(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 March 2001 (15.03.2001)

PCT

(10) International Publication Number
WO 01/18685 A2

(51) International Patent Classification⁷: G06F 17/21

(21) International Application Number: PCT/US00/24459

(22) International Filing Date:
5 September 2000 (05.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/152,766 3 September 1999 (03.09.1999) US

(71) Applicant *(for all designated States except US)*: **LEWIS, Robert** [US/US]; 11059 Oak Creek Drive, Lakeside, CA 92040 (US).

(71) Applicant and
(72) Inventor: **WHITBY, Evan** [US/US]; 15051 Zodiac Street NE, Forest Lake, MN 55025 (US).

(74) Agent: **STEINS, Karl, M.**; Steins & Associates, 2333 Camino del Rio South, Suite 120, San Diego, CA 92108 (US).

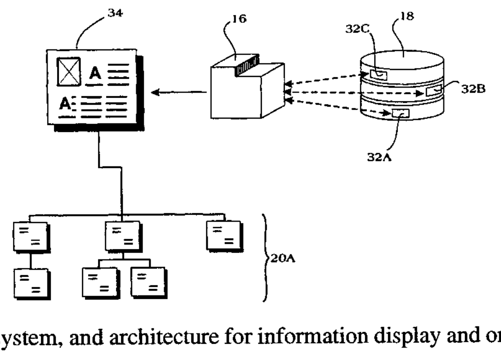
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: IMPROVED METHOD, SYSTEM, AND ARCHITECTURE FOR INFORMATION DISPLAY AND ORGANIZATION



(57) Abstract: An improved method, system, and architecture for information display and organization is disclosed. Also disclosed is a fully integrated, automated, dynamic, distributed, and comprehensive System Human interaction facilitation system (information display and organization system). This system is based on a distributed network architecture that allows for page creation to be done remotely from anywhere in the world to a central server that serves the world. This system is based on easy creation of informational pages consisting of simple predefined page parts. Individual pages are tagged according to the type of page and the location in a hierarchy of pages. Pages are registered into categorization schemes to facilitate retrieval according to category. Pages are also indexed based on text contained in the pages, page type, and initiator to allow for rapid and accurate retrieval of specific pages via searching. The overall hierarchy of pages created in one particular language is replicated across all working languages of the specific theme area. This system can be used to facilitate a wide variety of interactions, such as those between private individuals, those between individuals of the same or different businesses/organizations. The facilitated interactions can be for a variety of purposes, such as interacting in conjunction with a shared hobby or other interest, interacting in conjunction with the purchase by a private individual of a product or service from a business, and interacting in conjunction with the purchase of a product or service by individuals within the same business/organization or different businesses/organizations. The interactions that can be facilitated can be extended to the facilitation of human interaction across national, language, and other borders/barriers.

WO 01/18685 A2

IMPROVED METHOD, SYSTEM, AND ARCHITECTURE FOR INFORMATION DISPLAY AND ORGANIZATION

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to the facilitation of interactions between individuals and, more specifically, to the use of digital network-based informational interfaces to promote the interaction of individuals for purposes such as the exchange of information and facilitation of product purchases and sales, namely via an IMPROVED METHOD, SYSTEM, AND ARCHITECTURE FOR INFORMATION DISPLAY AND ORGANIZATION.

2. Description of Related Art

Human interactions have changed over time from exclusive face-to-face interactions (including the exchange of physical materials such as letters and the like), to include the use of direct real-time distance-separated interactions via voice or images over phone lines or airways, as well as the use of virtual real-time or time-delayed interactions via intermediary digital networked-based media (as can be found on the World Wide Web). The use of digital network-based facilitation of human interactions is becoming an increasingly important part of everyday life, but there are still many severe limitations on the ease of use, efficiency, accuracy, and reach of the facilitation process. Three of the most problematic processes of human interaction facilitation via digital network-based processes are the following.

1. The process by which an individual can initiate the facilitation process by indicating/describing what they are interested in, have to offer, etc.

2. The process by which an individual can locate information that has been made available by other individuals regarding their interests, offerings, etc.

3. The process by which national language, and other borders/barriers, across which there is a need for such facilitation of human interactions, are bridged.

What is needed is an integrated system and method for the easy, efficient, accurate, and far-reaching facilitation of human interactions via intermediary digital network-based media.

SUMMARY OF THE INVENTION

In light of the aforementioned problems associated with the prior systems and methods, it is an object of the present invention to provide an Improved method, system, and architecture for information display and organization.

The present invention contemplates a system whereby an individual (initiator) can easily, efficiently, and accurately create an intermediary digital network-based media description related to their interests, offerings, etc. This information can range from a simple one-line description of their interests, offerings, etc. to a massive and rationally structured set of pages that describe their interests, offerings, etc. Furthermore, this information or parts thereof can be logically registered into a categorization scheme that is designed to most rationally present the information to individuals (receivers) that are potentially interested. Furthermore, in the same categorization scheme, an individual (receiver) can indicate their areas of interest and hereby initiate interaction with other individuals (receivers or initiators). Furthermore, the information created by individuals (initiators) can be digitally searched based on specific types of pages, and content contained therein, by other individuals (receivers) who might be interested in finding such information. Finally, this information can be rationally structured and categorized similarly in an automated manner in other languages or formats that allow for it to cross national, language, and other borders/barriers.

Accordingly, it is an object of this invention to provide a method and means for easily creating pages of information based on simple page parts. It is another object of this invention to provide a method and means for creating a highly rational hierarchy of pages based on page types and page registration. It is still another object of this invention to provide a method and means for allowing pages to be registered into a categorization scheme. It is still another object of this invention to provide a method and means whereby an individual (receiver) can register their areas of interest into a categorization scheme. It is still another object of this invention to provide a multi-lingual information management system that replicates the page parts, pages, hierarchy, and categories of pages across multiple languages.

These and other objects may be fully understood through the description and accompanying drawings that follow.

BRIEF DESCRIPTION OF THE DRAWINGS

The objects and features of the present invention, which are believed to be novel, are set forth with particularity in the appended claims. The present invention, both as to its organization and manner of operation, together with further objects and advantages, may best be understood by reference to the following description, taken in connection with the accompanying drawings, of which:

Figure 1 is a diagram of the conventional client (browser)-server (web server) interaction in a system for generating web pages dynamically;

Figure 2 is an example of a web site structure showing the page hierarchy for a business-to-business interaction system;

Figure 3 is a diagram of a preferred embodiment of the Human Interaction Facilitation System of the present invention;

Figure 4 depicts the interaction between the database and the page creation application in creating an assembled page body;

Figures 5A, 5B, and 5C depict preferred embodiment of a Page Common Record, a Page Part Record, and a Page Identifier Record, respectively;

Figure 6 depicts the interaction between the Receiver and Initiator Access Interfaces and the Page Creation Application;

Figure 7 is an image of a preferred page creation tool interface;

Figures 8A and 8B are example pages of hierarchy navigation interfaces in English and Japanese, respectively;

Figure 9 depicts the opening of a particular page in a preferred page preview window;

Figure 10 depicts a page open in a preferred work space area;

Figure 11 is an example of a dialog box for setting basic page properties;

Figure 12 is an example of a dialog box for setting advanced page properties;

Figure 13 depicts a preferred tool for making combinations of page parts;

Figure 14 is a preferred scheme for registering a created page into the front-end categorization scheme;

Figure 15 is a preferred site search interface that allows for targeted searching against page types; and

Figure 16 is a depiction of a preferred collapsed site structural display, provided to highlight a dynamic category browsing scheme using parent-category/child-category relationships.

DETAILED DESCRIPTION

OF THE PREFERRED EMBODIMENTS

The following description is provided to enable any person skilled in the art to make and use the invention and sets forth the best modes contemplated by the inventors for carrying out their invention. Various modifications, however, will remain readily apparent to those skilled in the art, since the generic principles of the present invention have been defined herein specifically to provide an Improved method, system, and architecture for information display and organization.

The present invention can best be understood by initial consideration of Figures 2 and 3. Figure 2 is an example of a web site structure showing the page hierarchy for a business-to-business interaction system, and Figure 3 is a diagram of a preferred embodiment of the Human interaction facilitation system (information display and organization system) of the present invention.

While the Human Interaction Facilitation System (HIFS) described here can be adapted for use with any number of digital media formats (CD-ROMs, private digital networks, etc.), the present invention is particularly well suited for use with open networks such as the global World Wide Web infrastructure and readily available and easy to use interfaces such as web browser (e.g., those made by Microsoft, Netscape, etc.). As an example of this invention, the use of this HIFS on the World Wide Web with web browsers for facilitation of interactions between individuals in a high-tech business-to-business interaction setting on a global multi-national/multi-lingual scale will be described (see Figure 3).

To facilitate human interaction in the most efficient manner possible and on the largest scale possible, the now common web browser of a client computer is used to communicate with the HIFS that resides on a server computer. Communication between the web browser on the client computer and the HIFS is done on a global scale over the World Wide Web using standard TCP/IP protocols. In the current human interaction facilitation scheme, there are two main types of individuals involved. The first is the type of individual (initiator) who creates information in an effort to initiate interaction. The second is the type of individual (receiver) who seeks out such information and determines if interaction is desirable, and therefore should be commenced. The initiator can use any commonly available computer (e.g., an Intel CPU-based computer with the Windows operating system, where the Intel CPU is preferably a Pentium 100 MHz or higher CPU and the Windows operating system is Windows95 or higher). To access the facilitation system, the initiator can use a commonly available web browser (e.g., Netscape Navigator or Microsoft Internet Explorer, where the preferred version is 4.0 or higher). The receiver can use any commonly available computer (e.g., Intel CPU-based computer running the Windows operating system, where there is no particular preference regarding the speed of the CPU, but the preferred Windows operating system is Windows95 or higher). To access the facilitation system, the receiver can use any commonly available web browser, where there is no particular preference regarding the version of the browser. The facilitation system itself resides on a server computer running a web server program, a dynamic database-fed web page generation program, and a database program that houses the information created. The server computer operating system (OS) can be UNIX, Windows or any other suitable operating system, where the web server program can be any commonly available web server program (e.g., Microsoft IIS web server). Examples of applications for dynamic web generation are Allaire's Cold Fusion program and Microsoft's ASP program. The database program can be any database that can be accessed by the dynamic web generation program, for example Microsoft's Access, Microsoft's SQL, Oracle, Sybase, etc. Our commonly used server configuration is an Intel CPU-based server running WindowsNT, where the Intel CPU is preferably a 200 MHz or higher CPU, the RAM is 128 MB or higher, the web server program is Microsoft's IIS web server, the dynamic web generation program is Allaire's Cold Fusion program, and the database is Microsoft's Access. A more preferred database would be one from the robust enterprise-class of databases such as Microsoft's SQL, Oracle's Oracle, or the like.

The above constitutes the basic components of the HIFS, where the current invention focuses on three aspects. One, the server-side software program created to work in conjunction with dynamic web generation program. Two, the access interfaces that are loaded into the two types of main individual's web browsers. Three, the underlying database tables (their structures and relationships). The software program of the current invention allows for automated and globally distributed facilitation of interactions among an unlimited number of individuals who do not need to be interacting at the same time nor in the same human language.

The relationships and interaction between the various programs residing on a conventional server when employing a Page Creation application (e.g. one made by using "Cold Fusion") are clarified in Figure 1. Figure 1 is a diagram of a conventional system 8 involving client (browser) 14 -server (web server) 10 interaction to generate web pages dynamically. Interaction by an individual with the system 8 is typically initiated by clicking on a link, button, or other object that sends a request for a URL to the server. If Cold Fusion is present and the URL request refers to a file on the server that is identified as a Cold Fusion template (program file with the suffix .cfm, which is used by the Cold Fusion dynamic web generation program), then the .cfm template is read by the Cold

Fusion engine 16, which then executes all Cold Fusion commands present (i.e. those in the language referred to as CFML). The Cold Fusion templates are like normal HTML files, but they include a mix of HTML (can include JavaScript and other scripts) and CFML commands. The CFML commands can include a wide variety of SQL database query commands that allow the template to do things like a select query to pull a copy of data from the database. Other CFML commands can output the text (that has been pulled from the database by a select query) as formatted HTML. When this process is finished, a complete HTML-based file is produced that is then passed to the Web server 10. The Web server 10 in turn passes the HTML-based file on to the web browser 14 of the individual who made the initial URL request. The individual will then see the page that they requested as a normal HTML-formatted page. The strong advantage of using this approach is that the massive amount of information that is necessary to construct a large web site can be stored in the database. This approach allows for easy storage, manipulation, and retrieval of the information. Furthermore, the information can be selected from the database and output as normal HTML, all by using only one Cold Fusion template. Thus, one essentially traditional HTML page (used as a Cold Fusion template with some CFML commands mixed in) can be used to show an infinite number of HTML pages depending on the content fed to the template by the database select query. This approach allows for the creation and easy maintenance of massive informational web sites with minimum overhead. In particular, the look, feel, and content of a massive web site can be changed by changing only one (or a few) templates.

The advancement provided by the present invention involves the creation of an architecture, method and system that utilizes this standard web site generation language (e.g. Cold Fusion) in a unique way in order to provide hierarchical web site structure of virtually unlimited size and complexity.

The typical initiation point for using the HIFS (the improved system of the present invention) to facilitate interaction between individuals is the creation of informational pages that describe the initiator's interests, offerings, etc. Here, the example of a high-tech company describing its various products, services, licenses, and other offerings is used. The process is started when an initiator of the information accesses the server system via a web browser and logs onto the system. An attempted logon calls a security program consisting of HTML and CFML commands that query the user profile database to see if the logon information (name and password) are valid. If the query returns one valid result, then the profile parameters for that individual are obtained by a select query of the profile database. Key parameters for the individual are stored on the Cold Fusion server (as session variables that persist for a set period of time). These stored parameters for the individual allow for their session to be verified as an official secure session and to modify interfaces presented to the individual based on their preferences. Logon is followed by automatic loading of a JavaScript program and HTML or other scripting language into the initiator's browser. The look and feel of the interface can be automatically and appropriately modified to meet the preferences of the individual (e.g., preferred language, etc.), based on their profile data stored in the database. Logon results in the generation of a page creation and page hierarchy management tool within the browser that can then interact with itself and closely over the World Wide Web with the counterpart software program of this invention that resides on the server computer. All information created by the individual (initiator) is ultimately saved into the database on the server, which allows for rapid and easy manipulation by the initiator, collaboration with colleagues in other parts of the world, and rapid and easy access by the individuals using the information (receivers).

Before describing the page creation and page hierarchy management tool in detail, it is best to describe what pages in this system consist of and how they are handled. Complete pages that are created by use of the page creation and page hierarchy management system consist of three fundamental parts. Each part is stored in various forms within database tables. The three fundamental parts of a page are as follows.

1. Page Body data
2. Page Common data
3. Page Identifier data

As shown in Figure 3, the core unit of information that is created by the initiator when using the page creation tool in this system is the Page Body 34. To facilitate human interactions in a rapid and simple method, Page Bodies 34 can be rapidly assembled from simple predefined Page Parts 32 that do not require knowledge of programming or scripting languages to use (e.g., HTML, DHTML, XML, JavaScript, etc.). Page Parts 32 are such things as titles 32A, text 32B, images 32C, and other parts that are commonly found on normal web pages. As a simple alternative to using multiple Page Parts, full and completely formatted pages can be easily created by pasting existing HTML into a single text part. This is particularly useful when pre-existing HTML pages are available. Pages are typically created in sections that can have one or more page parts per section.

The layout of the Page Parts 32 within the Page Body 34 as viewed in the page creation tool is controlled by placement of the Page Parts 32 in standard HTML table cells, where the placement is automatically handled by the page creation tool based on the page positioning and part groupings specified for each part. This same table cell formatting is saved as a series of parameters with each of the parts into the database so that the entire page can be recreated with the proper layout by merely recalling the parts and their format parameters from the database and outputting the formatted page. Individual parts of a page are stored in individual data rows in a database table. A particular page part will be stored with parameters that identify the initiator's affiliation, the identity of the particular page body 34 (to distinguish it from other Page Bodies), the order of the part on the page (relative to the other parts that make up the page), the ID of the part type 32 (to indicate if it is text, image, etc.), parameters defining the style of how to show the part, parameters defining if the part should be shown, parameters that correspond to the table formatting needed to accurately show the part on the page, parameters that are associated with how to show the content of the part (e.g., if text, then these would be formatting parameters such as color, size, font, etc.), and the actual content of the part (the actual text, the actual image file name, etc.). An important feature of this approach is that the content is cleanly separated from its formatting, which allows for easy manipulation and modification of the content and the formatting. In an early version of the part structuring and storage in the database, a simple one-row two-column table structure was used. This, however, results in severe limitations on how the parts can be grouped and variously arranged, so a comprehensive table structuring algorithm was created that allows for complex table formations that support complex part groupings and even the inclusion of groups within groups.

Figure 4 is an additional depiction of the relationships between the database and the assembled page, wherein it is depicted that the individual parts for a particular page 34 can either be saved to or read from the database. Saving can be one of two types, insert new (for a new page) or update old (for an existing page). Saving parts for a new Page Body 34 is typically done by a database query that inserts the parts for the new Page Body 34 into the database. Saving parts for an existing Page Body 34 is typically done by a database query that updates (or deletes and then inserts) the changed parts into the database. Reading parts from the database is typically done by a database query that selects the parts from the database. These saving or reading processes are most easily done by using SQL database commands written directly into the templates (programs that are the object of this invention) that are used by the dynamic web generation program. Because there are potentially an unlimited number of parts that can be placed on a page, the SQL database commands that do the actual saving of a particular part are repeated within the template by looping over the routine for the appropriate number of times for each individual part while doing an include for each instance of the saving routine as needed for each individual part 32.

In a similar manner, the output of parts as a complete page in the workspace of a browser is accomplished by selecting the parts from the database via a select query and then looping over a routine that includes output routines for each part as formatted HTML or other script-based content. Page Bodies 34 created within the page creation tool are isolated (or "orphaned") content until they are assigned to Page Identifier Records (discussed herein below), which associates them with one or more page identities - this fixes their position within a larger hierarchy of pages created by the initiator.

If we turn to Figure 5, we can examine the nucleus of the novel method and system of the present invention. Figure 5A depicts a Page Common Record 51 (or a portion of one); the Page Common Record 51 comprises a plurality of Common Data

Components 53 that are related to all Page Part Records (see below) that comprise the Page Body 34 of a particular (assembled) page. Some of the types of information contained within the Page Common Record 51 are: the fundamental characteristics; language related data (e.g., the language used during the first creation of the page, the current language version of the page, etc.); and the history (e.g. creation date, modification date, ID of the person who created or modified the page, IP address of the computer used to edit the page, etc.) of a particular (assembled) page. By placing this information in one separate record per assembled page body 34 (rather than repeating it as necessary within Page Part Records, described below), efficiencies are obtained in the maintenance and update of the information contained within the Record 51.

Now turning to Figure 5B, we can examine the next component group that contributes to make an assembled page body 34. Figure 5B depicts a preferred embodiment of a Page Part Record 50, in a very simple form. The Page Part Record 50 is a record containing Page Part Data Components 52. The Page Part Data Components 52 contain the data and/or links to data that make up a particular assembled page body; this data is typically stored as one row of data per page part, with one parameter per row field. Examples of the stored data and parameters include: information that identifies it as belonging to a particular initiator and being part of a particular Page Body, indicates the Identification of this particular part; the type of page part that this Record 50 constitutes; and the content of the page part (or, perhaps an address or link to an external file to be displayed as the page part). Data is saved to or updated in the Page Common database tables (for Page Common Records) and Page Part database tables (for Page Part Records) by standard insert or update SQL queries in the save routine (CFML commands).

Finally turning to Figure 5C, we can examine the third (and potentially the most valuable) component of an assembled page body; that of the Page Identifier Record(s) 55. The Page Identifier Records 55 essentially provide navigational links between a display page (i.e., an assembled Page Body 34) and other (assembled) pages within the Web site. Pertinent information contained within the Page Identifier Data Components 57 (in addition to the company and page ID's) is the ID of the parent "page" to the instant "page." Unlike conventional (static) web site designs that include navigational links to related pages, the preferred design of the present invention only includes the parents of the displayed page. In order to determine whether or not to display the children of the displayed page (i.e. in a navigation section of the page body 34), it is a matter of conducting a simple database search for any pages listing the present page as their parent. By constructing the navigation section of the page body 34 in this manner, there is no possibility of having "dead" links (i.e. links to non-existent pages), since a link will display only if the child exists (otherwise the search would not include the child as a result). Furthermore, a substantial advantage of the instant design is that each "link" provided by a displayed Page Identifier Record 55 is separate from any other link; as a result, one particular Page Body 34 may have several options in terms of how many and which Page Identifier Records 55 will be displayed and how they will be displayed, depending upon a number of variables and conditions. In a conventional situation, the Page Identifier Records 55 displayed for a particular Page Body 34 will be the "parent" pages to the displayed page, and the "child" pages to the displayed page.

The Page Identifier Data Components 57 are assigned (i.e. Records 55 are created) when a Page Part Record 50 is created and then saved in total into the server database. Specifically speaking, the page identifier data consists of several parameters 57 that are stored in a database table, typically as one row of data per page, with one parameter per row field. Page identifier data is used to hold high-level information about the larger hierarchy of pages that might have been created by the initiator. This high-level identifier data is used to, among other purposes, identify the name and description of the page when reporting the existence of the page in search or browse results. It is used to identify the page as a particular type of page such as a product line page, a product description page, a product features page, etc., which is useful for focusing search results to specific types of pages. It is used to identify the page in terms of its registration positions(s) within the larger hierarchy of pages that have been created. That is, to identify the page as being registered underneath one or more other pages that are in turn registered into other pages, etc. This registration of pages into other pages creates the highly rational hierarchy of pages that make up the informational site that has been created. Registration is accomplished by first doing a database query of the default page hierarchy table to determine what page type(s) the current page is allowed to be registered into. Then the actual page hierarchy of the current individual's resulting pages are grouped by page type in alphabetical order and presented to the individual for selection as parent pages that the current page can be registered into. Here a known CFML routine called DoubleBox is used to present the resulting parent pages according to type to the individual, where multiple DoubleBoxes are used when there is more than one allowed page type. The individual then selects the appropriate page(s) and submits the results, which are sent to a save routine that inserts an entry into the Page Identifier table for each selected parent page. The save routine is looped over for each of the working languages of the site to simultaneously create the page hierarchy for the site in all working languages. In essence, then, each Page Body 34 is a combination of a single Page Common Record 51, one or more Page Part Records 50 and one or more Page Identifier

Records 55. Some of the key parameters contained within Page Identifier Records 55 are defined below:

1. Identifier: a unique number such as 2000-25 that uniquely identifies the page ("Page Identifier Parameter"). It is preferably comprised of two components:
    a. Co_ID: a unique number such as 2000 that uniquely identifies the initiator.
    b. Loc_ID: a unique number such as 25 that uniquely identifies the page in the initiator's hierarchy of pages.
2. Parent_Loc_ID: a number such as 20 that identifies the parent page into which the current page is registered ("Parent Registration Parameter").
3. Item_ID: a unique number such as 15 that uniquely points to the page body of a particular page ("Page Body ID Parameter").
4. Parent_Item_ID: a number such as 10 that uniquely identifies the parent page body of a particular page ("Parent Parameter").
5. Type: a number that identifies the page type of the current page ("Page Type Parameter"). The type of page is selected from a group of page types ("Page Type Group") that is provided to the initiator during the site design process; control of page types available for a particular new page being place in the hierarchy will prevent the use of incompatible pages, and will improve the overall functionality of the resultant site.

6. Type_Parent: a number that identifies the page type of the parent page ("Parent Type Parameter").

7. Name: a text field that holds the name of the current page ("Page Name Parameter").

8. Description: a text field that holds a brief description of the current page ("Description Parameter").

9. Children: a Yes/No field that indicates if the current page has child pages ("Child Parameter")

10. Name_Lang: a field that indicates the language of the current page's name and description ("Language Parameter").

11. On_Off: a Yes/No field that indicates if the current page can be viewed (by a receiver) ("Viewability Parameter").

12. Class: a number field that indicates the class of the current page (e.g., normal page, category page, form page, etc.) ("Class Parameter")

13. Administrative: A variety of other parameters that control various aspects of how the page appears or can be used (which company logo to use, the page's security, etc.) ("Administrative Parameter(s)").

Referring back to Figure 3, we can see that Page Type identities represent a high-level framework for a complete informational web site. This structure allows for very complex page hierarchies 20A to be easily created, modified, and manipulated, as well as for easy, precise, and accurate access to specific types of pages by other individuals (i.e. receivers) via the access interface. In this example, the name and description of individually created pages (of any of the specific page types), along with several related parameters associated with each page (page type, security, page header that should be used, how to show page links, etc.), are maintained in a database. The Page Bodies 34, via their Page Identifier Records 55, can also be registered in Categories 36 in order to provide the initiator and receiver with additional search and display options. The relationships between pages are also maintained in this database table as defined by the parent-page/child-page relationships. A particular page can have one or more parent pages and none, one, or more than one child page.

Multiple instances of pages can exist (i.e., a page registered into several other parent pages), where each instance of the page can have its own individualized name, description, and other related parameters (see Figure 4). Figure 4 is a database structure indicating page properties and relationships. The actual page content (combination of page parts) will typically be the same for all instances of a page registered into various parent pages, but variations of each page can also be accommodated depending on location, etc. Importantly, by using the Loc_ID, it is possible to register the same Page Body into different locations within the hierarchy of pages, but because each registration of the page has a unique identity (e.g., Loc_ID, name, description, security, etc.), they can be treated and viewed as individual pages. Here the Page Body would typically be the same for each instance of the page.

When a completed page is saved, its Identifier Records 55 are replicated across all working languages of the system. In the situation where a page is being saved in a non-English language, when the dialog boxes for inputting the name and description are displayed, the user (initiator) is presented with one set of boxes for the non-English input and one set for English input. The non-English data goes into the table for that language and the English goes into all other languages. In this way, the entire hierarchy of the informational site is replicated across all working languages even though some of the pages may not have translated content yet (i.e., no Page Body 34 exists for one or more specific languages). This insures that the overall structure of a web site is complete and consistent across all languages and that all language versions can be simultaneously managed from one language. The actual replication of the information across each language table is carried out simply with the use of a program that loops the data saving routine through each of the working languages. In such a situation, the table names are typically modified with the language identifier. For example, a table holding common page data would have the name of Item_Common_ENG, Item_Common_JPN, etc. The saving program loop would then check what the working languages are for a particular system and loop through the saving routine for each language. Importantly, the master list of working languages for each system is stored in a master file (called the Application.cfm file in the case of using Cold Fusion), so merely adding or deleting a language from the master list of working languages (and ensuring that the appropriate tables exist) instantly expands or contracts the languages for which the system is compatible.

Each page type can have its own set of default forms that indicate how that type of page should be structured when it is first created. These forms are like normal pages, but they are identified by their class number (which is located in the Identifier data table) as a form and are associated with a particular type of page based on their page type. These forms can be opened as default pages for a particular page type that is being newly created and used as is or modified to create a different page layout.

In the example used here for high-tech companies, several page types are used. For example, company description pages, product line pages, product description pages, product specification pages, etc. for a “business-to-business theme” area (see

Figure 2), where each page exists in a rational hierarchy of pages. That is, a product specification page would be registrable into product description pages, which in turn would be registrable into product line pages, which would in turn be registrable into the main company page. Thus, when a page is being saved as a new page, depending on the type of page that it is, the choice of pages that it can be registered into is pre-determined. This scheme creates a highly rational and easy to create/use hierarchy of pages. This approach introduces a very important simplification to linking pages compared to traditional hyperlinks of normal HTML-based systems. That is, a complex hierarchy of pages is created from the top down, where pages can only be registered into already existing pages higher up the hierarchy and only within the types of pages that they should be registered into. This creates a situation in which it is virtually impossible to have "dead" links as are commonly found on the World Wide Web. Furthermore, there is no need to create hard links within the body of a page because the information related to forward or backward links is maintained in the database table (where backward links are identified as parent pages if needed to create a browse trail leading to the current page), as well as all of the sibling or child pages that exist for the current page. Because all of the links can be generated dynamically when a particular page is requested, the links are always live links. If a particular page is removed from the page hierarchy, then the links referring to that page will no longer appear because the page is no longer the child of any pages.

Additional page types can be freely added by the individual to their page hierarchy or to the larger system hierarchy, where these individually created page types may or may not be specifically identifiable/recognizable by the larger HIFS.

The use of page types and parent-page/child-page relationships, along with the use of one or more form pages for a particular page type gives this scheme incredible efficiencies and flexibility for dealing with small or massive informational web sites. This level of simplicity and automation is critical for building a massive collection of informational web sites in a larger and very massive informational web site for the particular theme area being dealt with.

In order to facilitate rapid creation and editing of the site content and structure, the arrangement depicted in Figure 6 is provided. As can be seen, the Interface 14 (see Figure 1) can be separated into two distinct parts – the Initiator Access Interface 14A and the Receiver Access Interface 14B. During editing, the Initiator will interact with a Back-end Database 18A via the Page Creation Application 16. The Back-end Database 18A is, essentially a for-editing copy of the Front-end Database 18B. The Initiator can view either the Front-end or Back-end Database 18B and 18A, respectively, however, the Receiver can only access the Front-end Database 18B.

The interface that is loaded into the initiator's web browser consists mainly of two parts. One part is the normal HTML or other such script-based objects that appear within the workspace of the browser. The other part is the JavaScript (or other suitable language) that remains essentially invisible within the full script that is loaded into the browser. The JavaScript is used to operate the interface on the client-side of the software program, which allows for minimizing initiator interactions with the server and thus server-side overhead (which is important for being able to serve massive numbers of people on a global scale). The JavaScript part of the program is largely responsible for maintaining the state of the interface (e.g., which buttons should be active depending on the current activity of the initiator); initiating HTML or other JavaScript calls based on receiver actions, timed events, number of times that an event has occurred, etc.; and keeping track of changes made (by tracking in data arrays) so that the changes can be undone or transferred to the server database when a save to the server is initiated.

When an individual uploads the program over the World Wide Web, a simple interface for using the page/page-hierarchy creation/maintenance tool is loaded into their browser (see Figure 7). This interface has buttons to initiate different functions. For example, one button will allow an individual to create a new page of a specific type within their informational area. Another button opens the navigation tool that allows the individual to navigate through their informational area to select a page to work with. In this example, buttons are interface-state dependent and are available or unavailable for use depending on the state of the interface and what action the individual is currently doing. Other button and menuing systems can also be used.

Navigation around an exiting hierarchy of pages can be done in a number of ways, where one way resembling a file navigation system is used here as an example (see Figure 8A). Figure 8A is an example page hierarchy navigation interface in English. The individual can navigate in the initial language of their preference or change to other languages supported by the system (e.g., Japanese, see Figure 8B). Figure 8B is an example page hierarchy navigation interface in Japanese. The navigation program is based on parent-page/child-page relationships and the showing of child pages for the currently selected parent page based on standard SQL queries of the particular organizations hierarchy of pages. The representation of category-like pages (e.g., product lines vs. product description pages) is based on the Class of the particular page. Furthermore, regular pages will be represented by a small page icon (vs. a folder for category-like pages). When a particular regular page has child pages (e.g., a product description that has specification, feature, option, and other child pages), then this is shown by a cluster of pages where a single page was shown when no children were present. Alternatively, specific pages can be located by searching on name and/or description information, as well as full page content searching, all in the desired language. Furthermore, specific types of pages can be located by selecting the type of page desired (e.g., product description page).

Once a particular page has been located, it can be opened in a Preview Window 90 to confirm that it is the desired page (see Figure 9). Figure 9 depicts the opening of a particular page in the page preview window. The preview of a page in the preview window 90 is accomplished by a database query to select all of the parts of the selected page. These parts are then output with an output algorithm as mentioned earlier, which generates the preview of the page. The preview page is typically smaller than its normal corresponding page because the text and images have been reduced on the fly. This preview method provides a view of a page for confirmation, but does not take over a large portion of the browser's workspace. If this page is the desired one, then this page can be opened in the work space and modified, saved as a new page, page parts added or copied between this and other pages, etc. Additional pages can be opened in the workspace to facilitate work between multiple open pages. Additional pages can also be opened in place of (overwriting) currently open pages. Furthermore, a web page on the World Wide Web can be opened in the workspace to facilitate working between any web page on the World Wide Web and a page being created or modified in the system. When a page is opened in the workspace, the "Save As" icon becomes active and allows the currently opened page to be saved as a new page. This feature allows for rapid creation of an entire informational web site based on pre-existing pages. The underlying save routing is the same as that for saving a new page, where the Identifier information such as Name and Description are input and the parent page(s) are selected. The new page saving is completed with the insert query to create the appropriate Page Common Record, Page Part Record(s), and Page Identifier Record(s) for the new page and all of the various data are saved to the appropriate tables with an insert query.

When a page is opened in the workspace (see Figure 10), the individual horizontal bands of the part groupings can be seen. This grouping is used to clearly demarcate individual groupings of page parts (the coloration is accomplished by setting the color of the underlying table row in an alternating fashion for each consecutive group). Each part and each group has their own selection icon that allows the part or group to be selected for an operation (e.g., copy, paste over, delete, edit, and copy format in the case of an individual part). Small arrow icons on either side of the part selection icon allow for a part or group to be pasted before or after the part. An insertion arrow icon between groups allows for a copied or deleted group to be inserted between groups.

All of the above copy, paste, delete and copy format functions are executed via a JavaScript program that resides in the browser. This program tracks changes made in the parts and groups. When a save is executed, the JavaScript program grabs the most recent data in the database and adds/subtracts/moves the underlying data representing the parts of the page as appropriate to create the most current version of the page. This is then saved into the database. The JavaScript tracking of page changes also allows for undoing page changes if needed.

The page creation window of the interface has a number of icons that lead to functions that help in the creation of pages. For example, when a part 32 is selected, the JavaScript program activates the edit icon to allow for that part 32 to be edited. When the edit icon is clicked, an edit dialog box is opened that is dependent on the part 32 being edited (e.g., if it is a text part, then the text can be edited and the font characteristics modified). The modified part can then be saved back into the JavaScript program. Another icon calls a template that queries the Identifier table for the particular page being edited and present the current high-level identifying parameters for the current page in a dialog box. Here, both the basic and advanced properties of a particular page can be readily modified and saved back into the database through the use of another template that presents a dialog box for modification of the parameters and then executes an update query for the data that was modified.

The basic properties displayed for the page include the browse trail(s) indicating where the page is registered, On/Off (show or don't show on the World Wide Web) and deregister (remove this instance of the page) cluster of radio buttons, a name fill-in box, a description fill-in box, and links to the Advanced Properties settings and the category registration scheme (see Figure 11). This organization allows for rapid modification of all instances of the page.

Turning to Figure 12, we can see that for the Advanced Properties, such as the type of header used on the page when it is viewed by individuals (receivers), the way that links on the page will appear, the security on the page (who has access to that particular page), timed events (e.g., showing or not showing a page depending on a time or date associated event), and whether or not the page has additional functions (e.g., the product described on the page can be purchased), etc. can be individually set for a particular page or instance of a page. Another icon that can be selected for an open page is the translation icon. This opens the translation dialog box that allows for a particular page to be translated. Another icon calls the view page template that will reconstruct the current page from its individual parts and add the appropriate headers and navigation tools to the page so that the individual can confirm that the layout is as desired.

Pages are made of individual page parts that have characteristics that depend on the particular page part being used. Page parts include titles, paragraphs, images, etc. Each page can be easily created by placement of individual page parts on the page in the desired position, thereby eliminating the need for the individual to know specific or complicated programming languages or other protocols. Any individual page part can be edited, where the specific characteristics of that particular page part can be set (e.g., text can be set according to size, color, placement, font style, etc.). Page parts on a particular page can be copied, deleted, pasted, etc. among page parts on the same page or on other pages. Page parts in this particular example are positioned on the page according to an underlying HTML table structure, but other positioning schemes can equally well be used (e.g., DHTML). Combinations of parts can be easily created and added to pages as well, where a simple graphical tool for making simple or complex combinations of parts can be used (see Figure 13). Figure 13 depicts a tool for making combinations or groupings of page parts. Individual page parts are most conveniently stored in order of appearance on the page, but can be stored or used in other sequences depending on the need. Each part can have its associated properties stored in other fields of the same data row to indicate the content's properties (e.g., formatting parameters for text such as font style, color, size, etc.). In this way, a clean separation of the content (e.g., actual text) and its formatting (e.g., font size) is attained, which allows for easy data manipulation and movement to other systems and data structures as needed.

When an already existing page has been modified, it can be saved back into the back-end database system and additional actions done with it. For example, the page can be republished into the front-end database that will be viewed through the access interface by individuals (receivers). The back-end database can therefore be used as an intermediate staging area until pages or sets of pages being prepared have been completed and it is desired to have them be viewable in the front-end access interface. This allows for an entire site to be created or modified over time and published or republished in part or in its entirety at a specific time or different times in the future.

During the saving process, an intermediate workflow dialog is automatically entered into to guide the individual (initiator) through the options available to them when saving. For example, the initial creation of a page can be set to require the following approval of another individual before the further publishing or translation of the page can be authorized. This workflow process is easily set up with a dialog box that displays potential workflow members who are in the group of all individuals authorized to use the system, where the authorized individuals (initiators), their specialties, security levels, job functions, etc. are stored in a personnel database that is accessed through the workflow system. The actual routing of the workflow jobs is done via E-mail notification of a job needing attention or other comparable processes. At the end, an individual with authority to publish (i.e. to make the content accessible to "receivers" via the front-end access interface) and/or translate the information will be included to complete the workflow task and initiate the desired facilitation of interaction with other individuals, here via the front-end access interface.

The publishing procedure captures an image of the completed page consisting of all of the compiled parts as formulated with standard HTML or other scripts that are automatically generated by the system. The HTML capturing is done with a Cold Fusion template that uses the CFHTTP command to capture a particular URL's content as a single data string. The captured page content can include additional parts that are needed to complete the ultimate page that is published for viewing (i.e., is not limited to only the Page Body). The HTML rendition of the captured page is cleaned of unwanted scripts, spaces, and other unnecessary characters and stored in the front-end database that individuals (receivers) will be accessing. This publication process has two advantages. One, this allows for two versions of the information to exist (published page and work-in-progress page). Two, this allows for the published page to be pre-rendered and formatted, which minimizes processing overhead on the web server system that would be required to recreate the page on each viewing (page parts only need to be assembled into a single page once and can then be viewed multiple times as one single preprocessed unit of data). An alternative to publishing the page in the form of a database field entry is to store the page as a traditional HTML or equivalent page on the server.

Along with publishing a page in the same language, under the advancement of the present invention, the page can also be translated (and published) in a different language or languages. This translation procedure can take predominately one of two forms. One form, allows the individual (initiator) to translate the content of pages (and the web site) into a new language version of the page within the same interface (with or without the original version present at the same time). This is the same process as that used for regular page editing, where the individual will be editing into the target language. The second form of translation procedure involves submitting the translation to another individual (within that individual's own organization or in an outside organization). Translation by another individual can readily be accomplished by notifying (e.g., via E-mail) that individual of the need for a translation, or by sending (e.g., via E-mail) the page (i.e. its component part(s)) for translation to that individual. E-mail notification is done by the system when the individual is within the group of people that have access to the system. The system will automatically notify the individual by E-mail that they have a translation task to do and notify them of the page needing translation. Pages sent via E-mail for translation to an individual not within the authorized group of individuals who can use the system can be either the complete page or a simplified version of the page that contains only (or essentially only) the text to be translated. Typically, the page is rendered in a simplified form that contains simple tags marking the individual parts, but does not actually format the parts. When translations are completed by an individual, they are sent back to the system in the form of E-mail to an account checked by the system. When a completed translation job is detected, the file is read (e.g. using Cold Fusion) and an insert query is used to insert the new page into the appropriate parts table (and other tables are updated regarding the newly created page). Here, again, translated content will be treated as new page content when it returns, which will initiate any mandated workflow procedure. In addition to the above means of translation, the use of machine (i.e. automated) translation can also be used.

Pages within a particular informational site can be registered into categories in the front-end access interface (see Figure 14). Figure 14 depicts a preferred scheme for registering a created page into the front-end categorization scheme. Here categories are selected to capture the entire domain of the particular theme area being served. In our example here, we are using the 3,000 item high-tech product classification scheme that has been developed by CorpTech in the U.S. This categorization scheme then serves as the link between the information individual (initiator) who created the information and the individual (receiver) who will be seeking the information. The front-end registration procedure involves clicking on the registration link on the page creation interface (either in the Page Properties dialog box or on the Page Opening dialog box) and opening a modified version of the front-end category browsing scheme. This scheme can be readily browsed just like the normal front-end browsing scheme, but it also contains checkboxes that allow for the page being created 144 to be registered into the front-end category scheme 140. Upon clicking a registration check box for a particular category, a Cold Fusion routine within the browsing scheme is invoked that does an insert query into the registration table. This insert query inserts the Identifier for the current page and the category number for the selected category 142. This approach allows for a particular page 144 to be registered in multiple categories in the front-end and for any one particular category in the front-end to be associated with many different pages. The larger browsing scheme's page counter data is updated to reflect that addition of a new page to a particular category so that when the categories are browsed, the individual will be able to see how many pages are registered into a particular category. All registration inserts and registration count updates are reflected across all language tables by a program loop that carries out the routine for each of the working languages.

The front-end interface includes a search mechanism 150 that allows individuals (receivers) to search for specific pages within the larger system (see Figure 15). Figure 15 is a preferred embodiment of a search function that allows for targeted searching against page types. Searches can be either full text searches or searches against the name and description of pages (or potentially against other forms of keywords, etc.). Searches can easily be limited to a particular page type or done across a variety of combinations of page types, thus allowing for pinpoint location of a particular page or group of pages from one or a number of different informational sites contained within the larger system. To obtain rapid and flexible search results, the Verity search engine bundled with Cold Fusion is used. The Verity search engine allows for pre-indexing of the entire site in various forms (referred to as collections in the Verity terminology). Each collection can be for a particular type of page (e.g., product description pages only), for a particular combination of pages (e.g., product description pages only), for a particular combination of pages (e.g., product line pages, product description pages, and all product related pages), or all page types. The results are also preformatted with the name and description of the pages returned as hits, where the main organization/individual responsible for the page is also added to the hit links that are returned. Furthermore, the Verity search engine rank orders results based on calculated relevancy, so the percent relevancy is also output for the receiver's benefit in judging results. The Verity search engine now supports several human languages, so this can be used to search for information in a variety of human languages. The search results are also gathered into a pick list at the top of the access interface to that the receiver has one-click access to the results of their search. Upon clicking on a link associated with a particular search result (either original search result link or the link in a pick list search result summary), the particular page is instantly displayed.

Pages displayed in the front-end access interface are provided with a number of advanced navigational aids. For example, when one jumps deeply into a particular page within one of the large informational sites contained within the larger site, the potential for becoming disoriented on where one is is quite high. The system will automatically create a browse trail in reverse from the current page back to the parent organization's opening page. This is done by recursively doing a select query of the parent page based on the current page, then doing a select of the parent page for that parent page, etc. and outputting the results as a browse trail that leads back to the ultimate parent page for that particular informational site. Each browse trail link is the name of one of the intermediate parent pages and clicking on that link calls for that page to be immediately viewed. A particular informational site can also be browsed by "skimming" along the Identifier information only and not viewing actual Page Bodies. For example, when the Browse informational site button is clicked, the receiver is presented with a link to the top page for the site and links to the child pages for that page. Upon clicking one of the child pages, that page's child pages are then displayed and a browse trail from the top page to the current page is displayed. At any time the entire browse sequence can be displayed by expanding the browse results to show all intermediate children in the sequence. This can often lead to too much information, however, so the default setting is the collapsed version of the browse sequence, which is similar to the larger site browsing scheme 160 shown in Figure 16.

When an informational site has been entered, there is a navigation menu bar presented that shows a standard array of options for viewing that informational site. For example, search on informational site, browse on informational site, contact company, go to top page (company description page), go to the top product page, etc. This navigation menu bar is consistent across all informational sites, but dynamically generated with Cold Fusion so that each menu option is specific for the particular informational site being viewed. The Cold Fusion menu template does a database query to obtain the top page for each key menu item and then generates the menu bar with links that lead to the correct pages for each informational site. This consistent look and feel makes navigating a vast number of different informational sites a familiar and therefore efficient process.

When a page is being viewed, it is possible to immediately contact the company regarding the information on that particular page or regarding the company (or any other page for that matter). While receivers are looking at different pages, the page that they are currently viewing is recorded on the server as a session variable. When a receiver then clicks the contact company menu button, a dialog box is opened that indicates from which page they came. The dialog presents a short summary of the page (name and description) and presents radio button selections that allow the receiver to indicate the nature of their inquiry. There is also a fill-in box that allows them to define a specific inquiry. The automation of this process is a big aid for bridging language barriers because the receiver can be looking at the information in English, but sending the request to a Japanese organization. The system can convert the simple selections into the target language and thereby facilitate more efficient and smooth communication between the receiver and the initiator.

Another navigation aid is presented to site receivers to help them gather information in an efficient manner. Any page can be bookmarked into the interface, which stores a reference (name and Identifier) in a bookmark picklist. This can be continually added to and refined as needed. When desired, this can be sent via E-mail (through the use of a Cold Fusion template) to a desired recipient, which could be oneself, a colleague, or a number of people, all in an effort to alert an even larger number of people to specific content of potential interest within the site.

Another navigation aid is the find related pages function. When a page of interest is found, the Find Related Technologies link will check the registration scheme for the particular page (or recursively search its parent pages) to find where the page (or its parent pages) have been registered in the categorization scheme. The system will then recreate the browsing trails for each registration category and present these to the receiver.

The receiver can then jump into the categorization scheme and find all pages related by registration into the categorization scheme.

The browsing categories (high-tech product items in this particular example) are all related to each other by a parent-category/child-category relationship (see Figure 16). This relationship allows for the browsing scheme to automatically adjust to any number of category entries in width (any number of "sibling" categories) and any number of category entries in depth (child categories and subsequent child categories, etc.). This scheme allows for a single dynamic category display Cold Fusion template to meet any number of needs for dynamically displaying categories and situations where categories will be changing with time.

As the individual (receiver) browses on the categories, the related page name/descriptions or derivative information thereof can be viewed for any one particular category (see Figure 16). The actual pages registered into the categorization scheme are typically limited to product line pages (this minimizes the overhead of having to register all of the products of a particular product line into various categories and the confusion of presenting too many products to the individual (receiver) in any one category). The browsing scheme has a switch that allows the resulting related pages for a particular category to be set as either the company opening pages for that category, as the company opening pages for that category and all lower categories, as the product line pages for that category, or as the product line pages for that category and all lower categories. This flexibility allows the receiver to fine tune the type of pages that they are presented with. A key part that makes this functionality possible is that each category has its own unique number ("this category") and a list of itself and all lower categories ("this and all lower categories"). These category numbers are used to search the registration table for all appropriate pages and then display them to the receiver. For the setting of displaying the companies, this is accomplished by selecting all of the appropriate product pages and filtering these for unique company Identifiers and then querying the Identifier table for the appropriate company information and then displaying that to the receiver.

The preferred language of the receiver and the identity of the various Cold Fusion templates that are used to display information to the receiver are stored as session variables. This allows for the entire site to be instantly switched to another language and to be back at the spot where it was before switching, but now in the requested target language.

The various language versions of the information on page name, page description, and other parameters, as well as the actual page content made of page parts, as well as the registration of pages into other pages or into front-end access interface categories is replicated across all of the working language database tables supported by the system (number and specific languages can be freely set as needed). Different language versions of the various data forms can be maintained in the same table with language indicators or in different language specific tables. The system has a parameter that refers to the preferred language of the current individual (initiator or receiver) and this parameter determines which of the language tables or language fields to use. Once a language is selected, that individual can remain operating in their preferred language, but can locate information that was originally created by individuals (initiators) in other languages. For example, the front-end browsing categories can exist in multiple languages and a particular page created in one language would be registered into the appropriate categories in all languages. In the current example of this system, the default language is set to English, but a native speaker of Japanese seeking information in Japanese on a particular high-tech product would be able to browse the product categories in Japanese and see a listing of the particular companies making that product or of the specific products registered into that product category. The information displayed to the Japanese individual would be a combination of Japanese and English (English content is used as the default language, where the original page name, page description, and/or page content was not available in Japanese). When a particular page is being viewed, if the Page Body has not been translated in the currently viewed language, then the system will query the English table to retrieve the English Page Body. If the English Page Body does not exist, then the system will display the name and description of all of the child pages. If there are no child pages, then the system will display the description for the current page. If this is not available in the current language, then the English description will be presented. For English speakers or those preferring to see the information in English, all information would appear in English, where the only deficient content would be the actual page content if that was not yet available in English (i.e., the page name and page description are always entered in English, even by Japanese individuals (initiators). Content in the desired target language can also be provided via machine translation, either in advance (e.g. in batch mode) or "on the fly," as desired.

Receivers also have a means to alert initiators or other receivers of their areas of interests. Receivers can go into a category registration scheme that is similar to that that initiators used to register their pages into the categorization scheme. The receivers can click on the check boxes of categories of interest, which will register them into those categories. This will allow the latest related information from initiators that has been registered into the categorization scheme to be cross-correlated with the categories of interest to the receivers and an electronic newsletter generated on a personal level to match new information of interest from initiators to the areas of interest to receivers.

In summary, a method and means of creating an HIFS has been described for the theme area of high-tech business-to-business facilitation on a global scale (multi-lingual). The present invention provides for the simple distributed creation of a particular type of page from page parts, for the identification of the page (name and description), for the rational registration of pages of particular type into a hierarchy of pages and into a front-end categorization scheme. Individuals (receivers) using the front-end can then readily locate desired information and interact with the originating individual (initiator).

Those skilled in the art will appreciate that various adaptations and modifications of the just-described preferred embodiment can be configured without departing from the scope and spirit of the invention. Therefore, it is to be understood that, within the scope of the appended claims, the invention may be practiced other than as specifically described herein.

CLAIMS

What Is Claimed Is:

1. A system for information display, comprising:

initiator information access interface means for posting information for display;

receiver information access interface means for displaying said posted information; and

page creation means for assembling Assembled Page Bodies from Page Parts, said page creation means in communication with said initiator information access interface means and said receiver information access interface means, wherein each said assembling of an Assembled Page Body is responsive to at least one Page Part Record, each said Page Part Record stored in a page data storage means, said page data storage means in communication with said page creation means.

2. The system of Claim 1, wherein:

said initiator information access interface means comprises at least one programmable computer, each said computer further comprising processing means, data input means, data display means, means for communicating with said page creation means and an initiator information access interface application executed by said programmable computer;

said receiver information access interface means comprises at least one programmable computer, each said computer further comprising processing means, data input means, data display means, means for communicating with said page creation means and a receiver information access interface application executed by said programmable computer; and

said page creation means comprises at least one programmable computer, each said computer further comprising processing means, means for communicating with each said initiator and receiver information access interface means and a page creation application executed by said programmable computer.

3. The system of Claim 2, wherein:

each said Page Part Record comprises a plurality of Page Part Data Components.

4. The system of Claim 3, wherein each said Assembled Page Body is further responsive to at least one Page Identifier Record, said Page Identifier Record comprising Page Identifier Record Data Components, said Page Identifier Record Data Components comprising a Parent Page Parameter containing Parent Page Identity Data.

5. The system of Claim 4, wherein said Assembled Page Body is further responsive to a Page Common Record, said Page Common Record comprising Page Common Data Components comprising at least one Language Parameter, one said Language Parameter indicating the human language in which each said Assembled Page Body was originally created as a result of operation of said page production means.

6. The system of Claim 5, wherein said Page Identifier Record further comprise a Page Type Parameter, said Page Type Parameter identifying the type of Assembled Page Body to be created by operation of said Page Part Record, the value of said Page Type Parameter selected from a Page Type Group, said Page Type Group comprising a hierarchical set of pre-established values that define the fundamental relationships between all said types of Assembled Page Bodies.

7. The system of Claim 6, wherein said Page Identifier Record further comprises:

a Viewability Parameter for controlling whether or not a particular Assembled Page Body is viewable by a particular receiver; and

at least one Administrative Parameter for adjusting the appearance or access to an Assembled Page Body.

8. The system of Claim 7, wherein said Page Identifier Data Components further comprise:

a Page Identifier Parameter for identifying an Assembled Page Body;

a Linked Page Name Parameter for specifying the name of an Assembled Page Body which is the subject of a Displayed Link; and

a Description Parameter for describing said Displayed Link; and

9. An improved method for information display, comprising the steps of:

creating a hierarchically-structured set of data for display by:

creating a plurality of Page Part Records, said Page Part Records comprising a plurality of Page Part Data Components, and Page Identifier Records said Page Identifier Records comprising links to Assembled Page Bodies, and said Page Identifier Records comprising a Page Type Parameter, the value of each said Page Type Parameter in each said Page Identifier Record being selected from a hierarchical Page Type Group; and

generating, on user request, individual Assembled Page Bodies for each said Page Part Record through operation of page creation means for assembling Assembled Page Bodies from said Page Part Records.

10. The method of Claim 9, further comprising the step of creating at least one alternate hierarchically-structured set of data for display by:

creating alternate said Page Part Records for each said alternate set;

creating alternate Page Common Records for each said alternate set; and

creating alternate Page Identifier Records for each said alternate Page Part Records.

11. The method of Claim 10, further comprising the step of translating said alternate hierarchically-structured set of data for display into a foreign language by translating portions of said Page Part Records and portions of said Page Identifier Records.

12. The method of Claim 9, wherein said Page Part Record creating step further comprises said Type Groups defining the generic relationships between said

Assembled Page Bodies and said Page Identifier Records, said Page Identifier Records comprising Page Identifier Data Components, said Page Identifier Data Components describing explicit heirarchical relationships between Assembled Page Bodies.

13. A system for information display over a computer network, said network comprising:

web site server means for transmitting web site data information over said network;

page creation application means for transmitting web site data information to said web site server, said web site data information comprising data for assembling Assembled Page Bodies from Page Parts, wherein each said assembling of an Assembled Page Body is responsive to at least one Page Part Record, said Page Part Record comprising a plurality of Page Part Data Components, said Page Part Data Components comprising a Page Type parameter, the value of said Page Type parameter being selected from a Page Type Group, said page creation application means comprising executable statements being executed on a programmable computer, said page creation application means further in communication with said web site server means;

database server means for storing said web site data information in the form of web site data records for access by said page creation application means;

initiator information access interface means for inputting said web site data records for storage in said database server means, said initiator information access interface means in communication with said page creation application means over said computer network; and

receiver information access interface means for displaying said transmitted web site data information, said receiver information access interface means detachably in communication with said web site server means over said computer network.

14. The system of Claim 13, wherein each said Assembled Page Body is further responsive to at least one Page Identifier Record, each said Page Identifier Record comprising Page Identifier Record Data Components, said Page Identifier Record Data Components comprising a Parent Page Parameter containing Parent Page Identity Data.

15. The system of Claim 13, wherein said Assembled Page Body is further responsive to a Page Common Record, said Page Common Record comprising Page Common Data Components comprising at least one Language Parameter, one said Language Parameter indicating the human language in which each said Assembled Page Body was originally created as a result of operation of said page production means.

16. The system of Claim 14, wherein said Page Identifier Record further comprise a Page Type Parameter, said Page Type Parameter identifying the type of Assembled Page Body to be created by operation of said Page Part Record, the value of said Page Type Parameter selected from a Page Type Group, said Page Type Group comprising a hierarchical set of pre-established values that define the fundamental relationships between all said types of Assembled Page Bodies.

17. The system of Claim 16, wherein said Page Identifier Record further comprises:

a Viewability Parameter for controlling whether or not a particular Assembled Page Body is viewable by a particular receiver; and

at least one Administrative Parameter for adjusting the appearance or access to an Assembled Page Body.

18. The system of Claim 14, wherein said Page Identifier Data Components further comprise:

a Page Identifier Parameter for identifying an Assembled Page Body;

a Linked Page Name Parameter for specifying the name of an Assembled Page Body which is the subject of a Displayed Link; and

a Description Parameter for describing said Displayed Link; and

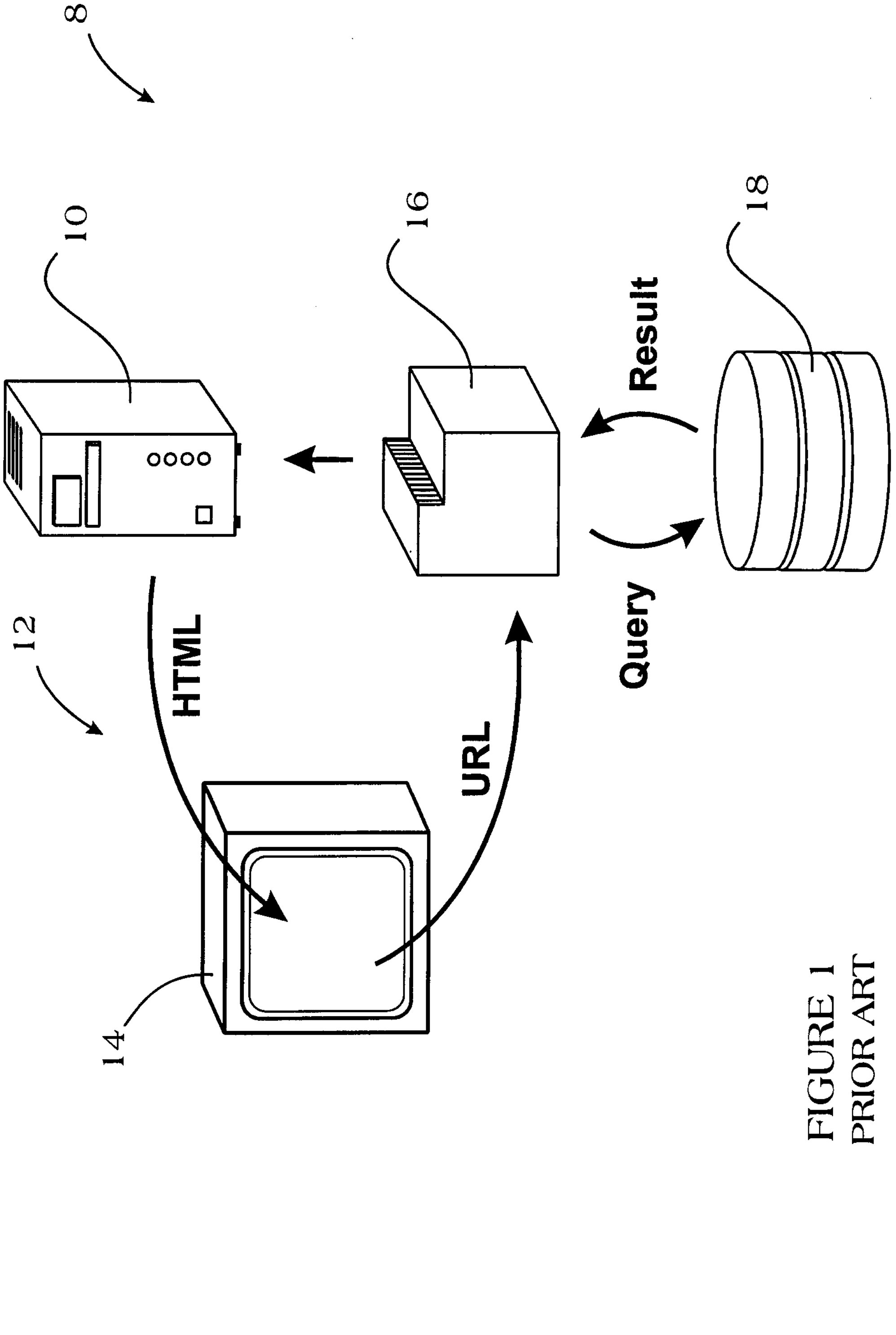


FIGURE 1
PRIOR ART

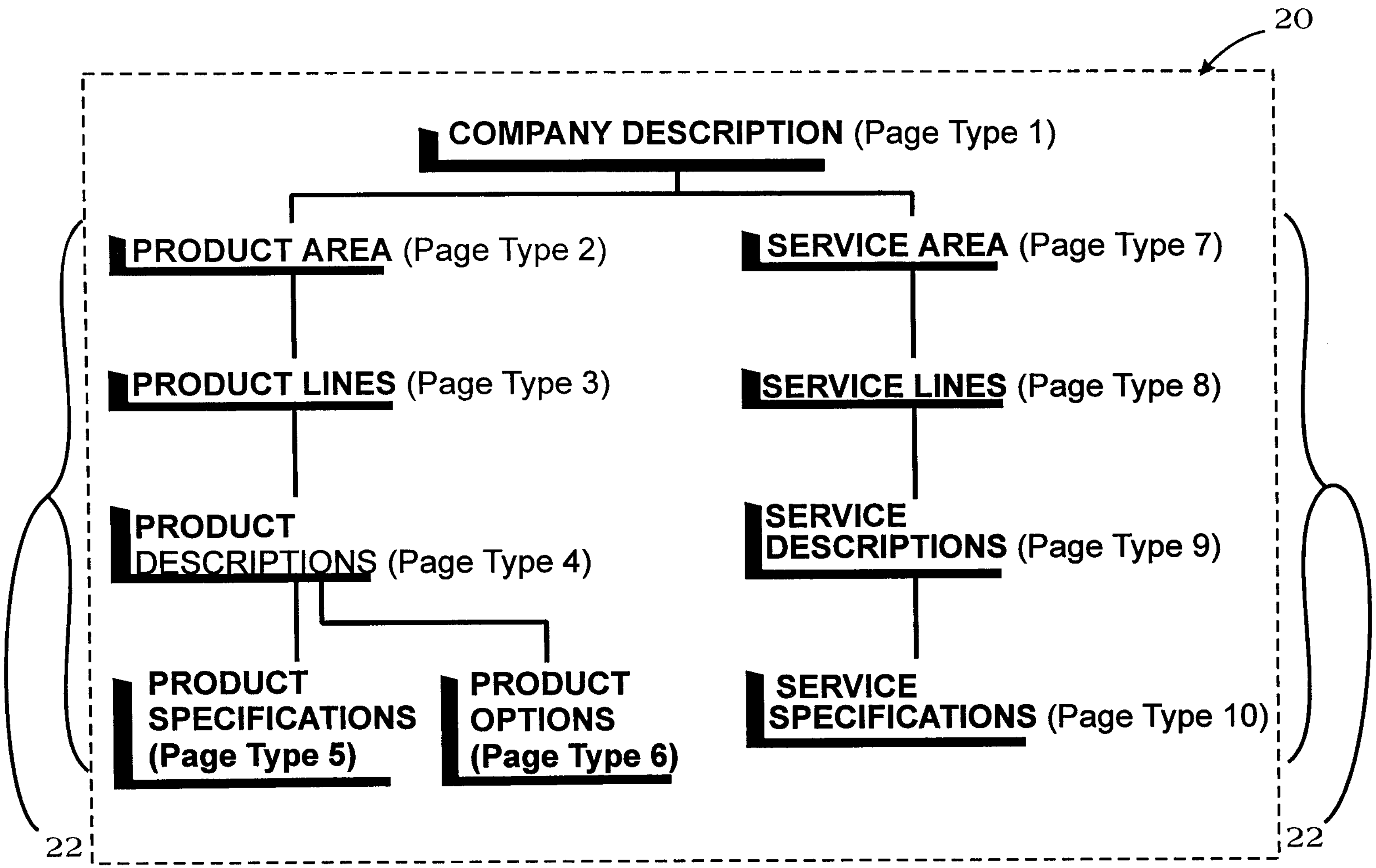

20
COMPANY DESCRIPTION (Page Type 1)
PRODUCT AREA (Page Type 2)
PRODUCT LINES (Page Type 3)
PRODUCT DESCRIPTIONS (Page Type 4)
PRODUCT SPECIFICATIONS (Page Type 5)
PRODUCT OPTIONS (Page Type 6)
SERVICE AREA (Page Type 7)
SERVICE LINES (Page Type 8)
SERVICE DESCRIPTIONS (Page Type 9)
SERVICE SPECIFICATIONS (Page Type 10)
22
22

FIGURE 2

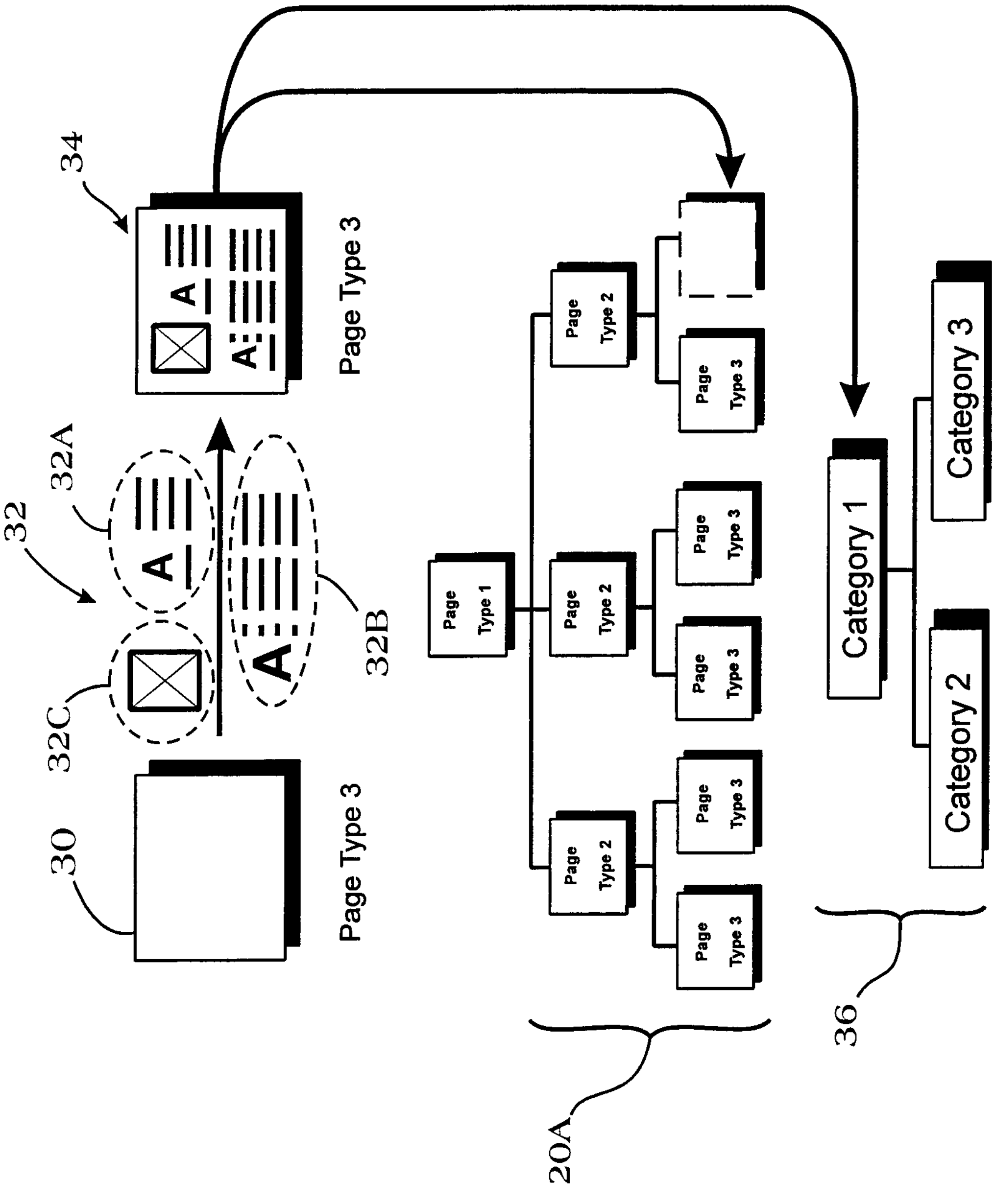

30
Page Type 3
32C
32
32A
32B
34
Page Type 3
Page Type 1
Page Type 2
Page Type 3
20A
Category 1
Category 2
Category 3
36

FIGURE 3

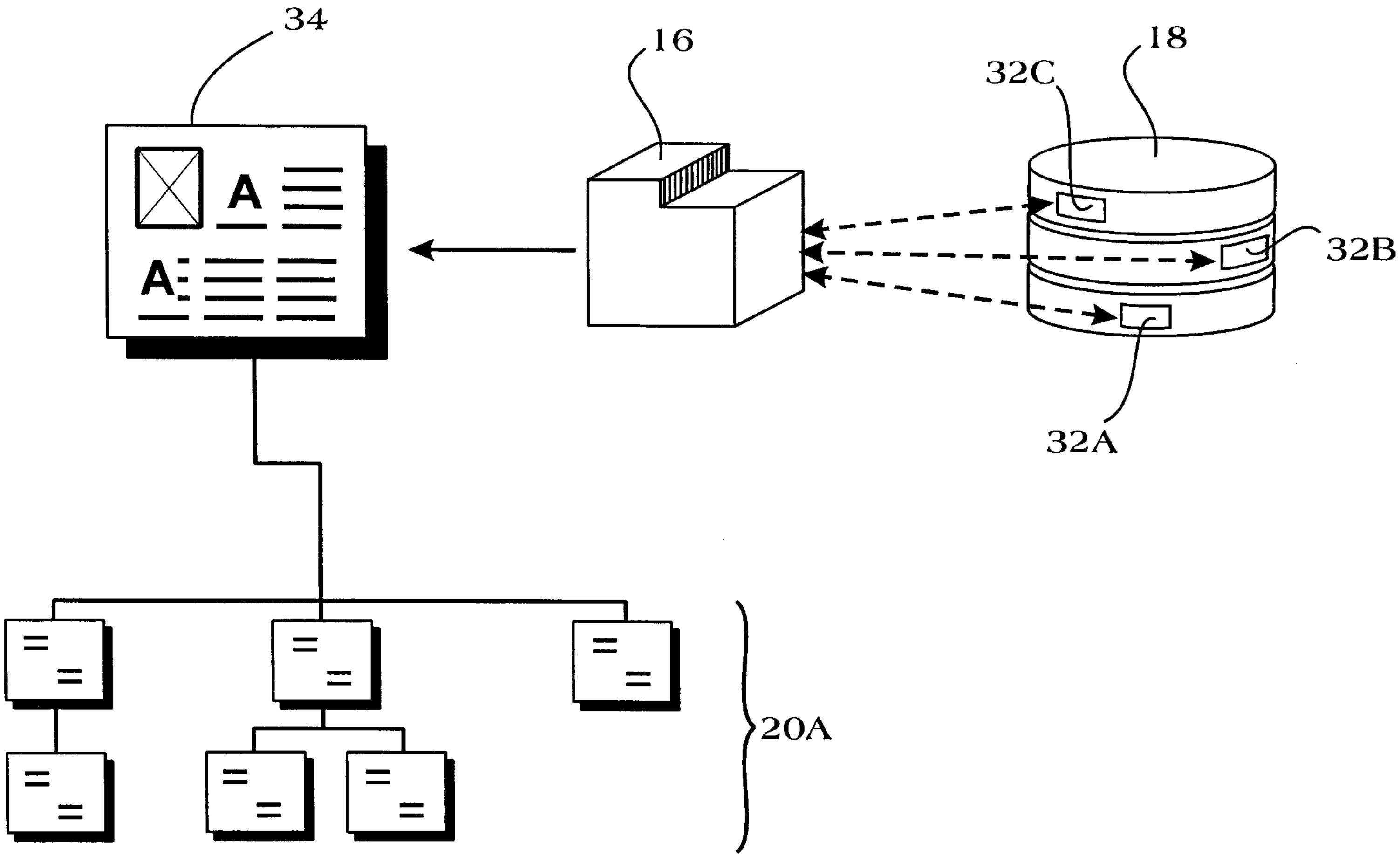


FIGURE 4

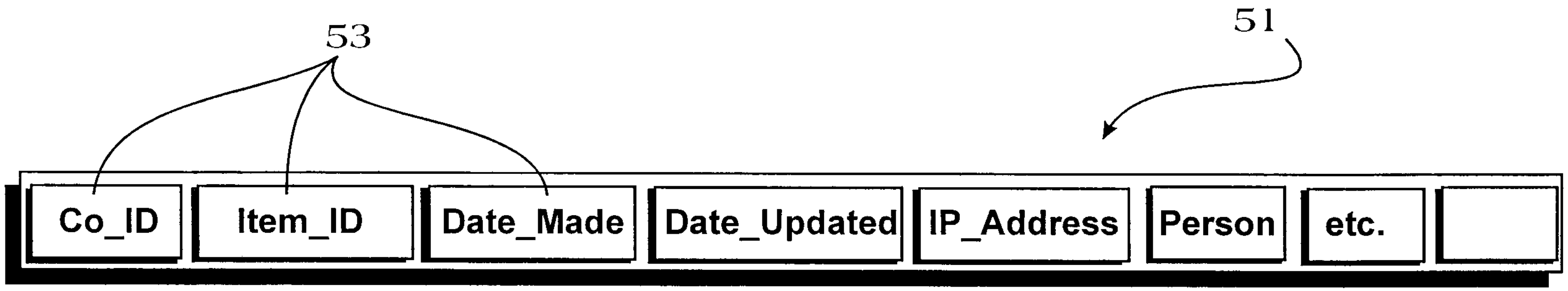


FIGURE 5A

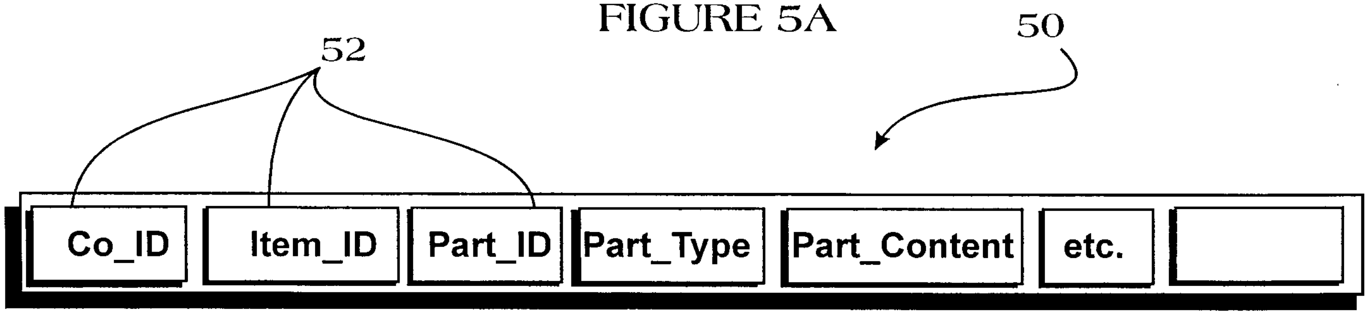


FIGURE 5B

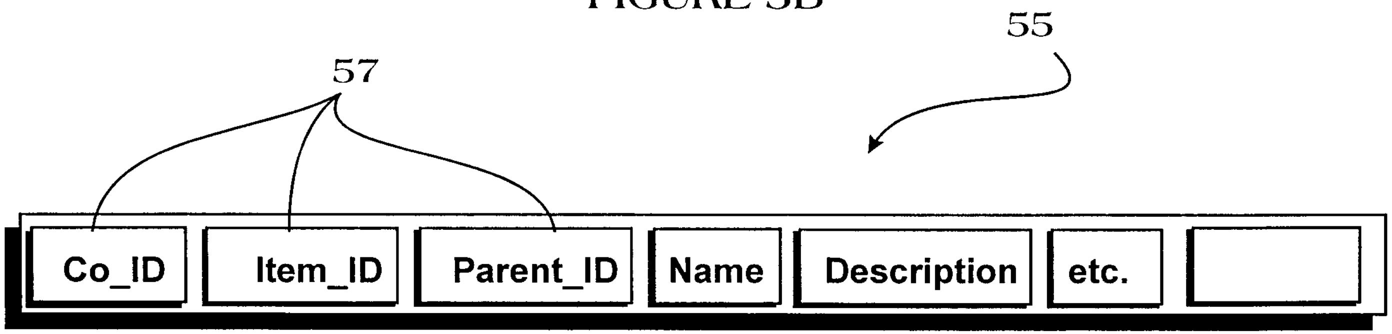


FIGURE 5C

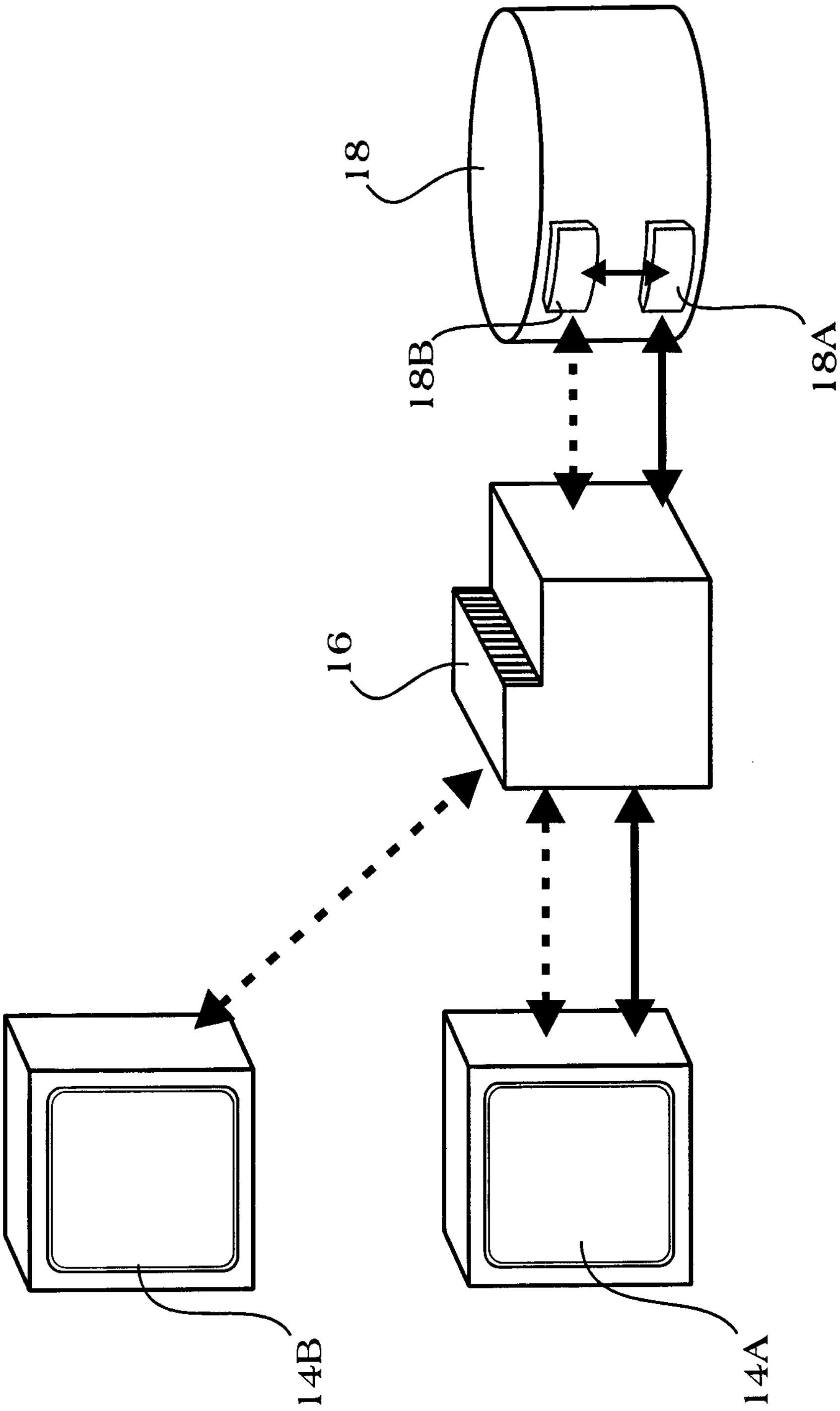


FIGURE 6

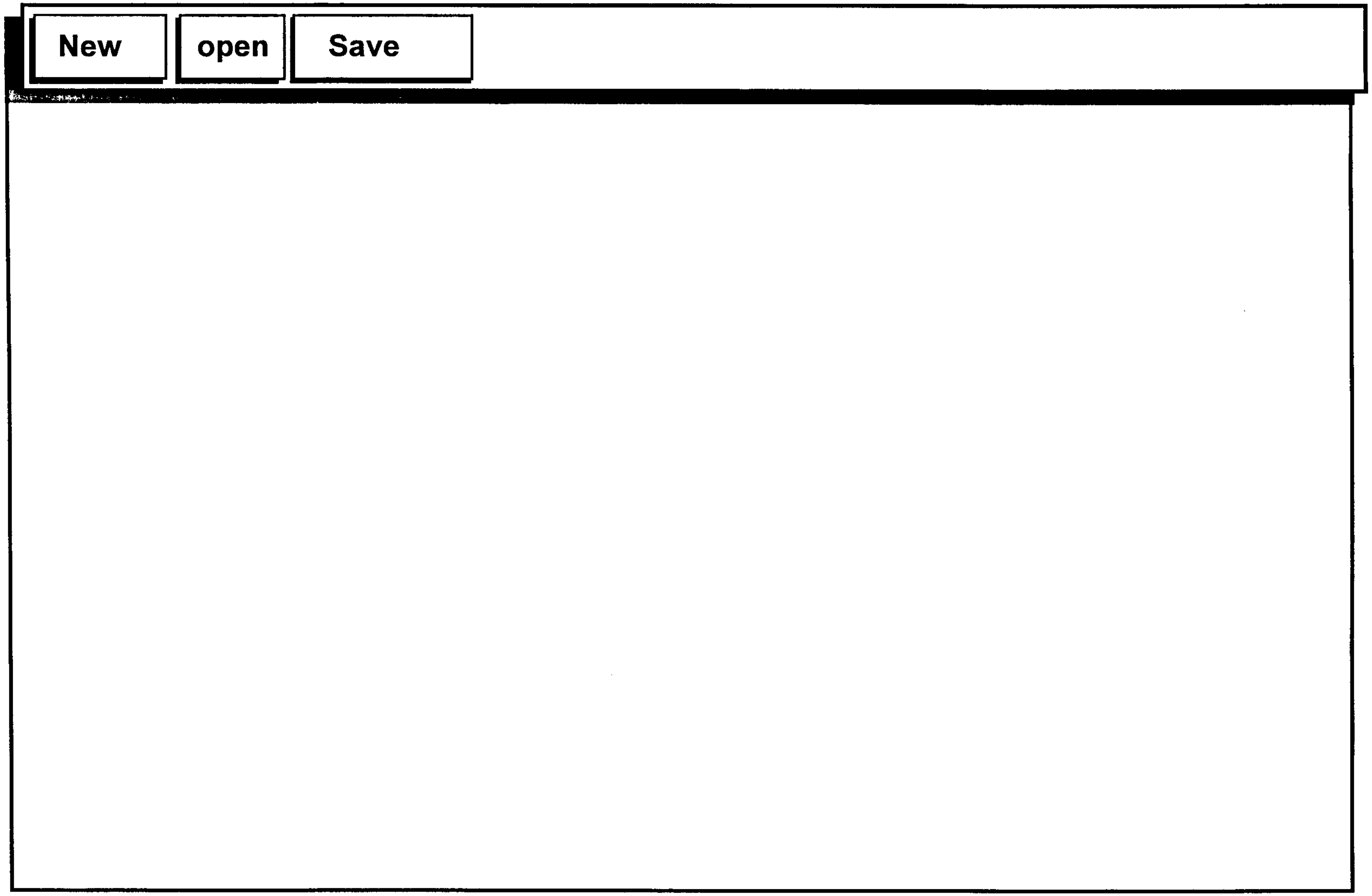


FIGURE 7

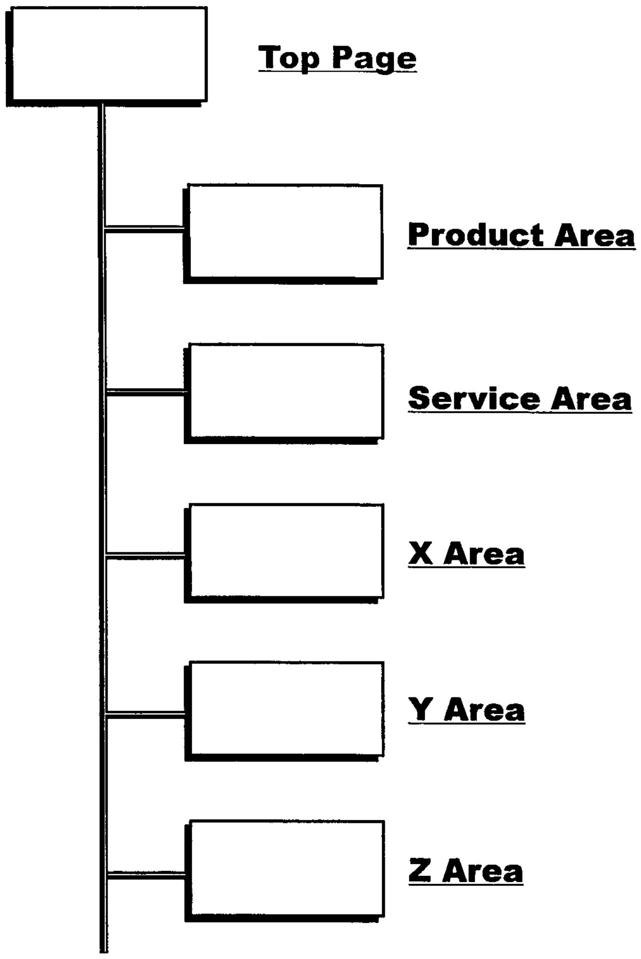


FIGURE 8A

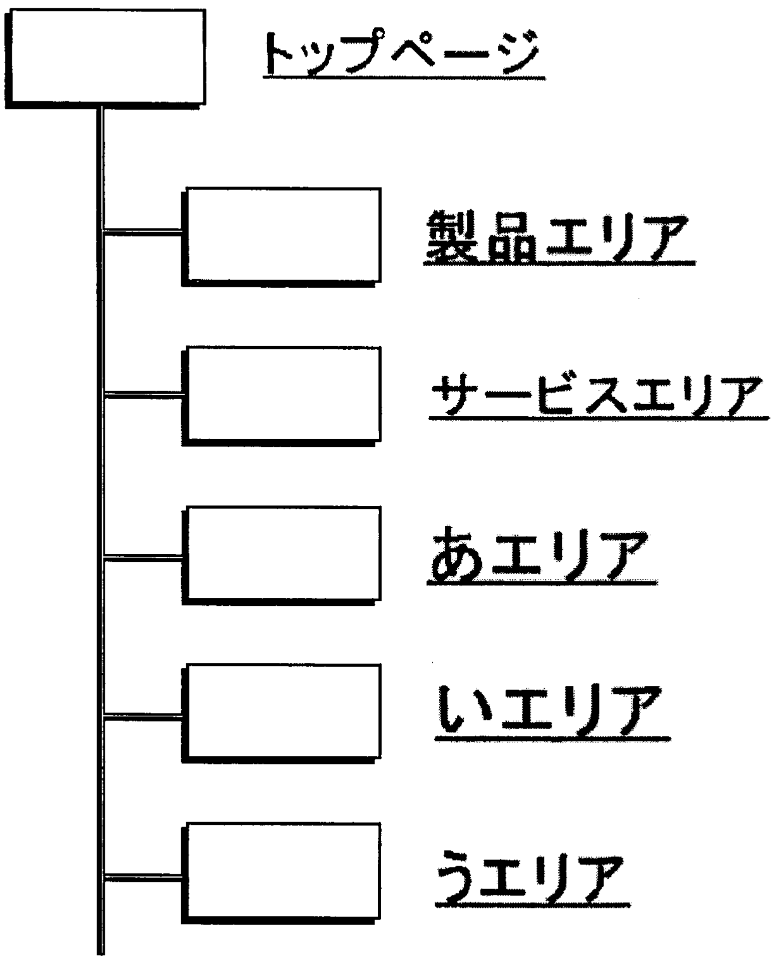


FIGURE 8B

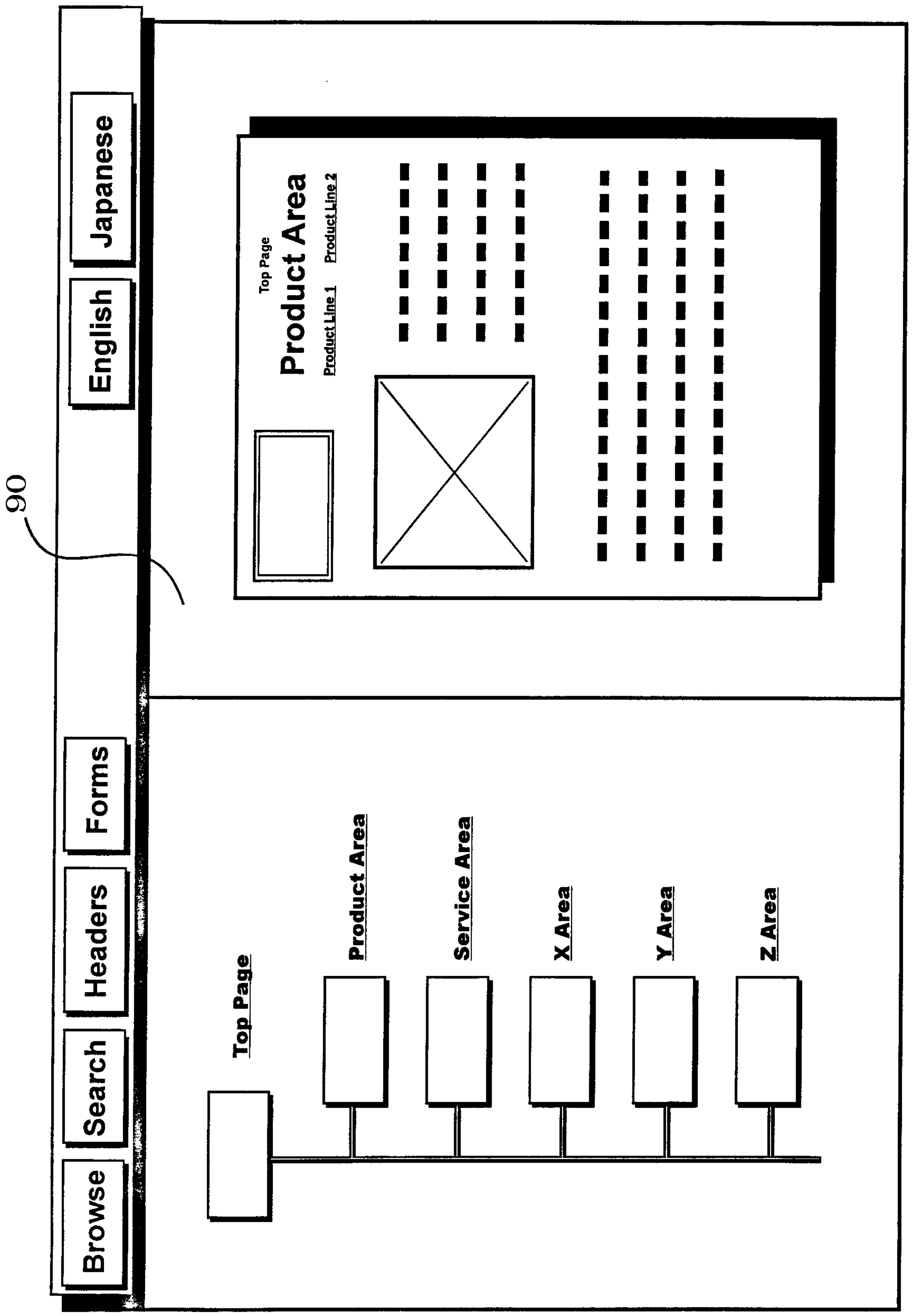

Browse
Search
Headers
Forms
English
Japanese
90
Top Page
Product Area
Service Area
X Area
Y Area
Z Area
Top Page
Product Area
Product Line 1
Product Line 2

FIGURE 9

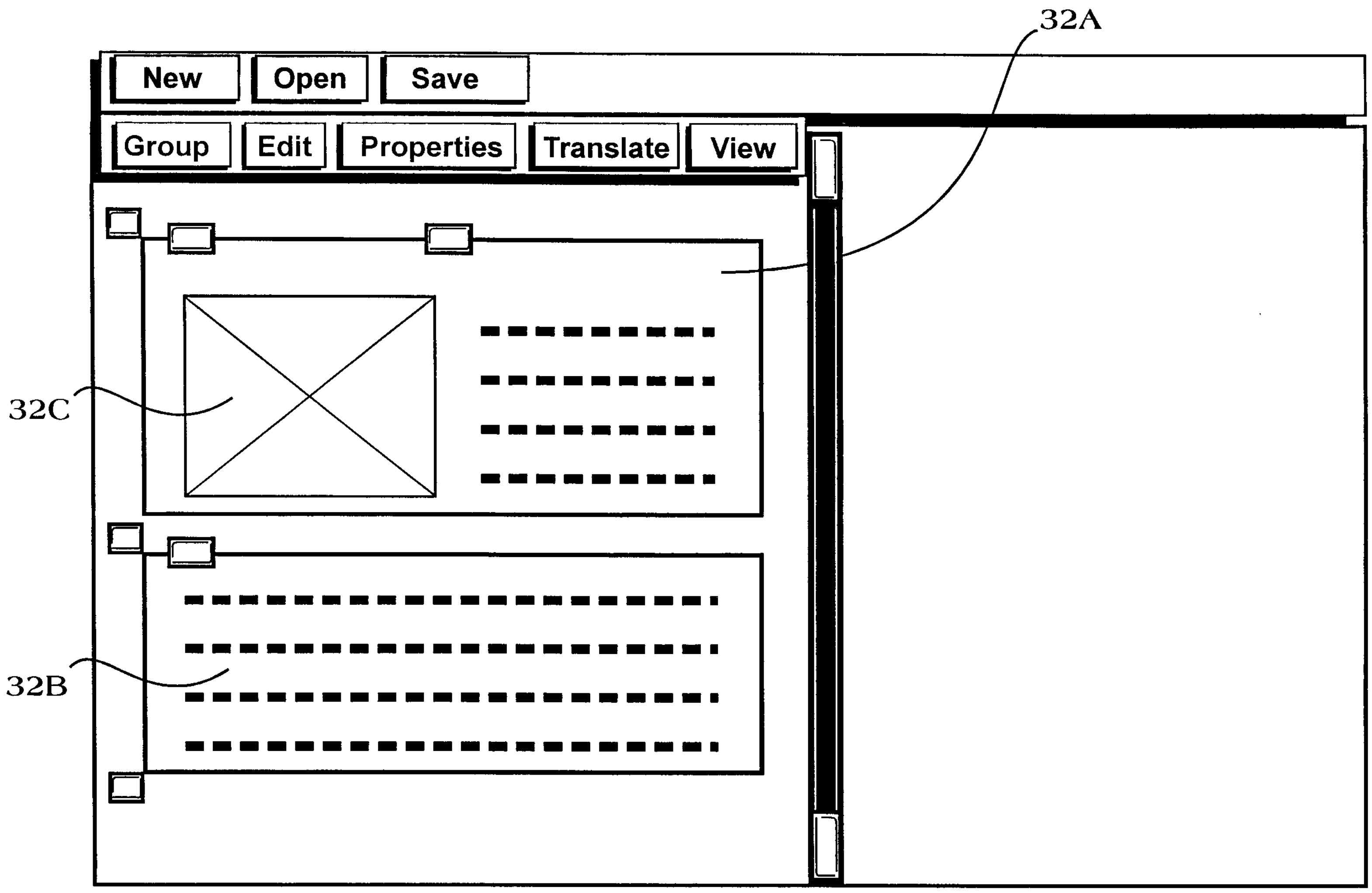
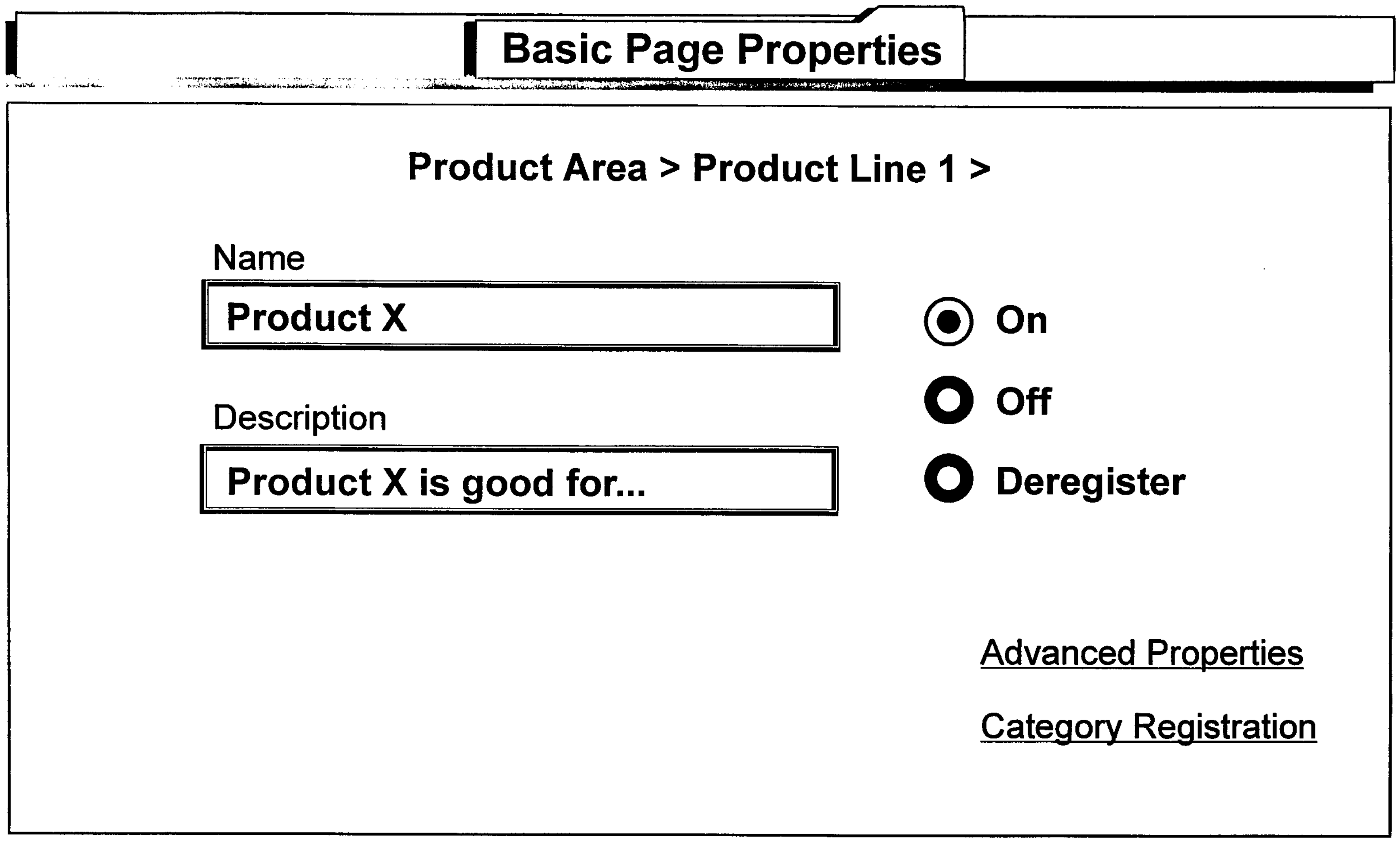

32A
New
Open
Save
Group
Edit
Properties
Translate
View
32C
32B

FIGURE 10



Basic Page Properties

Product Area > Product Line 1 >

Name

Product X

Description

Product X is good for...

● On

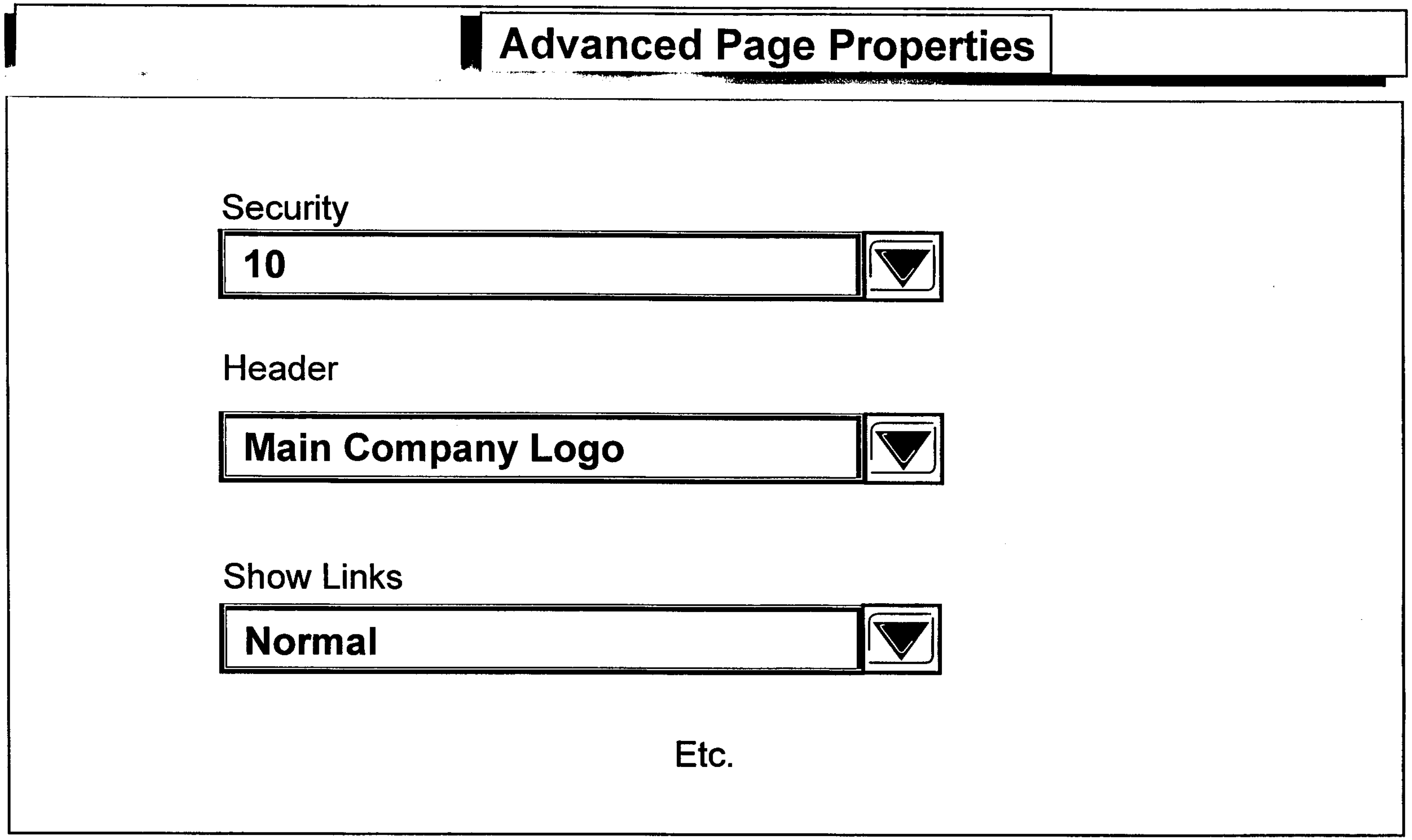
○ Off

○ Deregister

Advanced Properties

Category Registration

FIGURE 11



Advanced Page Properties

Security
10 ▼

Header
Main Company Logo ▼

Show Links
Normal ▼

Etc.

FIGURE 12

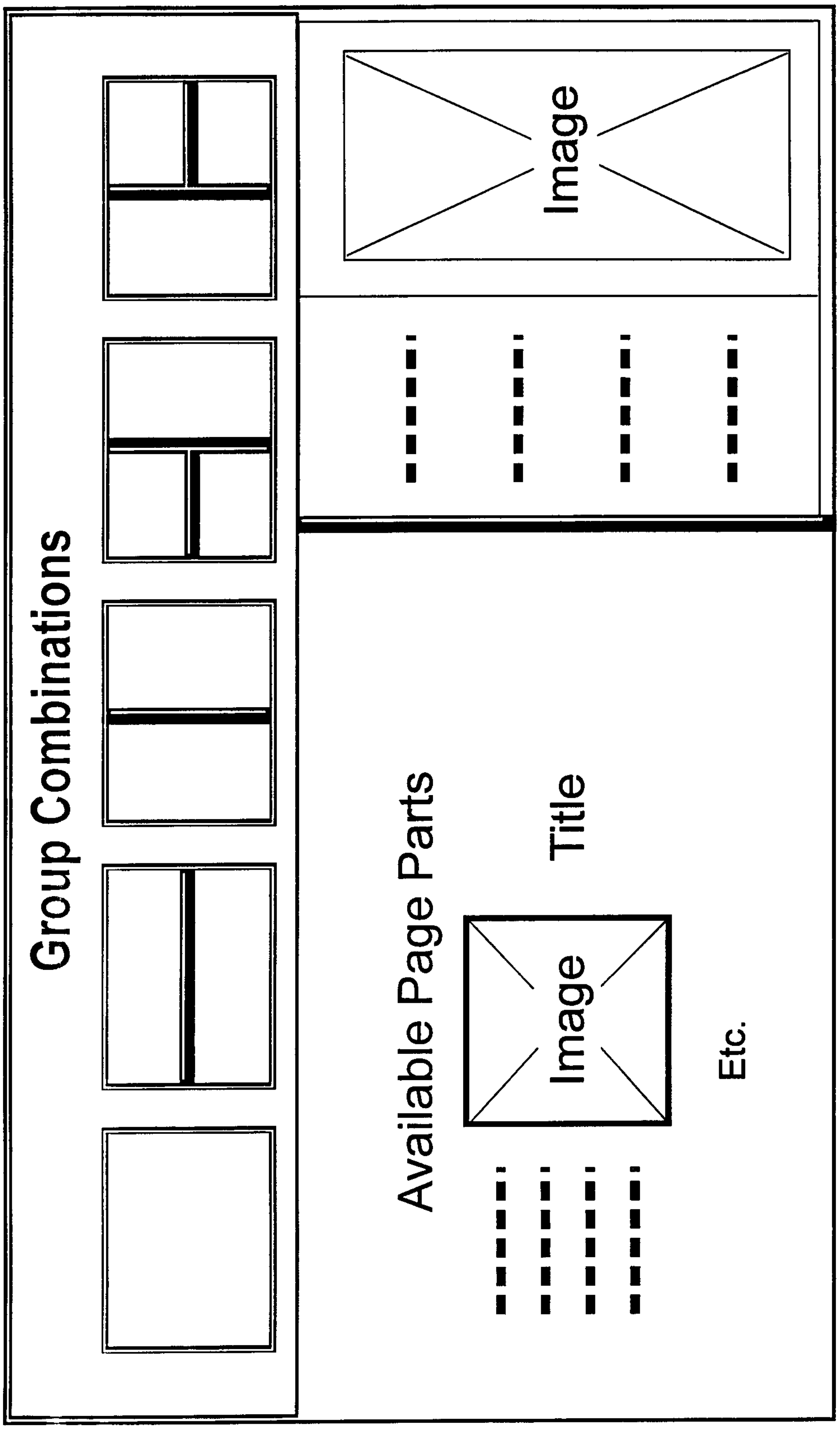

Group Combinations
Available Page Parts
Image
Title
Etc.
Image

FIGURE 13

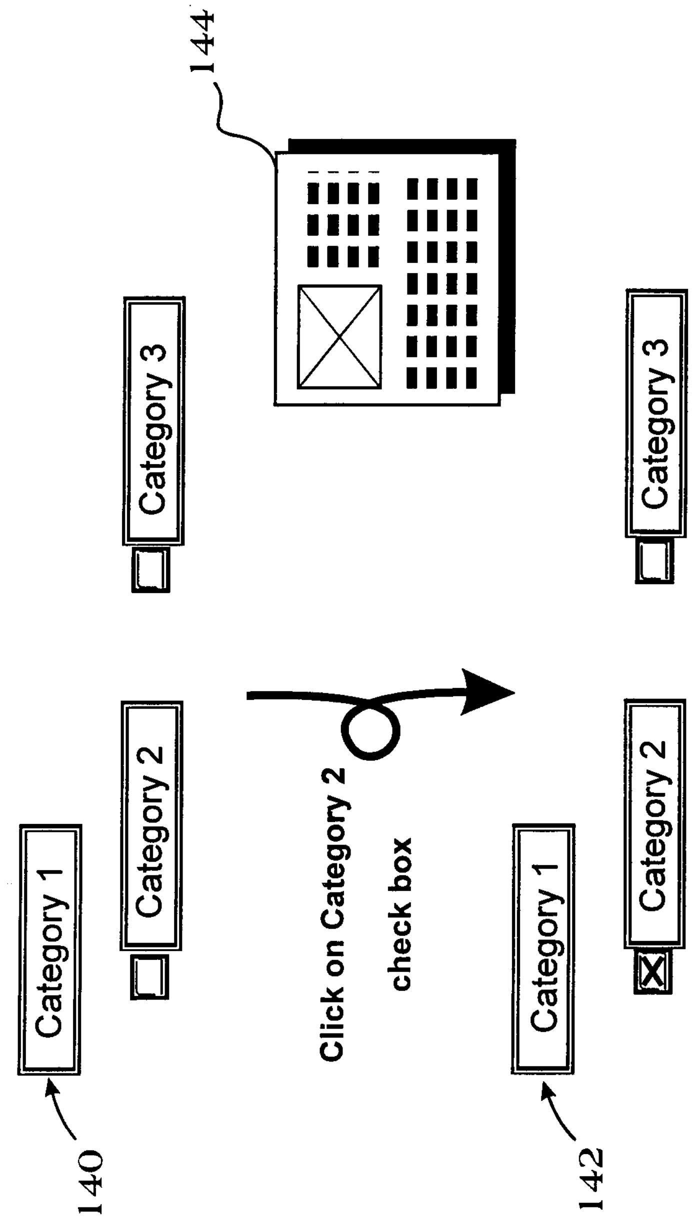
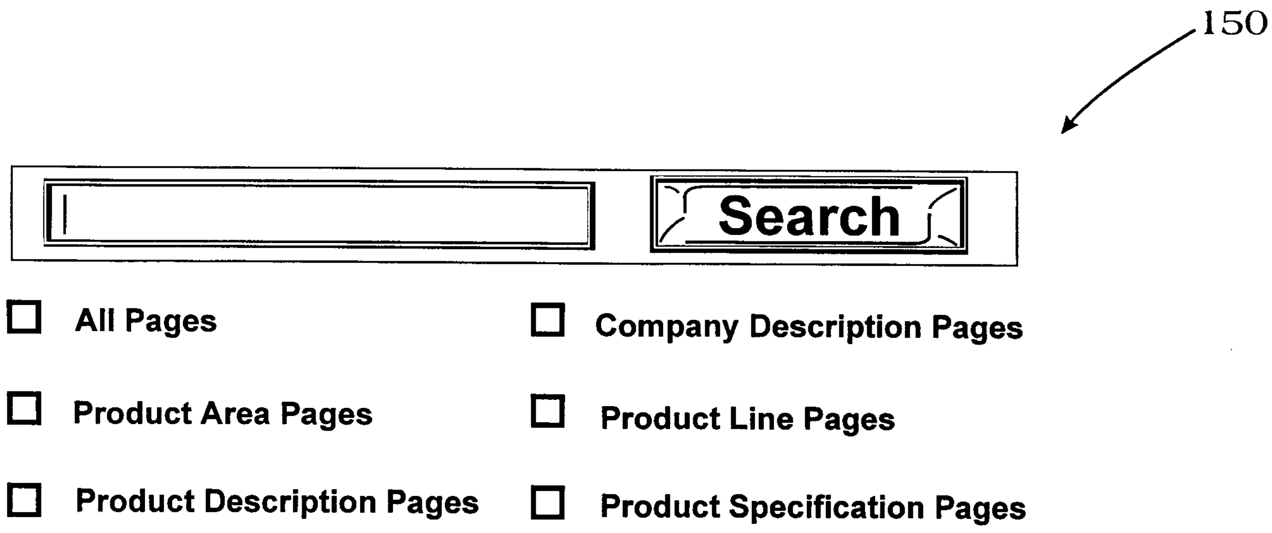

140
Category 1
Category 2
□
Category 3
□
Click on Category 2 check box
144
142
Category 1
Category 2
☒
Category 3
□

FIGURE 14


150
Search
□ All Pages
□ Company Description Pages
□ Product Area Pages
□ Product Line Pages
□ Product Description Pages
□ Product Specification Pages

Check box (es) of pages to be searched against

FIGURE 15

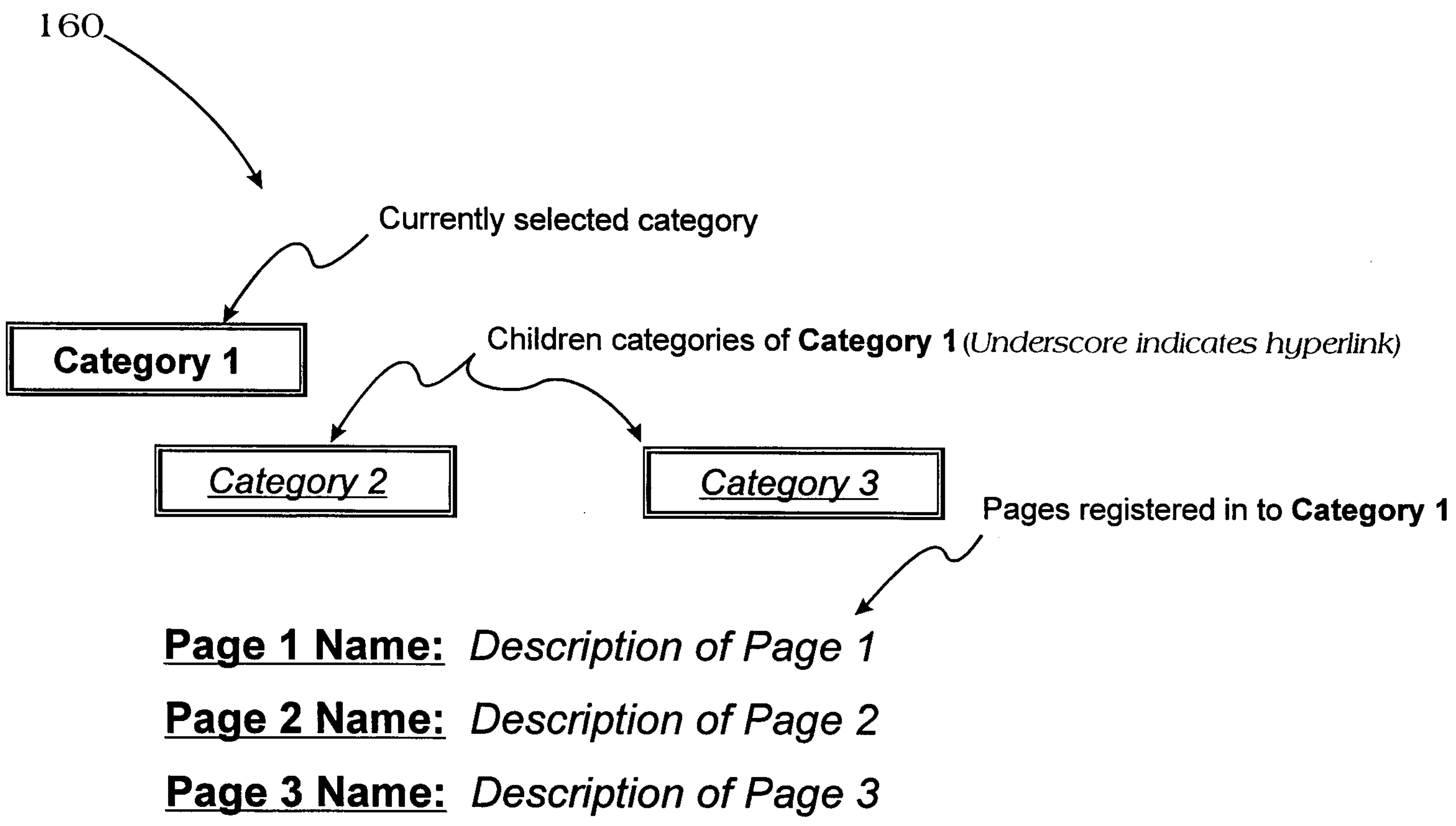

160
Currently selected category
Category 1
Children categories of **Category 1** *(Underscore indicates hyperlink)*
Category 2
Category 3
Pages registered in to **Category 1**
Page 1 Name: Description of Page 1
Page 2 Name: Description of Page 2
Page 3 Name: Description of Page 3

FIGURE 16