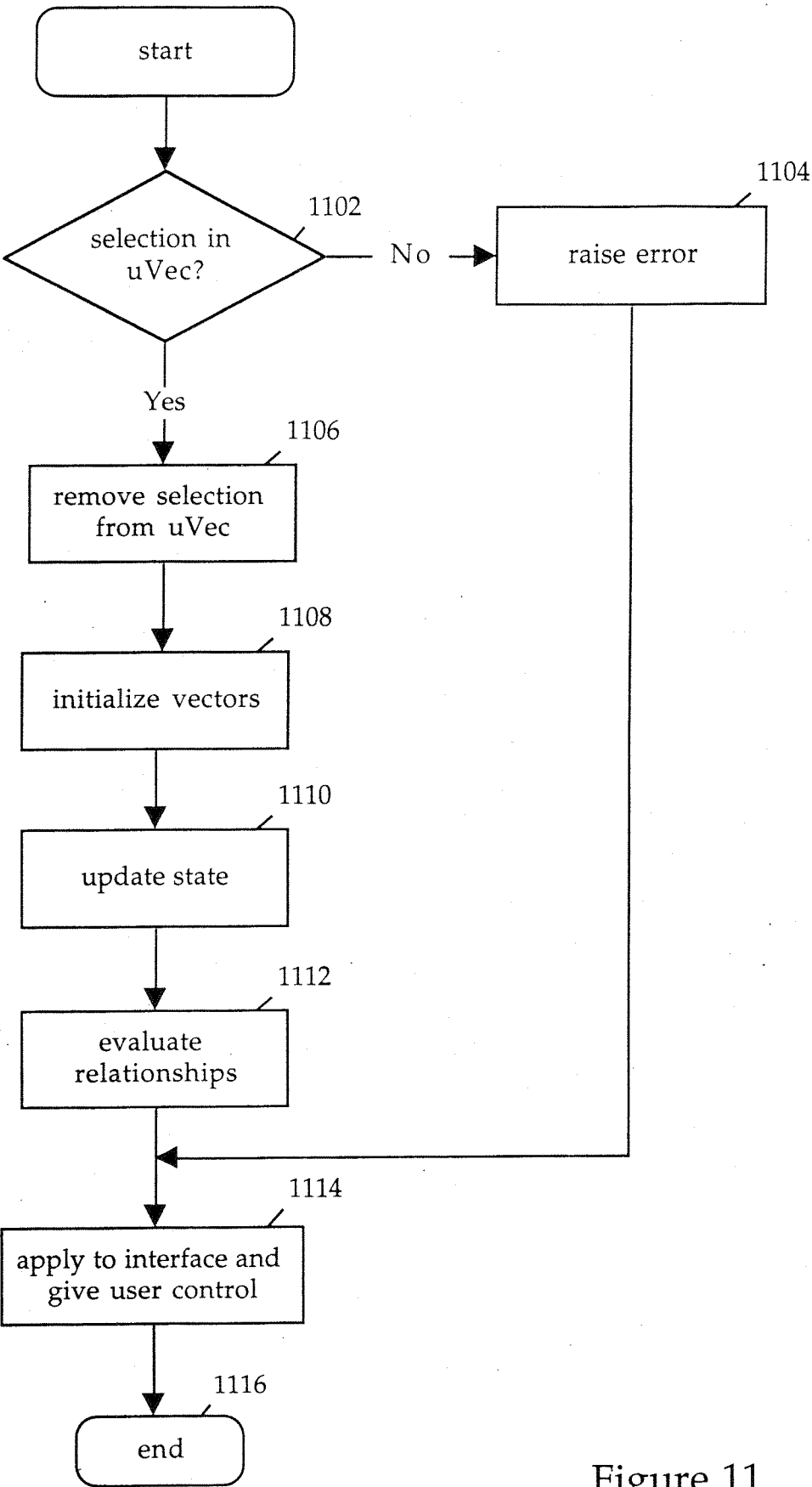


Figure 11

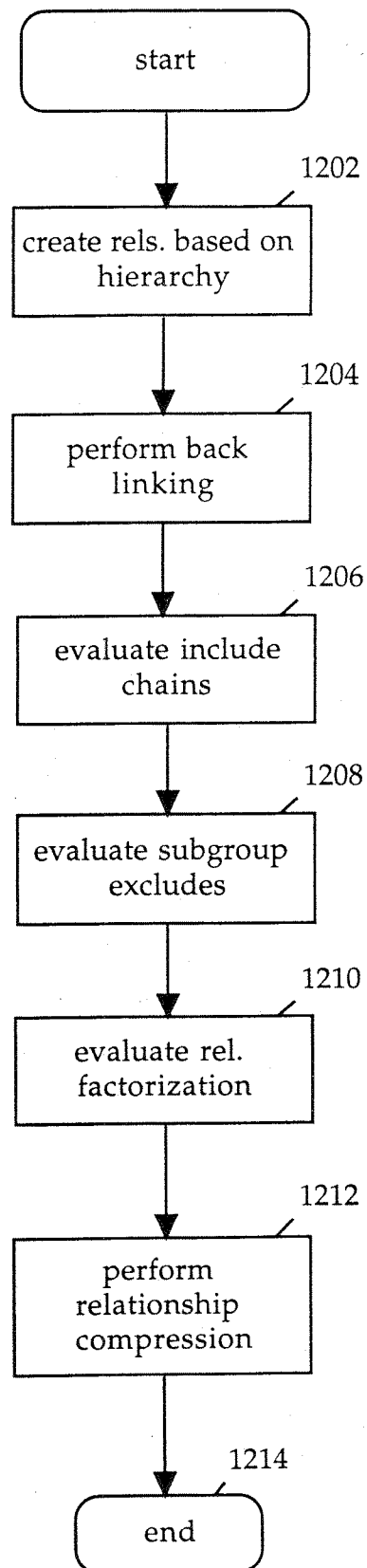


Figure 12

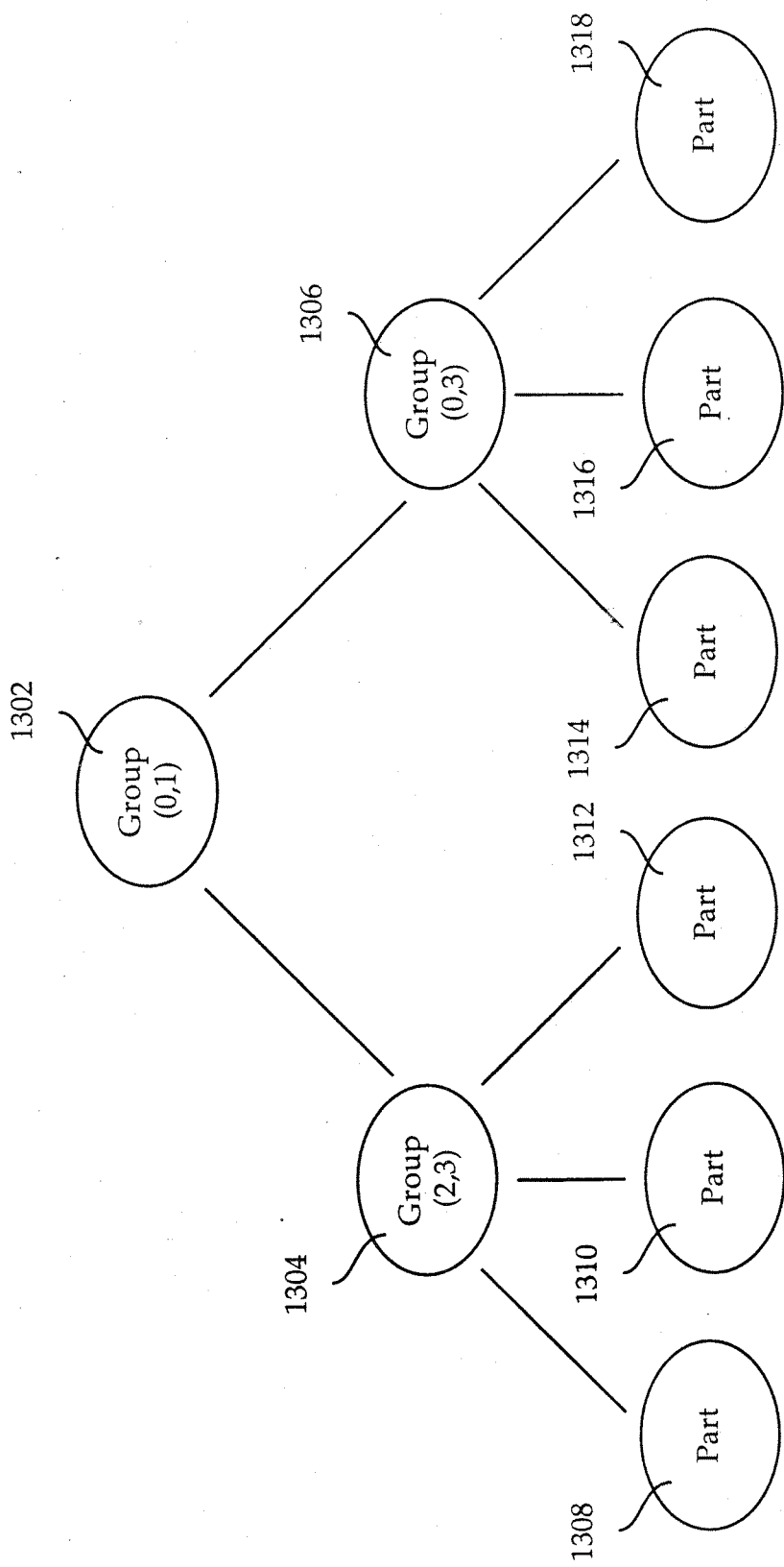


Figure 13A

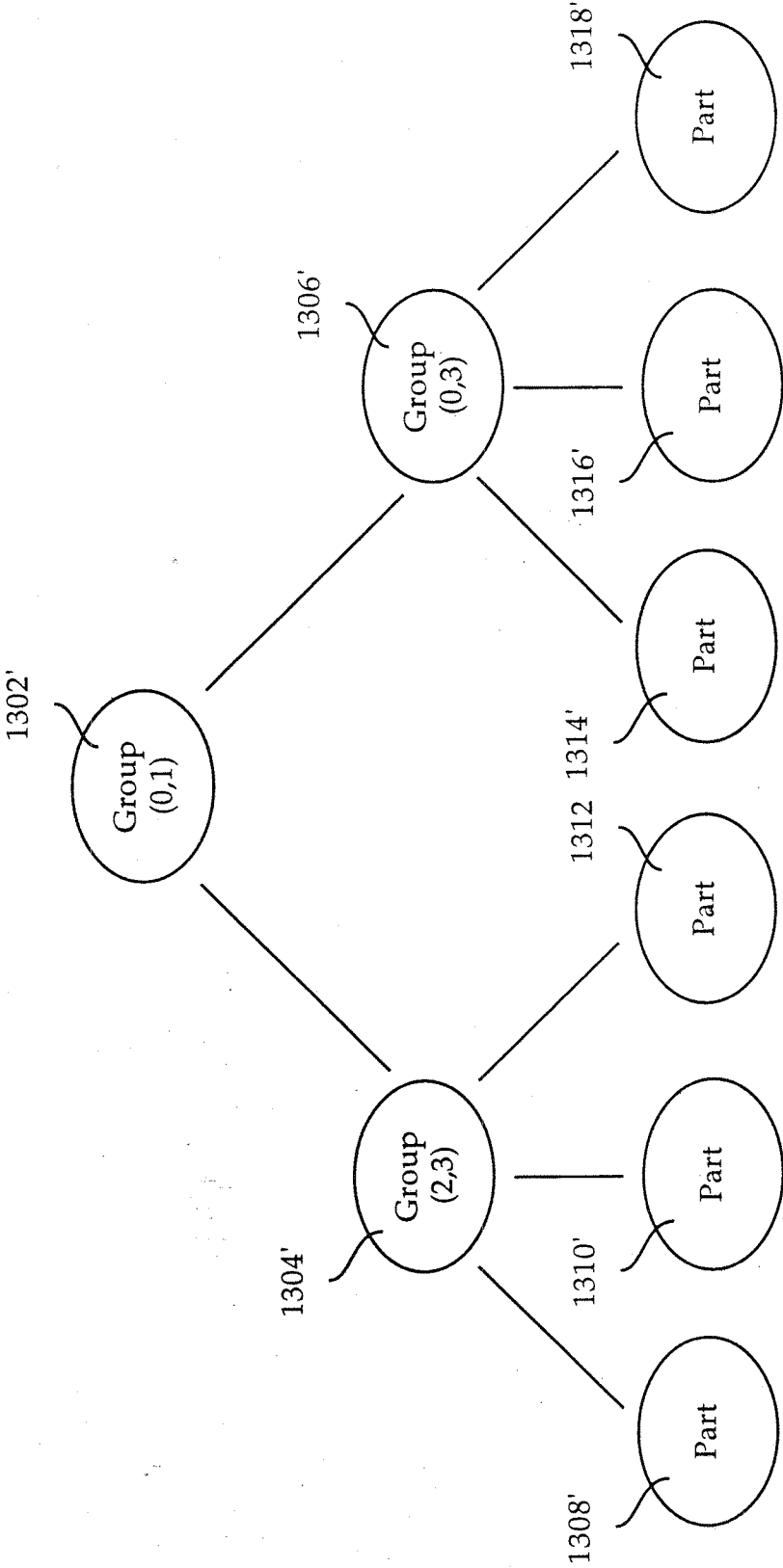


Figure 13B

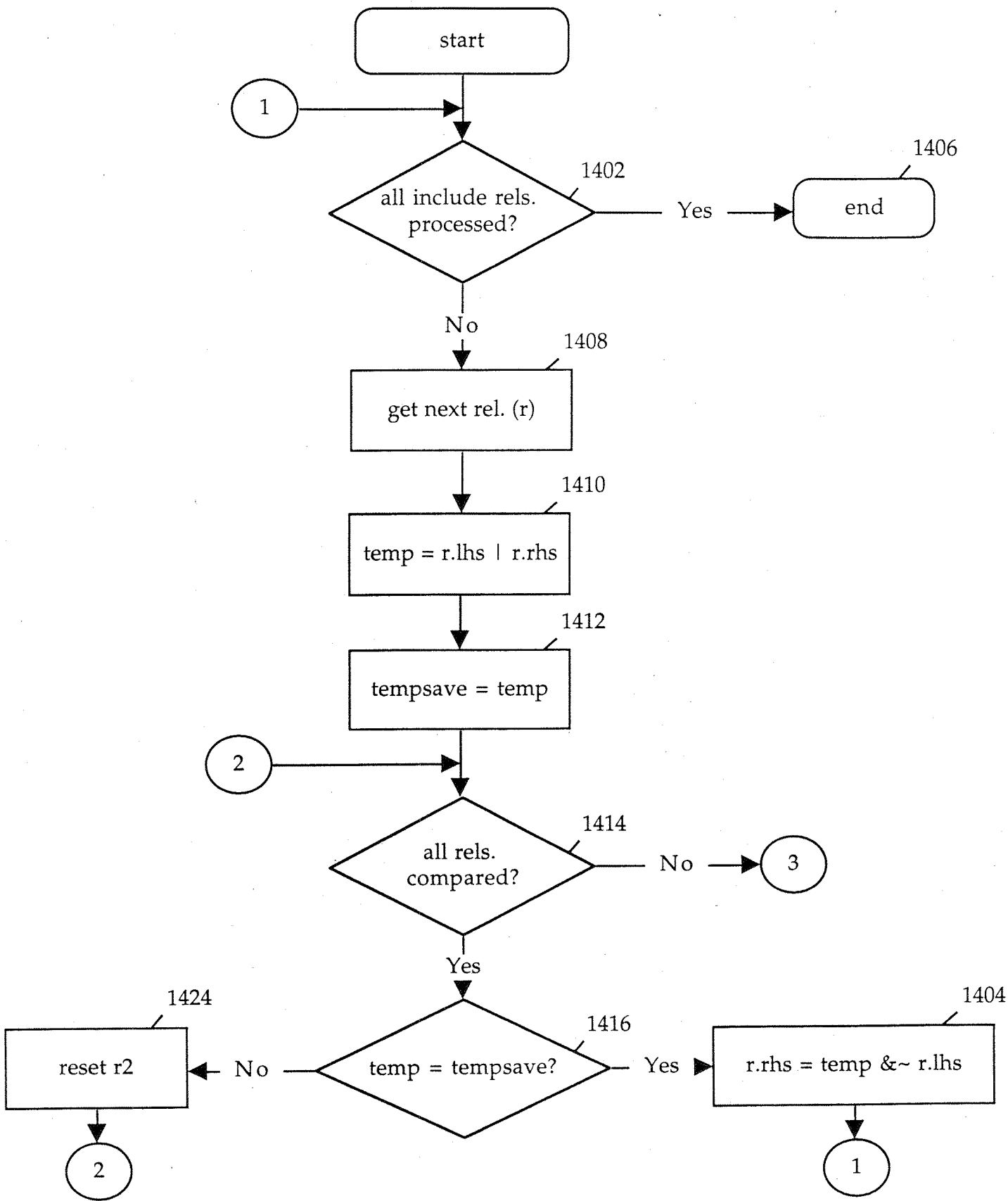


Figure 14A

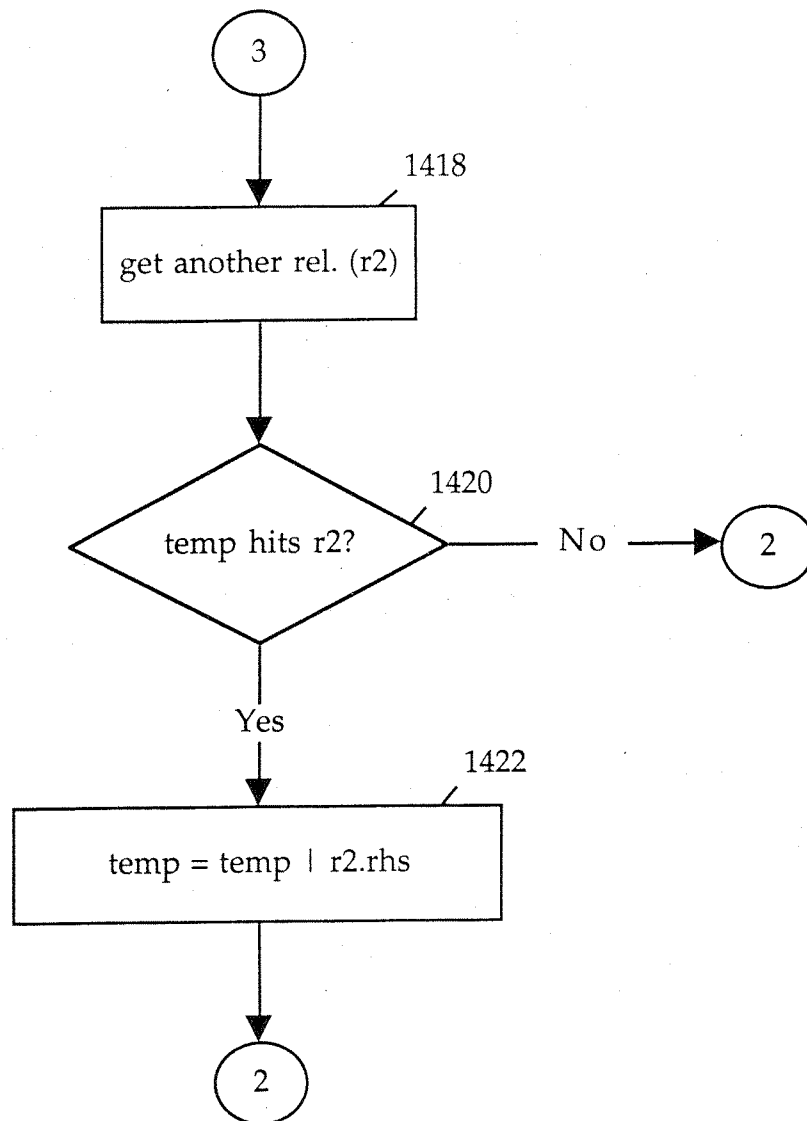


Figure 14B

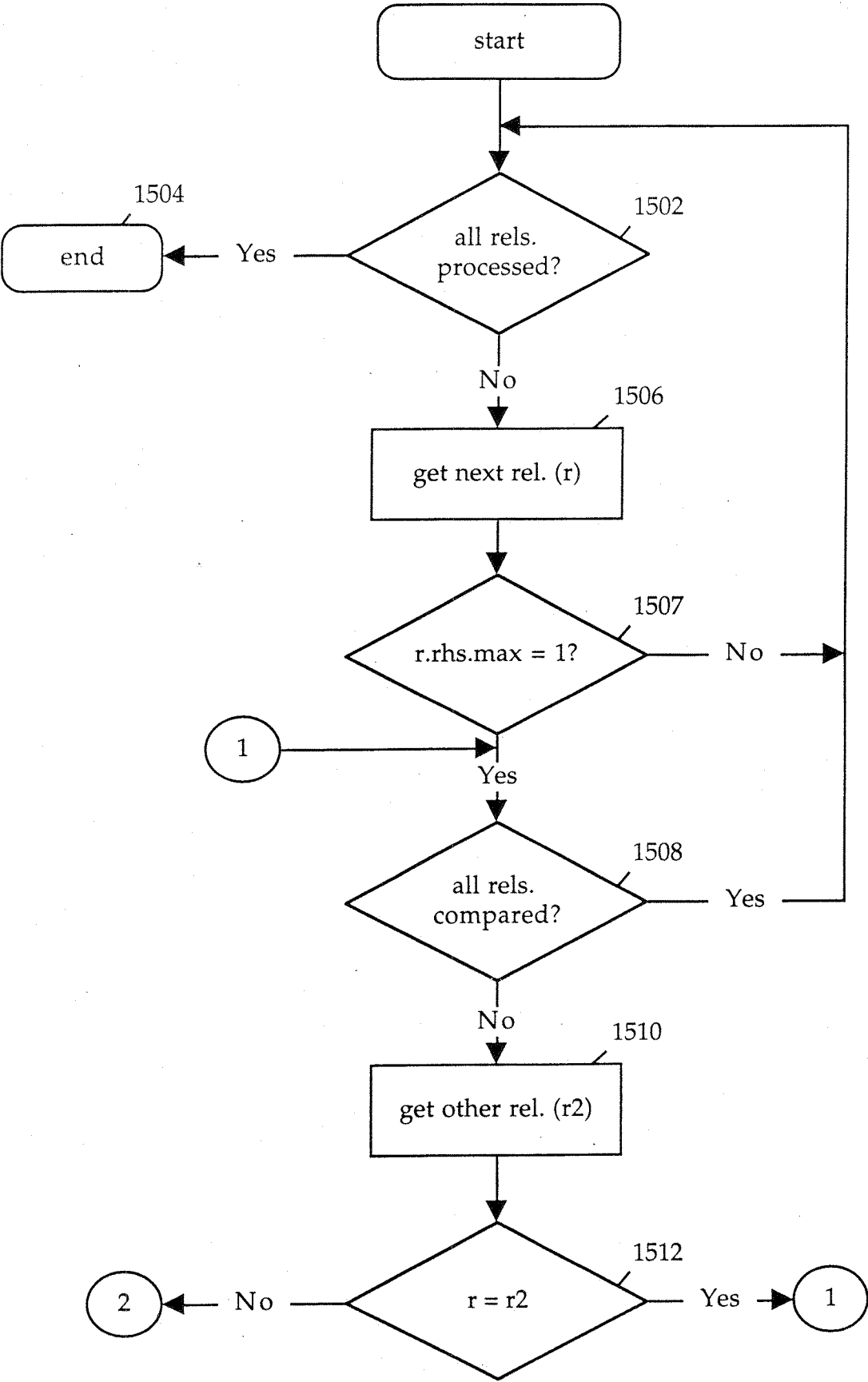


Figure 15A

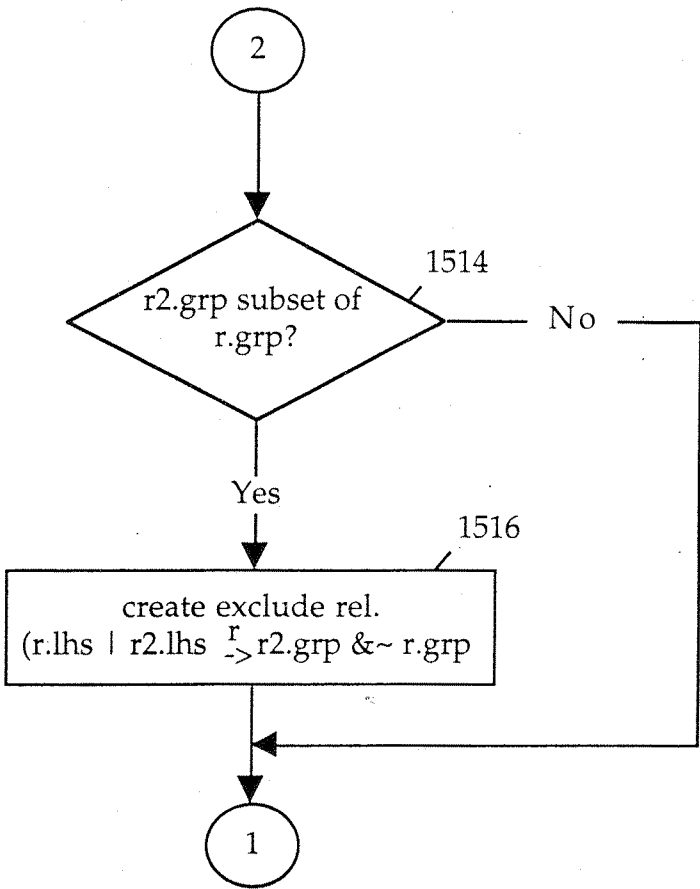


Figure 15B

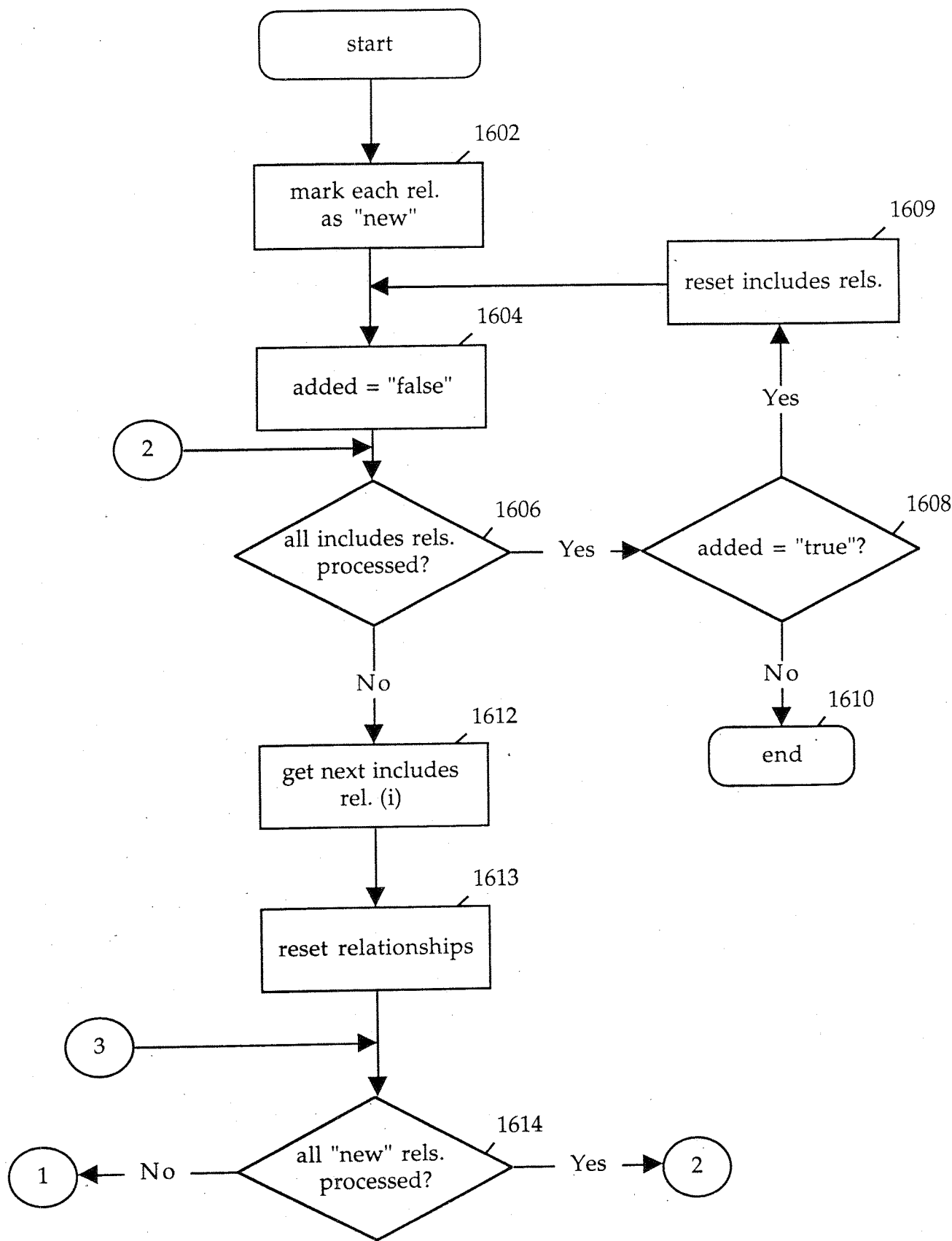


Figure 16A

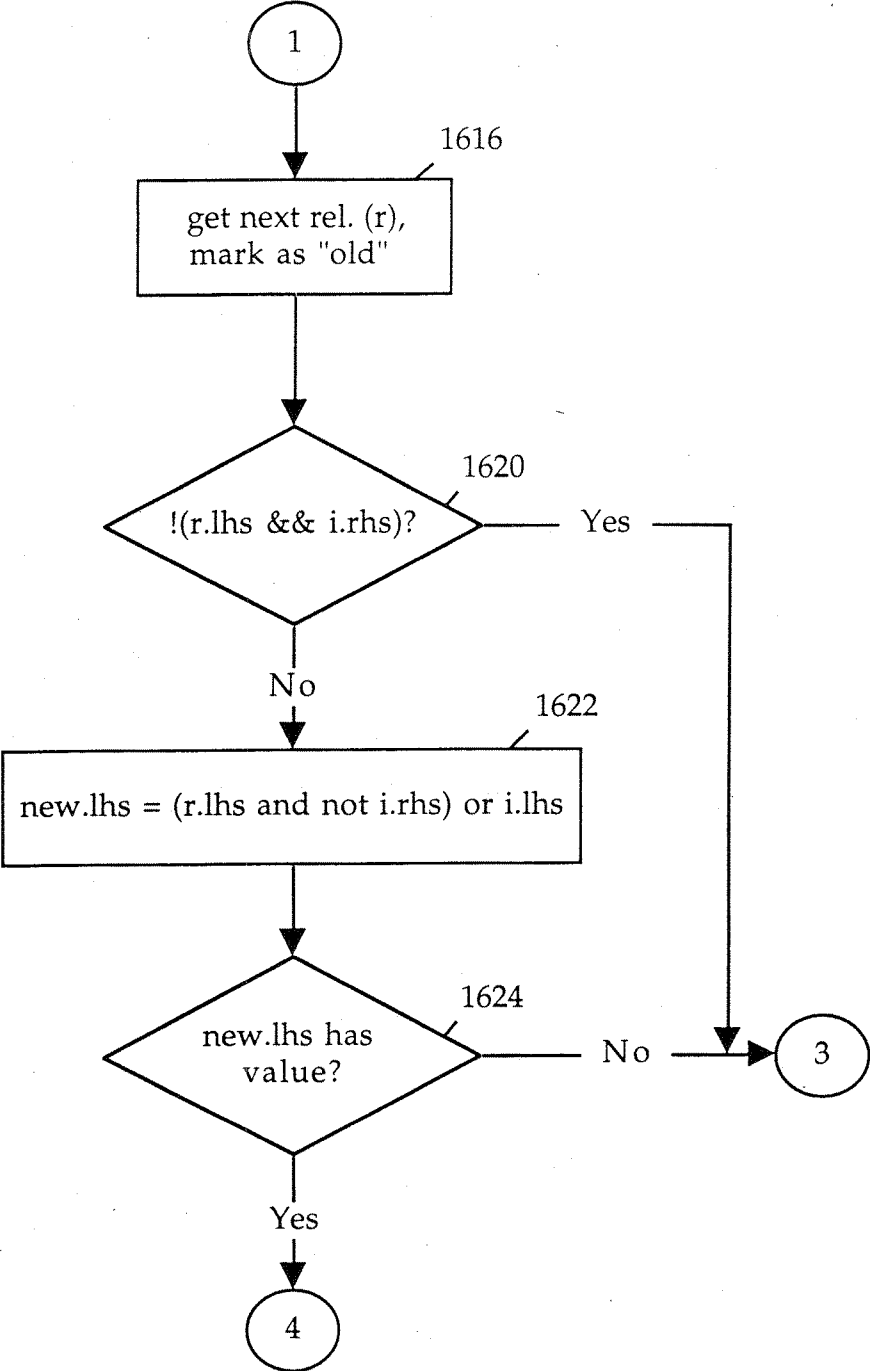


Figure 16B

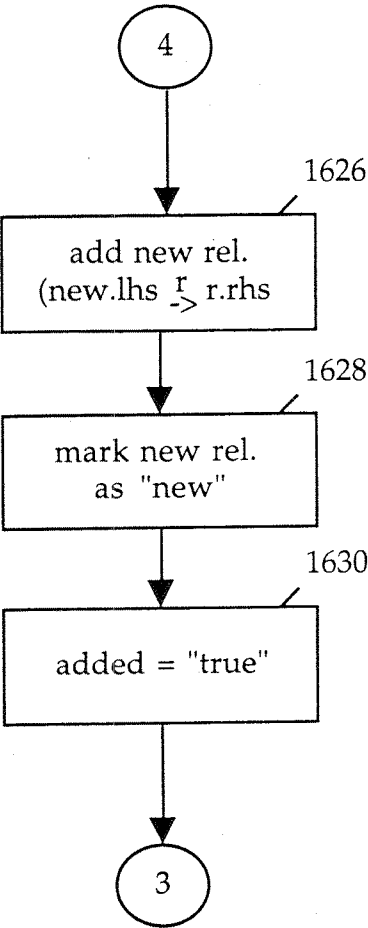


Figure 16C

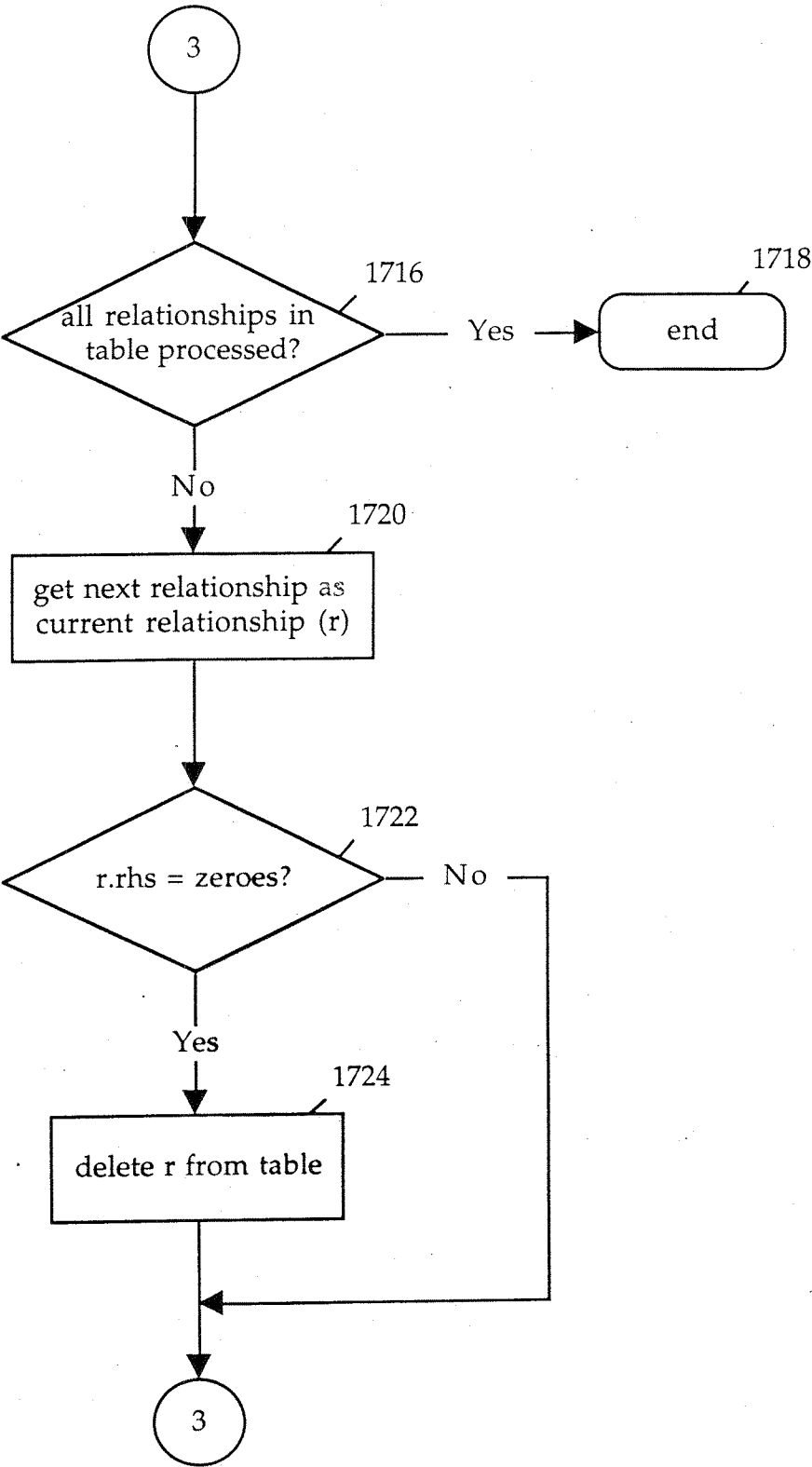


Figure 17B

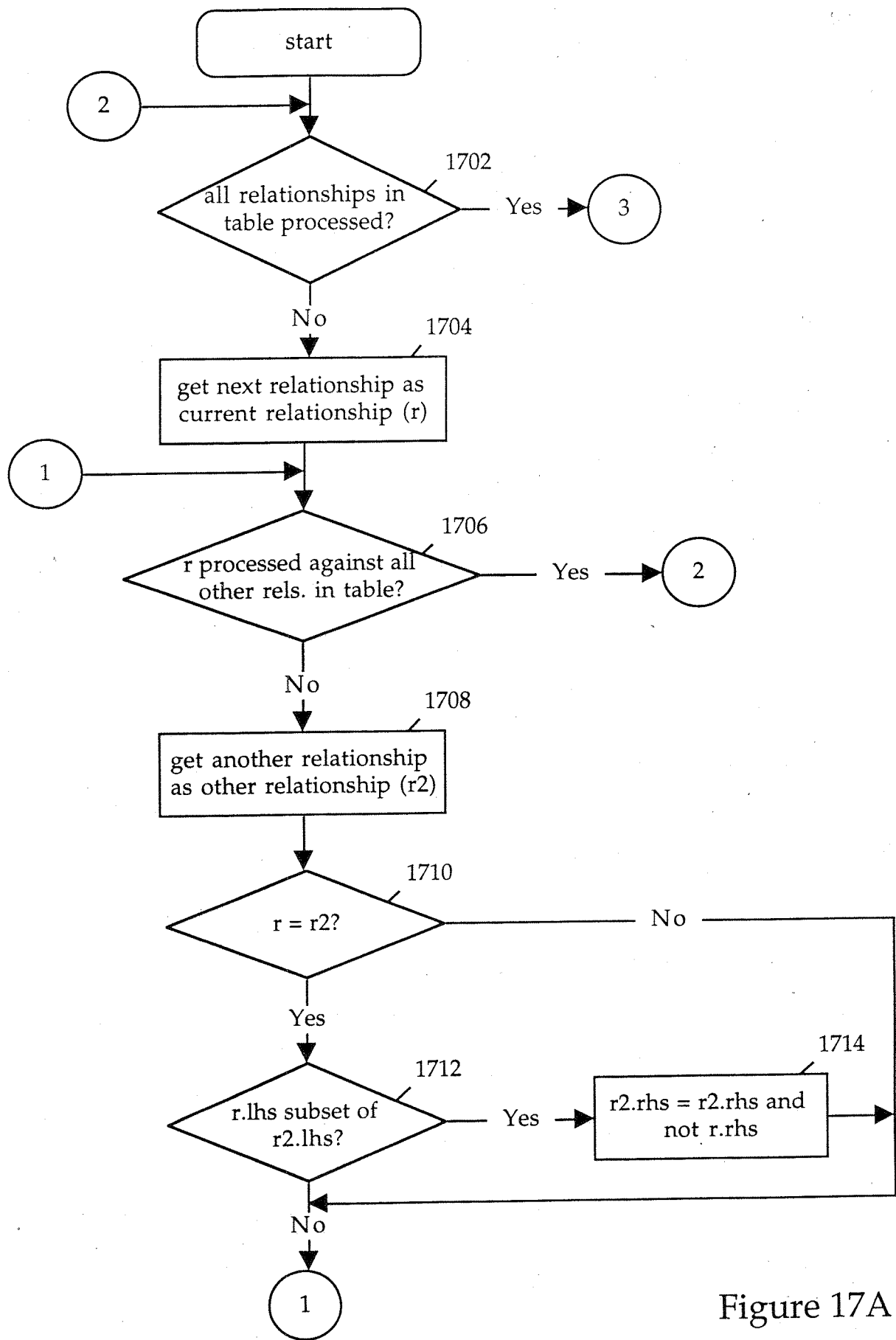


Figure 17A



UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office

Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
--------------------	-------------	-----------------------	------------------------

08/707,187 09/03/96 GUPTA N 85160,915

0212/1018

HECKER & HARRIMAN
2029 CENTURY PARK EAST SUITE 1600
LOS ANGELES CA 90067

0000

DATE MAILED: 10/18/96

NOTICE TO FILE MISSING PARTS OF APPLICATION
FILING DATE GRANTED

An Application Number and Filing Date have been assigned to this application. However, the items indicated below are missing. The required items and fees identified below must be timely submitted **ALONG WITH THE PAYMENT OF A SURCHARGE** for items 1 and 3-6 only of \$ 1300 for large entities or \$ 650 for small entities who have filed a verified statement claiming such status. The surcharge is set forth in 37 CFR 1.16(e).

If all required items on this form are filed within the period set below, the total amount owed by applicant as a ☒ large entity, ☐ small entity (verified statement filed), is \$ 1300.

Applicant is given **ONE MONTH FROM THE DATE OF THIS LETTER, OR TWO MONTHS FROM THE FILING DATE** of this application, **WHICHEVER IS LATER**, within which to file all required items and pay any fees required above to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- ☐ The statutory basic filing fee is: ☐ missing ☐ insufficient. Applicant as a ☐ large entity ☐ small entity, must submit \$ _____ to complete the basic filing fee.
- ☐ Additional claim fees of \$ _____ as a ☐ large entity, ☐ small entity, including any required multiple dependent claim fee, are required. Applicant must submit the additional claim fees or cancel the additional claims for which fees are due.
- ☒ The oath or declaration:
☒ is missing.
☐ does not cover items omitted at time of execution.


An oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date is required.
- ☐ The oath or declaration does not identify the application to which it applies. An oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.
- ☐ The signature(s) to the oath or declaration is/are: ☐ missing; ☐ by a person other than the inventor or a person qualified under 37 CFR 1.42, 1.43, or 1.47. A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.
- ☐ The signature of the following joint inventor(s) is missing from the oath or declaration:

An oath or declaration listing the names of all inventors and signed by the omitted inventor(s), identifying this application by the above Application Number and Filing Date, is required.
- ☐ The application was filed in a language other than English. Applicant must file a verified English translation of the application and a fee of \$ _____ under 37 CFR 1.17(k), unless this fee has already been paid.
- ☐ A \$ _____ processing fee is required since your check was returned without payment. (37 CFR 1.21(m)).
- ☐ Your filing receipt was mailed in error because your check was returned without payment.
- ☐ The application does not comply with the Sequence Rules. See attached Notice to Comply with Sequence Rules 37 CFR 1.821-1.825.
- ☐ Other.

Direct the response and any questions about this notice to, Attention: Application Processing Division, Special Processing and Correspondence Branch (703) 308-1202.

Attorney Ref. No.: 85160.915

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of )
GUPTA, Neeraj et al. 11/04/96)
Serial No: 08/707,187)
Filed: September 3, 1996)
For: METHOD AND APPARATUS)
FOR MAINTAINING AND)
CONFIGURING SYSTEMS)

RESPONSE TO NOTICE TO FILE MISSING PARTS

Honorable Commissioner of
Patents and Trademarks
Washington, D. C. 20231

Sir:

Responsive to the Notice to File Missing Parts of Application Filing Date Granted
mailed October 18, 1996, Applicant has enclosed the following:

1. Declarations and Powers of Attorney signed by the inventors (Messrs. Neeraj Gupta, Venky Veeraraghavan, and Ajay Agarwal);
2. A check in the amount of \$130.00 as surcharge fee set forth in 37 CFR 1.16(e);
3. A Copy of Missing Parts Notice;
4. Certificate of Mailing by FIRST CLASS Mail; and
5. A Return receipt postcard.

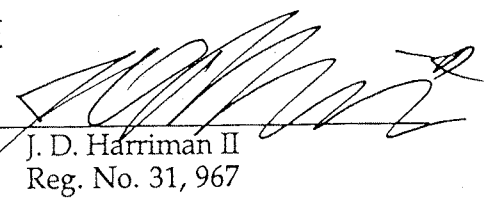
The Commissioner is hereby authorized to charge any additional fees, or credit any
overpayment associated with this communication to HECKER & HARRIMAN Deposit Account

No. 08-1520. A duplicate copy of this transmittal letter is also enclosed for this purpose.

Respectfully submitted,

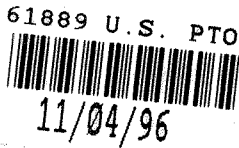
HECKER & HARRIMAN

Date: 10/31/96

By: 
J. D. Harriman II
Reg. No. 31, 967

2029 Century Park East, Suite 1600
Los Angeles, CA 90067
(310) 286-0377

0101



#3

Attorney Docket No. 85160.915

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below, next to my name,

I believe I am an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

**METHOD AND APPARATUS FOR MAINTAINING
AND CONFIGURING SYSTEMS**

the specification of which

XX Is attached hereto.
was filed on September 3, 1996 as
Application Serial No. 08/707,187
and was amended on _____
(if applicable)

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above. I do not know and do not believe that the same was ever known or used in the United States of America before my invention thereof, or patented or described in any printed publication in any country before my invention thereof or more than one year prior to this application, that the same was not in public use or on sale in the United States of America more than one year prior to this application, and that the invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months prior to this application.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56(a).

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119, of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

<u>Prior Foreign Application(s)</u>			<u>Priority Claimed</u>	
<u>(Number)</u>	<u>(Country)</u>	<u>(Day/Month/Year Filed)</u>	<u>Yes</u>	<u>No</u>
<u>(Number)</u>	<u>(Country)</u>	<u>(Day/Month/Year Filed)</u>	<u>Yes</u>	<u>No</u>
<u>(Number)</u>	<u>(Country)</u>	<u>(Day/Month/Year Filed)</u>	<u>Yes</u>	<u>No</u>
<u>(Number)</u>	<u>(Country)</u>	<u>(Day/Month/Year Filed)</u>	<u>Yes</u>	<u>No</u>

Attorney Docket No. 85160.915

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Serial No.)	(Filing Date)	(Status -- patented, pending, abandoned)
(Application Serial No.)	(Filing Date)	(Status -- patented,

I hereby appoint HECKER & HARRIMAN, a firm including: Gary A. Hecker, Reg. No. 31,023; J. D. Harriman II, Reg. No. 31,967; Christopher A. Mathews, Reg. No. 35,944; Frank Weyer, Reg. No. 33,050; Ross D. Snyder, Reg. No. 37,730; Carole A. Quinn, Reg. No. 39,000 and Jason Feldmar, Reg. No. 39,187 with offices located at 2029 Century Park East, Suite 1600, Los Angeles, California 90067, telephone (310) 286-0377, my attorneys with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.

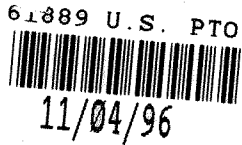
I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Joint Inventor NEERAJ GUPTA 1-00
Inventor's Signature: [Signature] Date: 10/21/96
Residence: 6804 N. CAP of TX HWY #123 Citizenship: CANADIAN
Austin, TX TX 78731 USA
(City, State) (Country)

Post Office Address: 6804 N. CAP of TX HWY # 123
Austin, TX 78731 U.S.A.

CERTIFICATE OF MAILING
This is to certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231 on:

OCT. 31, 1996
[Signature] 10/31/96
Signature Walter FEFER Date



Attorney Docket No. 85160.915

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:
My residence, post office address and citizenship are as stated below, next to my name,
I believe I am an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

**METHOD AND APPARATUS FOR MAINTAINING
AND CONFIGURING SYSTEMS**

the specification of which

XX Is attached hereto.
was filed on September 3, 1996 as
Application Serial No. 08/707,187
and was amended on _____
(if applicable)

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above. I do not know and do not believe that the same was ever known or used in the United States of America before my invention thereof, or patented or described in any printed publication in any country before my invention thereof or more than one year prior to this application, that the same was not in public use or on sale in the United States of America more than one year prior to this application, and that the invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months prior to this application.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56(a).

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119, of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

<u>Prior Foreign Application(s)</u>			<u>Priority Claimed</u>	
<u>(Number)</u>	<u>(Country)</u>	<u>(Day/Month/Year Filed)</u>	<u>Yes</u>	<u>No</u>
<u>(Number)</u>	<u>(Country)</u>	<u>(Day/Month/Year Filed)</u>	<u>Yes</u>	<u>No</u>
<u>(Number)</u>	<u>(Country)</u>	<u>(Day/Month/Year Filed)</u>	<u>Yes</u>	<u>No</u>
<u>(Number)</u>	<u>(Country)</u>	<u>(Day/Month/Year Filed)</u>	<u>Yes</u>	<u>No</u>

Attorney Docket No. 85160.915

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

_____ (Application Serial No.)	_____ (Filing Date)	_____ (Status -- patented, pending, abandoned)
_____ (Application Serial No.)	_____ (Filing Date)	_____ (Status -- patented,

I hereby appoint HECKER & HARRIMAN, a firm including: Gary A. Hecker, Reg. No. 31,023; J. D. Harriman II, Reg. No. 31,967; Christopher A. Mathews, Reg. No. 35,944; Frank Weyer, Reg. No. 33,050; Ross D. Snyder, Reg. No. 37,730; Carole A. Quinn, Reg. No. 39,000 and Jason Feldmar, Reg. No. 39,187 with offices located at 2029 Century Park East, Suite 1600, Los Angeles, California 90067, telephone (310) 286-0377, my attorneys with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

2-00

Full Name of Joint Inventor VENKY VEERARAGHAVAN

Inventor's Signature: *Venkatesh* Date: Oct 15, 1996

Residence: 11500 JOLLYVILLE RD #3711 Citizenship: INDIA

AUSTIN TX 78759 USA

(City, State) (Country)

Post Office Address: 11500 JOLLYVILLE RD #3711

AUSTIN TX 78759 USA

CERTIFICATE OF MAILING

This is to certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231 on:

OCT. 31, 1996

[Signature] 10/31/96

Signature *W. H. H. FEDERAL* Date



11/04/96

Attorney Docket No. 85160.915

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below, next to my name,

I believe I am an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

**METHOD AND APPARATUS FOR MAINTAINING
AND CONFIGURING SYSTEMS**

the specification of which

XX Is attached hereto.
was filed on September 3, 1996 as
Application Serial No. 08/707,187
and was amended on _____
(if applicable)

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above. I do not know and do not believe that the same was ever known or used in the United States of America before my invention thereof, or patented or described in any printed publication in any country before my invention thereof or more than one year prior to this application, that the same was not in public use or on sale in the United States of America more than one year prior to this application, and that the invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months prior to this application.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56(a).

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119, of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

<u>Prior Foreign Application(s)</u>			<u>Priority Claimed</u>	
<u>(Number)</u>	<u>(Country)</u>	<u>(Day/Month/Year Filed)</u>	<u>Yes</u>	<u>No</u>
<u>(Number)</u>	<u>(Country)</u>	<u>(Day/Month/Year Filed)</u>	<u>Yes</u>	<u>No</u>
<u>(Number)</u>	<u>(Country)</u>	<u>(Day/Month/Year Filed)</u>	<u>Yes</u>	<u>No</u>
<u>(Number)</u>	<u>(Country)</u>	<u>(Day/Month/Year Filed)</u>	<u>Yes</u>	<u>No</u>

Attorney Docket No. 85160.915

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Status -- patented,
pending, abandoned)

(Status -- patented,

I hereby appoint HECKER & HARRIMAN, a firm including: Gary A. Hecker, Reg. No. 31,023; J. D. Harriman II, Reg. No. 31,967; Christopher A. Mathews, Reg. No. 35,944; Frank Weyer, Reg. No. 33,050; Ross D. Snyder, Reg. No. 37,730; Carole A. Quinn, Reg. No. 39,000 and Jason Feldmar, Reg. No. 39,187 with offices located at 2029 Century Park East, Suite 1600, Los Angeles, California 90067, telephone (310) 286-0377, my attorneys with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

AJAY AGARWAL 3-00

Date: Oct 21, 1978

1603B ENFIELD RD

Citizenship: U.S.A

(City, State)

(Country)

1603B Enfield Rd.

Austin, TX 78703 U.S.A.

CERTIFICATE OF MAILING

This is to certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231 on:

Oct. 31, 1994

Signature MAHMOUD FERRIS Date _____



11/04/96

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark OfficeAddress: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

#3

APPLICATION NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
--------------------	-------------	-----------------------	------------------------

08/707,187 09/03/96 GUPTA N 85160.915

0212/1018

HECKER & HARRIMAN
2029 CENTURY PARK EAST SUITE 1600
LOS ANGELES CA 90067

0000

DATE MAILED:

10/18/96

**NOTICE TO FILE MISSING PARTS OF APPLICATION
FILING DATE GRANTED**

An Application Number and Filing Date have been assigned to this application. However, the items indicated below are missing. The required items and fees identified below must be timely submitted **ALONG WITH THE PAYMENT OF A SURCHARGE** for items 1 and 3-6 only of \$_____ for large entities or \$_____ for small entities who have filed a verified statement claiming such status. The surcharge is set forth in 37 CFR 1.16(e).

If all required items on this form are filed within the period set below, the total amount owed by applicant as a ☒ large entity, ☐ small entity (verified statement filed), is \$_____.

Applicant is given **ONE MONTH FROM THE DATE OF THIS LETTER, OR TWO MONTHS FROM THE FILING DATE** of this application, **WHICHEVER IS LATER**, within which to file all required items and pay any fees required above to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

1. ☐ The statutory basic filing fee is: ☐ missing ☐ insufficient. Applicant as a ☐ large entity ☐ small entity, must submit \$_____ to complete the basic filing fee.

2. ☐ Additional claim fees of \$_____ as a ☐ large entity, ☐ small entity, including any required multiple dependent claim fee, are required. Applicant must submit the additional claim fees or cancel the additional claims for which fees are due.

3. ☐ The oath or declaration:
☐ is missing.
☐ does not cover items omitted at time of execution.

An oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date is required.

4. ☐ The oath or declaration does not identify the application to which it applies. An oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.

5. ☐ The signature(s) to the oath or declaration is/are: ☐ missing; ☐ by a person other than the inventor or a person qualified under 37 CFR 1.42, 1.43, or 1.47. A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.

6. ☐ The signature of the following joint inventor(s) is missing from the oath or declaration:

_____ An oath or declaration listing the names of all inventors and signed by the omitted inventor(s), identifying this application by the above Application Number and Filing Date, is required.

7. ☐ The application was filed in a language other than English. Applicant must file a verified English translation of the application and a fee of \$_____ under 37 CFR 1.17(k), unless this fee has

already been paid.

8. ☐ A \$_____ processing fee is required since your check was returned without payment. (37 CFR 1.21(m)).

9. ☐ Your filing receipt was mailed in error because your check was returned without payment.

10. ☐ The application does not comply with the Sequence Rules. See attached Notice to Comply with Sequence Rules 37 CFR 1.821-1.825.

11. ☐ Other.

Direct the response and any questions about this notice to, Attention: Application Processing Division, Special Processing and Correspondence Branch (703) 308-1202.



UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office

Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NO.
08/707,187	09/03/96	GUPTA	N 85160.915

HECKER & HARRIMAN
SUITE 1600
2029 CENTURY PARK EAST
LOS ANGELES CA 90067

LM21/0106

EXAMINER

KRIESS, K

ART UNIT PAPER NUMBER

2787

4

DATE MAILED:

01/06/98

This is a communication from the examiner in charge of your application.
COMMISSIONER OF PATENTS AND TRADEMARKS

OFFICE ACTION SUMMARY

☒ Responsive to communication(s) filed on 9-3-96

☐ This action is **FINAL**.

☐ Since this application is in condition for allowance except for formal matters, **prosecution as to the merits is closed** in accordance with the practice under *Ex parte Quayle*, 1935 D.C. 11; 453 O.G. 213.

A shortened statutory period for response to this action is set to expire 3 month(s), or thirty days, whichever is longer, from the mailing date of this communication. Failure to respond within the period for response will cause the application to become abandoned. (35 U.S.C. § 133). Extensions of time may be obtained under the provisions of 37 CFR 1.136(a).

Disposition of Claims

☒ Claim(s) 1-59 is/are pending in the application.

Of the above, claim(s) _____ is/are withdrawn from consideration.

☒ Claim(s) 16-19 and 35-59 is/are allowed.

☒ Claim(s) ~~1-15, 16-19, 20-34~~ 1, 2, 15, 20, 21 and 35 is/are rejected.

☒ Claim(s) 3-14 and 22-33 is/are objected to.

☐ Claims _____ are subject to restriction or election requirement.

Application Papers

☒ See the attached Notice of Draftsperson's Patent Drawing Review, PTO-948.

☐ The drawing(s) filed on _____ is/are objected to by the Examiner.

☐ The proposed drawing correction, filed on _____ is ☐ approved ☐ disapproved.

☐ The specification is objected to by the Examiner.

☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

☐ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).

☐ All ☐ Some* ☐ None of the CERTIFIED copies of the priority documents have been

☐ received.

☐ received in Application No. (Series Code/Serial Number) _____

☐ received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

*Certified copies not received: _____

☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

☒ Notice of Reference Cited, PTO-892

☐ Information Disclosure Statement(s), PTO-1449, Paper No(s) _____

☐ Interview Summary, PTO-413

☒ Notice of Draftsperson's Patent Drawing Review, PTO-948

☐ Notice of Informal Patent Application, PTO-152

-- SEE OFFICE ACTION ON THE FOLLOWING PAGES --

Art Unit: 2787

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless --

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(c) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

2. Claims 1, 2, 15, 20, 21 and 34 are rejected under 35 U.S.C. 102(a), (b) (c) as being anticipated by Addesso et al.

3. As per claim 1, Addesso disclose the claimed:

“generating a definition”(Addesso’s models, col. 3, line 50);

“product relationships” (Addesso’s primitives, col. 3, line 7);

“generating a set of relationships” (Addesso’s relationships for associating entities, col. 3, line 8);

“conveyed graphically” (Addesso’s graphical interface, col. 3, line 6); and

“configuring said system” (Addesso’s creating modelling objects, col. 4, line 1).

4. As per claim 2, Addesso discloses the claimed:

“input” (Addesso’s low level user, col. 28, line 34);

Serial Number: 08/707,187

Page 3

Art Unit: 2787

“validating” (Addesso’s verification, col. 28, line 32); and

“Identifying a set of valid configuration options” (Addesso’s verifying compliance, col. 28, line 34).

5. As per claim 15, Addesso disclose the claimed:

“parts catalog” (Addesso’s primitive in database, col. 27, line 41).

6. As per claims 20, 21 and 34, Addesso discloses the claimed article of manufacture comprising computer usable medium and program code as is claimed. See col. 5, lines 11-15. Refer to discussion of claims 1, 2 and 15 for further details of how Addesso discloses the claimed features of claims 20, 21 and 34.

7. Claims 3-14 and 22-33 objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

8. Claims 16-19 and 35-59 are allowed.

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kevin A. Kriess whose telephone number is (703) 305-9668. The examiner can normally be reached on Monday-Thursday from 7:30 am to 6:00 pm.

Serial Number: 08/707,187

Page 4

Art Unit: 2787

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh, can be reached on (703) 305-9648. The fax phone number for this Group is (703) 308-5359.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-3900.

K.Kriess/vgj
December 31, 1997


KEVIN A. KRIESS
PRIMARY EXAMINER

Attachment _____

The drawings submitted with this application were declared informal by the applicant. Accordingly they have not been reviewed by a draftsperson at this time. When formal drawings are submitted, the draftsperson will perform a review.

Direct any inquiries concerning drawing review to the Drawing Review Branch (703) 305-8404.

FORM PTO-892 (REV. 2-92)		U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE		SERIAL NO. 68/707,187		GROUP ART UNIT 2787		ATTACHMENT TO PAPER NUMBER 4						
NOTICE OF REFERENCES CITED				APPLICANT(S) GUPTA et al.										
U.S. PATENT DOCUMENTS														
*		DOCUMENT NO.						DATE	NAME	CLASS	SUB-CLASS	FILING DATE IF APPROPRIATE		
A		5019961	5-91	Addesso et al.						364	192			
B		5586052	12-96	Iannuzzi et al.						364	512			
C		5659478	8-97	Pennisi et al.						364	468.01			
D		5355317	10-94	Talbot et al.						364	468			
E		5357440	10-94	Talbot et al.						364	467			
F		5019992	5-91	Brown et al.						364	468			
G														
H														
I														
J														
K														
FOREIGN PATENT DOCUMENTS														
*		DOCUMENT NO.						DATE	COUNTRY	NAME	CLASS	SUB-CLASS	PERTINENT SHTS. DWG.	PP. SPEC.
L														
M														
N														
O														
P														
Q														
OTHER REFERENCES (Including Author, Title, Date, Pertinent Pages, Etc.)														
	R													
	S													
	T													
	U													
EXAMINER		/ [Signature]		DATE		12-4-97								
* A copy of this reference is not being furnished with this office action. (See Manual of Patent Examining Procedure, section 707.05 (a).)														



Attorney Docket No. 85160.915

gp 2787.

#5
H.W.M.S
4/22/98

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of)
)
GUPTA, Neeraj et al.)
)
Serial No: 08/707,187)
)
Filed: September 3, 1996)
)
For: METHOD AND APPARATUS)
FOR MAINTAINING)
AND CONFIGURING)
SYSTEMS)

Examiner: K. A. KRIESS

Group Art Unit: 2787

RECEIVED
APR 21 98
GROUP 2600

INFORMATION DISCLOSURE STATEMENT

Assistant Commissioner for Patents
Washington, D. C. 20231

Attn: Examiner Kevin A. Kriess
Group Art Unit 2787

Sir:

Pursuant to 37 C.F.R. § 1.56 and § 1.98, Applicants hereby submit the following art in connection with the above-referenced patent application for "METHOD AND APPARATUS FOR MAINTAINING AND CONFIGURING SYSTEMS."

<u>U.S. Patent No.</u>	<u>Inventor(s)</u>	<u>Issue Date</u>
4,796,194	Atherton	Jan. 3, 1989

Applicants request that the disclosed reference be included with the references cited in this patent application and that it be printed in any patent that may issue from this patent application.

Attorney Docket No. 85160.915

Pursuant to 37 C.F.R. § 1.97(c), no fee is required as applicants enclose a Certification under 37 C.F.R. §1.97(e)(1). Applicants also enclose Form PTO-1449 and a copy of the reference disclosed pursuant to 37 C.F.R. § 1.98(a).

The Commissioner is hereby authorized to charge any fees associated with this communication to Hecker & Harriman Deposit Account No. 08-1520. A duplicate copy of this transmittal letter is also enclosed for this purpose.

Respectfully submitted,

HECKER & HARRIMAN

Date:

4/17/98

By:

J. D. Harriman II

Reg. No. 31,967

2029 Century Park East, Suite 1600
Los Angeles, CA 90067
(310) 286-0377

CERTIFICATE OF MAILING

This is to certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as EXPRESS MAIL with Label No. EM484492311US in an envelope addressed to: Assistant Commissioner for Patents, Washington, D. C., 20231 on :

April 17, 1998

Mario Federis

Date

04/17/98

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of
GUPTA, Neeraj et al.

Serial No.: 08/707,187

Filed: September 3, 1996

For: METHOD AND APPARATUS
FOR MAINTAINING
AND CONFIGURING
SYSTEMS



Examiner: K. A. Kriess

Group Art Unit: 2787

RECEIVED
APR 21 98
GROUP 2600

CERTIFICATION UNDER 37 C.F.R. § 1.97(e)(1)

Assistant Commissioner for Patents
Washington, D. C. 20231

Sir:

The undersigned hereby certifies that the item of information contained in the enclosed Information Disclosure Statement was cited in a communication from a PCT Searching Authority in a corresponding PCT patent application not more than three months prior to the filing of the statement.

Respectfully submitted,
HECKER & HARRIMAN

Date: 4/17/98

By: [Signature]
J. D. Harriman II
Reg. No. 31,967

2029 Century Park East, Suite 1600
Los Angeles, CA 90067
(310) 286-0377

CERTIFICATE OF MAILING

This is to certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as EXPRESS MAIL with Label No. EM484492311US in an envelope addressed to: Assistant Commissioner for Patents, Washington, D. C., 20231 on :

April 17, 1998
[Signature]
Mario Federis

Date 04/17/98

85160.915

08/707.187

INFORMATION DISCLOSURE CITATION IN AN APPLICATION

(Use several sheets if necessary)

Applicant
GUPTA, Neeraj et al.

Filing Date
Sept. 3, 1996

Group Art Unit	2787
----------------	------

U.S. PATENT DOCUMENTS

RECEIVED
APR 21 98
GROUP 2600

FOREIGN PATENT DOCUMENTS

Translation	
YES	NO

OTHER DOCUMENTS (including Author, Title, Date, Pertinent Pages, Etc.)

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

EXAMINER

DATE CONSIDERED _____

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to the applicant.



6/a
M. [Signature]
5-15-98.

Attorney Docket No.: 85160.915

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of)	
)	
GUPTA, N. et al.)	Examiner: Kriess, Kevin A.
)	
Serial No: 08/707,187)	Group Art Unit: 2787
)	
Filed: September 3, 1996)	
)	
For: METHOD AND APPARATUS FOR)	
MAINTAINING AND)	
CONFIGURING SYSTEMS)	

RECEIVED
98 MAY 14 PM 2:47
GROUP 2700

AMENDMENT AND RESPONSE

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

This is in response to the outstanding Examiner's Action in the present application. Please amend the present application as follows:

A

IN THE CLAIMS:

1. (Once amended) A method of configuring a system comprising the steps of:

generating a definition for said system, said definition containing one or more elements and being conveyed graphically using a set of product relationships, said product relationships identifying classifications for said one or more elements, said product relationships comprising an includes classification;

generating a set of part relationships between said one or more elements, said set of part relationships being conveyed graphically [using a set of part relationships]; and

configuring said system using said definition and said set of product relationships and said part relationships.

~~2. (Not amended) The method of claim 1 wherein said step of configuring further comprises the steps of:~~

~~receiving input from a configuration user;~~

~~validating said input based on said definition, said set of relationships, and a current configuration state; and~~

~~identifying a set of valid configuration options using said definition, said set of relationships and said current configuration state.~~

3. (Not amended) The method of claim 2 wherein said step of identifying further comprises the steps of:

determining whether a relationship in said set of relationships is active;

modifying said configuration to satisfy said active relationship when it is determined that a modification is needed;

making said relationship notActivatable when it is determined that said relationship is not active and the activation of said relationship would render the configuration invalid.

4. (Not amended) The method of claim 3 wherein said step of determining whether a relationship is active further comprises the step of determining whether the elements specified in the left-hand side of said relationship are present in said configuration.

5. (Not amended) The method of claim 4 wherein said step of determining whether the elements specified in the left-hand side of said relationship are present in said configuration further comprises the step of performing a bit-level comparison of the bits corresponding to the elements in said left-hand side of the relationship with a "selected" bit vector.

6. (Not amended) The method of claim 3 wherein said relationship is an includes relationship, said step of modifying further

comprises the step of including in said configuration elements identified in a right-hand side of said relationship.

7. (Not amended) The method of claim 3 wherein said relationship is an excludes relationship, said step of modifying further comprises the step of marking said elements identified in the right-hand side of said relationship as excluded and notSelectable.

8. (Not amended) The method of claim 3 wherein said relationship is a removes relationship, said step of modifying further comprises the step of removing said elements identified in the right-hand side of said relationship from said configuration.

9. (Not amended) The method of claim 3 wherein said relationship is a requires choice relationship, said step of modifying further comprises the steps of:

determining whether a plurality of elements identified in a right-hand side of said relationship are present in said configuration such that said relationship is satisfied;

modifying said configuration to satisfy said requires choice relationship when it is determined that only one path exists to satisfy said relationship and said relationship is not already satisfied.

10. (Not amended) The method of claim 3 wherein said step of making said relationship notActivateable further comprises the step of

marking one or more elements in said left-hand side of said relationship as notSelectable.

11. (Not amended) The method of claim 2 wherein said definition, each of said set of relationships, and said current configuration state are expressed as bit vectors having bits that correspond to elements.

12. (Not amended) The method of claim 2 wherein said step of validating further comprises the step of determining whether the relationships other than those marked notActivateable in said set of relationships can be satisfied.

13. (Not amended) The method of claim 2 wherein said step of generating a definition selects said one or more elements for inclusion in said definition, said input is made without regard for the order in which said one or more elements are selected for said definition.

14. (Not amended) The method of claim 2 wherein said configuration state identifies a set of selected elements and a set of selectable elements for said system, said input capable of selecting any member of said set of selectable elements and being made despite the order in which members of said set of selected elements were selected.

15. (Not amended) The method of claim 1 wherein a parts catalog is used to store information associated with said one or more elements, said information including an identifier and description.

16. (Not amended) A method for defining a system that can be configured comprising the steps of:

identifying an element of said system;

specifying said element as an included element when said element is automatically included in the configuration;

specifying said element as an optional element when said element is not a necessary component of said system;

specifying said element as a required choice when said element is a group that contains one or more members from which to choose.

17. (Not amended) The method of claim 16 wherein said element is a member of a first element set said first element set containing one or more elements, said method further comprising the steps of:

identifying a second element set of said system, said second element set containing one or more elements; and

specifying a relationship between said first element set and said second element set.

18. (Not amended) The method of claim 17 wherein said step of specifying further comprises the steps of:

specifying that said second element set is to be included in said system when said first element set is present in said system;

specifying that said second element set is to be excluded from said system when said first element set is present in said system; and

specifying that said second element set is to be removed from said system when said first element set is present in said system;

19. (Not amended) The method of claim 16 further comprising the steps of:

specifying a group of elements;

specifying a relationship between said first element set and said group of elements such that one or more elements in said group of elements must be chosen when said first element set is present in said system.

20. (Once amended) An article of manufacture comprising:

a computer usable medium having computer readable program code embodied therein for configuring a system comprising:

computer readable program code configured to cause a computer to generate a definition for said system, said definition containing one or more elements and being conveyed graphically using a set of product relationships, said product relationships identifying classifications for said one or more elements, said product relationships comprising an includes classification;

computer readable program code configured to cause a computer to generate a set of part relationships between said one or more elements, said

set of part relationships being conveyed graphically[using a set of part relationships]; and

computer readable program code configured to cause a computer to configure said system using said definition and said set of product relationships and said part relationships.

21. (Not amended) The article of manufacture of claim 20 wherein said computer readable program code configured to cause a computer to configure further comprises:

computer readable program code configured to cause a computer to receive input from a configuration user;

computer readable program code configured to cause a computer to validate said input based on said definition, said set of relationships, and a current configuration state; and

computer readable program code configured to cause a computer to identify a set of valid configuration options using said definition, said set of relationships and a current configuration state.

22. (Not amended) The article of manufacture of claim 21 wherein said computer readable program code configured to cause a computer to identify further comprises:

computer readable program code configured to cause a computer to determine whether a relationship in said set of relationships is active;

computer readable program code configured to cause a computer to modify said configuration to satisfy said active relationship when it is determined that a modification is needed;

computer readable program code configured to cause a computer to make said relationship notActivatable when it is determined that said relationship is not active and the activation of said relationship would render the configuration invalid.

23. (Not amended) The article of manufacture of claim 22 wherein said computer readable program code configured to cause a computer to determine whether a relationship is active further comprises computer readable program code configured to cause a computer to determine whether the elements specified in the left-hand side of said relationship are present in said configuration.

24. (Not amended) The article of manufacture of claim 23 wherein said computer readable program code configured to cause a computer to determine whether the elements specified in the left-hand side of said relationship are present in said configuration further comprises computer readable program code configured to cause a computer to perform a bit-level comparison of the bits corresponding to the elements in said left-hand side of the relationship with a "selected" bit-vector.

25. (Not amended) The article of manufacture of claim 22 wherein said relationship is an includes relationship, said computer readable program

code configured to cause a computer to modify further comprises computer readable program code configured to cause a computer to include in said configuration elements identified in a right-hand side of said relationship.

26. (Not amended) The article of manufacture of claim 22 wherein said relationship is an excludes relationship, said computer readable program code configured to cause a computer to modify further comprises computer readable program code configured to cause a computer to mark said elements identified in the right-hand side of said relationship as excluded and notSelectable.

27. (Not amended) The article of manufacture of claim 22 wherein said relationship is a removes relationship, said computer readable program code configured to cause a computer to modify further comprises computer readable program code configured to cause a computer to remove said elements identified in the right-hand side of said relationship from said configuration.

28. (Not amended) The article of manufacture of claim 22 wherein said relationship is a requires choice relationship, said computer readable program code configured to cause a computer to modify further comprises:

computer readable program code configured to cause a computer to determine whether a plurality of elements identified in a right-hand side of

said relationship are present in said configuration such that said relationship is satisfied;

computer readable program code configured to cause a computer to modify said configuration to satisfy said requires choice relationship when it is determined that only one path exists to satisfy said relationship and said relationship is not already satisfied.

29. (Not amended) The article of manufacture of claim 22 wherein said computer readable program code configured to cause a computer to make said relationship notActivateable further comprises computer readable program code configured to cause a computer to mark one or more elements in said left-hand side of said relationship as notSelectable.

30. (Not amended) The article of manufacture of claim 21 wherein said definition, each of said set of relationships, and said current configuration state are expressed as bit vectors having bits that correspond to elements.

31. (Not amended) The article of manufacture of claim 21 wherein said code configured to cause a computer to validate further comprises program code configured to cause a computer to determine whether the relationships other than those marked notActivateable in said set of relationships can be satisfied.

32. (Not amended) The article of manufacture of claim 21 wherein said code configured to cause a computer to generate a definition selects said one or more elements for inclusion in said definition, said input is made without regard for the order in which said one or more elements are selected for said definition.

33. (Not amended) The article of manufacture of claim 21 wherein said configuration state identifies a set of selected elements and a set of selectable elements for said system, said input capable of selecting any member of said set of selectable elements and being made despite the order in which members of said set of selected elements were selected.

34. (Not amended) The article of manufacture of claim 17 wherein a parts catalog is used to store information associated with said one or more elements, said information including an identifier and description.

35. (Not amended) An article of manufacture comprising
a computer usable medium having computer readable program code embodied therein for defining a system that can be configured comprising:

computer readable program code configured to cause a computer to identify an element of said system;

computer readable program code configured to cause a computer to specify said element as an included element when said element is automatically included in the configuration;

computer readable program code configured to cause a computer to specify said element as an optional element when said element is not a necessary component of said system;

computer readable program code configured to cause a computer to specify said element as a required choice when said element is a group that contains one or more members from which to choose.

36. (Not amended) The article of manufacture of claim 35 wherein said element is a member of a first element set said first element set containing one or more elements, said article of manufacture further comprising:

computer readable program code configured to cause a computer to identify a second element set of said system, said second element set containing one or more elements; and

computer readable program code configured to cause a computer to specify a relationship between said first element set and said second element set.

37. (Not amended) The article of manufacture of claim 36 wherein said computer readable program code configured to cause a computer to specify further comprises:

computer readable program code configured to cause a computer to specify that said second element set is to be included in said system when said first element set is present in said system;

computer readable program code configured to cause a computer to specify that said second element set is to be excluded from said system when said first element set is present in said system; and

computer readable program code configured to cause a computer to specify that said second element set is to be removed from said system when said first element set is present in said system;

38. (Not amended) The article of manufacture of claim 35 further comprising:

computer readable program code configured to cause a computer to specify a group of elements;

computer readable program code configured to cause a computer to specify a relationship between said first element set and said group of elements such that one or more elements in said group of elements must be chosen when said first element set is present in said system.

39. (Not amended) A method of configuring a system comprising the steps of:

translating a system definition from an external representation comprising a set of intuitive relationships between components of said

system definition into an internal representation comprising a set of internal relationships;

validating configuration input using said internal representation; and
modifying a configuration state based on said user input.

40. (Not amended) The method of claim 39 wherein said step of translating further comprises the step of:

removing a plurality of hierarchical levels from said external representation such that said internal representation is a single level.

41. (Not amended) The method of claim 40 wherein said step of translating further comprises the step of:

creating a set of relationships based on said hierarchical levels, each one in said set of relationships having a first set of components and a second set of components and a relationship operator.

42. (Not amended) The method of claim 41 wherein said relationship operator is chosen from the operators include, exclude, remove, and requires choice.

43. (Not amended) The method of claim 39 wherein each of said internal relationships has a left-hand side, a relationship operator and a right-hand side, said step of translating further comprises the steps of:

determining whether a group of components exists on said left-hand side of one of said internal relationships;

performing the following steps when a group of components exist on said left-hand side:

creating a new component for said group; and

creating a new relationship for each component of said group of components such that each component will include said new component.

44. (Not amended) The method of claim 39 wherein a type of said internal relationships is an include relationship, said include relationship identifies a second set of components that are included in said system when all of a first set of components exist in said system, said step of translating further comprises the steps of:

determining whether a subset of the union of said first and second sets of components in a first includes relationship comprises a first set of components in a second includes relationship; and

generating a combined includes relationship, said combined relationship comprises said first set of components of said first includes relationship and said second set of components from said first and second includes relationships when said subset of the union of said first and second sets of components in said first includes relationship comprises a first set of components in a second includes relationship.

45. (Not amended) The method of claim 39 wherein a type of said internal relationships is a requires choice relationship, said requires choice relationship identifies a second set of components from which a number of components must be chosen when a first set of components exist in said system definition, said step of translating further comprises the steps of:

determining whether a second set of components of a second requires choice relationship is a proper subset of a second set of components of a first requires choice relationship;

performing the following steps when said condition exists:

creating a new relationship such that said new relationship excludes a second set of components of said new relationship when a first set of components of said new relationship is selected, said new relationship's first set of components is the union of the first set of components of said first and second requires choice relationships and said new relationship's second set of components is the difference of the second set of components of said first and second requires choice relationship.

46. (Not amended) The method of claim 39 wherein a type of said internal relationships is an includes relationship, said includes relationship identifies a second set of components that are included in said system when all of a first set of components exist in said system, said step of translating further comprises the steps of:

determining whether the second set of components of an includes relationship are contained in a first set of components of another internal relationship;

replacing said second set of components of said includes relationship in said another internal relationship with said first set of components of said includes relationship when said second set of components of an includes relationship are contained in a first set of components of another internal relationship.

47. (Not amended) The method of claim 39 said internal representation comprises multiple types of relationships, each type of relationship having a first and second set of components and an operation, said operation is performed on said second set of components when said first set of components exist in said system.

48. (Not amended) The method of claim 47 wherein said step of translating further comprises the steps of:

determining whether a first and second relationship are the same type of relationship;

performing the following steps when they are the same type:

determining whether the first relationship's first set of components is a proper subset of the second relationship's first set of components;

removing the first relationship's second set of components that exist from the second relationship's second set of components when said proper subset exists;

removing said second relationship when said second relationship's second set of components is an empty set.

49. (Not amended) The method of claim 47 wherein said step of validating further comprises the steps of:

determining whether a component selected by said configuration input is selectable; and

invalidating said configuration input when said component is not selectable.

50. (Not amended) The method of claim 49 wherein said one of said internal relationships is an includes relationship, said includes relationship causing a second set of components to be included in said system when all members of a first set of components exist in said system, said second step of determining further comprises the step of:

determining whether any member of the second set of components cannot be included in said system.

51. (Not amended) The method of claim 49 wherein said one of said internal relationships is an excludes relationship, said excludes relationship causing a second set of components to be excluded from said system when all members of a first set of components exist in said system, said second step of determining further comprises the step of:

determining whether any member of the second set of components already exists in said system.

52. (Not amended) The method of claim 49 wherein said one of said internal relationships is an removes relationship, said removes relationship causing a second set of components to be removed from said system when all members of a first set of components exist in said system, said second step of determining further comprises the step of:

determining whether any member of the second set of components has been selected for inclusion in said system by a configuration user.

53. (Not amended) The method of claim 49 wherein said one of said internal relationships is a requires choice relationship, said requires choice relationship requiring a selection of a number of members of a second set of components when all members of a first set of components exist in said system, said second step of determining further comprises the step of:

determining whether said number of members of said second set of components cannot be selected for addition to said system.

54. (Not amended) The method of claim 47 wherein said step of modifying further comprises the steps of:

determining whether the performance of said operation of one of said internal relationships would result in an invalid configuration;

causing said one of said internal relationships to become inactivateable.

55. (Not amended) The method of claim 54 wherein said step of causing further comprises the step of excluding the last remaining member of said first set of components such that said last remaining member cannot be added to said system, said last remaining member being the only member of said first set of components that has not already been selected.

56. (Not amended) The method of claim 39 wherein said step of translating further comprises the step of:

mapping said intuitive relationships of said external representation into a set of relationships in said internal representation.

57. (Not amended) The method of claim 39 wherein said internal representation is stored as a set of bit vectors wherein each bit vector includes a bit that corresponds to one of said components of said system definition, the impact of changing the number of said components of said system definition thereby being minimal.

58. (Not amended) The method of claim 57 wherein said step of validating comprising a plurality of bit-level operations.

59. (Not amended) The method of claim 39 wherein said step of modifying further comprises the step of updating said configuration state to represent which of said components of said system definition are selectable, selected, included, and excluded.

Please add the following new Claims 60-85:

--60. A method of configuring a system comprising the steps of:

generating a definition for said system, said definition containing one or more elements and being conveyed graphically using a set of product relationships;

generating a set of part relationships between said one or more elements, said set of part relationships being conveyed graphically using a set of part relationships; and

receiving input from a configuration user;

validating said input based on said definition, said set of product relationships, said set of part relationships, and a current configuration state; and

identifying a set of valid configuration options using said definition, said set of product relationships, said set of part relationships and said current configuration state.--

--61. The method of claim 60 wherein said step of identifying further comprises the steps of:

determining whether a relationship in said set of relationships is active;

modifying said configuration to satisfy said active relationship when it is determined that a modification is needed;

making said relationship notActivatable when it is determined that said relationship is not active and the activation of said relationship would render the configuration invalid.--

--62. The method of claim 61 wherein said step of determining whether a relationship is active further comprises the step of determining whether the elements specified in the left-hand side of said relationship are present in said configuration.--

--63. The method of claim 62 wherein said step of determining whether the elements specified in the left-hand side of said relationship are present in said configuration further comprises the step of performing a bit-level comparison of the bits corresponding to the elements in said left-hand side of the relationship with a "selected" bit vector.--

--64. The method of claim 61 wherein said relationship is an includes relationship, said step of modifying further comprises the step of including in

said configuration elements identified in a right-hand side of said relationship.--

--65. The method of claim 61 wherein said relationship is an excludes relationship, said step of modifying further comprises the step of marking said elements identified in the right-hand side of said relationship as excluded and notSelectable.--

--66. The method of claim 61 wherein said relationship is a removes relationship, said step of modifying further comprises the step of removing said elements identified in the right-hand side of said relationship from said configuration.--

43
cont.
--67. The method of claim 61 wherein said relationship is a requires choice relationship, said step of modifying further comprises the steps of:

determining whether a plurality of elements identified in a right-hand side of said relationship are present in said configuration such that said relationship is satisfied;

modifying said configuration to satisfy said requires choice relationship when it is determined that only one path exists to satisfy said relationship and said relationship is not already satisfied.--

--68. The method of claim 61 wherein said step of making said relationship notActivateable further comprises the step of marking one or more elements in said left-hand side of said relationship as notSelectable.--

--69. The method of claim 60 wherein said definition, each of said set of relationships, and said current configuration state are expressed as bit vectors having bits that correspond to elements.--

--70. The method of claim 60 wherein said step of validating further comprises the step of determining whether the relationships other than those marked notActivateable in said set of relationships can be satisfied.--

--71. The method of claim 60 wherein said step of generating a definition selects said one or more elements for inclusion in said definition, said input is made without regard for the order in which said one or more elements are selected for said definition.--

63
cont.
--72. The method of claim 60 wherein said configuration state identifies a set of selected elements and a set of selectable elements for said system, said input capable of selecting any member of said set of selectable elements and being made despite the order in which members of said set of selected elements were selected.--

--73. An article of manufacture comprising:

a computer usable medium having computer readable program code embodied therein for configuring a system comprising:

computer readable program code configured to cause a computer to generate a definition for said system, said definition containing one or more elements and being conveyed graphically using a set of product relationships;

computer readable program code configured to cause a computer to generate a set of part relationships between said one or more elements, said set of part relationships being conveyed graphically using a set of part relationships; and

computer readable program code configured to cause a computer to receive input from a configuration user;

computer readable program code configured to cause a computer to validate said input based on said definition, said set of product relationships, said set of part relationships, and a current configuration state; and

computer readable program code configured to cause a computer to identify a set of valid configuration options using said definition, said set of product relationships, said set of part relationships and said current configuration state.--

--74. The article of manufacture of claim 73 wherein said computer readable program code configured to cause a computer to identify further comprises:

computer readable program code configured to cause a computer to determine whether a relationship in said set of relationships is active;

computer readable program code configured to cause a computer to modify said configuration to satisfy said active relationship when it is determined that a modification is needed;

computer readable program code configured to cause a computer to make said relationship notActivatable when it is determined that said relationship is not active and the activation of said relationship would render the configuration invalid.--

--75. The article of manufacture of claim 74 wherein said computer readable program code configured to cause a computer to determine whether a relationship is active further comprises computer readable program code configured to cause a computer to determine whether the elements specified in the left-hand side of said relationship are present in said configuration.--

Q3
cont.
--76. The article of manufacture of claim 75 wherein said computer readable program code configured to cause a computer to determine whether the elements specified in the left-hand side of said relationship are present in said configuration further comprises computer readable program code configured to cause a computer to perform a bit-level comparison of the bits corresponding to the elements in said left-hand side of the relationship with a "selected" bit-vector.--

--77. The article of manufacture of claim 74 wherein said relationship is an includes relationship, said computer readable program code configured to cause a computer to modify further comprises computer readable program

code configured to cause a computer to include in said configuration elements identified in a right-hand side of said relationship.--

--78. The article of manufacture of claim 74 wherein said relationship is an excludes relationship, said computer readable program code configured to cause a computer to modify further comprises computer readable program code configured to cause a computer to mark said elements identified in the right-hand side of said relationship as excluded and notSelectable.--

--79. The article of manufacture of claim 74 wherein said relationship is a removes relationship, said computer readable program code configured to cause a computer to modify further comprises computer readable program code configured to cause a computer to remove said elements identified in the right-hand side of said relationship from said configuration.--

Q3
cont.
--80. The article of manufacture of claim 74 wherein said relationship is a requires choice relationship, said computer readable program code configured to cause a computer to modify further comprises:

computer readable program code configured to cause a computer to determine whether a plurality of elements identified in a right-hand side of said relationship are present in said configuration such that said relationship is satisfied;

computer readable program code configured to cause a computer to modify said configuration to satisfy said requires choice relationship when it

is determined that only one path exists to satisfy said relationship and said relationship is not already satisfied.--

--81. The article of manufacture of claim 74 wherein said computer readable program code configured to cause a computer to make said relationship notActivateable further comprises computer readable program code configured to cause a computer to mark one or more elements in said left-hand side of said relationship as notSelectable.--

--82. The article of manufacture of claim 73 wherein said definition, each of said set of relationships, and said current configuration state are expressed as bit vectors having bits that correspond to elements.--

63
cont.
--83. The article of manufacture of claim 73 wherein said code configured to cause a computer to validate further comprises program code configured to cause a computer to determine whether the relationships other than those marked notActivateable in said set of relationships can be satisfied.--

--84. The article of manufacture of claim 73 wherein said code configured to cause a computer to generate a definition selects said one or more elements for inclusion in said definition, said input is made without regard for the order in which said one ore more elements are selected for said definition.--

a3
could.
--85. The article of manufacture of claim 73 wherein said configuration state identifies a set of selected elements and a set of selectable elements for said system, said input capable of selecting any member of said set of selectable elements and being made despite the order in which members of said set of selected elements were selected.--

REMARKS

Claims 1-59 are pending in the present application. The Examiner has allowed claims 16-19 and 35-59, objected to claims 3-14 and 22-33 and rejected claims 1, 2, 15, 20, 21 and 35. Applicant has amended Claims 1 and 20. Applicant has added new Claims 60-85. Reexamination and reconsideration of pending Claims 1-85 is respectfully requested.

Rejection of Claims 1, 2, 15, 20, 21, and 34 based on 35 U.S.C § 102(a), (b) and (e)

The Examiner rejects Claims 1, 2, 15, 20, 21 and 34 based on 35 U.S.C § 102(a), (b) and (e) as being anticipated by Addesso et al. (U. S. Patent No. 5,019,961) (Addesso).

Regarding Claim 1, the Examiner states that Addesso discloses generating a definition (citing Addesso's models at col. 3, line 50), product relationships (citing Addesso's primitives, col. 3, line 7), generating a set of relationships (citing Addesso's relationships for associating entities, col. 3, line 8), conveying graphically (citing Addesso's graphical interface, col. 3, line 6) and configuring a system (citing Addesso's creating modeling objects, col. 4, line 1). The Examiner states that Addesso discloses the features of Claim 20 and refers to the discussion of Claim 1.

Applicant respectfully disagrees. Applicant believes that Claims 1, 2, 15, 20, 21 and 34 are not anticipated by Addesso. Applicant believes that Addesso

does not teach, suggest or describe the claimed invention for at least the following reasons:

1. *Addesso's primitives do not teach, suggest or describe the claimed product relationships;*
2. *Addesso's models do not teach, suggest or describe generating a definition;*
3. *Addesso cannot and does not teach, suggest or describe configuring a system; and*
4. *For at least the foregoing reasons, Addesso does not teach, suggest or describe the claimed invention.*

Applicant provides the following discussion of these reasons:

1. Addesso's primitives do not teach, suggest or describe the claimed product relationships.

The Examiner states that the primitives described in Addesso disclose the claimed product relationships. Applicant respectfully disagrees. Addesso's primitives do not teach, suggest or disclose the claimed product relationships.

Addesso states (at col. 3, lines 7-9) that primitives are "entities, relationships for associating entities and attributes for setting conditions on entities and relationships." None of the primitives described in Addesso teach, suggest or describe the claimed product relationships.

Specifically, the relationships defined in Adesso's primitives are not the same as the claimed product relationships. According to Adesso (at col. 7, lines 34-35), a "relationship is an association among entities and perhaps other relationships." Adesso describes a send relationship (referring to Fig. 4) in which one entity is a sender, another entity is the receiver and another is the data that is being sent (see col. 7, lines 1-15). According to Adesso (at col. 7, lines 44-61 and Fig. 4):

"[e]ntities play roles in relationships. Thus, process 72 at the beginning of sends relationship 70 is playing the role of the "sender" of data to the process 76. At the same time, process 76 plays the role of the "receiver" of the data. The data 74 is playing the role of "what" is being sent. The meta model includes information that describes a valid combination of role players in an instance of a relationship. Such combinations are referred to as "player sets", and there can be one or more player sets for a relationship in the meta model. Each player set consists of role-player combinations, where the role specified must be played by the corresponding entity or relationship types when the relationship is created or modified. Consistent with the proper logical operation of the present system, each role in a relationship in the meta model must be accounted for. Therefore, the database will not allow a relationship to exist which does not fit into one of the player sets."

Thus, in Adesso, a relationship defines the interaction between players in a player set and the roles that a player has in a player set. In the send relationship, the players in the send player set play the role of a sender, a receiver or data. The players interact such that a sender sends data to a

receiver. The relationship primitive described in Addesso does not identify classifications for elements in a definition as in the claimed invention.

In contrast to Addesso, the claimed product relationships identify an includes classification for elements in a definition. Addesso does not teach, suggest or describe the claimed product relationships.

2. Addesso's models do not teach, suggest or describe generating a definition.

The Examiner states that Addesso's models disclose the step of generating a definition. Applicant respectfully disagrees.

Addesso's models are created from Addesso's primitives. As discussed above, Addesso's primitives do not teach, suggest or describe the claimed definition. Further, as stated in Addesso (at col. 3, lines 49-68):

[a] method is provided for creating modelling methodologies to enable a modeler to build evaluable models for intended applications. In accordance with the method, a set of primitives is provided including entities, relationships for associating entities, and attributes for setting conditions on entities and relationships. The primitives are stored in a model database, and modelling objects are created from the stored entities, relationships and attributes. These modelling objects are stored in the database, and views thereof are created. Logical relationships between elements of the graphical views and modelling objects are defined and stored in the database. Through graphical views and the logical relationships, a user is provided with access to the modelling objects to enable models to be created for an intended application. Model

creation is accomplished by manipulating views to edit entities and relationships in the modelling database in accordance with a methodology defined by the types of modelling objects provided, and the relationships and attributes applicable thereto.

Thus, according to Addesso, a model is created by editing a view. A view is created from modeling objects that are created from Addesso's primitives (i.e., entity, relationship and attribute primitives). Thus, the user creates a model by editing the entities and relationships contained in a view.

Therefore, according to Addesso, a user modifies modeling objects that are created from the entity, relationship and attribute primitives described in Addesso. As previously discussed, the primitives described in Addesso do not teach, suggest or describe the claimed product relationships. Since Addesso's primitives do not teach, suggest or describe the claimed product relationships, Addesso's method of editing the primitives cannot generate the claimed definition that contain one or more elements conveyed graphically using the claimed product relationships. In Addesso, the model is generated by editing views of modeling objects that are created from Addesso's primitives. Therefore, Addesso cannot teach, suggest or describe the generation of a definition.

As stated above, the claimed step generates a definition that includes elements that are conveyed graphically using product relationships that identify classifications for elements and comprise an includes classification. Addesso does not teach, suggest or describe these product relationships.

Therefore, Addesso does not teach, suggest or describe the claimed step of generating a definition.

3. Addesso cannot and does not teach, suggest or describe configuring a system.

The Examiner states that Addesso's creation of modeling objects discloses the step of configuring. Applicant respectfully disagrees. Since Addesso does not teach, suggest or describe the claimed product relationships, Addesso's creation of modeling objects cannot disclose configuring a system as claimed.

According to Addesso (at col. 2, lines 67-68), modeling objects are defined in terms of primitives (i.e., entities, relationships and attributes). As discussed above, Addesso's primitives do not teach, suggest, or describe the claimed production relationships. According to Addesso, relationship primitives define the interaction between players in a player set and the roles that a player has in a player set. Addesso's relationship primitives do not identify an includes classification. Therefore, Addesso's relationship primitives do not teach, suggest or describe the claimed product relationships. Further, Addesso cannot teach, suggest or describe a step of configuring a system using the claimed product relationships.

Addesso's modeling objects are defined by primitives that do not teach, suggest or describe the claimed product relationships. Therefore, the creation of modeling objects as described in Addesso does not teach, suggest or describe

the step of configuring. Further, as discussed above, Addesso does not teach, suggest or describe the claimed product relationships. Therefore, Addesso cannot teach, suggest or describe the step of configuring a system using product relationships.

4. For at least the foregoing reasons, Addesso does not teach, suggest or describe the claimed invention.

For at least the foregoing reasons, Applicant believes that Claims 1 and 20 are allowable over Addesso. Further, Applicant believes that Claims 2-15 and 21-34 are allowable for the same reasons.

Objection to Claims 3-14 and 22-33

The Examiner objects to claims 3-14 and 22-33 as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Applicant has rewritten Claims 2 and 21 in independent form including all of the limitations of the base claim and any intervening claims (i.e., new Claims 60 and 73). Claims 3-14 and 22-33 are rewritten as Claims 61-72 and 74-85 dependent on Claims 60 and 73, respectively.

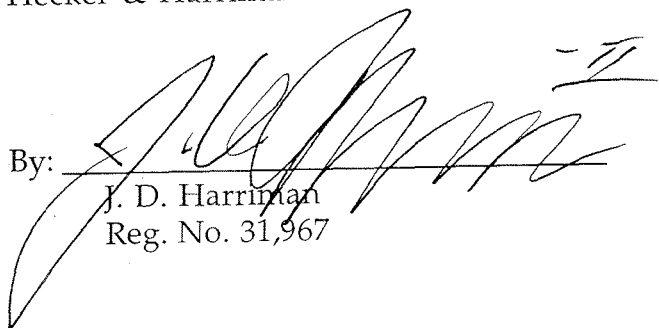
Conclusion

For the foregoing reasons, Applicant contends that none of the references cited, either alone or in combination, teach, describe, or suggest the

present invention. Applicant contends that pending Claims 1-85 are allowable.

Respectfully submitted,
Hecker & Harriman

Date: 5/6/98

By: 
J. D. Harriman
Reg. No. 31,967

2029 Century Park East
Suite 1600
Los Angeles, CA 90067
(310) 286-0377

File No.: 85160.915

CERTIFICATE OF MAILING

This is to certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as First Class Mail in an Envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on:

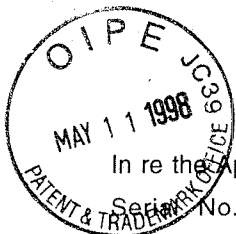
May 6, 1998


Signature

Date

Lillian E. Rodriguez

5-6-98



GAU 2787

Case/Docket No. 85160.915

4

#6

In re the Application of: GUPTA, N. et al.

Serial No.: 08/707,187

Filed: September 3, 1996

For: METHOD AND APPARATUS FOR MAINTAINING AND CONFIGURING SYSTEMS

Assistant Commissioner for Patents
Washington, D.C. 20231

Sirs:

Transmitted herewith is an Amendment and Response in the above-identified application.

Small entity status of this Application under 37 CFR 1.9 and 1.27 has been established by a verified statement previously submitted.

A verified statement to establish small entity status under 37 CFR 1.9 and 1.27 is enclosed.

No additional fee is required.

The fee has been calculated as shown below:

	(Col. 1)		(Col. 2)	(Col. 3)	SMALL ENTITY		OTHER THAN A SMALL ENTITY	
	Claims Remaining After Amd		Highest No. Previously Paid For	Present Extra	Rate	Add'l Fee	Rate	Add'l Fee
Total Claims	* 85	Minus	** 59	26	X 11	\$0	X 22	\$572
Indep. Claims	* 7	Minus	*** 5	2	X 41	\$0	X 82	\$164
<input type="checkbox"/> First Presentation of Multiple Dependent Claims					+ 135	\$0	+ 270	\$0
					Total Add'l Fee	\$0	Total Add'l Fee	\$736

- * If the entry in Col. 1 is less than the entry in Col. 2, write "0" in Col. 3.
- ** If the "Highest No. Previously Paid For" IN THIS SPACE is less than 20, write "20" in this space.
- *** If the "Highest No. Previously Paid For" IN THIS SPACE is less than 3, write "3" in this space The "Highest No. Previously Paid For" (Total or Independent) is the highest number found from the equivalent box in Col. 1 of a prior amendment or the number of claims originally filed.

XXX The Commissioner is hereby authorized to charge payment of any additional fees associated with this communication, or credit any overpayment, to Deposit Account Number 08-1520. A duplicate of this authorization is enclosed.

XXX Applicant hereby petitions for an extension of time of ONE month, pursuant to Rule 1.136.

XXX A check in the amount of \$110.00 is attached for processing fees under 37 CFR 1.17.

XXX A check in the amount of \$736.00 is attached for presentation of additional claims.

Respectfully submitted,

HECKER & HARRIMAN

J.D. Harriman II
Reg. No. 31,967

CERTIFICATE OF MAILING

This is to certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on: May 6, 1998.

Signature: Lillian E. Rodriguez

Date

RECEIVED
98 MAY 14 PM 2:47
GROUP 2700

05/13/1998 RABARCA 00000050 08707187

01 FC:103 572.00 OP
02 FC:102 164.00 OP
03 FC:115 110.00 OP

Date: May 6, 1998

2029 Century Park East
Suite 1600
Los Angeles, California 90067
(310) 286-0377



**UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office**

Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NO.
--------------------	-------------	-----------------------	---------------------

08/707,187 09/03/96 GUPTA N 85160.915

LM21/0522

HECKER & HARRIMAN
SUITE 1600
2029 CENTURY PARK EAST
LOS ANGELES CA 90067

EXAMINER

KRIESS, K
ART UNIT PAPER NUMBER

2787

7

DATE MAILED: 05/22/98

This is a communication from the examiner in charge of your application.
COMMISSIONER OF PATENTS AND TRADEMARKS

NOTICE OF ALLOWABILITY

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance and Issue Fee Due or other appropriate communication will be mailed in due course.

☒ This communication is responsive to Amendment filed 5-11-98

☒ The allowed claim(s) is/are 1-85

☐ The drawings filed on _____ are acceptable.

☐ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).

☐ All ☐ Some* ☐ None of the CERTIFIED copies of the priority documents have been

☐ received.

☐ received in Application No. (Series Code/Serial Number) _____

☐ received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

*Certified copies not received: _____

☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

A SHORTENED STATUTORY PERIOD FOR RESPONSE to comply with the requirements noted below is set to EXPIRE THREE MONTHS FROM THE "DATE MAILED" of this Office action. Failure to timely comply will result in ABANDONMENT of this application. Extensions of time may be obtained under the provisions of 37 CFR 1.136(a).

☐ Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL APPLICATION, PTO-152, which discloses that the oath or declaration is deficient. A SUBSTITUTE OATH OR DECLARATION IS REQUIRED.

☒ Applicant MUST submit NEW FORMAL DRAWINGS

☐ because the originally filed drawings were declared by applicant to be informal.

☒ including changes required by the Notice of Draftperson's Patent Drawing Review, PTO-948, attached hereto or to Paper No. 4.

☐ including changes required by the proposed drawing correction filed on _____, which has been approved by the examiner.

☐ including changes required by the attached Examiner's Amendment/Comment.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the reverse side of the drawings. The drawings should be filed as a separate paper with a transmittal letter addressed to the Official Draftperson.

☐ Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Any response to this letter should include, in the upper right hand corner, the APPLICATION NUMBER (SERIES CODE/SERIAL NUMBER). If applicant has received a Notice of Allowance and Issue Fee Due, the ISSUE BATCH NUMBER and DATE of the NOTICE OF ALLOWANCE should also be included.

Attachment(s)

☐ Notice of References Cited, PTO-892

☒ Information Disclosure Statement(s), PTO-1449, Paper No(s). _____

☐ Notice of Draftperson's Patent Drawing Review, PTO-948

☐ Notice of Informal Patent Application, PTO-152

☐ Interview Summary, PTO-413

☐ Examiner's Amendment/Comment

☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material

☐ Examiner's Statement of Reasons for Allowance



UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office

NOTICE OF ALLOWANCE AND ISSUE FEE DUE

LN21/0522

HECKER & HARRIMAN
SUITE 1600
2029 CENTURY PARK EAST
LOS ANGELES CA 90067

APPLICATION NO.	FILING DATE	TOTAL CLAIMS	EXAMINER AND GROUP ART UNIT	DATE MAILED
08/707,107	09/03/96	035	KRIESS, K 2787	05/22/98
First Named Applicant	GUPTA, NEERAJ			

TITLE OF INVENTION METHOD AND APPARATUS FOR MAINTAINING AND CONFIGURING SYSTEMS

ATTY'S DOCKET NO.	CLASS-SUBCLASS	BATCH NO.	APPLN. TYPE	SMALL ENTITY	FEE DUE	DATE DUE
85160.915	364-468.090	E36	UTILITY	NO	\$1320.00	06/24/98

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED.

THE ISSUE FEE MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED.

HOW TO RESPOND TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

- A. If the status is changed, pay twice the amount of the FEE DUE shown above and notify the Patent and Trademark Office of the change in status, or
- B. If the status is the same, pay the FEE DUE shown above.

If the SMALL ENTITY is shown as NO:

- A. Pay FEE DUE shown above, or
- B. File verified statement of Small Entity Status before, or with, payment of 1/2 the FEE DUE shown above.

II. Part B-Issue Fee Transmittal should be completed and returned to the Patent and Trademark Office (PTO) with your ISSUE FEE. Even if the ISSUE FEE has already been paid by charge to deposit account, Part B Issue Fee Transmittal should be completed and returned. If you are charging the ISSUE FEE to your deposit account, section "4b" of Part B-Issue Fee Transmittal should be completed and an extra copy of the form should be submitted.

III. All communications regarding this application must give application number and batch number. Please direct all communications prior to issuance to Box ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PART B—ISSUE FEE TRANSMITTAL

Complete and mail this form, together with applicable fees, to:

Box ISSUE FEE
Assistant Commissioner for Patents
Washington, D.C. 20231

1421820
RECEIVED
501-3
MAY 27 1998

MAILING INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE. Blocks 1 through 4 should be completed where appropriate. All further correspondence including the Issue Fee Receipt, the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Legibly mark-up with any corrections or use Block 1)

HECKER & HARRIMAN
SUITE 1600
2029 CENTURY PARK EAST
LOS ANGELES CA 90067

LM21/0522

JUL 10 1998

Note: The certificate of mailing below can only be used for domestic mailings of the Issue Fee Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing.

Certificate of Mailing

I hereby certify that this Issue Fee Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Box Issue Fee address above on the date indicated below, w/label no. EL111266466US

Mario Federis

(Depositor's name)

(Signature)

(Date)

03/16/98

APPLICATION NO.	FILING DATE	TOTAL CLAIMS	EXAMINER AND GROUP ART UNIT	DATE MAILED
08/707,187	09/03/96	085	KRIESS, K 2787	05/22/98
First Named Applicant	GUPTA, NEERAJ			

TITLE OF INVENTION: METHOD AND APPARATUS FOR MAINTAINING AND CONFIGURING SYSTEMS

ATTY'S DOCKET NO.	CLASS-SUBCLASS	BATCH NO.	APPLN. TYPE	SMALL ENTITY	FEE DUE	DATE DUE
2 85160.915	364-468.090	E36	UTILITY	NO	\$1320.00	08/24/98

1. Change of correspondence address or indication of " Fee Address" (37 CFR 1.363). Use of PTO form(s) and Customer Number are recommended, but not required.

☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47) attached.

2. For printing on the patent front page, list (1) the names of up to 3 registered patent attorneys or agents OR, alternatively, (2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1. HECKER & HARRIMAN

2. _____

3. _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)
PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. Inclusion of assignee data is only appropriate when an assignment has been previously submitted to the PTO or is being submitted under separate cover. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

TRILOGY DEVELOPMENT GROUP, INC.

(B) RESIDENCE: (CITY & STATE OR COUNTRY)

Austin, Texas

Please check the appropriate assignee category indicated below (will not be printed on the patent)

☐ individual ☒ corporation or other private group entity ☐ government

4a. The following fees are enclosed (make check payable to Commissioner of Patents and Trademarks):

☒ Issue Fee

☒ Advance Order - # of Copies 10

4b. The following fees or deficiency in these fees should be charged to:

DEPOSIT ACCOUNT NUMBER 08-1520

(ENCLOSE AN EXTRA COPY OF THIS FORM)

☒ Issue Fee

☒ Advance Order - # of Copies 10

The COMMISSIONER OF PATENTS AND TRADEMARKS IS requested to apply the Issue Fee to the application identified above.

(Authorized Signature)

J. D. Harriman

(Date)

7/10/98

NOTE: The Issue Fee will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the Patent and Trademark Office.

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending on the needs of the individual case. Any comments on the amount of time required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, D.C. 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND FEES AND THIS FORM TO: Box Issue Fee, Assistant Commissioner for Patents, Washington D.C. 20231

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

07/15/1998 SHARKLIL 00000106 08707187

01 F01142
02 F01361

1020.00 DP
30.00 DP

Atty. Docket No. 85160.915

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of)	
)	
GUPTA, N. et al.)	
)	Examiner: K. KRIESS
Serial No: 08/707,187)	
)	Group Art Unit: 2787
Filed: September 3, 1996)	
)	Batch No.: E36
For: METHOD AND APPARATUS)	
FOR MAINTAINING)	
AND CONFIGURING)	
SYSTEMS)	
_____)	

PAYMENT OF ISSUE FEE

Assistant Commissioner for Patents
BOX ISSUE FEE
Washington, D. C. 20231

Sir:

This is in response to the Notice of Allowance dated May 22, 1998. Applicants have enclosed the following:

1. A check in the amount of \$1,350.00 as Issue Fee due (\$1,320.00), and advance order fee (\$30.00);
2. Part B of the Transmittal Letter with a Certificate of Mailing by EXPRESS MAIL;
3. A return receipt postcard.

Atty. Docket No. 85160.915

The Commissioner is hereby authorized to charge payment of any additional fees associated with this communication or credit any overpayment to Hecker & Harriman Deposit Account No. 08-1520. A duplicate of this authorization is enclosed for this purpose.

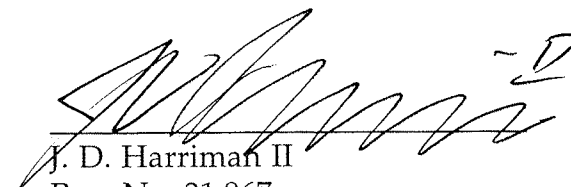
Respectfully submitted,

HECKER & HARRIMAN

Date:

7/10/98

By:

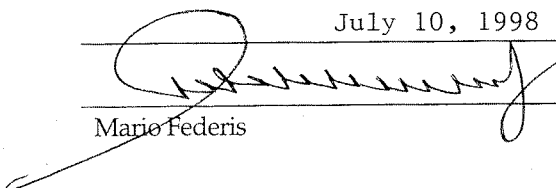

J. D. Harriman II
Reg. No. 31,967

1925 Century Park East, Suite 2300
Los Angeles, CA 90067
(310) 286-0377

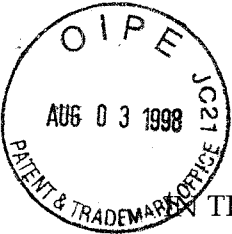
CERTIFICATE OF MAILING

This is to certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as EXPRESS MAIL with Label No. EL11266466US in an envelope addressed to: Assistant Commissioner for Patents, BOX ISSUE FEE, Washington, D. C., 20231 on :

July 10, 1998


Mario Federis

07/10/98
Date



27112

#8 B/

Atty. Docket No. 85160.915

THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of
GUPTA, N. et al.
Serial No: 08/707,187
Filed: September 3, 1996
For: METHOD AND APPARATUS
FOR MAINTAINING
AND CONFIGURING SYSTEMS

Examiner: K. KRIESS
Group Art Unit: 2787
Batch No.: E36

SUBMITTAL OF FORMAL DRAWINGS

Assistant Commissioner for Patents
Washington, D. C. 20231

Attn.: Official Draftsperson

Sir:

This is in response to the Notice of Allowance dated May 22, 1998. Applicants have enclosed the following:

1. Twenty Four (24) sheets of formal drawings;
2. A Certificate of Mailing by First Class Mail; and
3. A return receipt postcard.

The Commissioner is hereby authorized to charge any fees associated with this communication to Hecker & Harriman Deposit Account No. 08-1520. A duplicate copy of this transmittal letter is enclosed for this purpose.

Respectfully submitted,

HECKER & HARRIMAN

Date: 7/28/98

By: J. D. Harriman II

Reg. No. 31,967

1925 Century Park East
Suite 2300
Los Angeles, California 90067

CERTIFICATE OF MAILING

This is to certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as First Class Mail in an Envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on:

July 28, 1998

Mario Federis

Signature

Date

FORD 1013

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

5825651

1/24

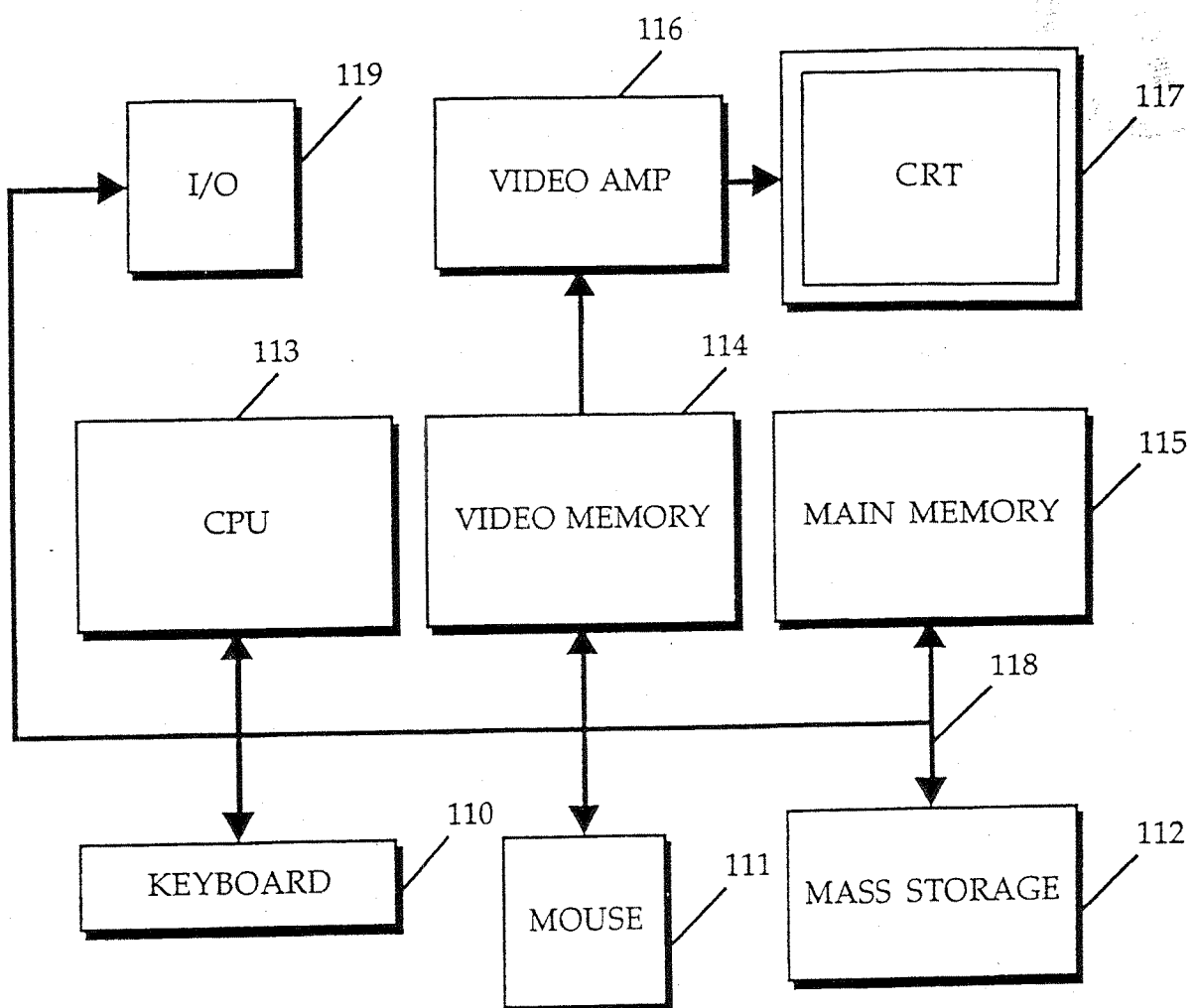


FIG. 1

APPROVED	O.G.	G.
BY	CLASS	SUBCLASS
DRAFTSMAN		

2/24

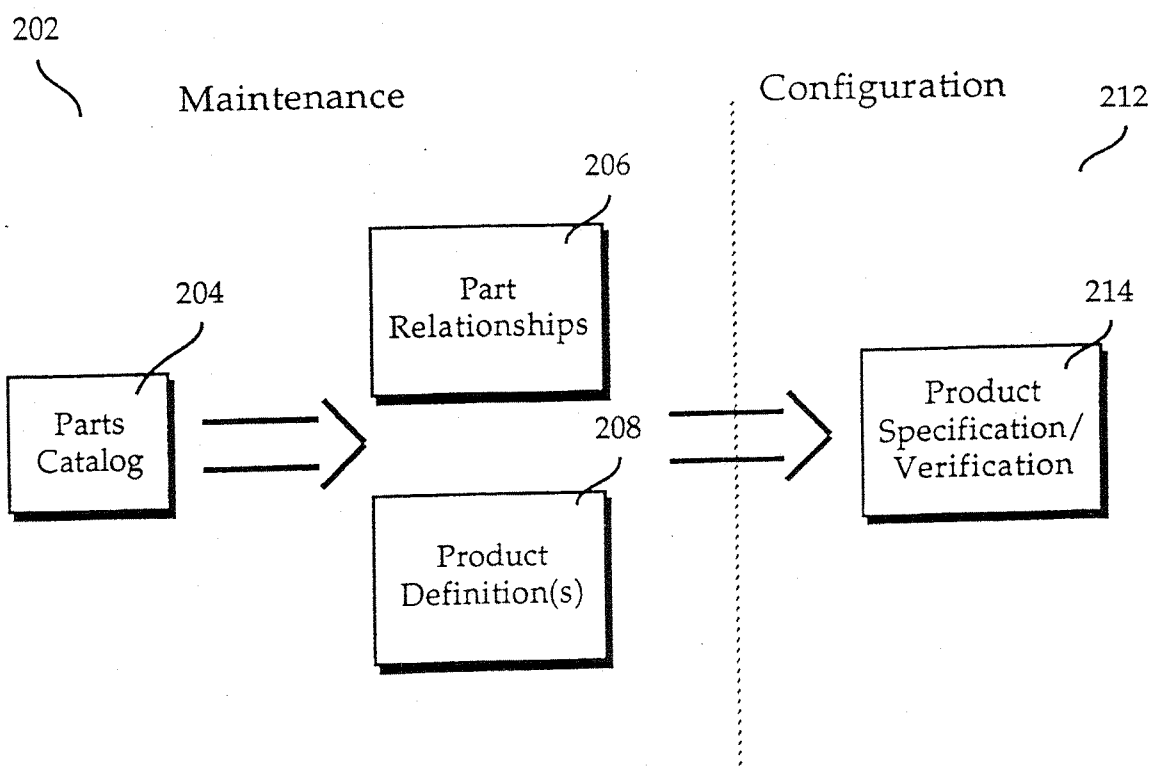


FIG. 2

APPROVED	O.C. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

3/24

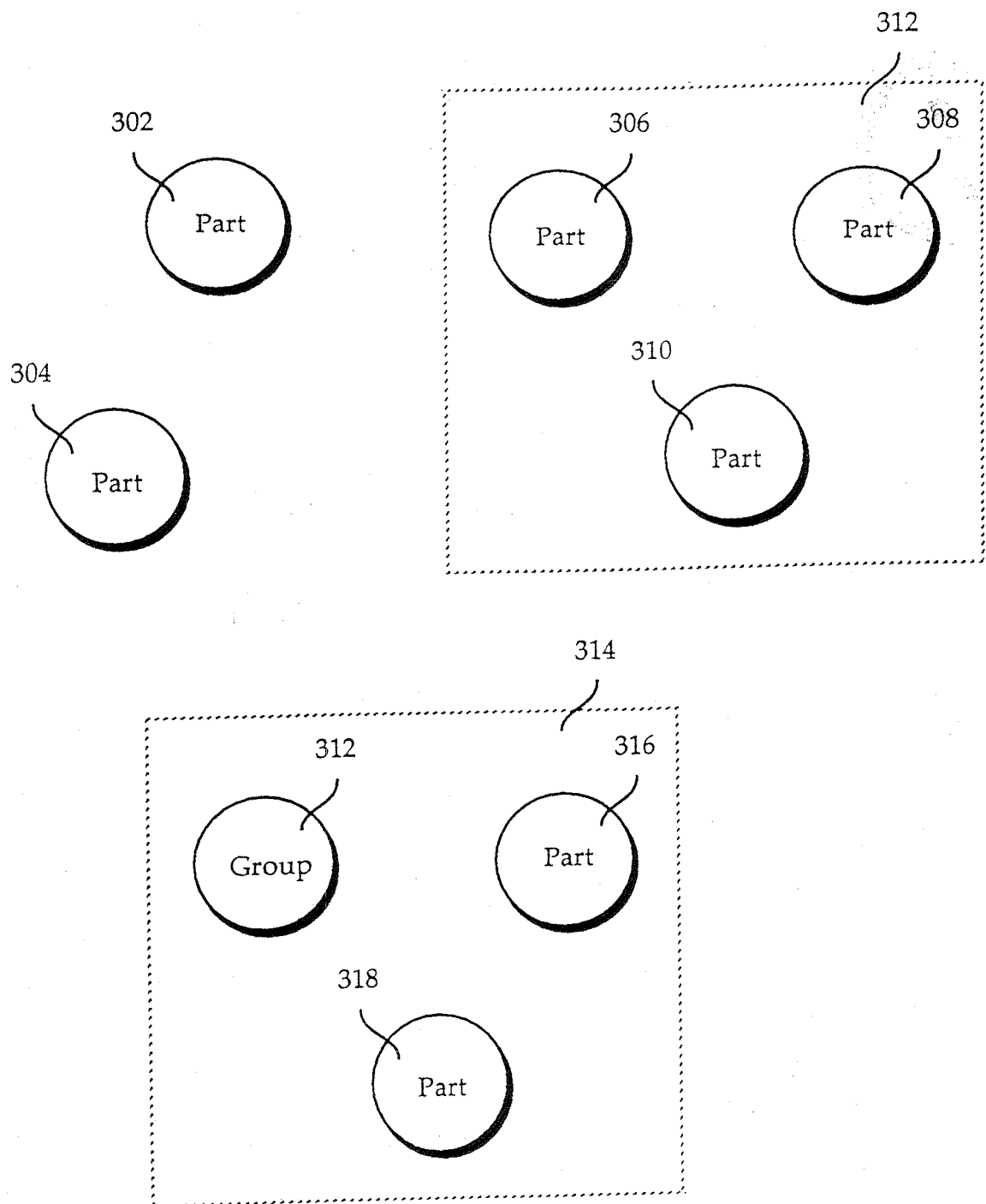


FIG. 3

APPROVED	O.G. -IG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

4/24

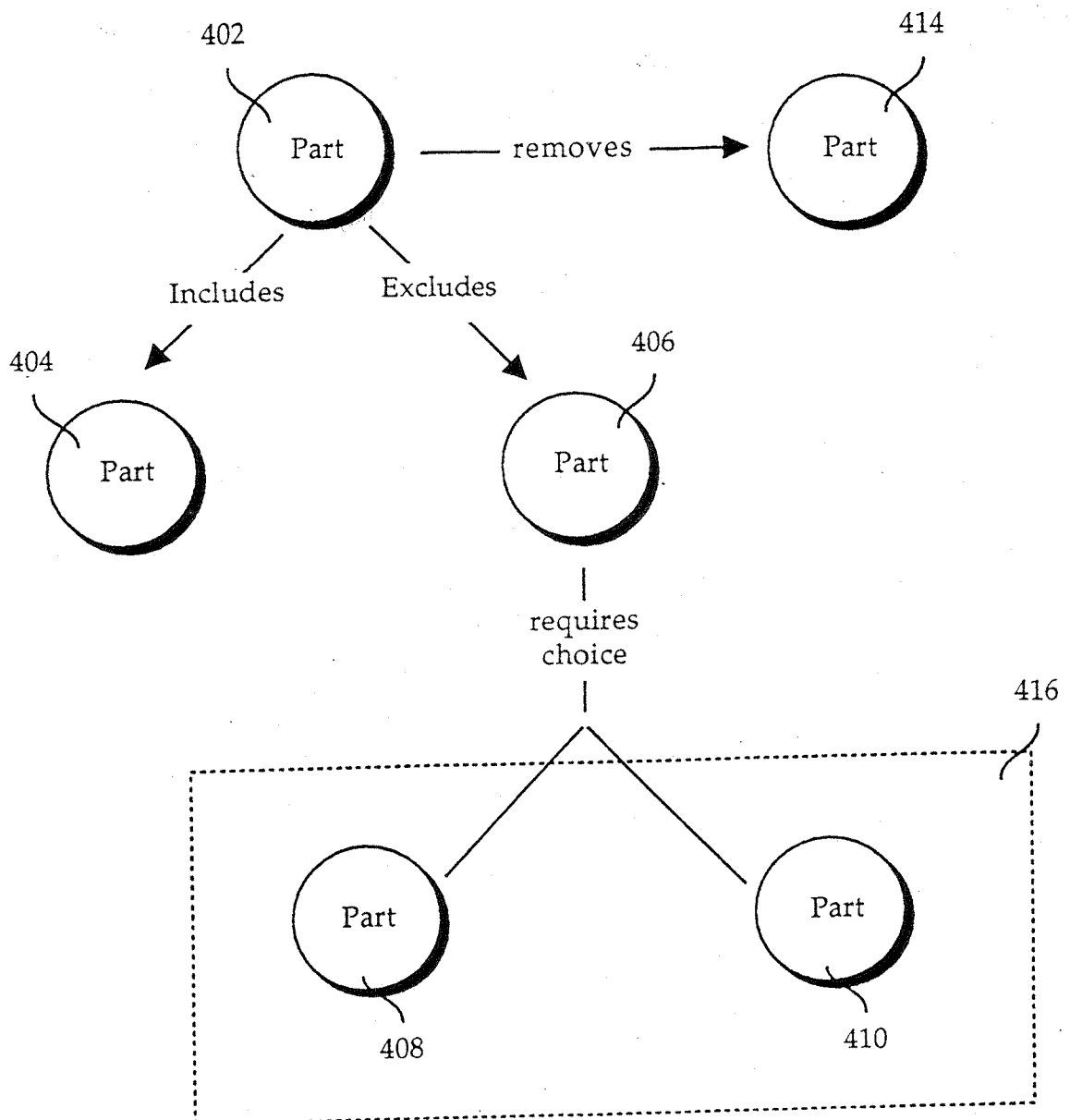


FIG. 4

APPROVED	FIG.
BY	CLASS SUBCLASS
WESTMAN	

5/24

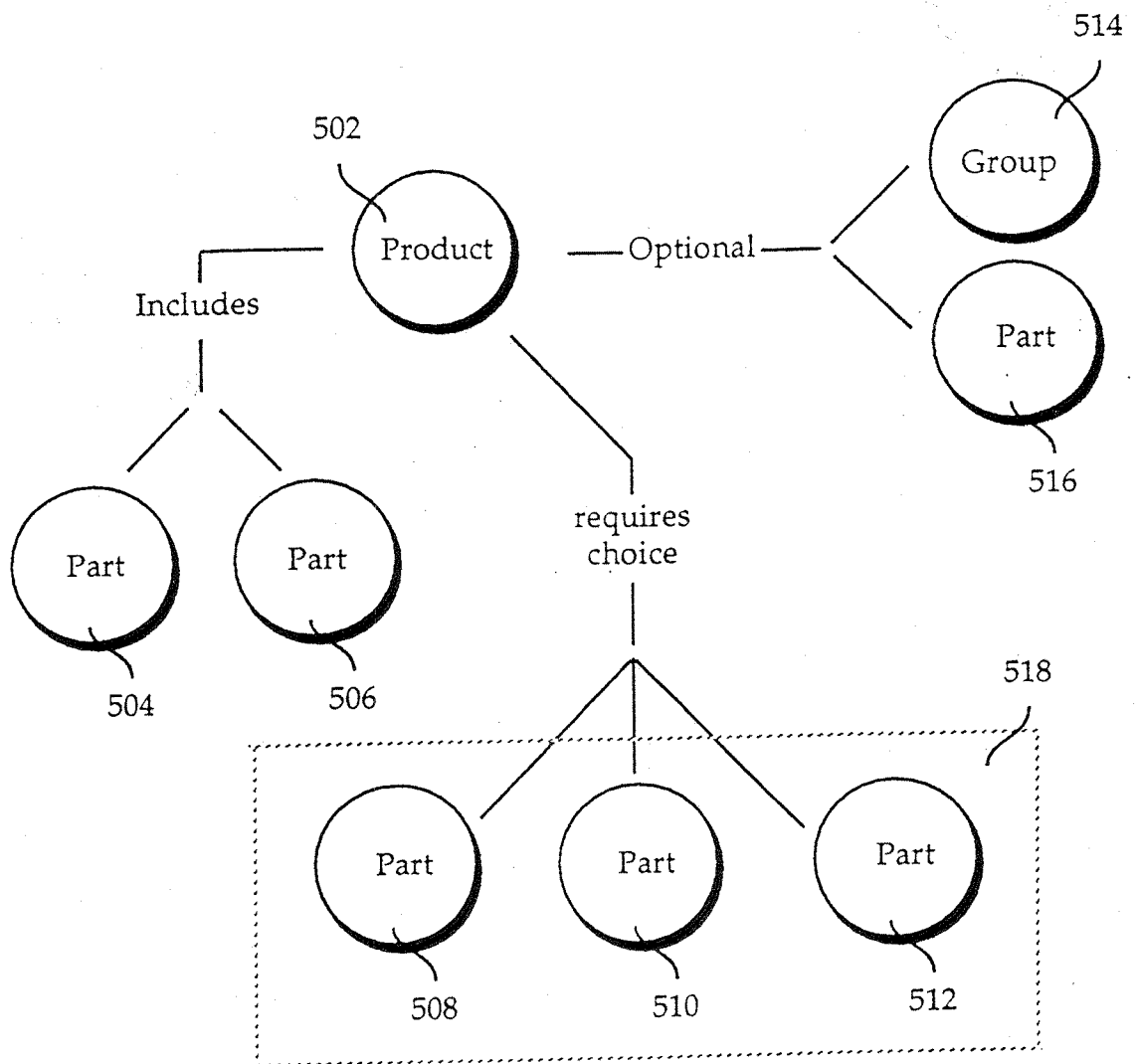


FIG. 5

APPROVED	D. L. G.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

6/24

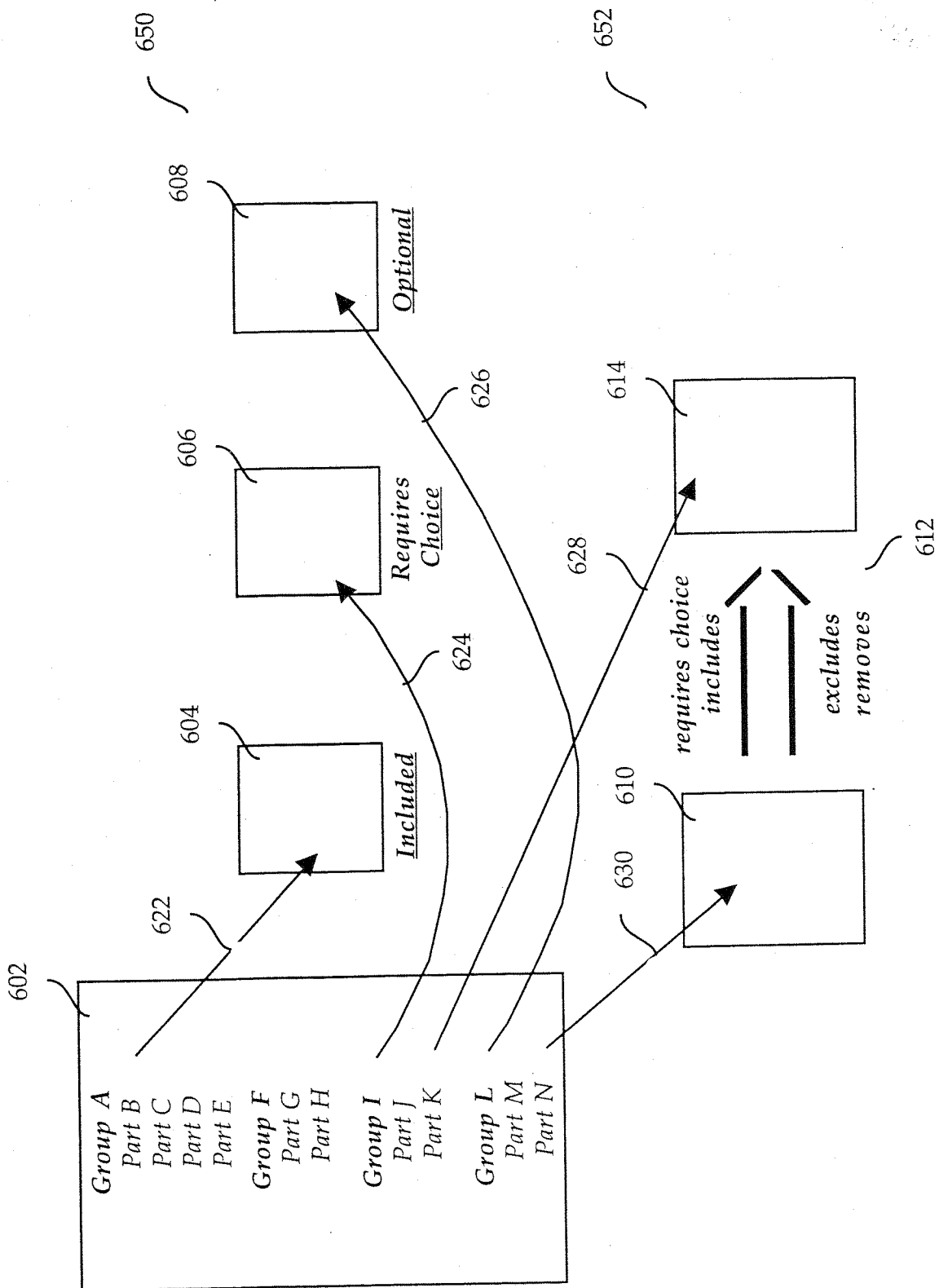


FIG. 6

7/24

APPROVED	DATE FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

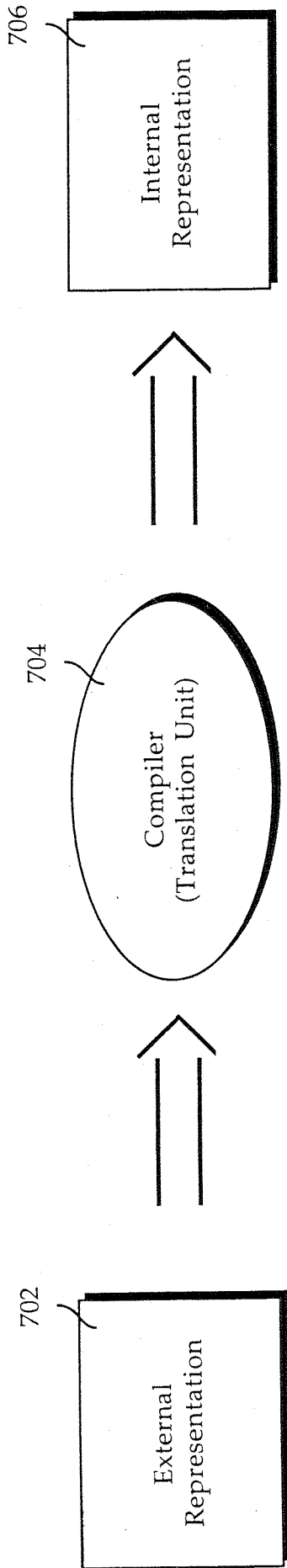


Figure 7

8/24

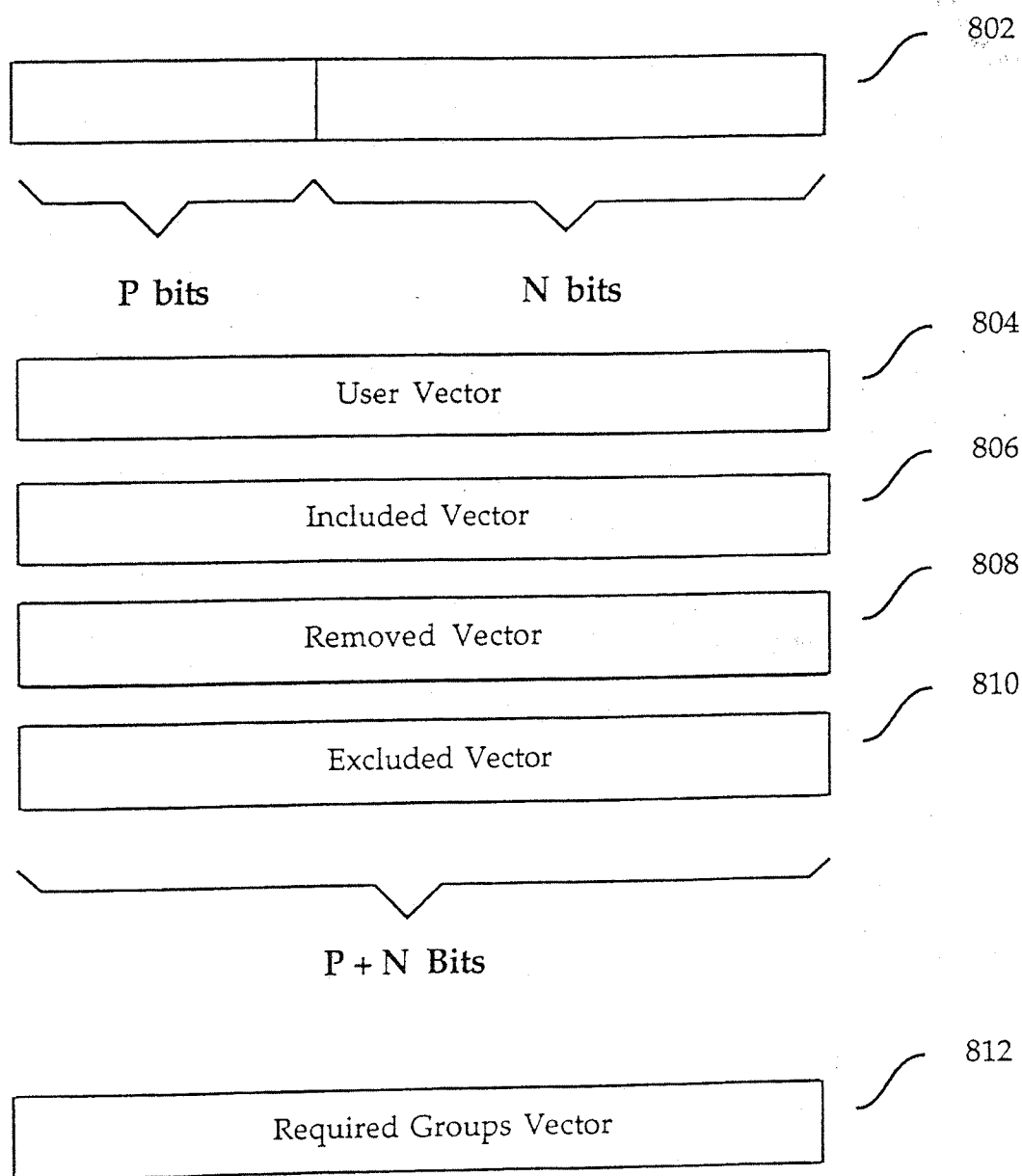


FIG. 8A

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

9/24

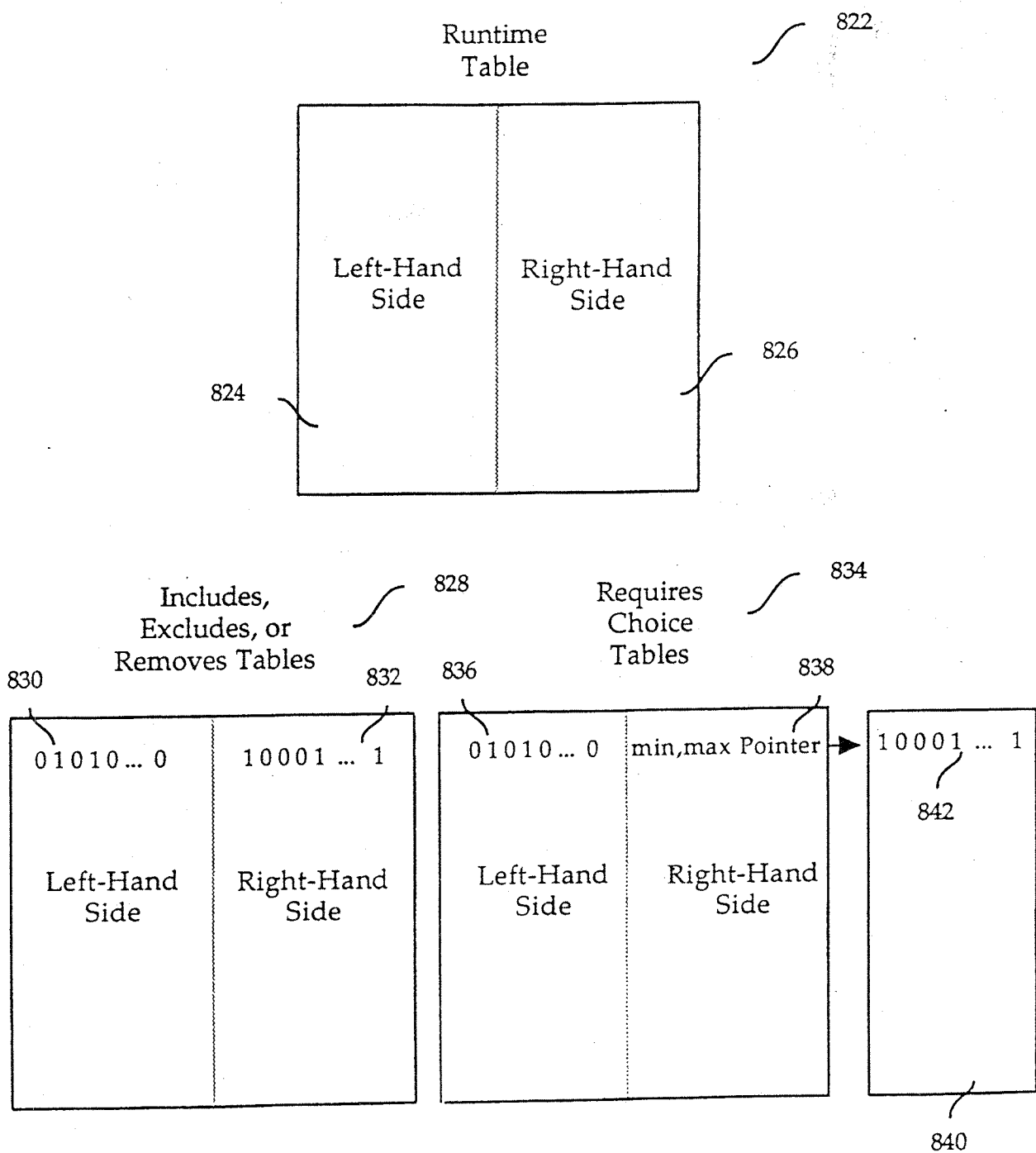


FIG. 8B

10/24

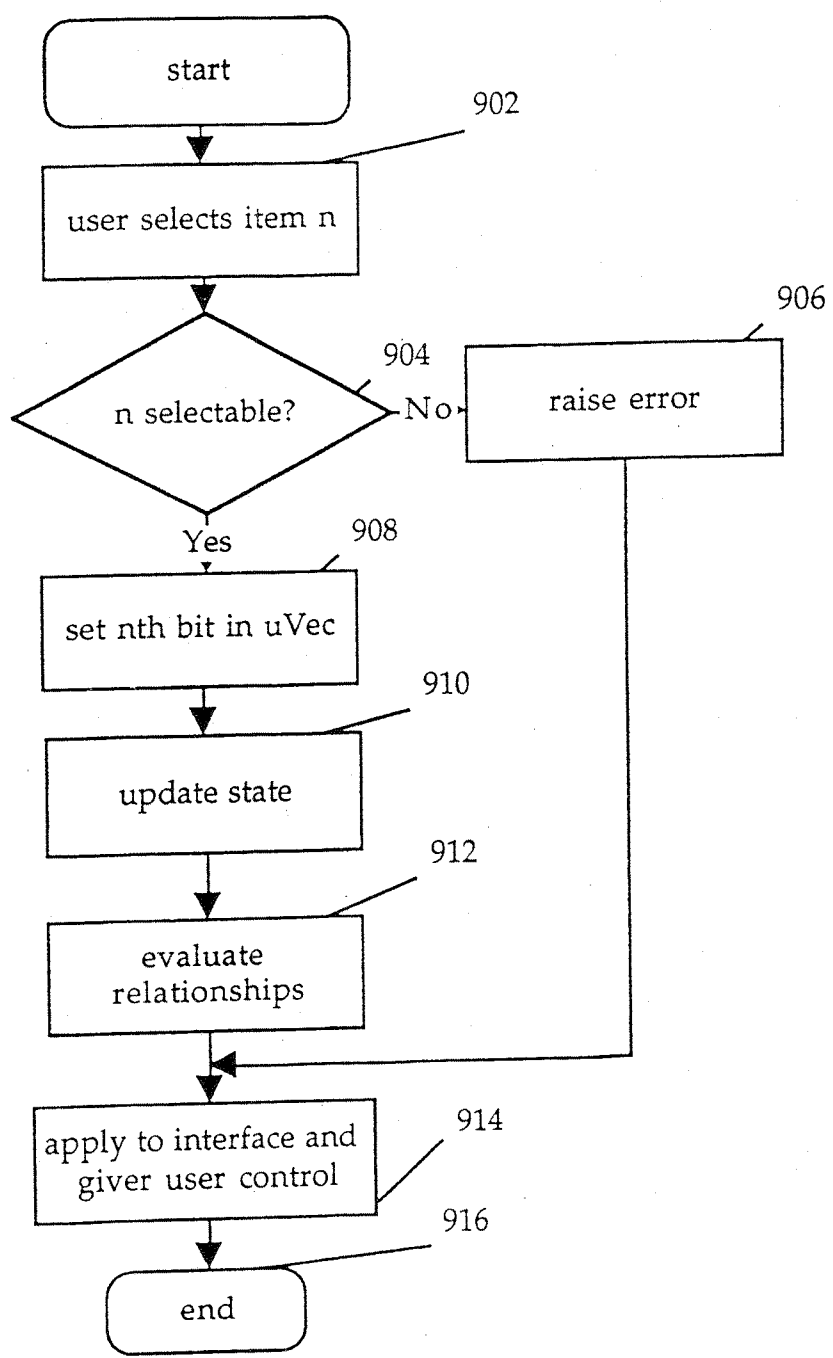


FIG. 9

11/24

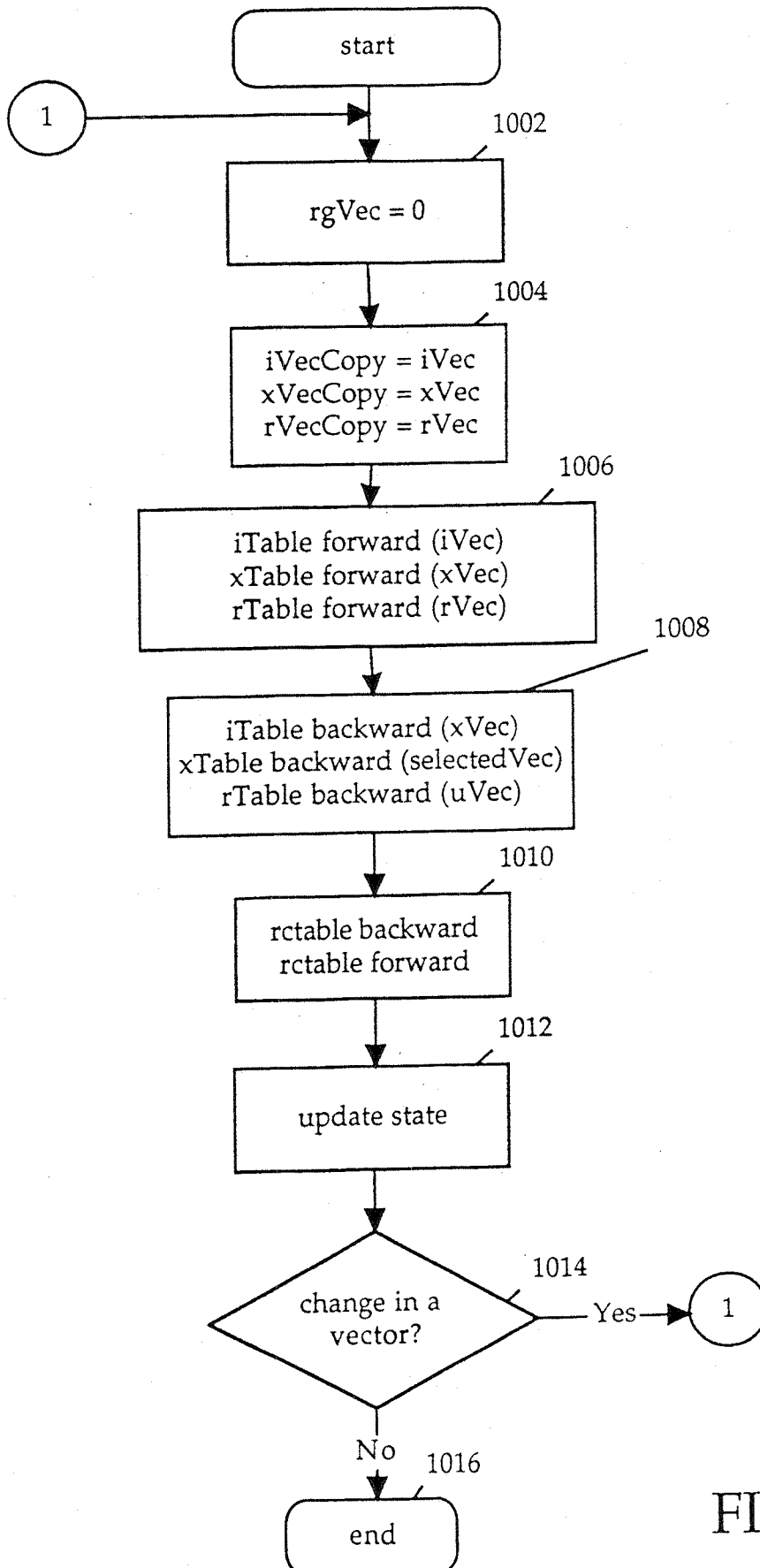


FIG. 10

12/24

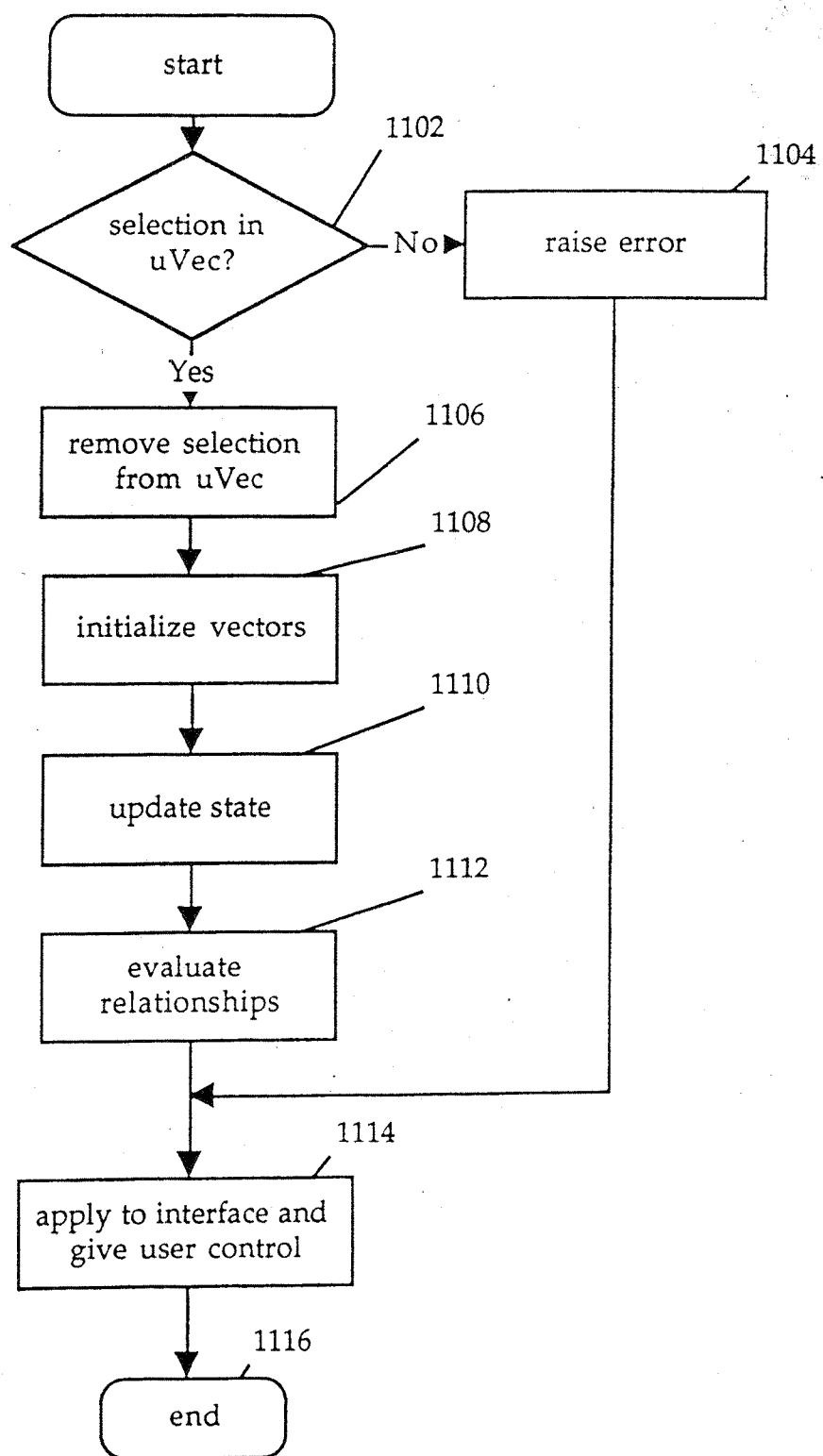


FIG. 11

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

13/24

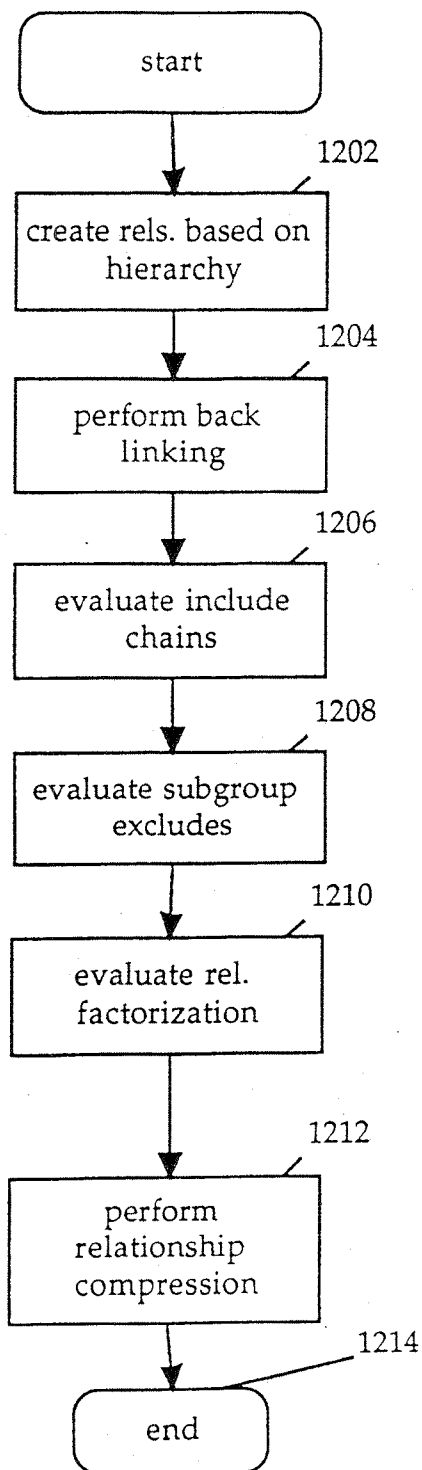


FIG. 12

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

14/24

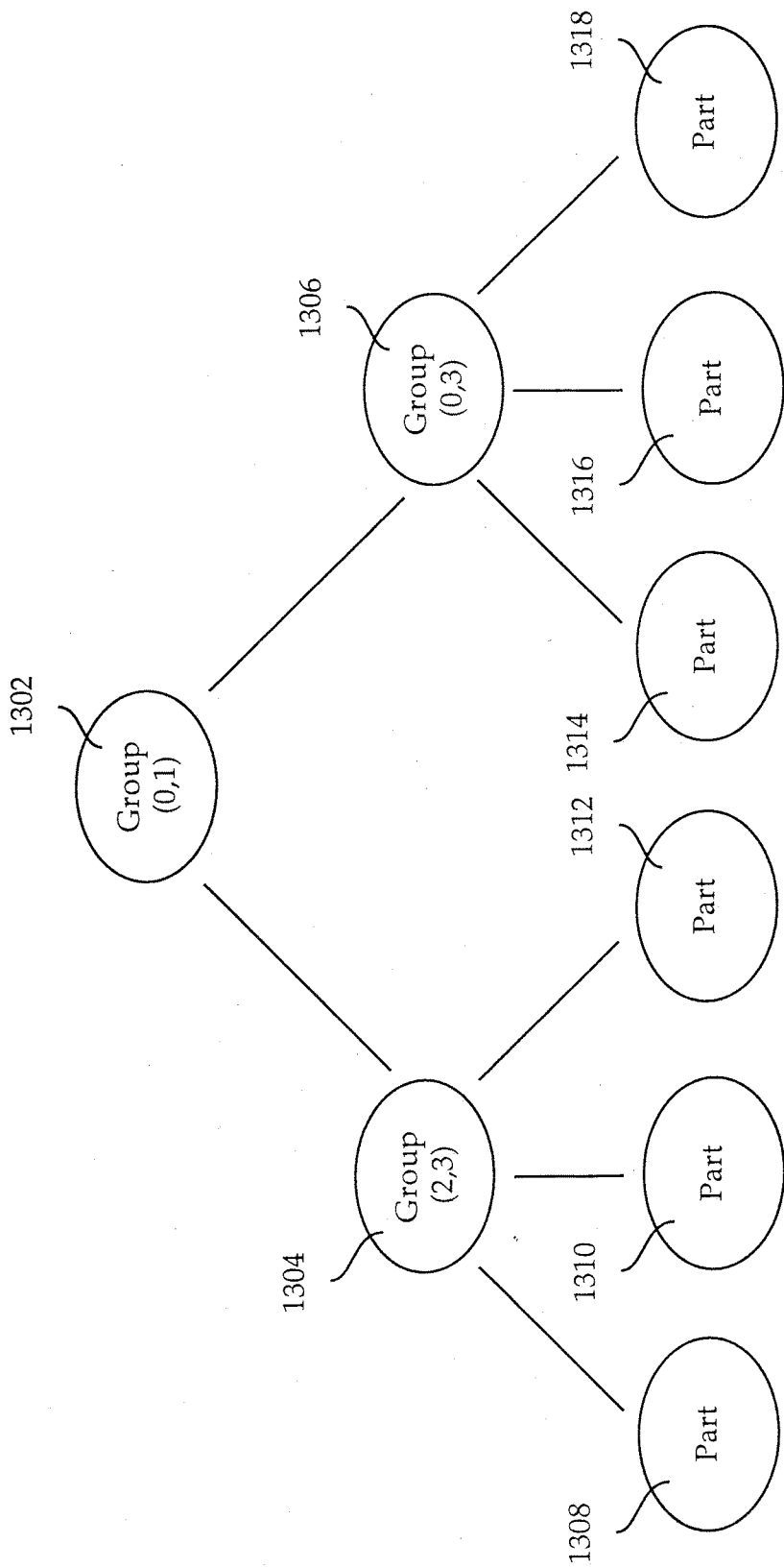


FIG. 13A

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

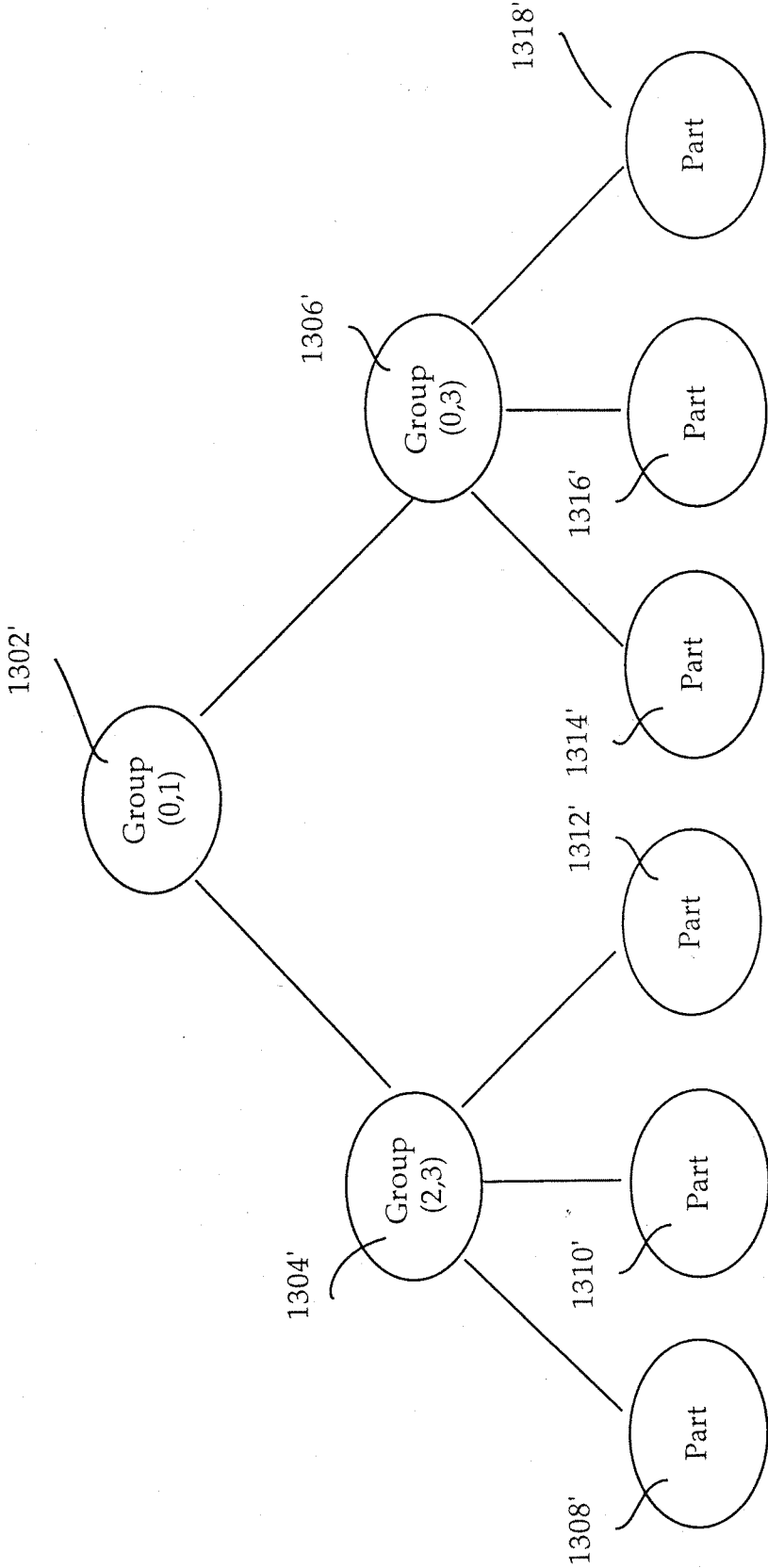


FIG. 13B

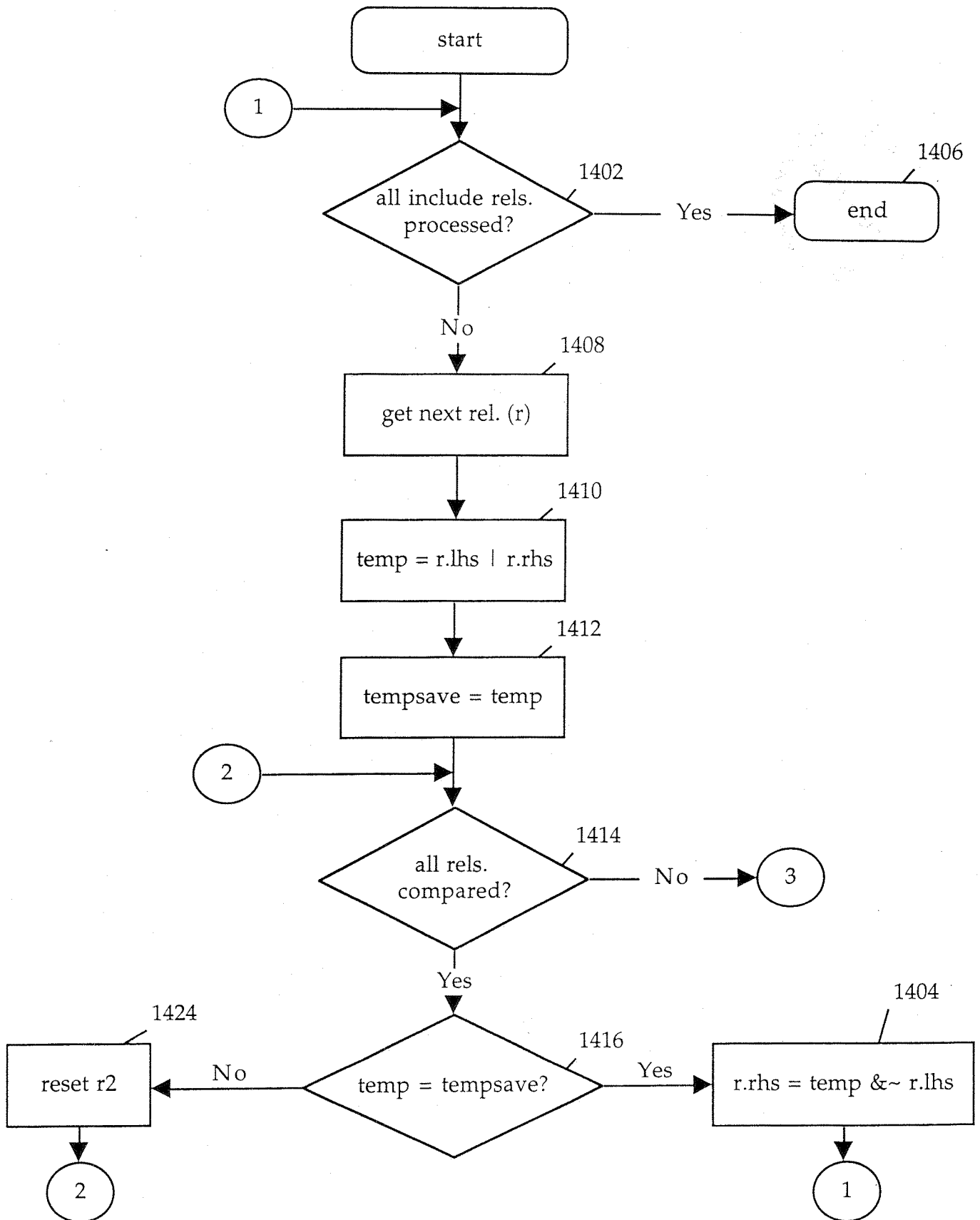


Fig. 14A

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

17/24

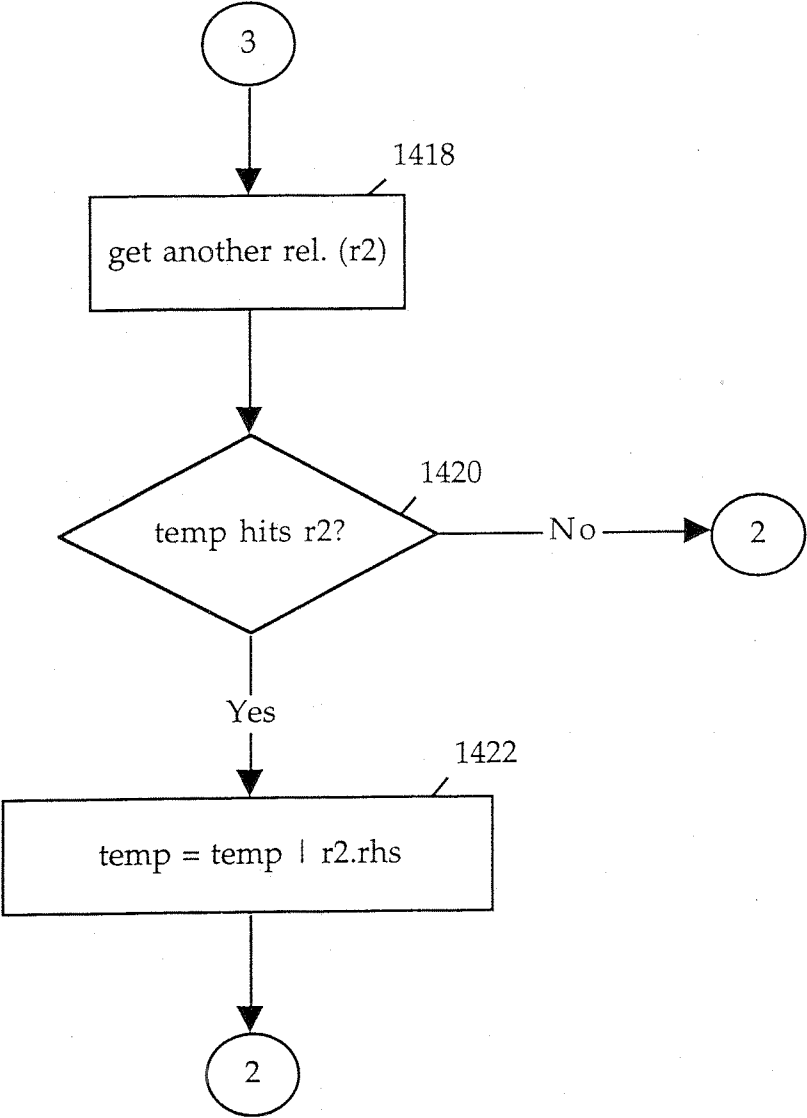


FIG. 14B

18/24

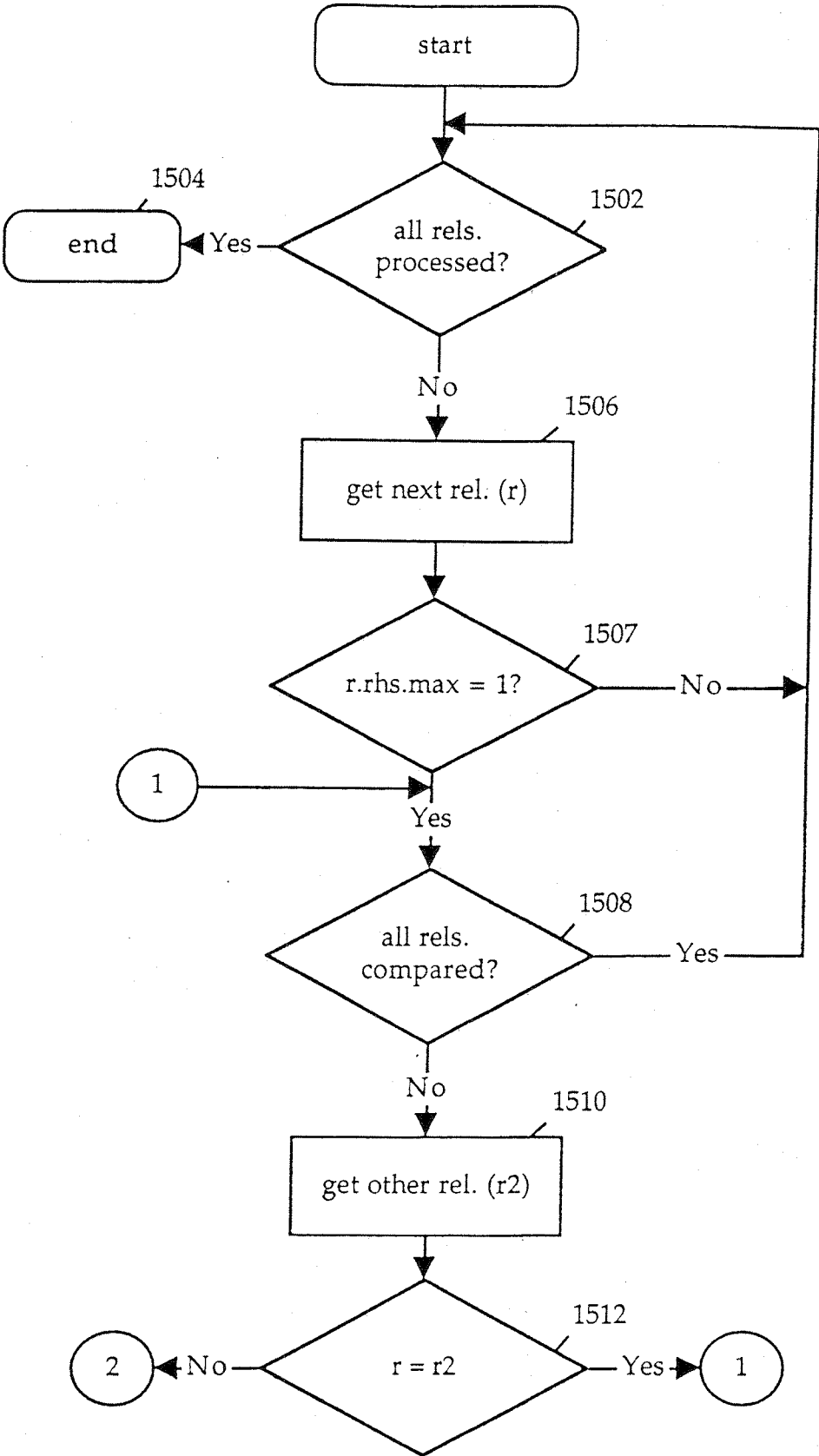


FIG. 15A

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

19/24

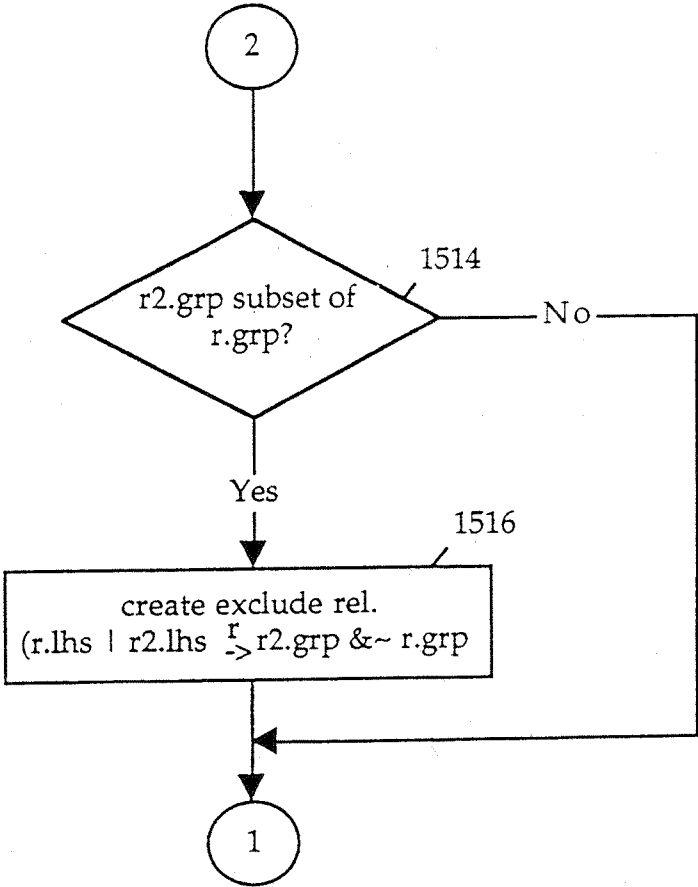


FIG. 15B

20/24

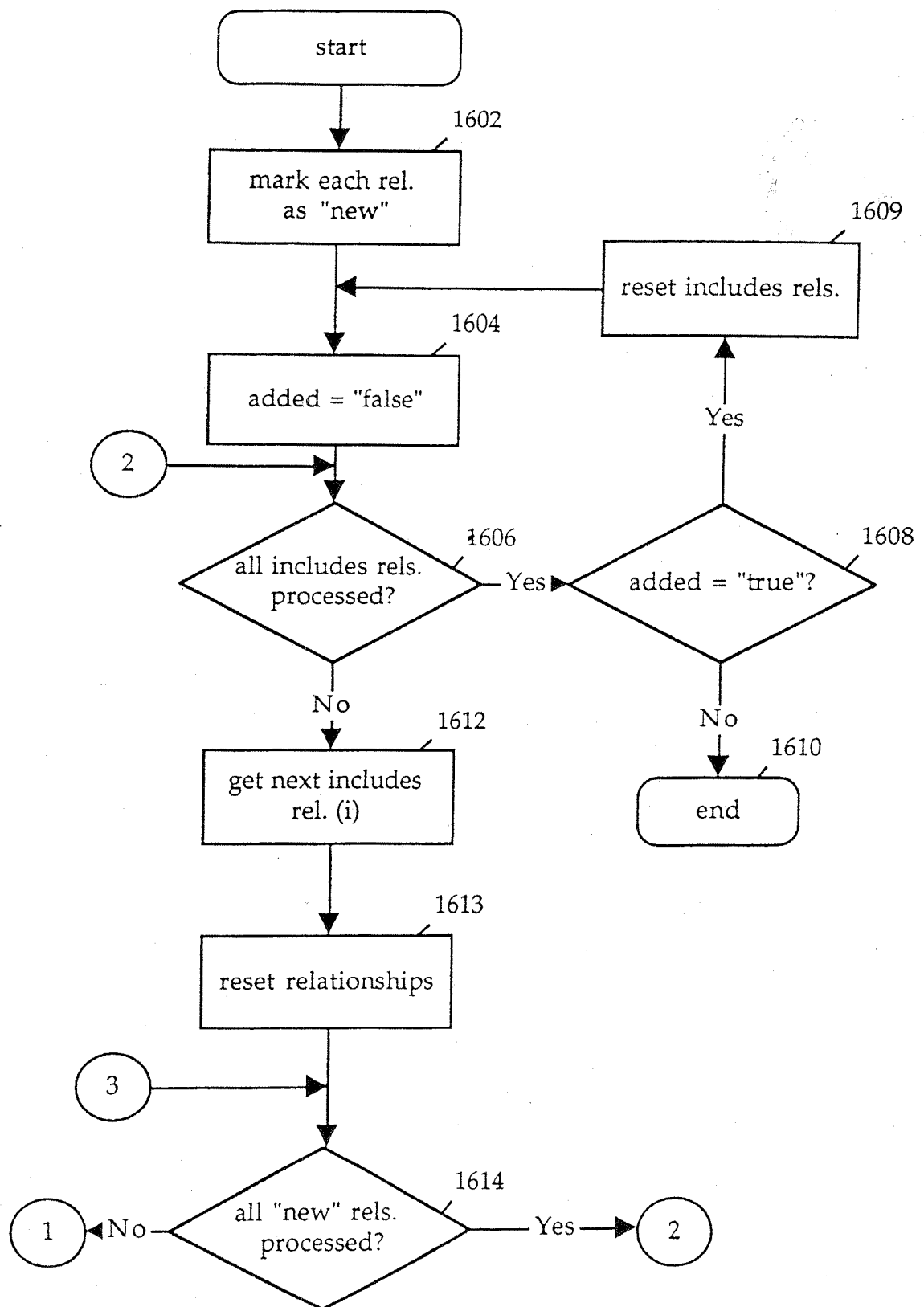


FIG. 16A

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

21/24

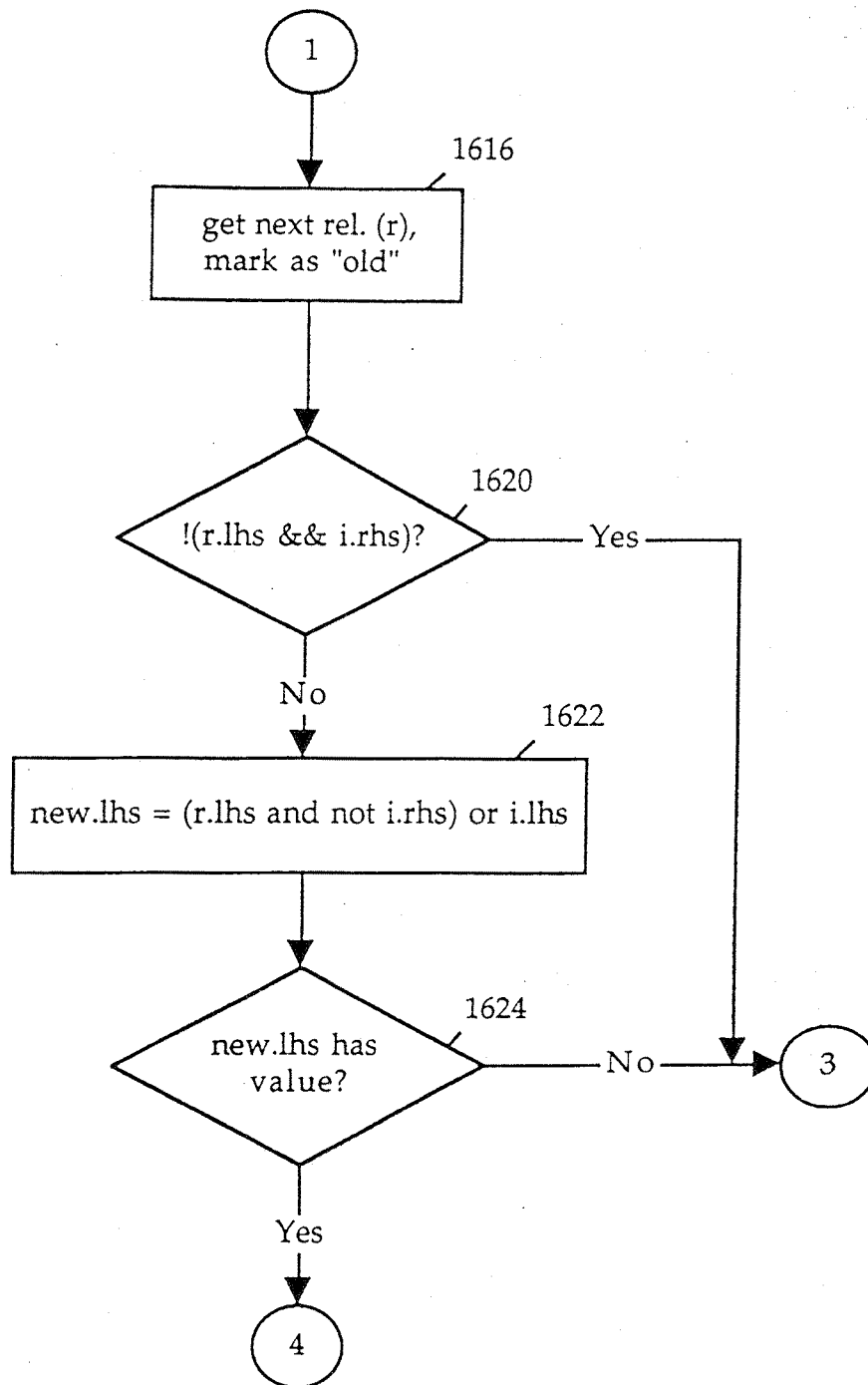


FIG. 16B

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

22/24

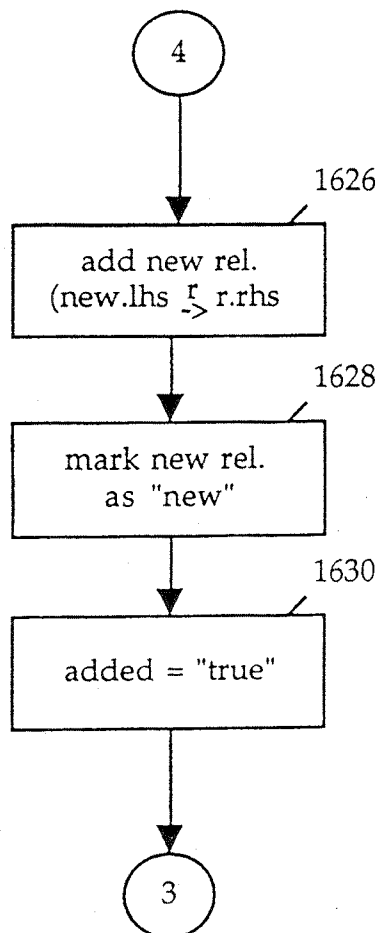


FIG. 16C

23/24

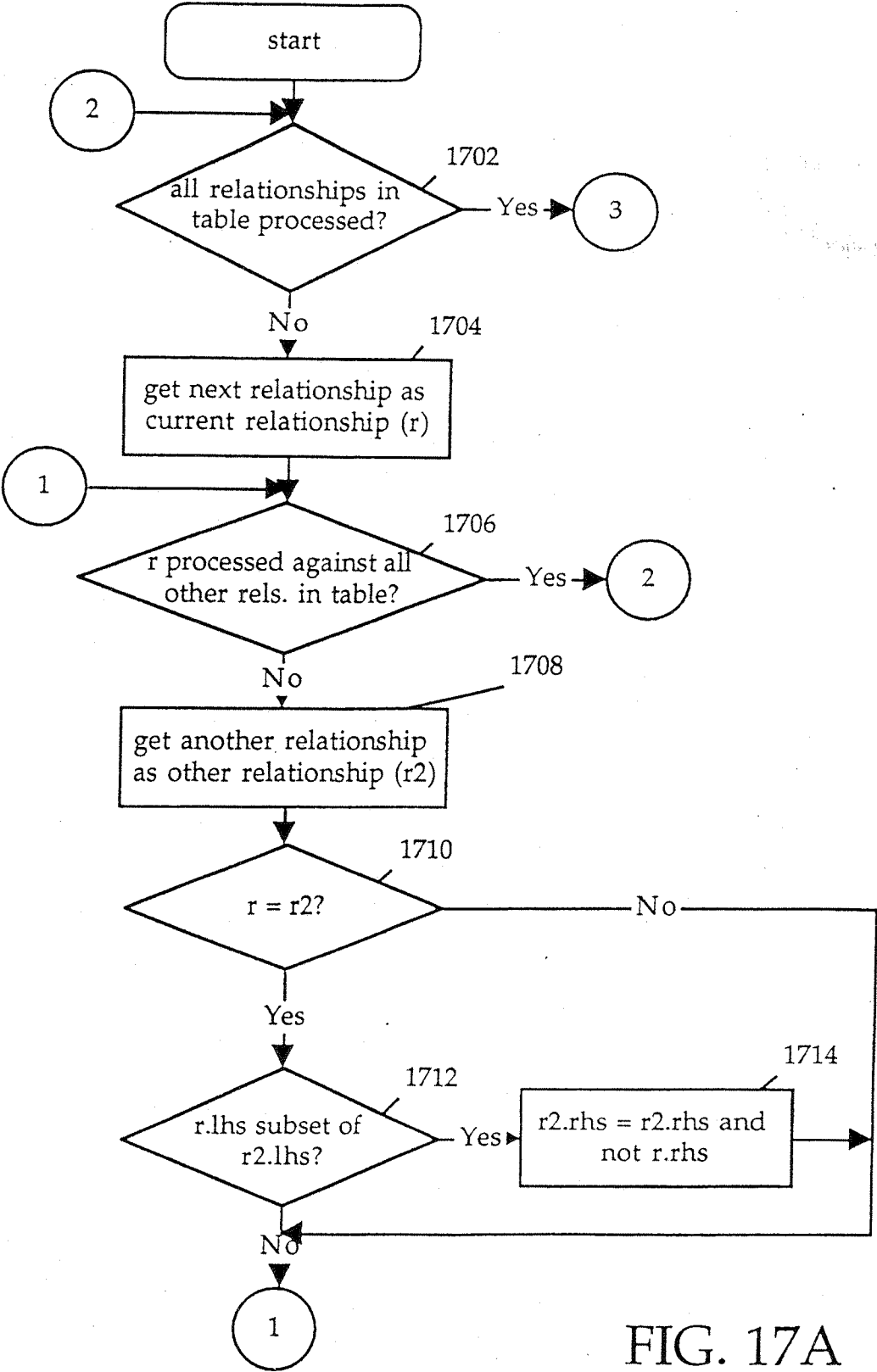


FIG. 17A

APPROVED	O.G. FIG.	
BY	CLASS	SUBCLASS
DRAFTSMAN		

24/24

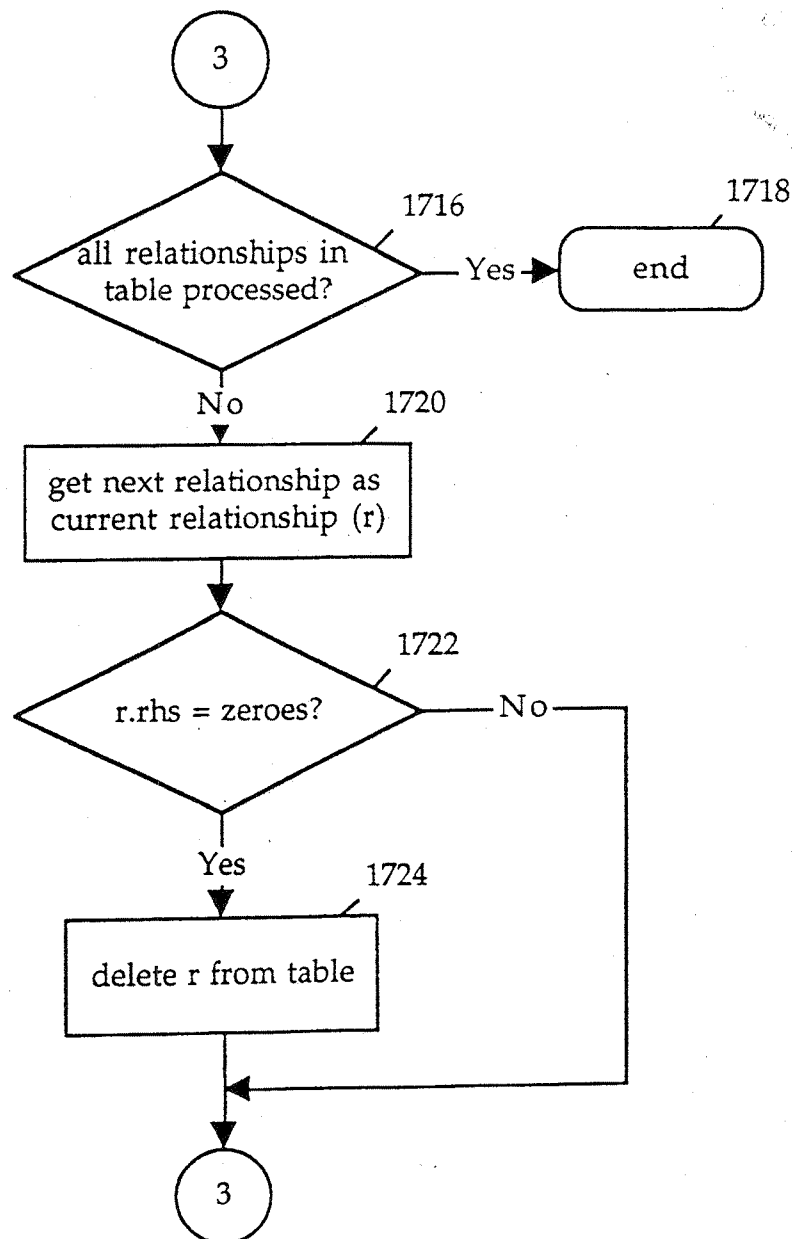


FIG. 17B

The
United
States
of
America



PTO UTILITY GRANT

Paper Number 9

The Commissioner of Patents
and Trademarks

Has received an application for a patent for a new and useful invention. The title and description of the invention are enclosed. The requirements of law have been complied with, and it has been determined that a patent on the invention shall be granted under the law.

Therefore, this

United States Patent

Grants to the person(s) having title to this patent the right to exclude others from making, using, offering for sale, or selling the invention throughout the United States of America or importing the invention into the United States of America for the term set forth below, subject to the payment of maintenance fees as provided by law.

If this application was filed prior to June 8, 1995, the term of this patent is the longer of seventeen years from the date of grant of this patent or twenty years from the earliest effective U.S. filing date of the application, subject to any statutory extension.

If this application was filed on or after June 8, 1995, the term of this patent is twenty years from the U.S. filing date, subject to an statutory extension. If the application contains a specific reference to an earlier filed application or applications under 35 U.S.C. 120, 121 or 365(c), the term of the patent is twenty years from the date on which the earliest application was filed, subject to any statutory extension.

Bence Lehman
Commissioner of Patents and Trademarks

Allie M. Person
Attest

Form PTO-1584 (Rev. 2/97)

SOLICITOR

TO: Commissioner of Patents and Trademarks 2121 Crystal Dr., Crystal Park, Suite 1100 Arlington, VA 22202	NOV 02 1999	REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK
--	-------------	---

U.S. PATENT & TRADEMARK OFFICE

In Compliance with 35 § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the WDTX on the following ☒ Patents or ☐ Trademarks:

DOCKET NO. <u>A-99-cv-690</u>	DATE FILED <u>10/22/99</u>	U.S. DISTRICT COURT <u>Western District of Texas - Austin</u>
PLAINTIFF <u>Trilogy Software, Inc.</u>		DEFENDANT <u>CarsDirect.Com, Inc. & Autodata Marketing Systems, Inc.</u>
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
<u>1 5,825,651</u>	<u>10.20.98</u>	<u>Trilogy Software, Inc.</u>
2		
3		
4		
5		

In the above-entitled case, the following patent(s) have been included:

DATE INCLUDED	INCLUDED BY <input type="checkbox"/> Amendment <input type="checkbox"/> Answer <input type="checkbox"/> Cross Bill <input type="checkbox"/> Other Pleading		
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK	
1			
2			
3			
4			
5			

In the above-entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE
-------	-------------------	------



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
PO Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	PATENT NUMBER	GROUP ART UNIT	FILE WRAPPER LOCATION
08/707,187	5825651	2787	9200 4 10031 1003 104

Change of Address/Power of Attorney

The following fields have been set to Customer Number 33438 on

- Correspondence Address

The address of record for Customer Number 33438 is:

HAMILTON & TERRILE, LLP
P.O. BOX 203518
AUSTIN, TX 78720

The Practitioners of record for Customer Number 33438 are:

PTO INSTRUCTIONS:

Please take the following action when the correspondence address has been changed to a customer number:

- 1) Add 'ADDRESS CHANGE TO CUSTOMER NUMBER' on the next available content line of the File Jacket.
- 2) Put a line through the old address on the File Jacket and enter the Customer Number as the new address.
- 3) File this Notice in the File Jacket.

Please take the following action when the correspondence address has NOT been changed:

- 1) File this Notice in the File Jacket