PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 17/30	A1	(11) International Publication Number: WO 99/10820
		(43) International Publication Date: 4 March 1999 (04.03.99)

(21) International Application Number: PCT/IB98/01553

(22) International Filing Date: 19 August 1998 (19.08.98)

(30) Priority Data:

60/058,946	22 August 1997 (22.08.97)	US
09/134,502	14 August 1998 (14.08.98)	US

(71) Applicant: SAP AKTIENGESELLSCHAFT [DE/DE]; Neurottstrasse 16, D–69190 Walldorf (DE).

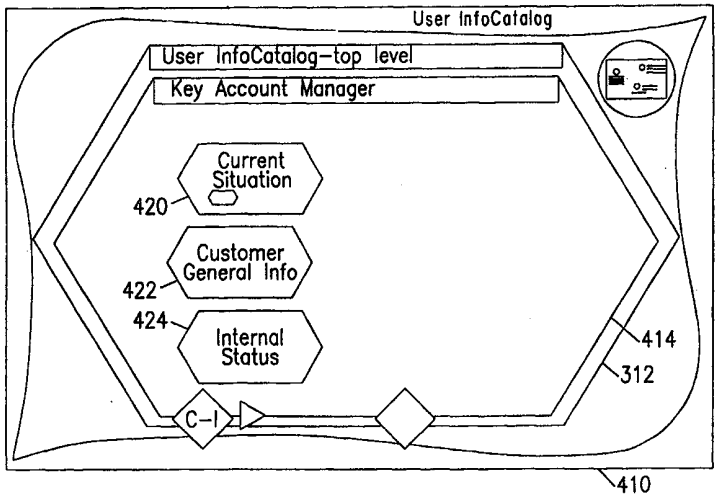
(72) Inventor: EBERT, Peter; Felchesgasse 3, D–64292 Darmstadt (DE).

(74) Agents: PFEIFER, Hans–Peter et al.; Beiertheimer Allee 19, D–76137 Karlsruhe (DE).

(81) Designated States: AU, CA, IL, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: BROWSER FOR HIERARCHICAL STRUCTURES



(57) Abstract

A browser for viewing data representing text, pictures, sounds, virtual reality worlds, links, or other objects, and organized in a hierarchical structure is provided to display and interact with links to the data. A retrieved set of data is displayed in a lens, and subsequently retrieved sets of data that are deeper in the hierarchical structure are displayed in further stacked lenses, creating a visual representation of the hierarchy. A label on each lens provides information on the data originally displayed in the lens. A user may choose to view the originally displayed contents of any lens on the screen. A user may also modify a personal hierarchical data structure comprising links to sets of data in the first hierarchical structure. In doing so, a user creates an entirely new hierarchical structure that customizes the organization of data according to personal preferences and takes advantage of natural association to reduce the number of perceived levels in the new hierarchical data structure.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

PATENT APPLICATION

Browser For Hierarchical Structures

Applicant hereby claims the benefit of the earlier filing date of Provisional Patent Application No. 60/058,946 entitled "Browser for Hierarchical Structures", filed August 22, 1997, pursuant to 35 U.S.C. § 119(e).

FIELD OF THE INVENTION

This invention relates to browsers as a means of viewing data and files structures. More particularly, this invention discloses a browser for viewing data wherein a retrieved set of data is displayed in a lens, and subsequently retrieved sets of data that are deeper in the hierarchical structure are displayed in further stacked lenses, creating a visual representation of the hierarchy.

BACKGROUND

With the tremendous advances in computer processing power and network capabilities over the past decade, the popularity of graphical user interfaces for viewing information over a network has increased substantially. Whereas in the past most networks used text displays to convey data from a server to a user, most interfaces today use graphics to convey information in a manner that is not only more pleasing to the eye, but also allows data to be manipulated and presented in ways not possible with pure text.

The first graphical user interfaces for networks were primitive and not entirely intuitive. A user would have to type complex commands to select the data to be displayed or to change the manner the data is displayed. Searches also required complex commands that were not very different from those used for text-based interfaces. However, searches were often the only way to find the

desired data, since these primitive interfaces were not able to take full advantage of the links, to the extent that they existed, to quickly move a user from one set of data to another related set of data.

Today, browsers with graphical user interfaces are often used for viewing data across a network. Examples of popular browsers being used today are Netscape Navigator and Microsoft Explorer. Today's browsers can take greater advantage of links between related sets of data, and use these links to navigate among the various sets, including going up and down data hierarchies and revisiting previously viewed data.

While browsers and their graphical user interfaces are becoming increasingly sophisticated, the underlying data itself has become more complex. Data may be organized into huge databases of information that is spread across a network. Data may be organized into objects, including applications and related files, or groups thereof. This data may be organized in larger hierarchies, often many levels deep.

These hierarchically organized objects and groups usually consist of textual data, icons, or two-dimensional images. New development tools, such as JAVA, allow browsers to display more sophisticated objects, such as moving pictures or moving text. However, even these objects are limited by the underlying descriptive language used to create them. For example, today's browsers are not capable of displaying interactive VRML (Virtual Reality Markup Language) worlds.

It has become a challenge to present data in a clear and efficient manner through networks. Browsers can display only a limited amount of information on a user's screen at any given time. If a user chooses to use one lens, or window, to display a given set of data, the data disappears when a user subsequently displays other sets of data. If a user then wishes to revisit a prior set of data, it is usually necessary to visit all the links between the most recently viewed set of data and the desired prior set of data. Although most browsers allow

users to create bookmarks for a given set of data, bookmarks are not automatically created, and require extra effort on a user's part. Many browsers also keep a list of recently accessed data, but users are then required to have the list displayed on the screen (taking up valuable screen space) and then remember which set of data is the desired set.

Some browsers allow users to have multiple lenses, or windows, each with a different set of data, displayed on the screen at any given time. However, as the multiple windows start overlapping each other, a user can easily lose track of the order in which the data was visited. Alternatively, a user may wish to recall the hierarchy of the sets of data previously visited, which is also easily forgotten when multiple windows are displayed.

On the other hand, a user may wish to have easy access to links to frequently accessed data. As noted above, most browsers allow a user to create bookmarks. However, as the number of bookmarks increases, it becomes more difficult for the user to quickly find a particular bookmark. This problem may be alleviated somewhat by having a hierarchical structure of bookmarks. However, this is far from a perfect solution, as an increasingly complicated hierarchical structure of bookmarks makes it even more difficult for a user to find a particular link, and time is wasted going through the various levels of the hierarchical structure. Thus, there is also a need to organize links in a manner that makes them readily accessible.

SUMMARY OF THE INVENTION

The present invention relates to an improvement over classic network browsers. The browser being disclosed provides easy access to previously accessed data.

It is an object of this invention to convey data in an efficient manner by displaying the data according to the data's hierarchical structure.

It is another object of this invention to give a user easy access to a hierarchical data structure by

providing access to data at several levels within the hierarchical data structure simultaneously.

It is another object of this invention to provide labels for lenses displaying data at various levels within a hierarchical data structure.

It is another object of this invention to organize the display of hierarchical data in a logical manner that is pleasing to the eye.

It is another object of this invention to provide a means for a user to define a personal hierarchical data structure.

It is another object of the invention to use links in a personal hierarchical data structure in conjunction with natural association to access data in another hierarchical data structure having more levels than the personal hierarchical data structure.

It is another object of the invention to utilize colors, shapes, or other attributes to indicate natural association of links in a personal hierarchical data structure.

It is another object of the invention to use links in a personal hierarchical data structure in conjunction with natural association to access data, wherein the data is obtained through channels.

It is another object of the invention to allow exception reporting in conjunction with a personal hierarchical data structure.

It is another object of the invention to organize data into objects and groups of objects that can be represented as three-dimensional animations and VRML-worlds.

It is another object of the invention to display a context-sensitive menu when a cursor is placed on or near an object.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a view of a computer screen showing an embodiment of the display for a browser.

Fig. 2 is a flow chart showing how the display of lenses in a browser is determined.

Fig. 3a is a view of a computer screen showing an example of a lens displaying a graphical representation of a category at a first level.

Fig. 3b is a tree structure representation of the category being represented in Fig. 3a.

Fig. 4a is a view of a computer screen showing an example of a lens displaying a graphical representation of a category at a second level.

Fig. 4b is a tree structure representation of the category being represented in Fig. 4a.

Fig. 5a is a view of a computer screen showing an example of a lens displaying a graphical representation of a category at a third level.

Fig. 5b is a tree structure representation of the category being represented in Fig. 5a.

Fig. 6a is a view of a computer screen showing an example of a lens displaying a graphical representation of a category at a fourth level.

Fig. 6b is a tree structure representation of the category being represented in Fig. 6a.

Fig. 7 is an example of a simplified object model showing how a hierarchical browser may be programmed.

Fig. 8 is a view of a computer screen showing an embodiment of the user favorites screen containing graphical representations of user-created links.

Fig. 9a is a view of a computer screen showing an embodiment of a display containing graphical representations of user-created links, wherein a popup menu is displayed.

Fig. 9b is a tree structure representation of the user-created links shown in Fig. 9a.

Fig. 10a is a view of a computer screen showing an embodiment of a display containing graphical representations of user-created links, wherein a link has been moved.

Fig. 10b is a tree structure representation of the user-created links shown in Fig. 10a.

Fig. 11a is a view of a computer screen showing an embodiment of a display containing graphical

representations of user-created links, wherein the physical representations of the links have been moved on the display without changing the hierarchical structure of the links.

Fig. 11b is a tree structure representation of the user-created links shown in Fig. 11a.

Fig. 12 is a view of a computer screen showing an alternative embodiment of the user favorites screen containing graphical representations of user-created links.

Fig. 13 is an example of a simplified object model showing how a user favorites screen containing graphical representations of user-created links may be programmed.

DETAILED DESCRIPTION OF THE INVENTION

For clarity and simplicity, the invention will be described in the context of SAP software, including the marketed R/3 System. A brief description of the R/3 System follows.

R/3 can be described primarily as an online transaction processing system designed to provide integrated processing of all business routines and transactions. It includes enterprise-wide, integrated solutions, as well as specialized applications for individual, departmental functions. R/3 mirrors all of the business-critical processes of the enterprise - finance, manufacturing, sales, and human resources. It also offers various analytical capabilities to supplement the transaction processing function.

Recent paradigm shifts have forced companies to optimize all business processes along the net-value-added chain through the use of modern data processing and enterprise-wide information management. In order to become successful, companies often network with customers and suppliers such that products and services of high quality can be made available with minimal delay in response to customer demands. R/3 achieves these goals with online integration of data in the company.

The R/3 System is based on SAP's client/server architecture which separates the database, application, and presentation components for greater flexibility. This

enables enterprises to take advantage of the various benefits of the architecture, including the capability to run across a variety of today's most popular UNIX-based hardware platforms. The R/3 System is designed so that the application systems and system functions are decoupled by a clear layer of architecture, with an application layer and a basis layer. The multilayer architecture of the R/3 System allows optimal load distribution, even in large installations with thousands of users. It has a modular structure with methods for controlling master-slave relationships between individual software components. Special servers linked by communications networks can be used for certain tasks without losing the integration of data and processes in the overall system. Implementation of separate servers for particular tasks makes optimal use of the performance potential and the different cost structures of available hardware architecture. This allows applications to be partitioned into different services and run on different servers. The fundamental services of the R/3 system (graphical presentation services, application services for handling of the application logic, database services for storage and recovery of business data, etc.) are partitioned and may evolve separately while maintaining interoperability.

It will be appreciated that the invention may be practiced using different hardware and different operating systems, and is not restricted to use with R/3. In fact, the invention is designed to be platform-independent.

Furthermore, the invention may be designed as modules that can be imbedded in a container. Thus, Active-X Control or OCX may be used, though the invention should not be construed as being limited to such implementations.

Preferred Embodiments of the Browser

It will be assumed that a hierarchically structured database exists. For illustrative purposes, within this database reside sets of data, including four sets of data with the following category names assigned to them: "User Info Catalog", "Companies", "Computer Technology", and "SAP". "User Info Catalog" is linked to

"Companies", which is one hierarchical level below "User Info Catalog". Similarly, "Companies" is linked to "Computer Technology", which is another hierarchical level below. Finally, "Computer Technology" is linked to "SAP", which is a further hierarchical level below.

With reference to Fig. 1, screen 10 shows a preferred embodiment of the invention. Lenses 12, 14, 16, and 18 display graphical representations of hierarchies of sets of data. Although the lenses illustrated in the figures are substantially hexagonal in shape, the lenses may be in the form of rectangles or other shapes in other preferred embodiments. Lens 12 is used to display a graphical representation of the category "User Info Catalog". Since "Companies" is at a deeper hierarchical level than "User Info Catalog", the graphical representation of "Companies" is displayed in the smaller lens 14. At the next level, lens 16 is used to display a graphical representation of "Computer Technology". Finally, lens 18 is used to display a graphical representation of "SAP". Each lens also displays the label or name of the category or set of data associated with that particular lens. Since lens 18 is the smallest lens displayed, it is also currently displaying a graphical representation of the set of data associated with "SAP". The set of data graphically represented in lens 18 contains objects, which may include text, icons, categories, applications, executable objects, such as executable reports, two-dimensional images, three-dimensional images, and VRML worlds.

Lens 18 is displaying a graphical representation of three objects 20, 22, and 24 that are categories one hierarchical level lower than "SAP". Objects 20, 22, and 24 are graphically represented as icons, but selecting one of these will cause a new, smaller lens to appear within lens 18 to display a graphical representation of the category selected. Lens 18 is also displaying graphical representations of objects 30, 32, 34, and 36. In a preferred embodiment, moving a cursor near one of these objects may cause a context-sensitive popup menu to appear

with a list of possible commands to be selected. For example, moving a cursor near object 34 causes popup menu or context menu 37 to appear. Since object 34 is an executable object in the example, popup menu or context menu 37 has an "execute" option. In a preferred embodiment, a popup menu or context menu is predefined and is automatically displayed in a format based on the object type and user authorization. Optionally, such a menu could be hierarchical.

A user may go to a deeper level in the hierarchical data structure by selecting a set of data graphically represented on screen 10 that has a deeper hierarchical level than the set of data currently being graphically represented. For example, selecting one of objects 20, 22, or 24 will bring a user to a deeper level. In a preferred embodiment, a graphical representation of an object that is a category will indicate whether the category contains, at the next deeper level, an additional object. For example, symbol or icon 25 and symbol or icon 26 indicate that at least one object that is a category and at least one object that is not a category exist one hierarchical level deeper than object 20. Similarly, symbol or icon 28 indicates that at least one category exists one hierarchical level deeper than object 22.

There are numerous methods by which a user may select an object graphically represented on screen 10. The most popular method used today is to use a mouse to move a mouse cursor to the graphical representation of the object being selected, and clicking on the mouse button. It will be appreciated that other methods, such as making use of a keyboard, are available.

Conversely, a user may wish to go to a higher level in the hierarchical data structure. This may be accomplished by selecting one of the lenses displayed behind the most recently displayed lens. For example, in Fig. 1 a user may select lens 12, 14, or 16 to display the contents of the selected lens. One way to select a lens that is already displayed on screen 10 is to move a mouse cursor to the displayed portion of the lens. For example,

a user may view the objects previously displayed in lens 14 by placing the mouse cursor in the general area of point 15 and clicking the mouse button. Alternatively, a user may press a key or key combination on the keyboard, such as an <Alt> key and left arrow key combination, to go up one level. This works in a manner similar to the "Back" button found in many popular browsers today. Selecting a lens on screen 10 will displace all smaller lenses on screen 10.

In a preferred embodiment, a user may navigate within a lens using a mouse, keyboard controls, or other means. For example, a mouse may be used to scroll in any direction within a lens. The left and right arrow keys of a keyboard may be used to scroll left or right, or to select objects having a graphical representation in a lens. Other keys may be used to select objects, for example by pressing the key corresponding to the first letter of the name of an object. In another preferred embodiment, buttons 40 and 42 may be used to scroll through objects. For example, selection of button 40 will cause the graphical representations of objects 20, 22, and 24 to disappear and to be replaced by graphical representations of other categories one hierarchical level lower than "SAP" to appear. Similarly, selection of button 42 will cause the graphical representations of objects 30, 32, 34, and 36 to disappear and to be replaced by graphical representations of additional objects.

With reference to Fig. 2, a preferred embodiment for determining the size of lenses is shown in flowchart 100. A computer makes a request 102 for a set of data. Upon receipt 104 of the requested set of data, a comparison 106 is made between the hierarchical level of the received set of data and the hierarchical level of the set of data currently graphically represented within the smallest lens displayed on screen 10. If the received set of data is at a deeper hierarchical level, the creation 108 of a lens that is smaller than the currently displayed lens or lenses is commenced. If the received set of data is not at a deeper hierarchical level, then removal 112 of the lens or lenses displaying graphical representations of any sets of

data that are at the same hierarchical level as the received set of data or at a deeper hierarchical level than the received set of data is commenced. After removal 112 occurs, creation 108 of a lens that is smaller than the remaining displayed lens or lenses is commenced. After the creation 108 of a lens, display 110 of a graphical representation of the received set of data commences within the lens created in creation 108. The computer is now ready to make a request 102 for another set of data.

It will be appreciated that flowchart 100 is only one preferred embodiment that there are other possible methods for determining the size of lenses and for displaying them in a fashion consistent with the invention. For example, in another preferred embodiment, if the computer has the information necessary to determine the hierarchical level of a requested set of data, it may be possible to make a comparison 106 before receipt 104 of the requested set of data. For example, if a requested set of data is associated with a URL that indicates the level of the data (such as a URL that uses slash characters to indicate a directory structure), it may be possible to determine the hierarchical level of the requested set of data before it arrives.

In another preferred embodiment, removal 112 of a displayed lens or lenses is not necessary if the creation 108 of a lens and display 110 of a graphical representation of the received set of data are accomplished in such a way that other lens or lenses are blocked off the screen 10.

Examples of Browser Displaying Hierarchical Data

It will again be assumed that a hierarchically structured database exists. For illustrative purposes, within this database reside sets of data, including four sets of data with the following category names assigned to them: "User InfoCatalog", "Key Account Manager", "Current Situation", and "Financial Situation". "User InfoCatalog" is linked to "Key Account Manager", which is one hierarchical level below "User InfoCatalog". Similarly, "Key Account Manager" is linked to "Current Situation", which is another hierarchical level below. Finally,

"Current Situation" is linked to "Financial Situation", which is a further hierarchical level below.

With reference to Fig. 3a, screen 310 shows lens 312, which is displaying a graphical representation of the category "User InfoCatalog". Lens 312 is displaying a graphical representation of three objects 320, 322, and 324 that are categories one hierarchical level lower than "User InfoCatalog". Objects 320, 322, and 324 are graphically represented as icons, but selecting one of these will cause a new, smaller lens to appear within lens 312 to display a graphical representation of the category selected. Fig. 3b shows a tree structure representation 390 of category or object "User InfoCatalog". Note that Fig. 3b indicates that additional objects or categories exist one hierarchical level lower than "User InfoCatalog", but their graphical representations are not currently shown on screen 310. Selection of button 340, which is analogous to button 40 described above, will cause the graphical representations of other categories one hierarchical level lower than "User InfoCatalog" to appear.

Suppose a user selects object 322, which happens to be category "Key Account Manager". With reference to Fig. 4a, the result is screen 410 showing lens 414, which is displaying a graphical representation of the category "Key Account Manager". Lens 414 is displaying a graphical representation of three objects 420, 422, and 424 that are categories one hierarchical level lower than "Key Account Manager". Objects 420, 422, and 424 are graphically represented as icons, but selecting one of these will cause a new, smaller lens to appear within lens 414 to display a graphical representation of the category selected. Fig. 4b shows a tree structure representation 490 of category or object "Key Account Manager".

Now suppose a user selects object 420, which is category "Current Situation". With reference to Fig. 5a, the result is screen 510 showing lens 516, which is displaying a graphical representation of the category "Current Situation". Lens 516 is displaying a graphical representation of three objects 520, 522, and 524 that are

categories one hierarchical level lower than "Current Situation". Objects 520, 522, and 524 are graphically represented as icons, but selecting one of these will cause a new, smaller lens to appear within lens 516 to display a graphical representation of the category selected. Fig. 5b shows a tree structure representation 590 of category or object "Current Situation".

Finally, suppose a user selects object 522, which is category "Financial Situation". With reference to Fig. 6a, the result is screen 610 showing lens 618, which is displaying a graphical representation of the category "Financial Situation". Lens 618 is displaying a graphical representation of object 620 that is a category one hierarchical level lower than "Financial Situation". Lens 618 is also displaying graphical representations of objects 630, 632, and 634, which, in a preferred embodiment, are not categories. Fig. 6b shows a tree structure representation 690 of category or object "Financial Situation".

While screen 610 is being displayed, a user may select another lens to return to a prior screen. By way of example, a user may select lens 414 (by, in a preferred embodiment, clicking in an area between the borders of lens 516 and lens 414), resulting in screen 410 being displayed.

Example of a Simplified Object Model of the Browser

A method of creating the browser described above is to use object-oriented programming. A preferred method is to use an object-oriented programming editor, such as LINGO. It is to be appreciated by those skilled in the art that C++ and other languages having object-oriented programming capabilities may be used to create the browser described above. It should also be appreciated that programming languages and tools are constantly evolving, and such programming languages and tools may also be well-suited for the creation of the browser.

To avoid confusion, the word "Object" is capitalized in the following paragraph to distinguish it from the word "object" previously defined.

Referring to Fig. 7, simplified Object Model 700 showing how the browser may be programmed is shown. Browser Manager Object 710 builds the graphic visualization of the browser and coordinates user interaction with browser controls. Data Cache Object 720 allows for temporary storage of sets of retrieved data, which are obtained through Data Interface Object 730. Lens Manager Object 740 creates a lens for displaying graphical representations of sets of data. Such sets of data may include objects that are categories, which are controlled by Category Object 750, and may include objects that are not categories, which objects are controlled by Document Object 760. Drag and Drop Object 770 allows a user to select an object and drag it to another location (this is described below). Finally, Control Object 780 could coordinate the options or commands in a popup menu to be displayed, or any other user interaction, and coordinates the taking of the proper action when an option or command is selected.

It is to be appreciated that simplified Object Model 700 is but one way to organize the implementation of the browser.

A Preferred Embodiment of the User Favorites Screen

A user may wish to retain quick access to one or more objects without having to follow the links of the hierarchical data structure of the database. In a preferred embodiment, quick access is accomplished via a user favorites screen 50 shown in Fig. 8. A preferred method for switching from screen 10 to user favorites screen 50 is by selecting cluster view icon 44 shown in Fig. 1. A preferred method for switching from user favorites screen 50 to screen 10 is by selecting icon 88. A preferred method alternative to these preferred methods is to display both screen 10 and user favorites screen 50 on one physical display simultaneously.

A link to an object may be created in user favorites screen 50 by the following preferred method. A user selects the desired object - object 32, by way of

example. The user places a mouse cursor on object 32, presses on a mouse button, drags the cursor over to cluster view icon 44, and releases the mouse button. User favorites screen 50 is then displayed, and the user moves the mouse cursor to a desired location on user favorites screen 50. The user then releases the pressed mouse button. The process of selecting an object with a mouse cursor, pressing on a mouse button, dragging the cursor to another location, and releasing the mouse button is referred to as "drag and drop". It will be appreciated that there are numerous other methods by which a link to an object may be created in user favorites screen 50.

In an alternative preferred embodiment, user favorites screen 50 may be used in conjunction with hierarchical structures without using the browser disclosed herein. For example, any hierarchical data controller that allows selection of objects within a hierarchical data structure may be used to select objects for the purpose of creating links to the selected objects in user favorites screen 50. An alternative preferred embodiment wherein a user favorites screen is used in conjunction with channels is described later.

User favorites screen 50 currently displays a graphical representation of a link to object 30. This is an example of a graphical representation of a link to an object that currently also has a graphical representation on screen 10. A user may access object 30 via screen 10 or via user favorites screen 50. Optionally, the graphical representation of an object that also has a graphical representation of a link to the object on user favorites screen 50 receives a graphical attribute indicating that the object has been selected as a user favorite object.

In a preferred embodiment, a link created on user favorites screen 50 has the same graphical representation as the pertinent object on screen 10. In an alternative preferred embodiment, the graphical representation of a link created on user favorite screen 50 inherits attributes, such as color, shading, or shape, associated with a cluster object link on user favorite screen 50. By

way of example, a user may use a mouse to drag the graphical representation of the link to object 30 over to the graphical representation of cluster object link 60 to cause the graphical representation of the link to object 30 to inherit the shading of the graphical representation of cluster object link 60.

In a preferred embodiment, a user may create hierarchies for the links graphically represented on user favorites screen 50. For example, a cluster object link, such as cluster object link 60, 62, or 64, may be created by clicking on button 86. The user may also type in a name for a cluster object link (such as the name "Workbooks" given to cluster object 60). In the example on user favorites screen 50, cluster object link 60 is one hierarchical level above object links 70, 72, 74 and is also linked to them. A link is created between a cluster object link, such as cluster object link 60, and one or more object links, such as object links 70, 72, 74, by using a mouse to drag the graphical representations of object links 70, 72,74 to the graphical representation of cluster object link 60. Optionally, a user may create, move, rename, and remove cluster object links and object links. In a preferred embodiment, moving a cursor near the graphical representation of an object link may cause a context-sensitive popup menu to appear with a list of possible commands or options to be selected. For example, moving a cursor near the graphical representation of object link 75 causes popup menu or context menu 77 to appear. Since object link 75 is a link to an executable object in the example, popup menu or context menu 77 has an "execute" option and a "preview" option, in addition to "rename" and "remove" options. In a preferred embodiment, a popup menu or context menu is predefined and is automatically displayed in a format based on the object type and user authorization. For example, "remove" appears as an option only when there is authorization to remove the object link. Optionally, a popup menu or context menu could be hierarchical.

In a preferred embodiment, user favorites screen 50 may be saved (e.g., stored on a server and/or on computer-readable storage media) in such a way that the user-specified relative positions of the graphical representations of object links and cluster object links as they appear on user favorites screen 50 are also saved.

In a preferred embodiment, user favorites screen 50 may be used in conjunction with exception reporting. For example, an administrator may define an object to be associated with exception reporting, wherein certain conditions associated with an object will cause a user to be alerted to the occurrence of an exception. For example, a user may be alerted by having the graphical representation of an object link flash.

In a preferred embodiment, user favorites screen 50 is capable of displaying different shelves of graphical representations of links. In Fig. 8, user favorites screen 50 is currently displaying a first shelf 81. By selecting button 82 or button 84, a user may select other shelves that display graphical representations of other links. By pressing button 80, a user may then return to displaying shelf 81. In a preferred embodiment, the graphical representation of an object link may be moved from one shelf to another shelf. A user may also move the graphical representation of a cluster object link from one shelf to another shelf, in which case the graphic representations of all the object links associated with that cluster object link also move to the other shelf. In another preferred embodiment, the shelf with the graphical representation of an object link associated with exception reporting may flash when alerting a user of an exception event.

Since the hierarchical organization of links whose graphical representations are displayed on user favorites screen 50 is displayed in a manner of natural association, the hierarchical nature of the links is not as readily perceived by a user. Since a user may create a cluster object link, such as cluster object link 60, and cause object links, such as object links 70, 72, and 74 to be linked to the cluster object link, a hierarchical

relationship is created that is displayed as a collection of proximate graphical representations of cluster object links and object links on the screen. The examples below illustrate some of the advantages of natural association.

Examples of User Favorite Screens

The following examples illustrate the versatility of user favorites screens.

Referring to Fig. 9a, user favorites screen 950 currently displays graphical representations of cluster object links 960, 962, 964. Object link 970 is linked to cluster object link 960 (the display of popup menu or context menu 977 indicates that a cursor is near the graphical representation of object link 970). Object links 972, 973 are linked to cluster object link 962, while object links 974, 975, 976 are linked to cluster object link 964. Fig. 9b shows a tree structure representation 990 of the cluster object links and object links whose graphical representations are displayed on user favorites screen 950. The graphical representations make the natural associations between the various groups of links apparent in a way that is not possible with a tree structure. The physical layout of the links, which may be defined by a user, obviates the need to display the hierarchical structure of the links.

In a preferred embodiment, if a user were to move the graphical representation of object link 974 by, for example, moving a cursor to the graphical representation of object link 974, pressing a mouse button, dragging the graphical representation of object link 974 over to touch the graphical representation of cluster object link 962, then dragging the graphical representation of object link 974 to a point on user favorites screen 950 that is just above the graphical representation of object link 972, and releasing the mouse button, the result would look like user favorites screen 1050 in Fig. 10a. This operation would remove the link between object link 974 and cluster object link 964, and create a link between object link 974 and cluster object link 962. Fig. 10b shows a tree structure representation 1090 of the cluster object links and object

links whose graphical representations are displayed on user favorites screen 1050.

In a preferred embodiment, an attribute of the graphical representation of object link 974 would be altered by the operation described above. By way of example only, assume that the graphical representation of cluster object link 962 is shaded red while the graphical representation of cluster object link 964 is shaded green. The graphical representation of object link 974 would be shaded green in user favorites screen 950, but would automatically switch to a shade of red in user favorites screen 1050.

To further illustrate the versatility of the user favorites screen, user favorites screen 1150 in Fig. 11a shows a different physical layout for the graphical representations of the links that are graphically represented in user favorites screen 1050. Fig. 11b shows a tree structure representation 1190 of the cluster object links and object links whose graphical representations are displayed on user favorites screen 1150. It is apparent that tree structure representation 1090 and tree structure representation 1190 are the same. When a user drags and drops the graphical representation of a cluster object link, the graphical representations of any object links that are linked to that cluster object link are also moved. For example, moving the graphical representation of cluster object link 960 from the top half of user favorites screen 1050 to the bottom half of user favorite screen 1150 causes the graphical representation of object link 970 to move as well. Furthermore, the drag and drop technique may also be used to move a graphical representation of an object link within the vicinity of a cluster object link. Thus, it is a simple matter to move the graphical representations of object links 975, 976 from the right of the graphical representation of cluster object link 964 (as in user favorites screen 1050) to the left of the graphical representation of cluster object link 964 (as in user favorites screen 1150). In a preferred embodiment, this operation would require two drag and drop operations. For

example, a user may drag and drop the graphical representation of object link 975 and subsequently drag and drop the graphical representation of object link 976. A further advantage of being able to customize the layout of the physical representation of objects is that a user, if it is so desired, may create a pseudo-hierarchy of object links simply by indenting graphical representations of certain object links relative to graphical representations of other object links.

Another Preferred Embodiment of the User Favorites Screen

An alternative embodiment of the user favorites screen is illustrated by user favorites screen 1250 in Fig. 12. In this preferred embodiment, cluster object link 1260 is linked to object links 1270, 1271, 1272, 1273, 1274. The graphical representations of these links share the same shape, which in user favorites screen 1250 is a triangular shape. The graphical representations of other cluster object links, and the graphical representations of object links linked to them, have other shapes. For example, the graphical representation of cluster object link 1262 has a diamond shape, and the graphical representation of cluster object link 1264 has a circular shape.

In a preferred embodiment, user favorites screen 1250 is used in conjunction with channels 1295, which supply sets of data. Links to objects in public or semi-private channels may be created and organized utilizing a user favorites channel. A user may use icon 1296 to display a user favorites channel. For example, if a user clicks on icon 1296, options to display user favorites channel 1, user favorites channel 2, or user favorites channel 3 appear (these options are analogous to buttons 80, 82, 84 used to select shelves for display in Fig. 8).

A user may create a link to an object in another channel by dragging the object to icon 1296. As with user favorites screen 50, once an object link has been created and is graphically represented in user favorites screen 1250, it may be linked to a cluster object link. It is to be appreciated that the operations that may be performed on

user favorites screen 50 as described above may also be performed on user favorites screen 1250.

Example of a Simplified Object Model of the User Favorites Screen

A method of creating the user favorites screen described above is to use object-oriented programming. A preferred method is to use an object-oriented programming editor, such as LINGO. It is to be appreciated by those skilled in the art that C++ and other languages having object-oriented programming capabilities may be used to create the cluster favorites screen described above. It should also be appreciated that programming languages and tools are constantly evolving, and such programming languages and tools may also be well-suited for the creation of the cluster favorites screen.

To avoid confusion, the word "Object" is capitalized in the following paragraph to distinguish it from the word "object" previously defined.

Referring to Fig. 13, simplified Object Model 1300 showing how the user favorites screen may be programmed is shown. Shelf Manager Object 1310 controls all instances of cluster objects and objects associated with the currently displayed shelf or user favorites channel. Stage manager object 1315 controls all the visible action on the screen and coordinates event and information streams between the acting controls. Data Cache Object 1320 allows for temporary storage of sets of retrieved data, which are obtained through Data Interface Object 1330. Cluster Object 1350 is an instance of a cluster object link, while Document Object 1360 is an instance of an object link that is not linked to a cluster object link. Document Object 1365 is an instance of an object link that is linked to a cluster object link. Drag and Drop Object 1370 allows a user to select an object or cluster object and drag it to another location. Finally, Control Object 1380 coordinates the options or commands in a popup menu to be displayed, or any other user interaction, and coordinates the taking of the proper action when an option or command is selected.

It is to be appreciated that simplified Object Model 1300 is but one way to organize the implementation of the user favorites screen.

Alternative Preferred Embodiments

In a preferred embodiment, the sets of data described above reside across a network of computers. The sets of data that reside on the Internet, or any sets of data that reside in more than one database, may be considered parts of one large database. In an alternative preferred embodiment, sets of data may reside on one computer. It will be appreciated that the invention may be practiced on one computer or over a network of computers. It will also be appreciated that a computer may receive sets of data from its own storage medium or cache, and is not limited to receiving data from other computers.

While there have been shown and described and pointed out fundamental novel features of the invention as applied to embodiments thereof, it will be understood that various omissions and substitutions and changes in the form and details of the invention, as herein disclosed, may be made by those skilled in the art without departing from the spirit of the invention. It is expressly intended that all combinations of those elements and/or method steps which perform substantially the same function in substantially the same way to achieve the same results are within the scope of the invention. It is the intention, therefore, to be limited only as indicated by the scope of the claims appended hereto.

CLAIMS

1. A method of displaying and interacting with links to data, which data is organized in a hierarchical data structure, the method comprising the steps of:

(a) sending a first request from a graphical user interface on a computer to provide a first set of data, said first set of data being in said hierarchical data structure and having a position in said hierarchical data structure and having a level associated with said position of first set of data in said hierarchical data structure;

(b) receiving in said computer said first set of data in response to said first request;

(c) converting said first set of data to a graphical representation for visual display;

(d) displaying said graphical representation for visual display of said first set of data in a first lens displayed by said graphical user interface;

(e) sending a second request from said graphical user interface to provide a second set of data, said second set of data being in said hierarchical data structure and having a position in said hierarchical data structure and having a level associated with said position of second set of data in said hierarchical data structure;

(f) receiving in said computer said second set of data in response to said second request;

(g) comparing said level of said second set of data with said level of said first set of data to determine which level is deeper within said hierarchical data structure;

(h) converting said second set of data to a graphical representation for visual display based upon said comparing said level of said second set of data with said level of said first set of data; and

(i) displaying said graphical representation for visual display of said second set of data in a second lens displayed by said graphical user interface stacked within and on top of said first lens only when said level of said second set of data is deeper than said level of said first set of data in said hierarchical data structure.

2. The method of displaying and interacting with links to data of claim 1, wherein said first lens and said second lens are substantially hexagonal in shape.

3. The method of displaying and interacting with links to data of claim 1, wherein at least one of said first set of data and second set of data contains a representation of a three-dimensional animation.

4. The method of displaying and interacting with links to data of claim 1, wherein at least one of said first set of data and second set of data contains a representation of a virtual reality world.

5. The method of displaying and interacting with links to data of claim 1, wherein:

at least one of said graphical representation for visual display of said first set of data and said graphical representation for visual display of said second set of data contains a graphical representation for visual display of a category;

said category has a position in said hierarchical data structure and has a level associated with said position of said category; and

said graphical representation for visual display of said category indicates whether a third set of data having a position at a deeper level than said level associated with said position of said category exists.

6. A method of displaying and interacting with links to data in an R/3 system, which data is organized in a hierarchical data structure, the method comprising the steps of:

(a) sending a first request from an R/3 graphical user interface on a computer to provide a first set of data, said first set of data being in said hierarchical data structure and having a position in said hierarchical data structure and having a level associated with said position of first set of data in said hierarchical data structure;

(b) receiving in said computer said first set of data in response to said first request;

(c) converting said first set of data to a graphical representation for visual display;

(d) displaying said graphical representation for visual display of said first set of data in a first lens displayed by said R/3 graphical user interface;

(e) sending a second request from said R/3 graphical user interface to provide a second set of data, said second set of data being in said hierarchical data structure and having a position in said hierarchical data structure and having a level associated with said position of second set of data in said hierarchical data structure;

(f) receiving in said computer said second set of data in response to said second request;

(g) comparing said level of said second set of data with said level of said first set of data to determine which level is deeper within said hierarchical data structure;

(h) converting said second set of data to a graphical representation for visual display based upon said comparing said level of said second set of data with said level of said first set of data; and

(i) displaying said graphical representation for visual display of said second set of data in a second lens displayed by said R/3 graphical user interface stacked within and on top of said first lens only when said level of said second set of data is deeper than said level of said first set of data in said hierarchical data structure.

7. A computer system comprising a display, a central processor, a graphics processor, a memory, an input device, and a graphical user interface, wherein:

said memory has a hierarchical data structure, said hierarchical data structure having a first set of data and a second set of data, said first set of data having a position in said hierarchical data structure and having a level associated with said position of said first set of data in said hierarchical data structure, said second set of data having a position in said hierarchical data structure and having a level associated with said position

of said second set of data in said hierarchical data structure;

said graphical user interface requests said first set of data;

said graphics processor receives said first set of data in response to said request for said first set of data;

said graphics processor converts said first set of data to a graphical representation for visual display;

said graphical user interface displays said graphical representation for visual display of said first set of data in a first lens on said display;

said graphical user interface requests said second set of data;

said graphical processor receives said second set of data in response to said request for said second set of data;

said central processor compares said level of said second set of data with said level of said first set of data to determine which level is deeper within said hierarchical data structure;

said graphics processor converts said second set of data to a graphical representation for visual display based upon said comparing said level of said second set of data with said level of said first set of data; and

said graphical user interface displays said graphical representation for visual display of said second set of data in a second lens stacked within and on top of said first lens on said display only when said level of said second set of data is deeper than said level of said first set of data.

8. The computer system of claim 7, wherein said first lens and said second lens are substantially hexagonal in shape.

9. The computer system of claim 7, wherein said central processor and said graphics processor are the same processor.

10. The computer system of claim 7, wherein at least one of said first set of data and second set of data contains a representation of a three-dimensional animation.

11. The computer system of claim 7, wherein at least one of said first set of data and second set of data contains a representation of a virtual reality world.

12. The computer system of claim 7, wherein:

at least one of said graphical representation for visual display of said first set of data and said graphical representation for visual display of said second set of data contains a graphical representation for visual display of a category;

said category has a position in said hierarchical data structure and has a level associated with said position of said category; and

said graphical representation for visual display of said category indicates whether a third set of data having a position at a deeper level than said level associated with said position of said category exists.

13. The computer system of claim 7, wherein said graphical user interface operates in conjunction with an instance of SAP Business Information Warehouse.

14. The computer system of claim 7, wherein said graphical user interface displays a context-sensitive pop-up menu, based on attributes of said second set of data and based on an authorization level.

15. The computer system of claim 7, wherein said graphical user interface is imbedded in a container.

16. A method of displaying data organized in a hierarchical data structure, the method comprising displaying a plurality of lenses, each lens having a size, wherein:

each said lens is associated with a set of data, each said set of data being in said hierarchical data structure and each having a position in said hierarchical data structure and each having a level associated with said position of said set of data in said hierarchical data structure; and

said size of each said lens being dependent on said level of said set of data associated with said lens, wherein a lens associated with a set of data having a deeper level has a smaller size than a lens associated with a set of data having a shallower level.

17. The method of displaying data organized in a hierarchical data structure of claim 16, wherein each said lens has a label.

18. The method of displaying data organized in a hierarchical data structure of claim 16, further comprising selecting any one of said plurality of lenses using a graphical user interface, displaying said any one of said plurality of lenses, and displaying said set of data associated with said any one of said plurality of lenses.

19. The method of displaying data organized in a hierarchical data structure of claim 16, wherein at least one of said plurality of lenses is associated with data containing a representation of a three-dimensional animation.

20. The method of displaying data organized in a hierarchical data structure of claim 16, wherein at least one of said plurality of lenses is associated with data contains a representation of a virtual reality world.

21. The method of displaying data organized in a hierarchical data structure of claim 16, wherein each said lens is substantially hexagonal in shape.

22. A computer system for displaying and interacting with data organized in a hierarchical data structure, which comprises:

(a) means for sending a first request from a graphical user interface on a computer to provide a first set of data, said first set of data being in said hierarchical data structure and having a position in said hierarchical data structure and having a level associated with said position of first set of data in said hierarchical data structure;

(b) means for receiving in said computer said first set of data in response to said first request;

(c) means for converting said first set of data to a graphical representation for visual display;

(d) means for displaying said graphical representation for visual display of said first set of data in a first lens displayed by said graphical user interface;

(e) means for sending a second request from said graphical user interface to provide a second set of data, said second set of data being in said hierarchical data structure and having a position in said hierarchical data structure and having a level associated with said position of second set of data in said hierarchical data structure;

(f) means for receiving in said computer said second set of data in response to said second request;

(g) means for comparing said level of said second set of data with said level of said first set of data to determine which level is deeper within said hierarchical data structure;

(h) means for converting said second set of data to a graphical representation for visual display based upon said comparing said level of said second set of data with said level of said first set of data; and

(i) means for displaying said graphical representation for visual display of said second set of data in a second lens displayed by said graphical user interface stacked within and on top of said first lens only when said level of said second set of data is deeper than said level of said first set of data in said hierarchical data structure.

23. The computer system of claim 22, wherein said first lens and said second lens are substantially hexagonal in shape.

24. The computer system of claim 22, wherein at least one of said first set of data and second set of data contains a representation of a three-dimensional animation.

25. The computer system of claim 22, wherein at least one of said first set of data and second set of data contains a representation of a virtual reality world.

26. The computer system of claim 22, wherein:

at least one of said graphical representation for visual display of said first set of data and said graphical representation for visual display of said second set of data contains a graphical representation for visual display of a category;

said category has a position in said hierarchical data structure and has a level associated with said position of said category; and

said graphical representation for visual display of said category indicates whether a third set of data having a position at a deeper level than said level associated with said position of said category exists.

27. The computer system of claim 22, wherein said graphical user interface operates in conjunction with an instance of SAP Business Information Warehouse.

28. The computer system of claim 22, wherein said graphical user interface displays a context-sensitive pop-up menu, based on attributes of said second set of data and based on an authorization level.

29. The computer system of claim 22, wherein said graphical user interface is imbedded in a container.

30. A method of displaying and interacting with links to data, which data is organized in a first hierarchical data structure, the method comprising the steps of:

(a) creating a link to a set of data in said first hierarchical data structure;

(b) placing said link into a second hierarchical data structure; and

(c) converting said link to a graphical representation for visual display, said graphical representation for visual display of said link being based upon position of said link in said second hierarchical data structure.

31. The method of displaying and interacting with links to data of claim 30, wherein said graphical representation for visual display of said link inherits attributes of a graphical representation for visual display

of at least one other link in said second hierarchical data structure.

32. The method of displaying and interacting with links to data of claim 30, wherein said set of data contains a representation of a three-dimensional animation.

33. The method of displaying and interacting with links to data of claim 30, wherein said set of data contains a representation of a virtual reality world.

34. The method of displaying and interacting with links to data of claim 30, further comprising the step of storing the position of said graphical representation for visual display of said link, relative to positions of graphical representations for visual display of other links, on computer-readable storage media.

35. The method of displaying and interacting with links to data of claim 30, further comprising the step of displaying a context-sensitive pop-up menu, based on type of said set of data and based on an authorization level, when a cursor is placed near said graphical representation for visual display of said link.

36. The method of displaying and interacting with links to data of claim 30, further comprising the step of obtaining said first set of data in said first hierarchical data structure through a channel.

37. The method of displaying and interacting with links to data of claim 30, further comprising the step of reporting an exception associated with said set of data in said first hierarchical data structure by altering said graphical representation for visual display of said link.

38. A computer system comprising a display, a central processor, a graphics processor, a memory, an input device, and a graphical user interface, wherein:

said memory has a first hierarchical data structure and a second hierarchical data structure, said first hierarchical data structure having a set of data, said set of data having a position in said first hierarchical data structure and having a level associated with said position of said set of data in said first hierarchical data structure;

said graphical user interface requests said set of data;

said graphics processor receives said set of data in response to said requesting said set of data;

said graphics processor converts said set of data to a graphical representation for visual display;

said central processor creates a link to a subset of said set of data;

said central processor places said link into said second hierarchical data structure; and

said graphics processor converts said link to a graphical representation for visual display, said graphical representation for visual display of said link being based upon position of said link in said second hierarchical data structure.

39. The computer system of claim 38, wherein said graphical representation for visual display of said link inherits attributes of a graphical representation for visual display of at least one other link in said second hierarchical data structure.

40. The computer system of claim 38, wherein said central processor and said graphics processor are the same processor.

41. The computer system of claim 38, wherein said set of data contains a representation of a three-dimensional animation.

42. The computer system of claim 38, wherein said set of data contains a representation of a virtual reality world.

43. The computer system of claim 38, wherein the position of said graphical representation for visual display of said link, relative to positions of graphical representations for visual display of other links, is stored on computer-readable storage media.

44. The computer system of claim 38, wherein said graphics processor creates a context-sensitive pop-up menu, based on type of said subset of said set of data and based on an authorization level, when a cursor is placed

near said graphical representation for visual display of said link.

45. The computer system of claim 38, wherein said graphics processor receives said set of data through a channel.

46. The computer system of claim 38, wherein said central processor reports an exception associated with said subset of said set of data by having said graphics processor alter said graphical representation for visual display of said link.

47. A method of displaying and linking data organized in a first hierarchical data structure, the method comprising the steps of:

(a) displaying a set of data in said first hierarchical data structure;

(b) selecting a subset of said set of data;

(c) creating a link to said subset of said set of data;

(d) placing said link into a second hierarchical data structure; and

(e) converting said link to a graphical representation for visual display, said graphical representation for visual display of said link being based upon position of said link in said second hierarchical data structure.

48. The method of displaying and linking data organized in a first hierarchical data structure of claim 47, wherein said set of data contains a representation of a three-dimensional animation.

49. The method of displaying and linking data organized in a first hierarchical data structure of claim 47, wherein said set of data contains a representation of a virtual reality world.

50. The method of displaying and linking data organized in a first hierarchical data structure of claim 47, further comprising the step of storing the position of said graphical representation for visual display of said link, relative to positions of graphical representations

for visual display of other links, on computer-readable storage media.

51. The method of displaying and linking data organized in a first hierarchical data structure of claim 47, further comprising the step of displaying a context-sensitive pop-up menu, based on type of said subset of said set of data and based on an authorization level, when a cursor is placed near said graphical representation for visual display of said link.

52. The method of displaying and linking data organized in a first hierarchical data structure of claim 47, further comprising the step of obtaining said set of data in said first hierarchical data structure through a channel.

53. The method of displaying and linking data organized in a first hierarchical data structure of claim 47, further comprising the step of reporting an exception associated with said subset of said set of data by altering said graphical representation for visual display of said link.

54. A computer system for displaying and interacting with links to data, which data is organized in a first hierarchical data structure, which comprises:

(a) means for creating a link to a set of data in said first hierarchical data structure;

(b) means for placing said link into a second hierarchical data structure; and

(c) means for converting said link to a graphical representation for visual display, said graphical representation for visual display of said link being based upon position of said link in said second hierarchical data structure.

55. The computer system of claim 54, wherein said graphical representation for visual display of said link inherits attributes of a graphical representation for visual display of at least one other link in said second hierarchical data structure.

56. The computer system of claim 54, wherein said set of data contains a representation of a three-dimensional animation.

57. The computer system of claim 54, wherein said set of data contains a representation of a virtual reality world.

58. The computer system of claim 54, further comprising means for storing the position of said graphical representation for visual display of said link, relative to positions of graphical representations for visual display of other links, on computer-readable storage media.

59. The computer system of claim 54, further comprising means for displaying a context-sensitive pop-up menu, based on type of said set of data and based on an authorization level, when a cursor is placed near said graphical representation for visual display of said link.

60. The computer system of claim 54, further comprising means for obtaining said set of data in said first hierarchical data structure through a channel.

61. The computer system of claim 54, further comprising means for reporting an exception associated with said set of data by altering said graphical representation for visual display of said link.

62. A method of displaying and interacting with links to data in an R/3 system, which data is organized in a first hierarchical data structure, the method comprising the steps of:

(a) sending a first request from an R/3 graphical user interface on a computer to provide a first set of data, said first set of data being in said first hierarchical data structure and having a position in said first hierarchical data structure and having a level associated with said position of first set of data in said first hierarchical data structure;

(b) receiving in said computer said first set of data in response to said first request;

(c) converting said first set of data to a graphical representation for visual display;

(d) displaying said graphical representation for visual display of said first set of data in a first lens displayed by said R/3 graphical user interface;

(e) creating a link to a subset of said set of data;

(f) placing said link into a second hierarchical data structure; and

(g) converting said link to a graphical representation for visual display in a user favorites screen, said graphical representation for visual display in a user favorites screen of said link being based upon position of said link in said second hierarchical data structure.

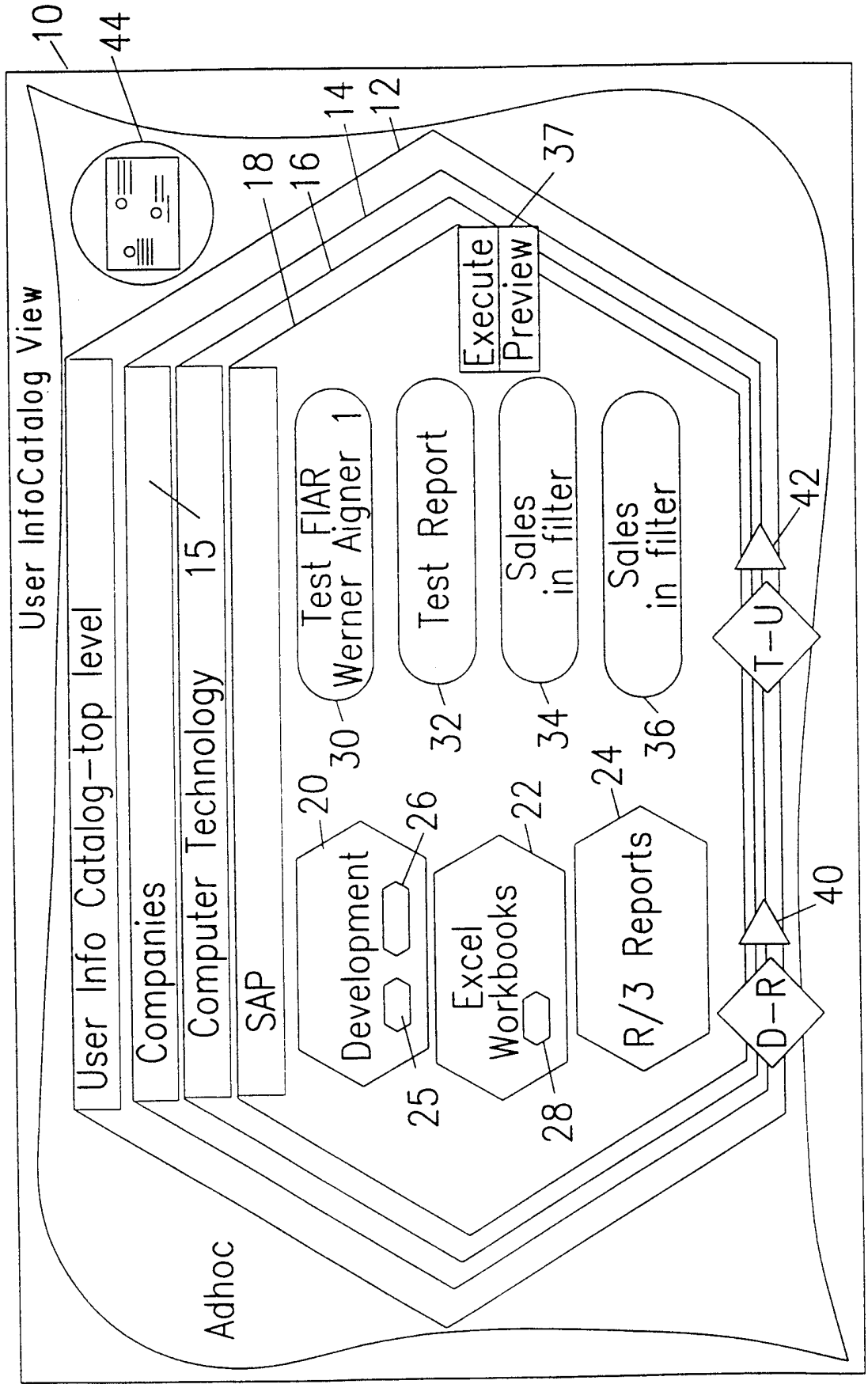

10
44
User InfoCatalog View
User Info Catalog-top level
Companies
Computer Technology 15
SAP
Adhoc
18
16
14
12
37
Execute
Preview
Test FIAR
Werner Aigner 1
30
Test Report
32
Sales
in filter
34
Sales
in filter
36
20
Development
26
25
22
Excel
Workbooks
28
24
R/3 Reports
T-U
42
D-R
40

Fig. 1

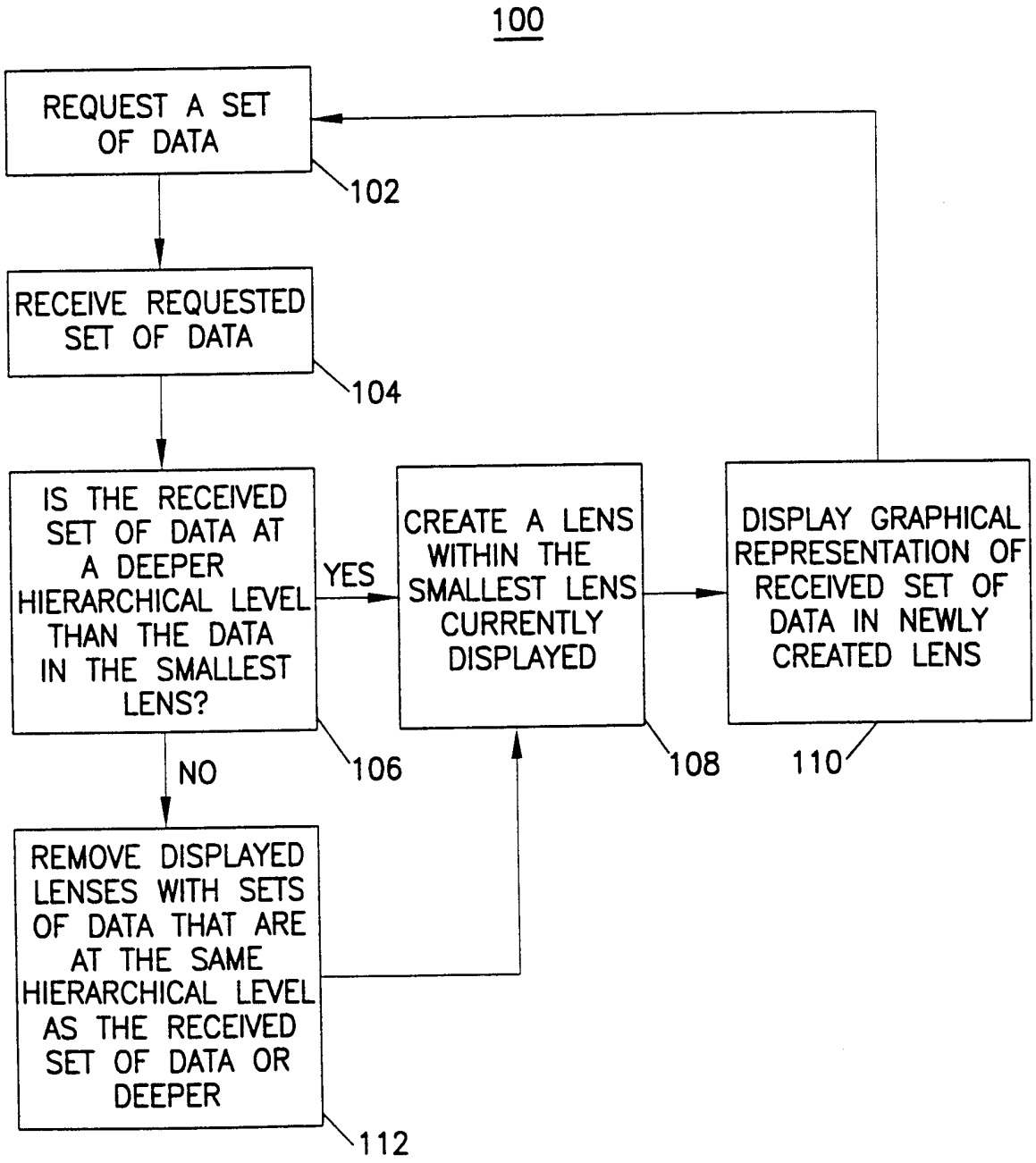

100
REQUEST A SET OF DATA
102
RECEIVE REQUESTED SET OF DATA
104
IS THE RECEIVED SET OF DATA AT A DEEPER HIERARCHICAL LEVEL THAN THE DATA IN THE SMALLEST LENS?
YES
NO
106
CREATE A LENS WITHIN THE SMALLEST LENS CURRENTLY DISPLAYED
108
DISPLAY GRAPHICAL REPRESENTATION OF RECEIVED SET OF DATA IN NEWLY CREATED LENS
110
REMOVE DISPLAYED LENSES WITH SETS OF DATA THAT ARE AT THE SAME HIERARCHICAL LEVEL AS THE RECEIVED SET OF DATA OR DEEPER
112

Fig. 2

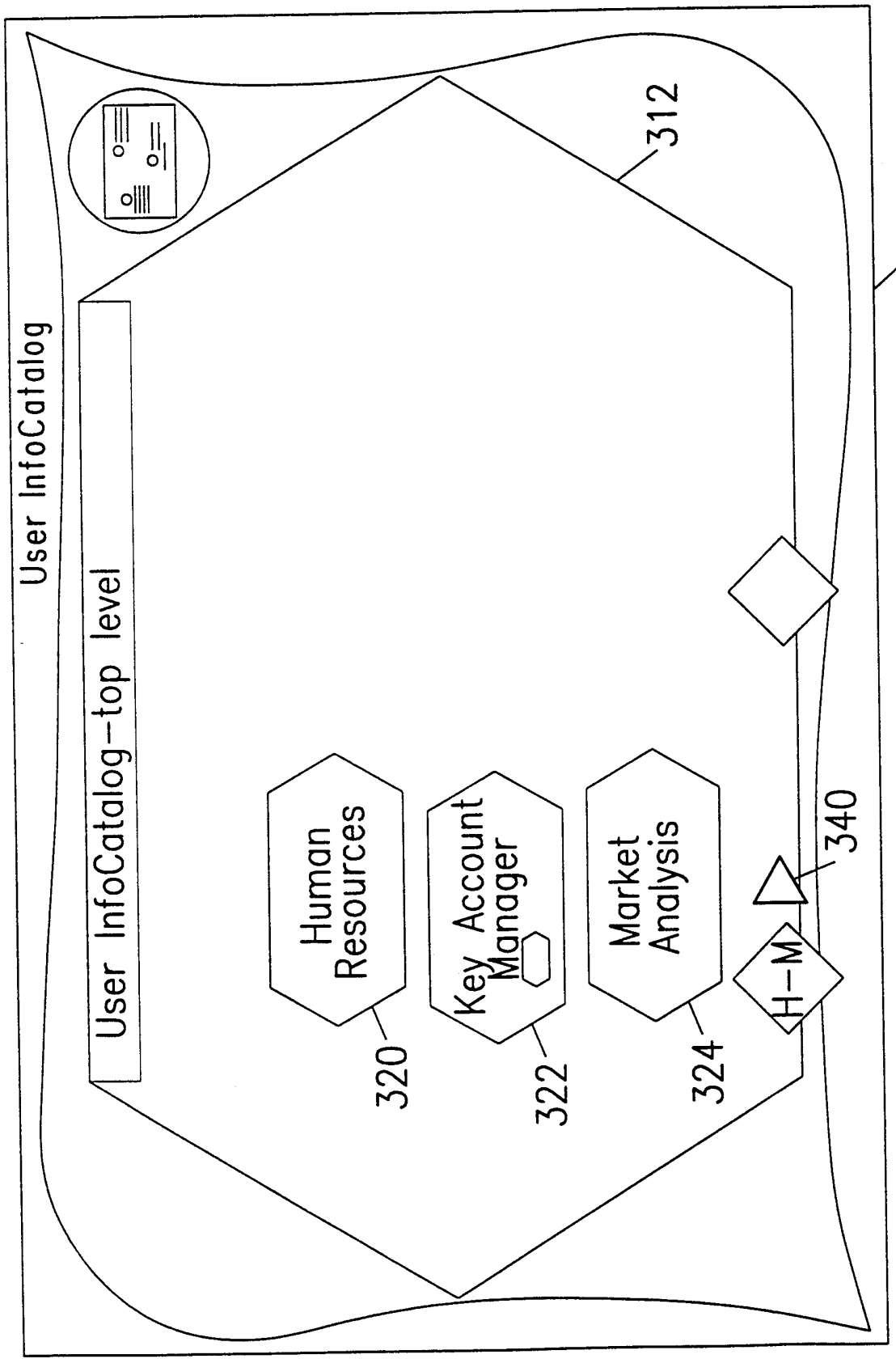

User InfoCatalog
User InfoCatalog–top level
312
310
Human Resources
320
Key Account Manager
322
Market Analysis
324
H–M
340

Fig. 3a

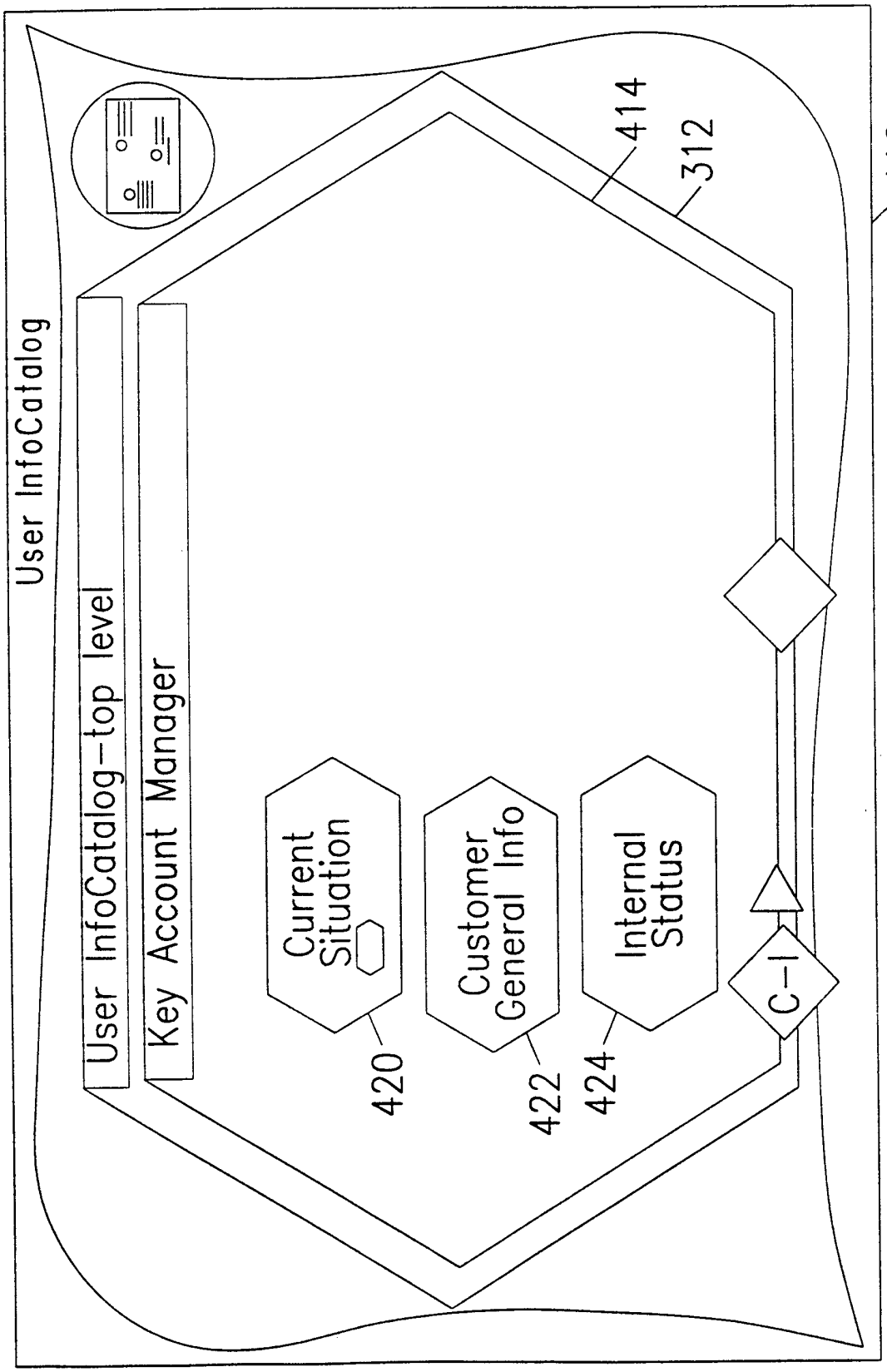

User InfoCatalog
User InfoCatalog–top level
Key Account Manager
Current Situation
Customer General Info
Internal Status
C–I
420
422
424
414
312
410

Fig. 4a

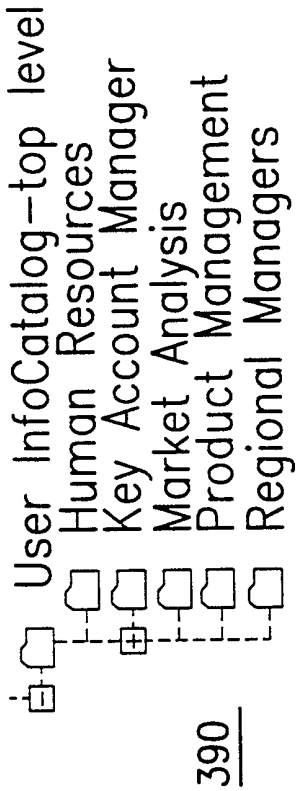

User InfoCatalog–top level
Human Resources
Key Account Manager
Market Analysis
Product Management
Regional Managers
390

Fig. 3b

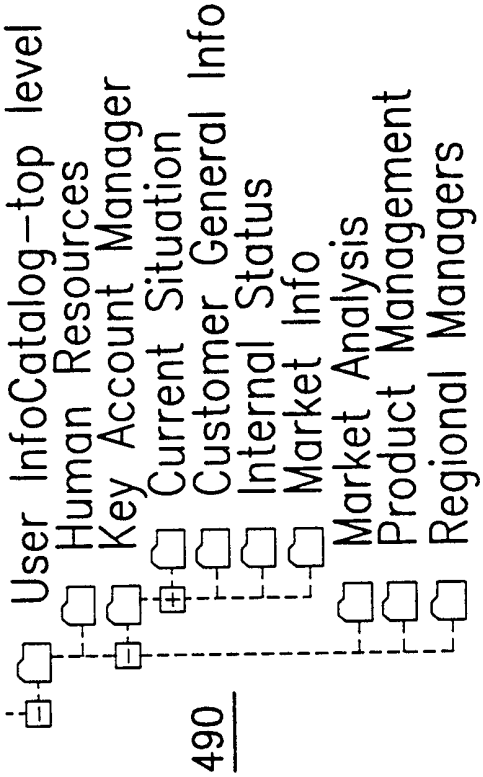

User InfoCatalog–top level
Human Resources
Key Account Manager
Current Situation
Customer General Info
Internal Status
Market Info
Market Analysis
Product Management
Regional Managers
490

Fig. 4b

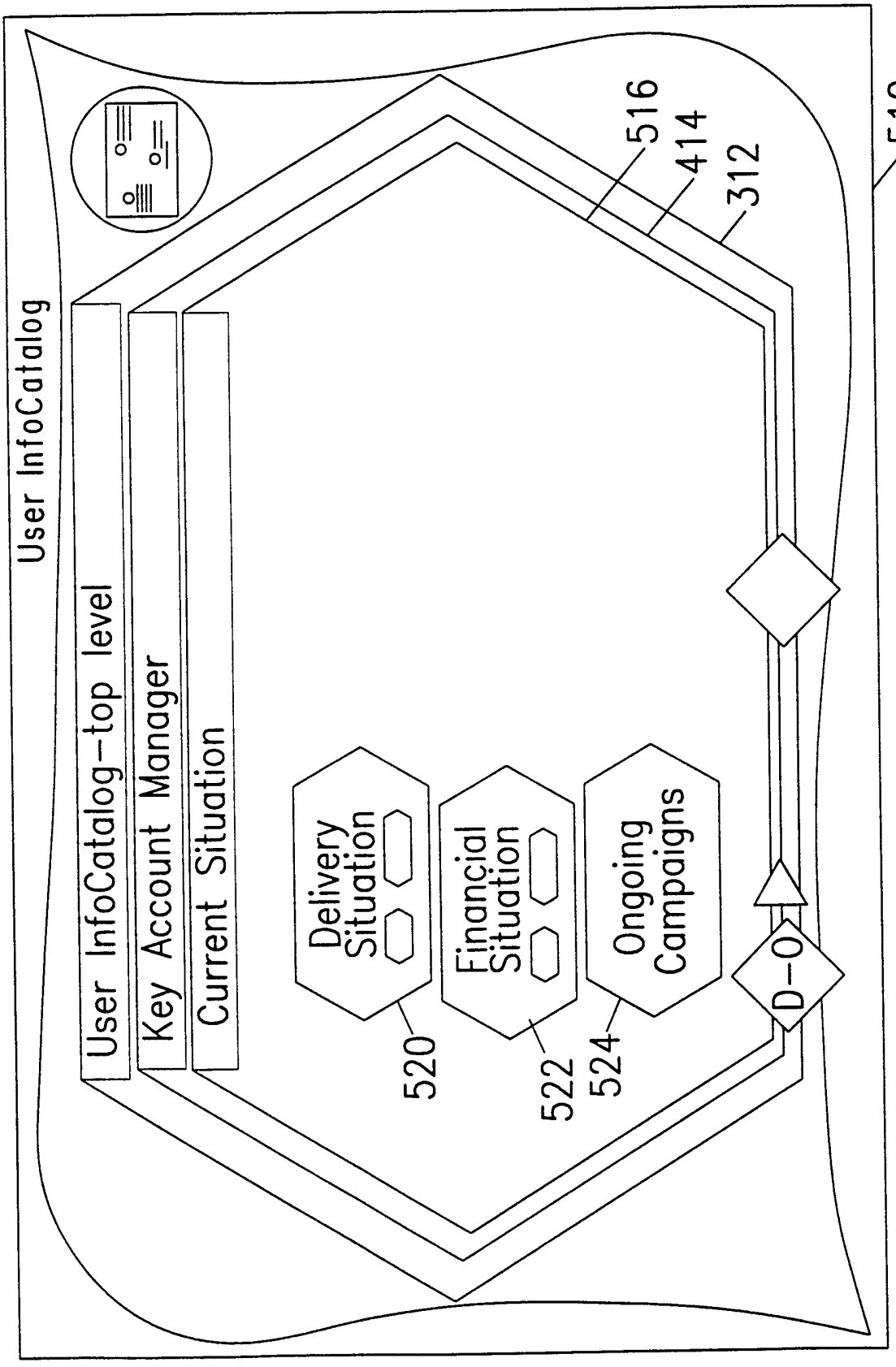

User InfoCatalog
User InfoCatalog–top level
Key Account Manager
Current Situation
Delivery Situation
Financial Situation
Ongoing Campaigns
D–O
516
414
312
510
520
522
524

Fig. 5a

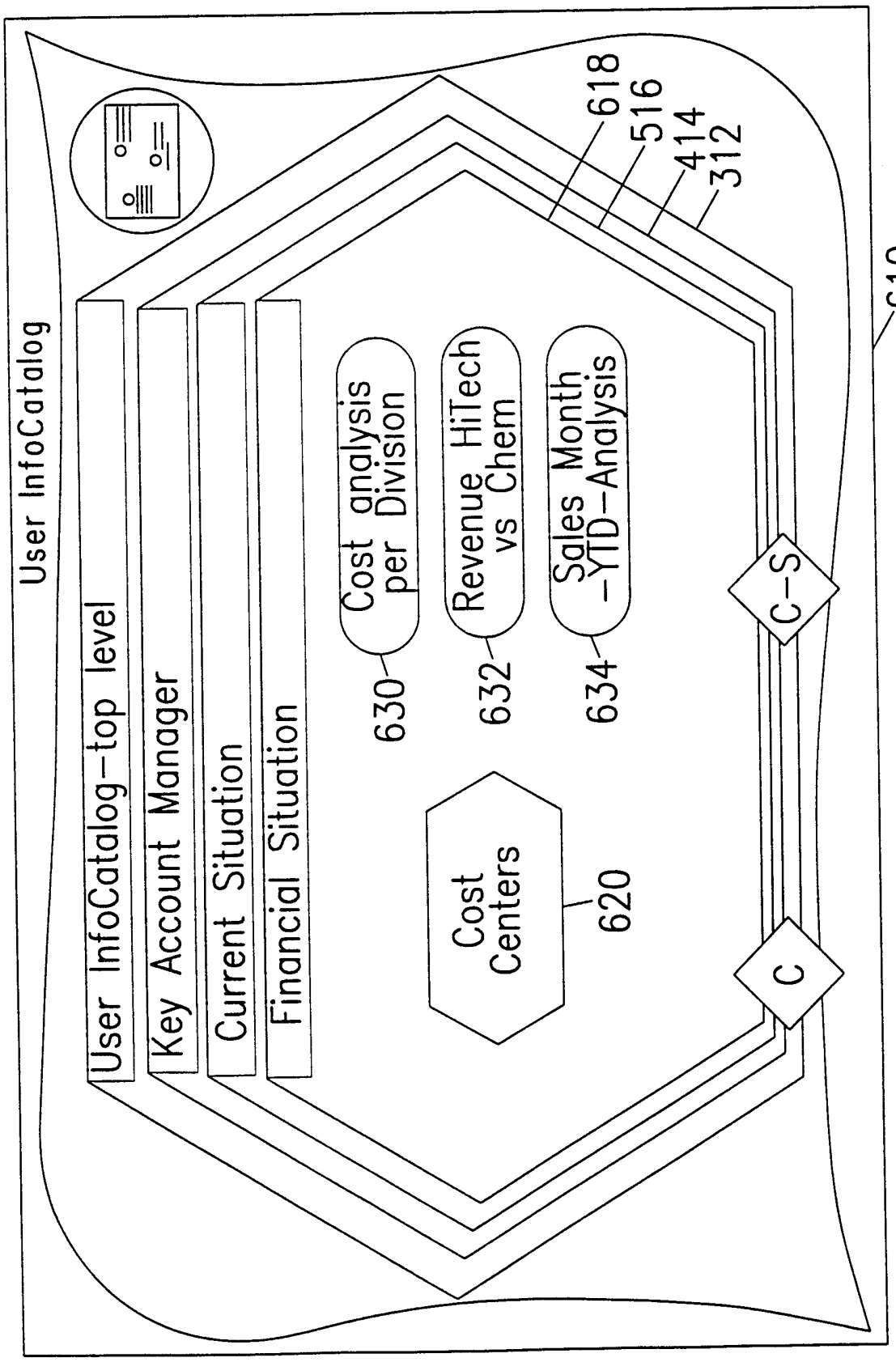

User InfoCatalog
User InfoCatalog–top level
Key Account Manager
Current Situation
Financial Situation
618
516
414
312
610
Cost analysis per Division
630
Revenue HiTech vs Chem
632
Sales Month –YTD–Analysis
634
Cost Centers
620
C–S
C

Fig. 6a

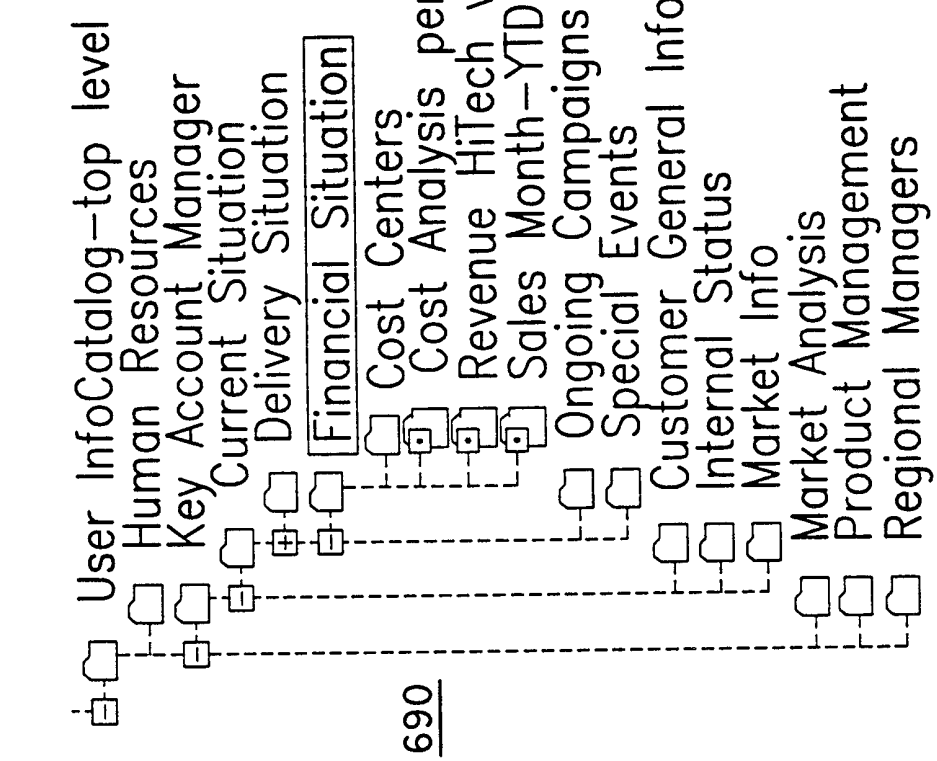

User InfoCatalog–top level
Human Resources
Key Account Manager
Current Situation
Delivery Situation
Financial Situation
Ongoing Campaigns
Special Events
Customer General Info
Internal Status
Market Info
Market Analysis
Product Management
Regional Managers
690
User InfoCatalog–top level
Human Resources
Key Account Manager
Current Situation
Delivery Situation
Financial Situation
Cost Centers
Cost Analysis per Division
Revenue HiTech vs Chem
Sales Month–YTD–Analysis
Ongoing Campaigns
Special Events
Customer General Info
Internal Status
Market Info
Market Analysis
Product Management
Regional Managers

Fig. 6b

Fig. 5b

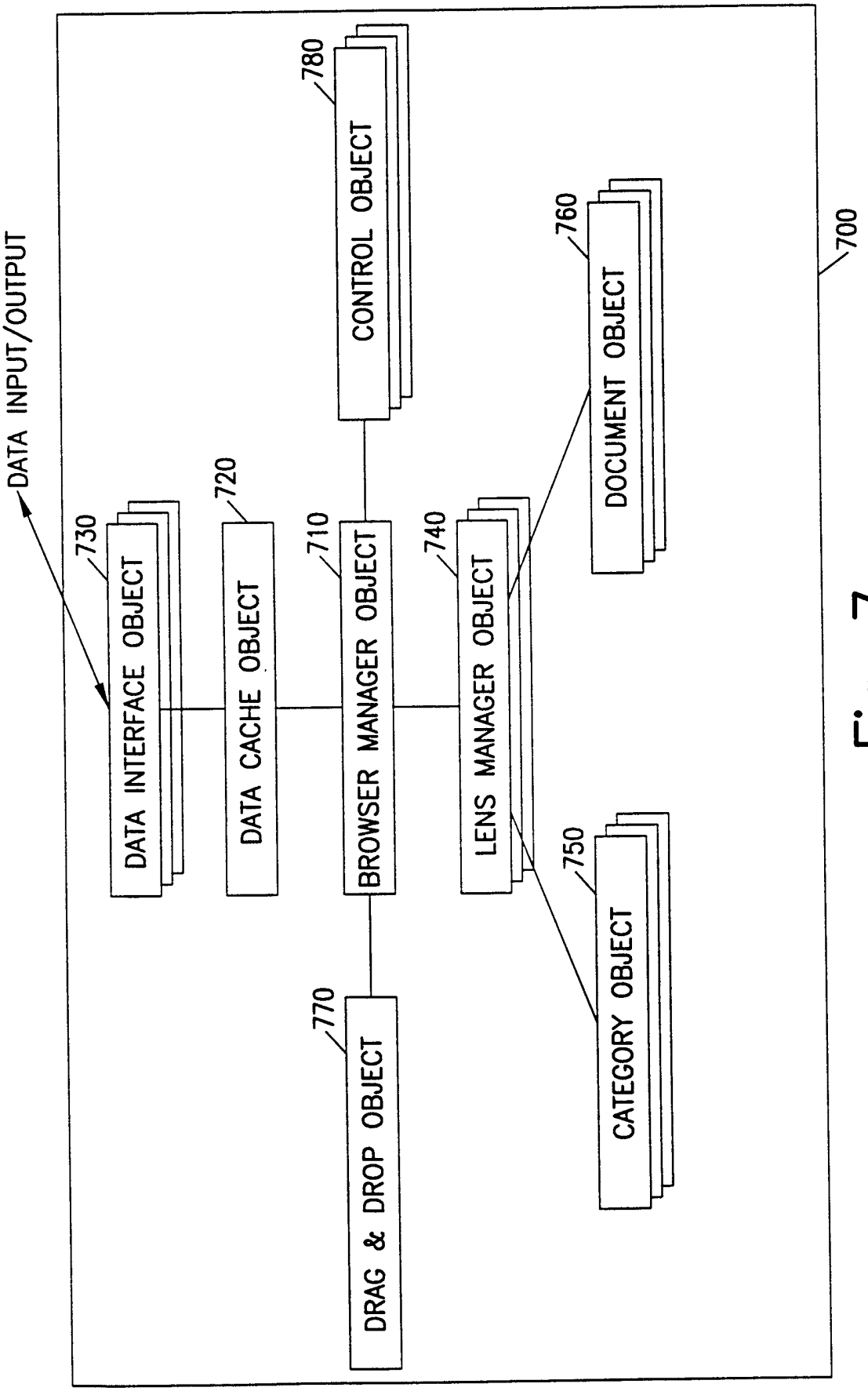

DATA INPUT/OUTPUT
700
780
CONTROL OBJECT
760
DOCUMENT OBJECT
720
730
710
740
DATA INTERFACE OBJECT
DATA CACHE OBJECT
BROWSER MANAGER OBJECT
LENS MANAGER OBJECT
750
CATEGORY OBJECT
770
DRAG & DROP OBJECT

Fig. 7

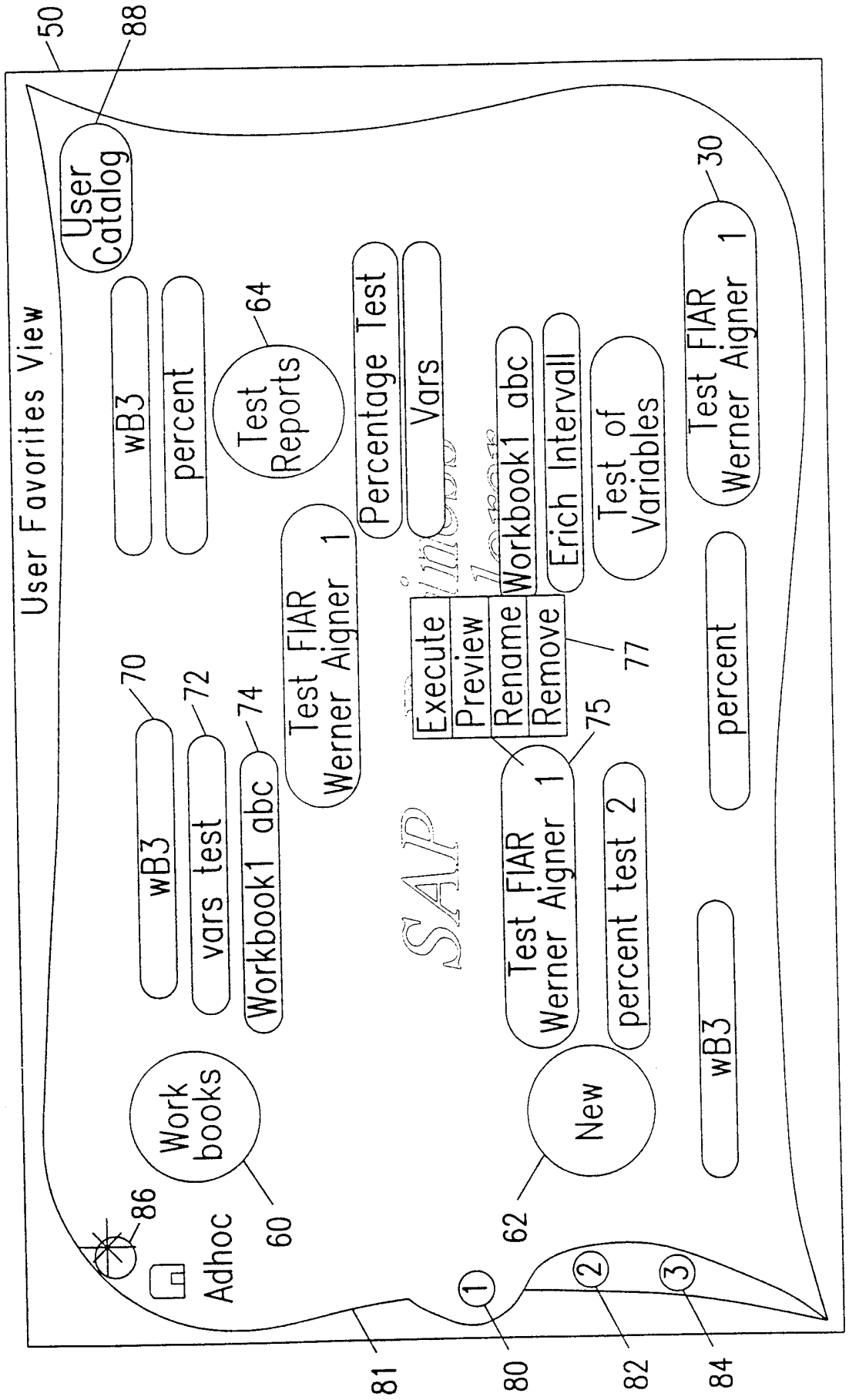

User Favorites View
50
88
User Catalog
wB3
percent
64
Test Reports
Percentage Test
Vars
Test FIAR Werner Aigner 1
Execute
Preview
Rename
Remove
Workbook1 abc
Erich Intervall
Test of Variables
30
Test FIAR Werner Aigner 1
70
wB3
72
vars test
74
Workbook1 abc
Test FIAR Werner Aigner 1
75
77
percent test 2
percent
wB3
Work books
New
62
60
86
Adhoc
1
2
3
81
80
82
84

Fig. 8

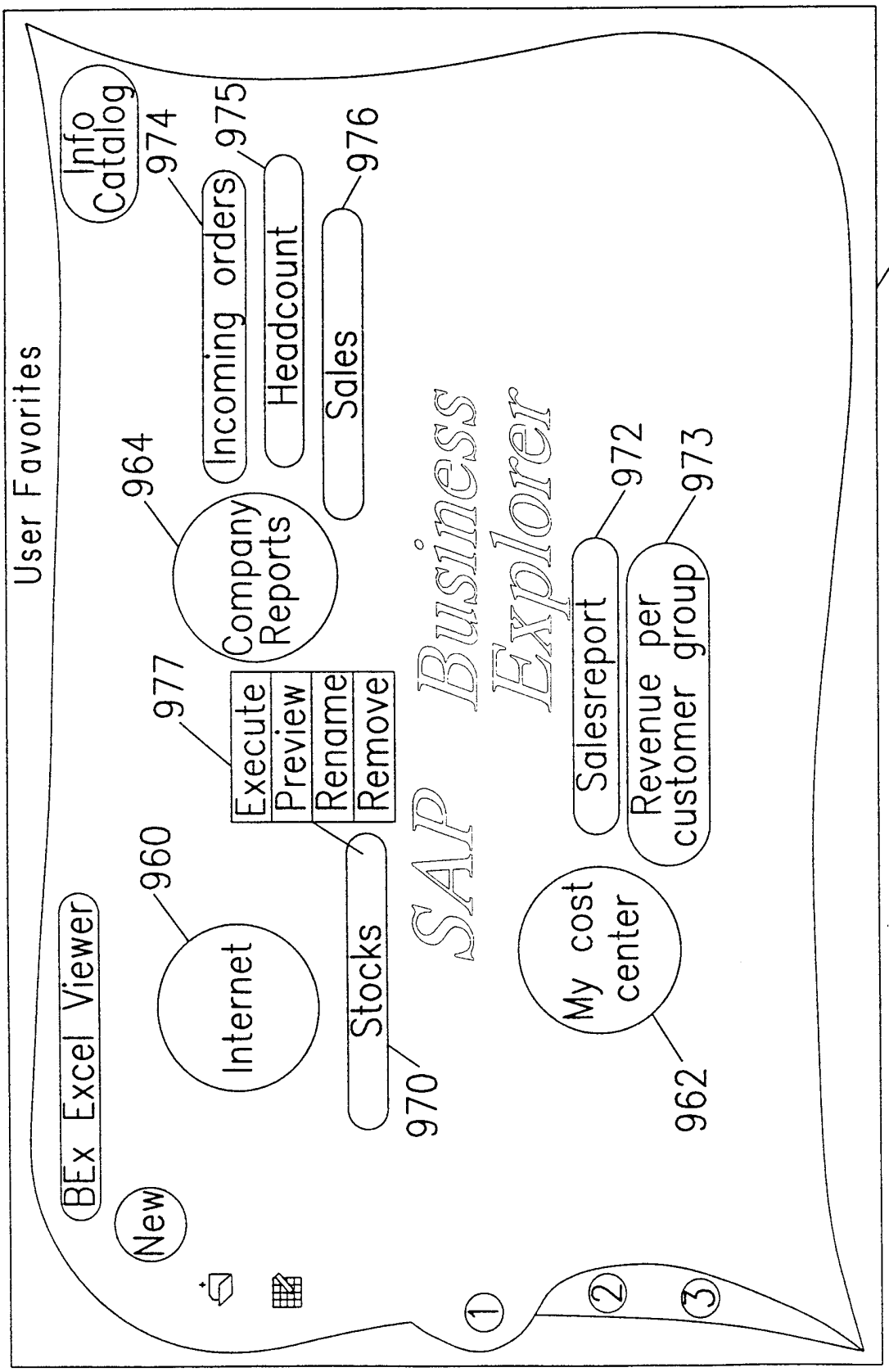

User Favorites
BEx Excel Viewer
New
960
Internet
Stocks
970
977
Execute
Preview
Rename
Remove
964
Company Reports
974
Info Catalog
975
Incoming orders
Headcount
976
Sales
SAP Business Explorer
My cost center
962
972
Salesreport
973
Revenue per customer group
950
1
2
3

Fig. 9a

Fig. 10b

Fig. 9b

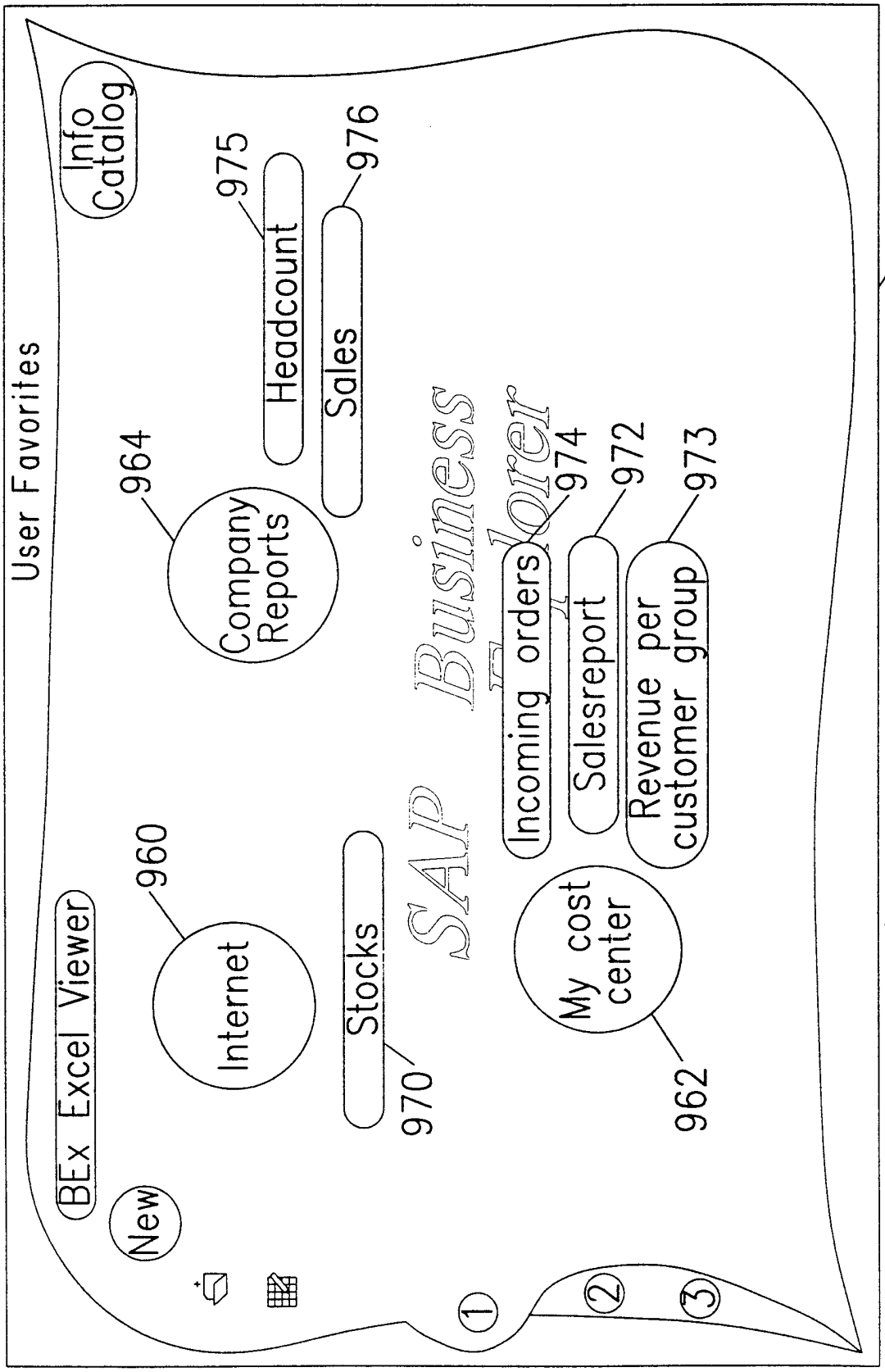

User Favorites
Info Catalog
975
976
Headcount
Sales
964
Company Reports
Business Explorer
974
Incoming orders
972
Salesreport
973
Revenue per customer group
960
Internet
Stocks
970
SAP
My cost center
962
BEx Excel Viewer
New
1
2
3
1050

Fig. 10a

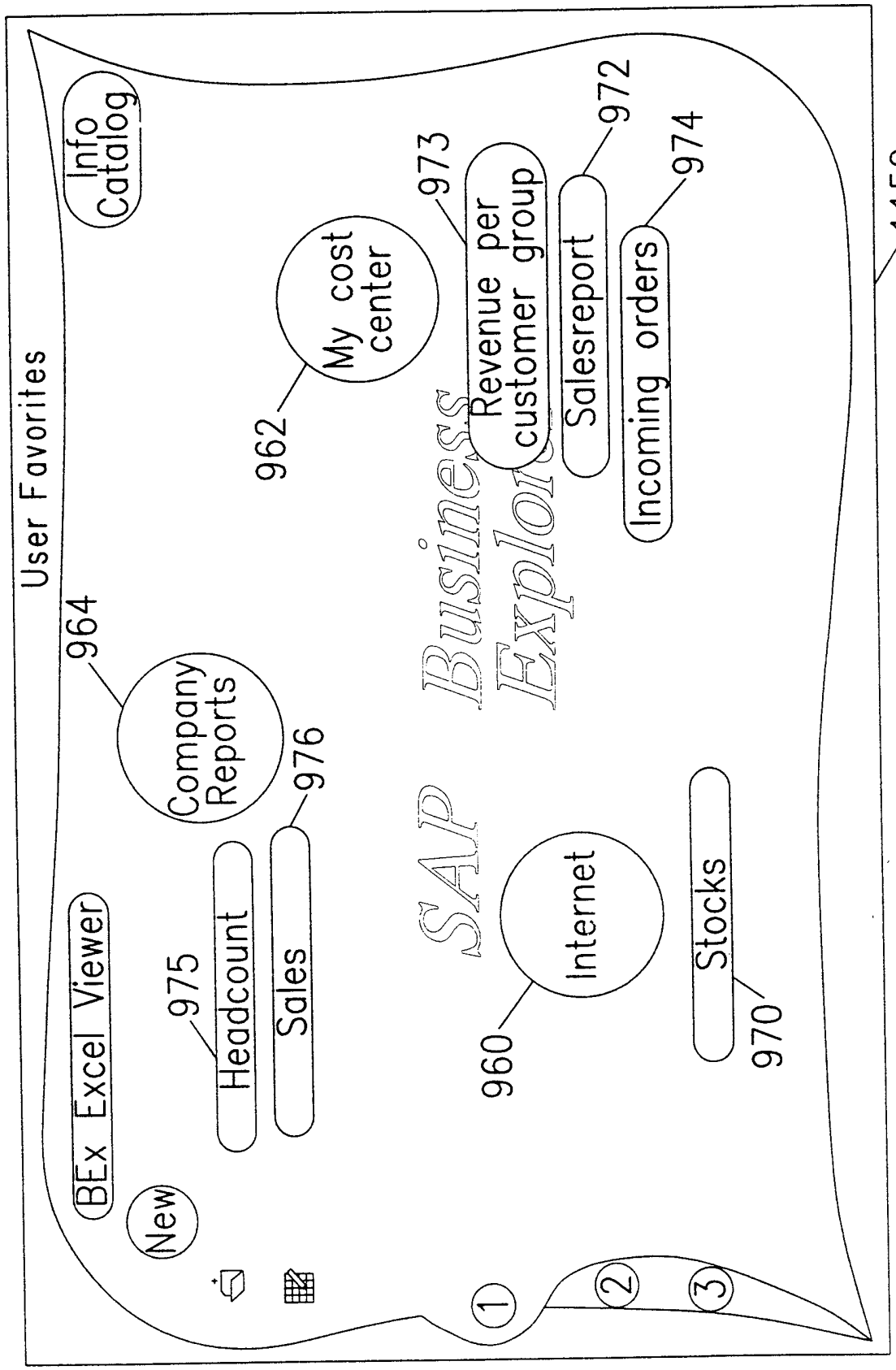

User Favorites
Info Catalog
964
Company Reports
976
BEx Excel Viewer
975
Headcount
Sales
New
962
My cost center
973
Revenue per customer group
972
Salesreport
974
Incoming orders
SAP Business Explorer
960
Internet
Stocks
970
1150
1
2
3

Fig. 11a

Fig. 12a	Fig. 12b
Fig. 12c	Fig. 12d

Fig. 12

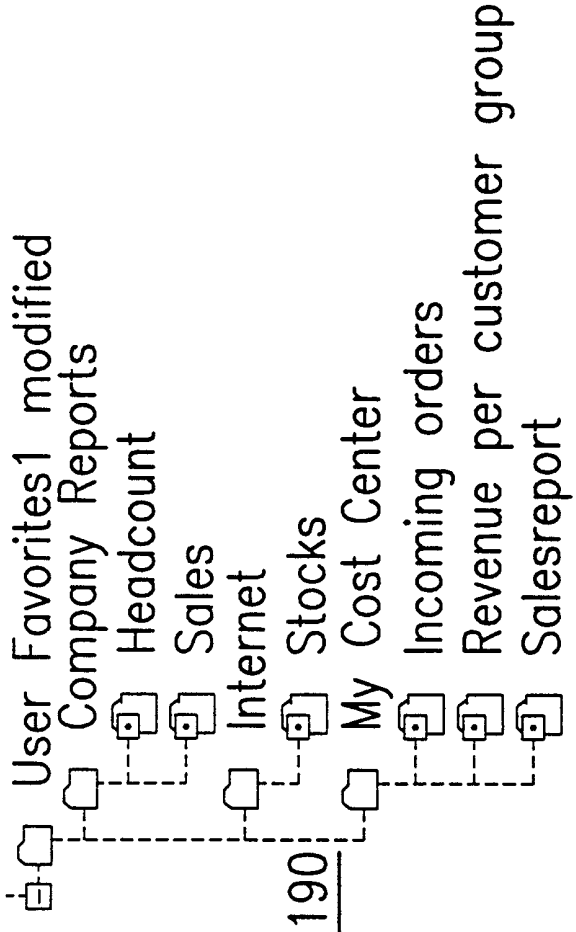


Fig. 11b

Fig. 12a

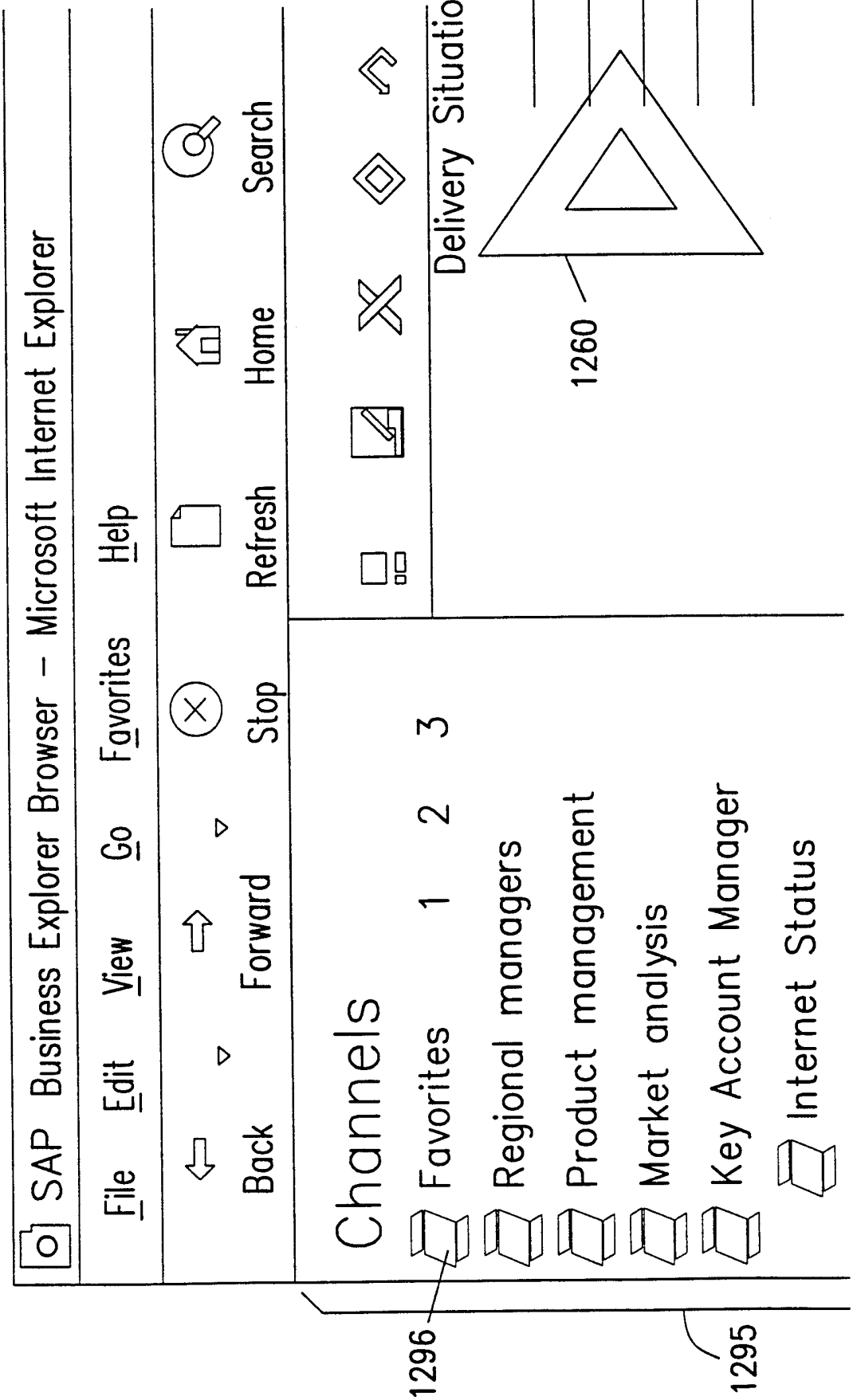

SAP Business Explorer Browser - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search
Delivery Situation
1260
Channels
Favorites 1 2 3
Regional managers
Product management
Market analysis
Key Account Manager
Internet Status
1296
1295

Fig. 12b

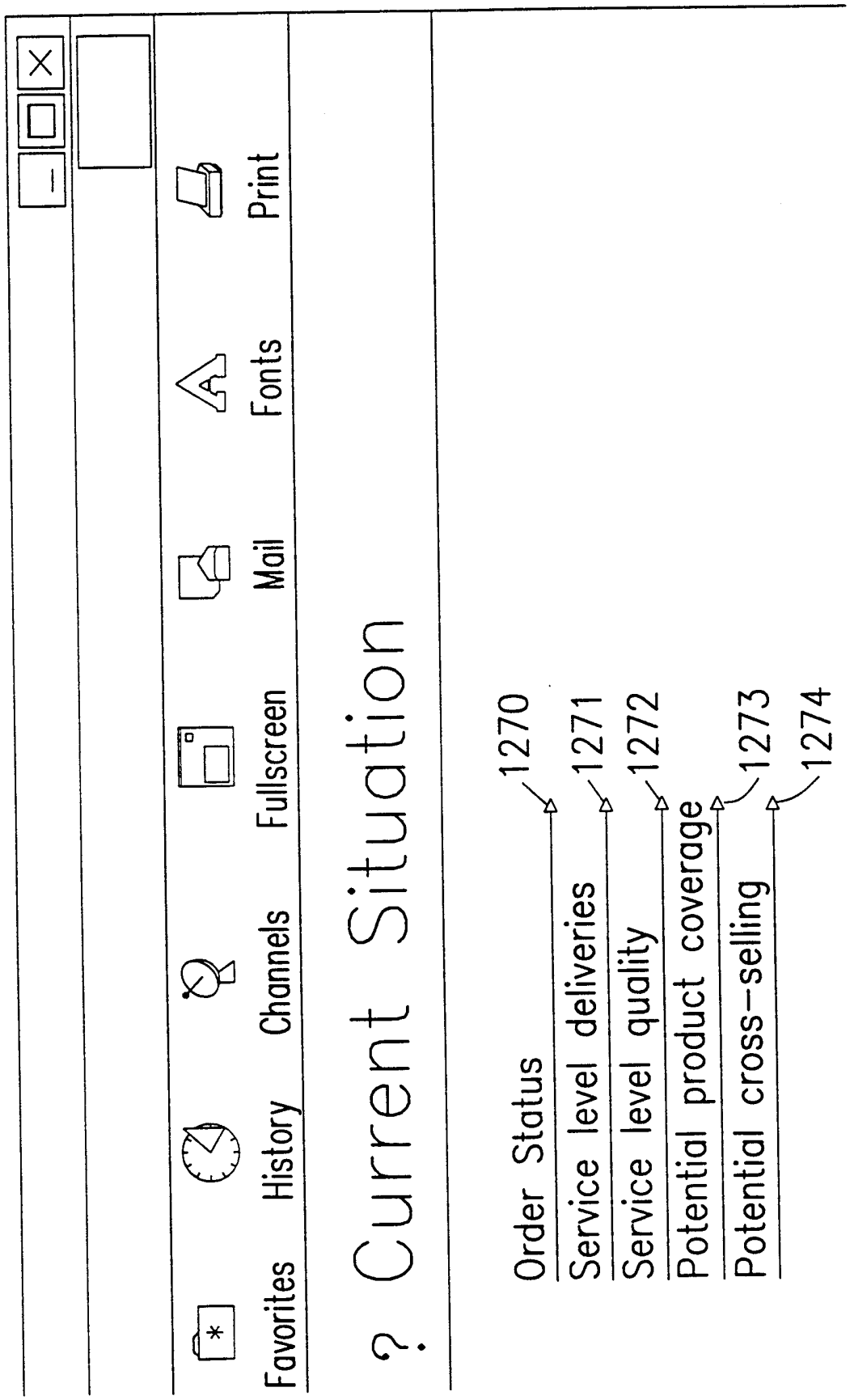

Favorites
History
Channels
Fullscreen
Mail
Fonts
Print
? Current Situation
Order Status
1270
Service level deliveries
1271
Service level quality
1272
Potential product coverage
1273
Potential cross-selling
1274

Fig. 12c

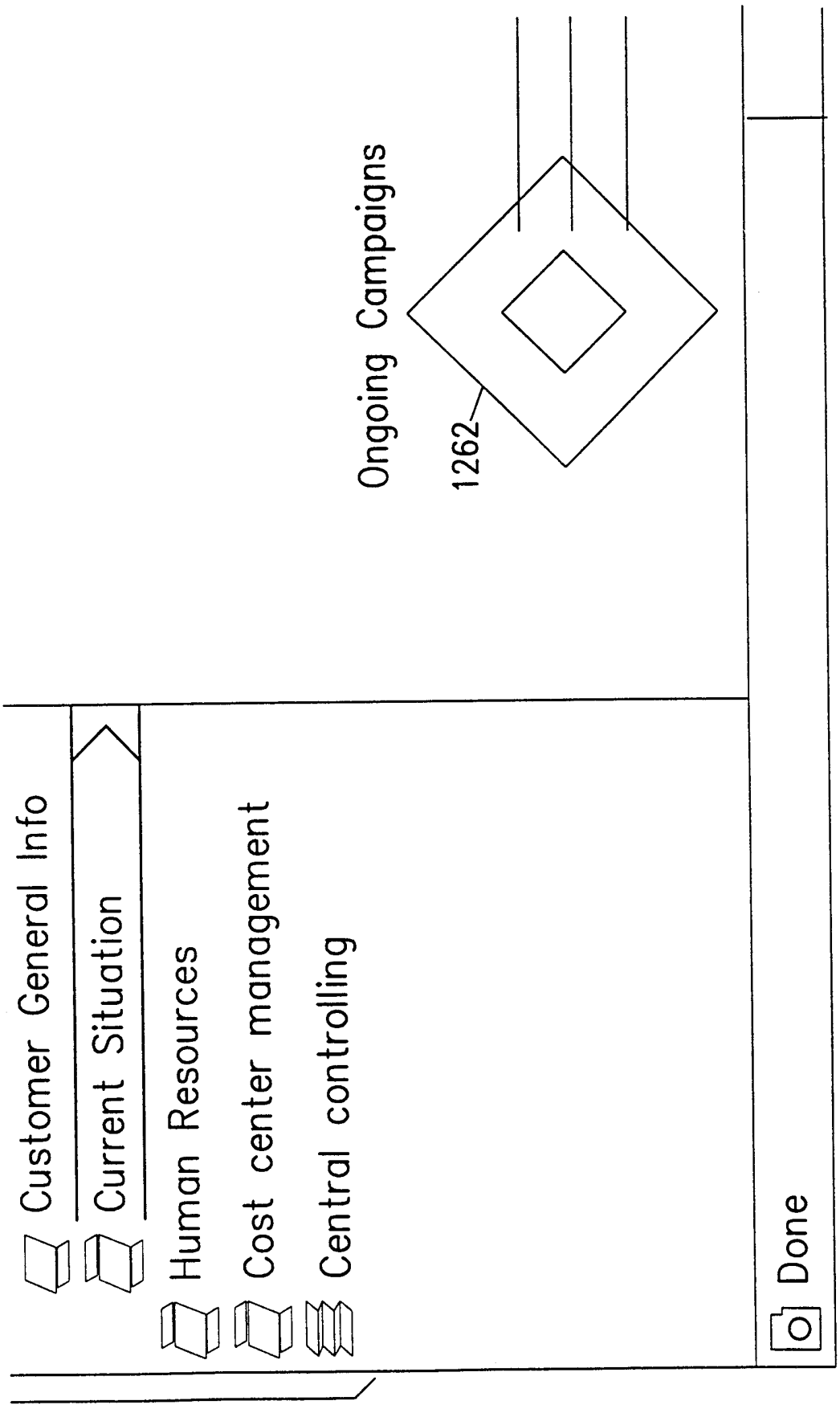
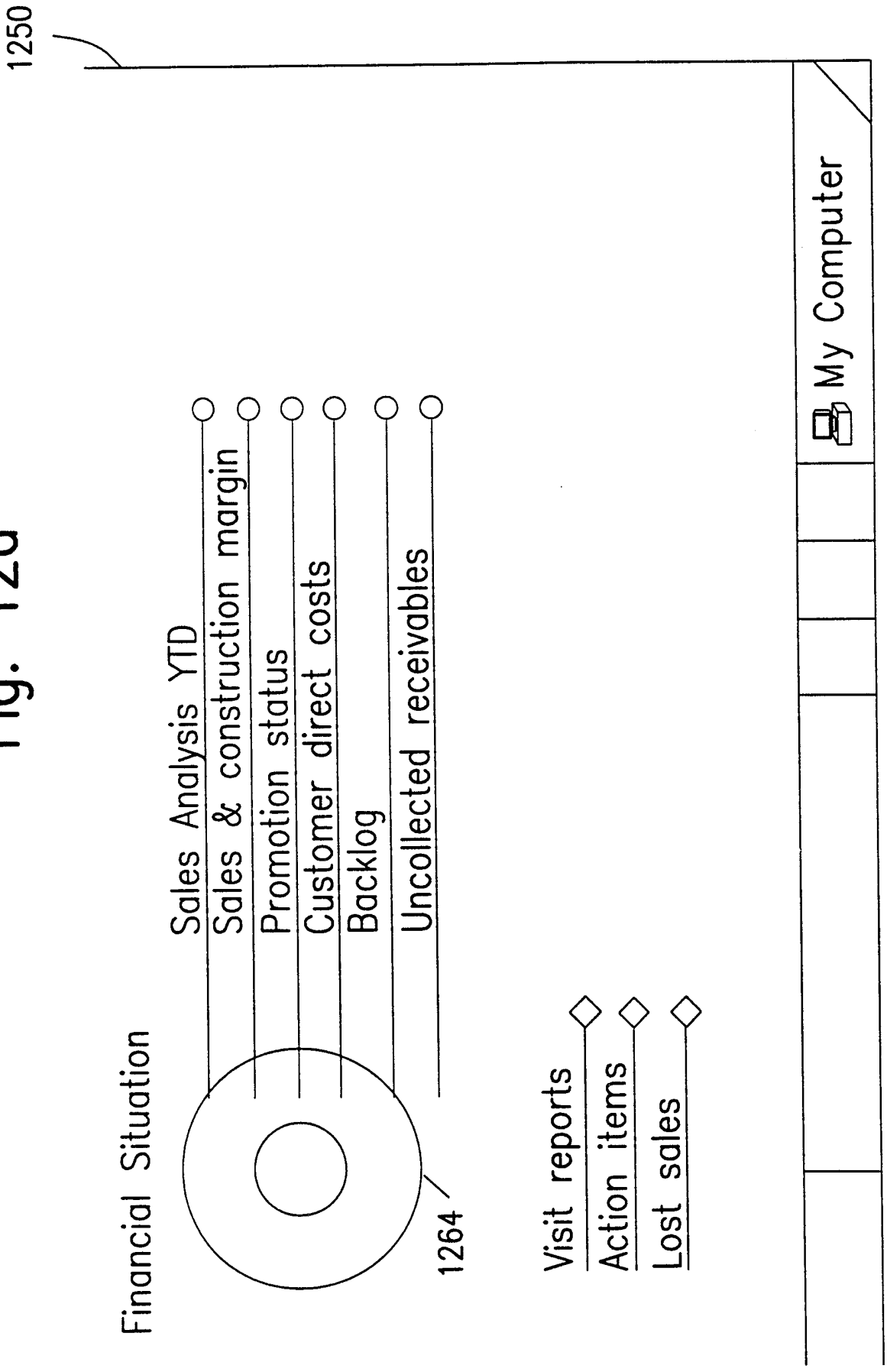

Customer General Info
Current Situation
Human Resources
Cost center management
Central controlling
Done
Ongoing Campaigns
1262

Fig. 12d


1250
My Computer
Financial Situation
Sales Analysis YTD
Sales & construction margin
Promotion status
Customer direct costs
Backlog
Uncollected receivables
1264
Visit reports
Action items
Lost sales

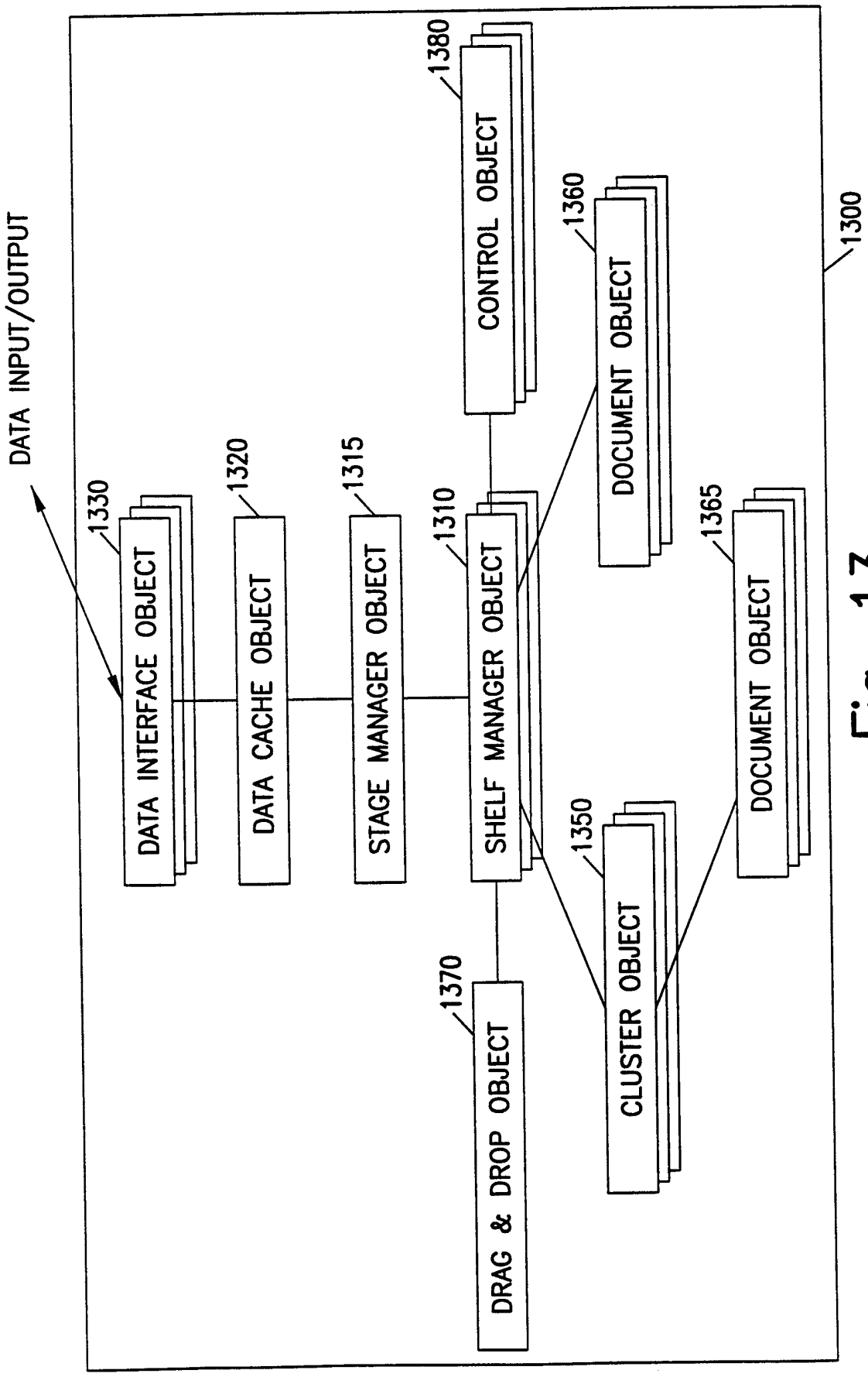

DATA INPUT/OUTPUT
1330
DATA INTERFACE OBJECT
1320
DATA CACHE OBJECT
1315
STAGE MANAGER OBJECT
1310
SHELF MANAGER OBJECT
1380
CONTROL OBJECT
1360
DOCUMENT OBJECT
1365
DOCUMENT OBJECT
1350
CLUSTER OBJECT
1370
DRAG & DROP OBJECT
1300

Fig. 13

INTERNATIONAL SEARCH REPORT

Inter ›nal Application No
PCT/IB 98/01553

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F17/30

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KUMAR H P ET AL: "Browsing hierarchical data with multi-level dynamic queries and pruning" INTERNATIONAL JOURNAL OF HUMAN-COMPUTER STUDIES, JAN. 1997, ACADEMIC PRESS, UK, vol. 46, no. 1, pages 103-124, XP002088976 ISSN 1071-5819 see the whole document	1-62
A	WO 96 02889 A (IKONIC INTERACTIVE INC) 1 February 1996 see the whole document	1-62

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 December 1998	14/01/1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Katerbau, R

1

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

..nformation on patent family members

Inter ..onal Application No
PCT/IB 98/01553

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9602889 A	01-02-1996	US 5544354 A	06-08-1996
		AU 698237 B	29-10-1998
		AU 3143495 A	16-02-1996
		CA 2195311 A	01-02-1996
		GB 2304432 A	19-03-1997