RESOURCE AND PROJECT MANAGEMENT SYSTEM

Related Applications

The application claims the benefit of U.S. Provisional Application No. 60/094,912, filed July 31, 1998, and incorporated herein by reference.

Technical Field

The present invention generally relates to management of projects in the development of graphic images, and more particularly to tracking and managing of projects and brand identity in the development of graphic images for product packaging within the pre-press industry.

Background of the Invention

There are a variety of software products designed specifically to manage a wide range of file formats. Aldus Fetch was the first product of this type to appear and was for use on an Apple MACINTOSH computer. It was designed to catalog a variety of image and application file formats such as EPS, TIFF, Adobe ILLUSTRATOR files, and other proprietary and non-proprietary file formats. Another prior art product is Kudos Image Browser. This product was simpler to use, but not as feature rich as Aldus Fetch. Both of these products could manage files for an individual user or a small group of users, but did not expand into a workgroup or company very well, let alone an enterprise.

CUMULUS was a product designed to solve the inadequacies of Aldus Fetch and Kudos Image browser and represented a beginning of the next generation of image management products. Cumulus used a client/server architecture to expand image management into the workgroup and company. Instead of opening a file on a local hard drive that stored all the image management information, the Cumulus client software connected to a database hosted by the Cumulus server software. Cumulus also provided

a hierarchal keyword structure to make it easier to organize and manage the keywords used to describe images. Though Cumulus made it easy to share this image information with an entire company, its performance degraded as the number of images in the database grew.

All of the previously mentioned products use proprietary database strategies. This makes those products extremely difficult to integrate with other products, due to the inability of a user to obtain the stored information directly. These products are also very file centric. In other words, these products can only catalog files, generate previews, and extract file specific data, such as creation date, file type, file size, etc. The products had no way to store production information pertaining to those files, such as: information pertaining to which jobs the files were used in, or what fonts/type faces and images were used to create the application files. In practice, this information, or meta data, is the most important.

The next generation of image management products incorporated prepress workflow management tools. This class of products included: Architype's Media Bank and Media Assets. These products continued to use proprietary database technologies, but incorporated workflow enhancements to help facilitate prepress production, specifically in the catalog market. These products however were very expensive. Additionally, these products are geared toward the catalog production market, which does not primarily use the same software tools as the packaging market.

Another downside to these products is that they require conformity to their approach of managing the pre-press workflow. This offers very little flexibility for an industry that is so heavily influenced by the rapid changes in technology.

Finally, the latest class of image management products are based on open standards-based technologies. Telescope from North Plaines Systems and Imation's Media Manager are two products that fall into this category. The primary differentiating feature of these products is that in addition to storing file information such as file name, file size, file type, creation date, etc, they also have fourteen customizable fields of

information. Both of these products store the file information and custom fields in an SQL compliant database such as Microsoft SQL Server, or Oracle. This allows for direct database access with custom applications or database reporting tools such as Crystal Reports, or Microsoft Query. This also allows for delivery of file information from within a Web browser.

However, despite the open standards-based nature of Media Manager and Telescope, these products still fall short in one area. They are all file centric products. In other words, in order to enter any data in the database, you must first catalog a file. Unfortunately, there is substantial amount of meta data that exists prior to the creation of a file and independent of a file.

The present invention is directed to solving the problems of the prior art. The present invention uses a standard relational database, but increases the functionality beyond the prior art by creating database structures that can store meta data without requiring the presence of a file or image. The present invention also provides an advanced security model. Furthermore, the present invention allows the use of APIs (Application Programming Interfaces) which allows third parties to create custom searches or functions that act on the contents of a shopping cart.

The present invention also allows an unlimited number of customizable data fields at the brand, product, job, project, file, event, job, and user areas of each, within the database. The present invention also implements a distributed file management approach. This means that files are not stored in a database, but rather the database keeps track of what file server the files reside on. Therefore, files could be located on one computer or on a network of computers anywhere in the world.

Finally, the present invention is a workflow tool designed to manage an entire production process from product conceptualization and design, to prepress production, to the converting of printed packages and promotions using common terminology. These advantages, and other advantages provided within the present specification, solve these and other problems.

Summary of the Invention

The present invention provides a system for maintaining, accessing, and archiving digital assets. In an embodiment, the system integrates file and image management functions with the storage of information about the files and images. In addition to providing access to the graphic assets themselves, the system also associates the assets with other information whereby a user can track completed work and define new projects. Further, system information is stored in a data hierarchy that is complimentary to the natural business processes.

Brief Description of the Figures

FIGURE 1 is a functional block diagram of an embodiment of an operating structure suitable for implementing the present invention and having a display;

FIGURE 2 is an illustration of a screen provided on the display of FIGURE 1 and having a "Navigate" link;

FIGURE 3 is an illustration of a brand hierarchy structure for organizing asset data associated with brands, products, jobs, and files;

FIGURE 4 is an illustration of a project hierarchy structure for organizing production data associated with projects, jobs, and events;

FIGURE 5 is an illustration depicting an intersection at the job level between the hierarchies of FIGURES 3 and 4;

FIGURE 6 depicts a top-level brand page provided on the display of FIGURE 1 and having sub-brand links for a "Consumer Brand Name" and an "Institutional Brand Name";

FIGURE 7 depicts a Brand List page having sub-brands links and being displayed in response to selecting the sub-brands link for the "Consumer Brand Name" in FIGURE 6;

FIGURE 8 depicts a Brand List page having sub-brands and product links and being displayed in response to selecting one of the sub-brands links in FIGURE 7;

FIGURE 9 depicts a Product List page having sub-products links and being displayed in response to selecting one of the product links in FIGURE 8;

FIGURE 10 depicts a Product List page having jobs links and being displayed in response to selecting one of the sub-products links in FIGURE 9;

FIGURE 11 depicts a Job List page having a event links and files links and being displayed in response to selecting one of the jobs links of FIGURE 10;

FIGURE 12 depicts a File List page and being displayed in response to selecting on of the files links of FIGURE 11;

FIGURE 13 is a top-level Navigation page containing a plurality of objects and being displayed in response to a user selecting a "Navigate" link of FIGURE 6;

FIGURE 14 is a Navigation page wherein one object of FIGURE 13 has been expanded in response to a user selecting the object;

FIGURE 15 is a Navigation page depicting an object expansion of FIGURE 14 in response to a user selection;

FIGURE 16 is a Navigation page depicting an object expansion of FIGURE 15 in response to a user selection;

FIGURE 17 is a Navigation page depicting an object expansion of FIGURE 16 in response to a user selection;

FIGURE 18 is a Navigation page depicting an object contraction of FIGURE 17 in response to a user selection;

FIGURE 19 is a Navigation page depicting an object expansion of FIGURE 18 in response to a user selection;

FIGURE 20 is a Search page for accepting a plurality of search and display criteria and being displayed in response to a user selecting a "Products" link of FIGURE 6;

FIGURES 22 through 24 illustrate the search criteria supported by the Search page of FIGURE 20;

FIGURE 25 through 26 illustrate the display criteria supported by the Search page of FIGURE 22;

FIGURE 27 illustrates an example of a search using the Search Page of FIGURE 20;

FIGURE 28 depicts a Product Search List and being displayed as a result of selecting a "Search" link within the Search page of FIGURE 27;

FIGURES 29 through 34 are Advanced Search pages that are displayed, at least in part, in response to selecting an "Advanced" link within the Search page of FIGURE 20;

FIGURE 35 is a Detail page that is displayed as a result of a user selecting a an object displayed on another page;

FIGURE 36 is an Edit Brand page that is displayed as a result of a user selecting an "Edit" link within the Detail page of FIGURE 35;

FIGURE 37 is an Attribute List page that is displayed as a result of a user selecting an "Asset Attribute" link within ;

FIGURE 38 is an Attribute Detail page that is displayed in response to a user selecting an attribute in the Attribute List page of FIGURE 37;

FIGURE 39 is an Edit Attribute page that is displayed in response to a user selecting an "Edit Attribute" link in the Attribute Detail page of FIGURE 38;

FIGURE 40 is an Add Property page that is displayed in response to a user selecting an "Add Property" link in the Attribute Detail page of FIGURE 38;

FIGURE 41 is a Remove Property page that is displayed in response to a user selecting a "Remove Property" link in the Attribute Detail page of FIGURE 38;

FIGURE 42 is an Add Attribute page that is displayed in response to a user selecting an "Add Attribute" link in the Asset Attribute page of FIGURE 37;

FIGURE 43 is another Add Attribute page that is displayed in response to a user selecting a “Search” link in FIGURE 42;

FIGURE 44 is an "Add New Attribute" page that is displayed in response to a user selecting a "New Attribute" link in either FIGURES 42 or 43;

FIGURE 45 is a "Remove Attribute" page that is displayed in response to a user selecting a "Remove Attribute" link in the Asset Attribute page of FIGURE 37;

FIGURE 46 is a "Contact List" page that is displayed in response to a user selecting a "Contact" link in the Detail Page of FIGURE 35;

FIGURE 47 is a data table provided by the "File List" page of FIGURE 12 wherein selections made by a user;

FIGURE 48 illustrates a "Contents of Primary Cart" page displayed in response to a user selecting a "Shopping Cart" link in FIGURE 12;

FIGURE 49 is a "Cart List" page that is displayed in response to a user selecting a "Carts" link in FIGURE 48;

FIGURE 50 is an "Add Cart" page that is displayed in response to a user selecting an "Add Cart" link in FIGURE 49;

FIGURE 51 is an illustrative "Cart List" page that is displayed in response to a user selecting an "Add Cart" link in FIGURE 50;

FIGURE 52 is a "Contents of a User's Cart" page that is displayed in response to a user selecting a user defined cart link in FIGURE 51;

FIGURE 53 is a "Load Files into a User's Cart" page that is displayed in response to a user selecting a "Load" link in FIGURE 52;

FIGURE 54 is a "Contents of a User's Cart" page that is displayed in response to a user selecting a cart link in "Load Files into a User's Cart" page of FIGURE 53;

FIGURE 55 is a "Project List" page that is displayed in response to a user selecting a "Projects" link in FIGURE 6;

FIGURE 56 is an "Add Project" page that is displayed in response to a user selecting an "Add Project" link in the "Project List" page of FIGURE 55;

FIGURE 57 is an "Project Detail" page that is displayed in response to a user selecting an project detail link in the "Project List" page of FIGURE 55;

FIGURE 58 is an "Event List" page that is displayed in response to a user selecting an "Events" link in the "Project Detail" page of FIGURE 55;

FIGURE 59 is an "Edit Project" page that is displayed in response to a user selecting an "Edit" link in the "Project Detail" page of FIGURE 57;

FIGURE 60 is a "Contact List" page that is displayed in response to a user selecting a "Contacts" link in the "Project Detail" page of FIGURE 57;

FIGURE 61 is an "Event Detail" page that is displayed in response to a user selecting an event name link in the "Event List" page of FIGURE 58;

FIGURE 62 is a "Login" page that is provided on the display of FIGURE 1 when a user wants to enter the system;

FIGURE 63 is an "Administration" page that is provided in response to a user selecting an "Admin" link that can be provided on the "Top-Level" page of FIGURE 6;

FIGURE 64 is a "Company List" page that is provided in response to a user selecting a "Companies" link in the "Administration" page of FIGURE 63;

FIGURE 65 is an "Add Company" page that is provided in response to a user selecting an "Add Company" link in the "Company List" page of FIGURE 64;

FIGURE 66 is a "Company Detail" page that is provided in response to a user selecting a company name link of the "Company List" page of FIGURE 64;

FIGURE 67 is an "Edit Company" page that is provided in response to a user selecting an "Edit" link in the "Company Detail" page of FIGURE 66;

FIGURE 68 is a "Contact List for a Company" page that is provided in response to a user selecting a "Contacts" link in the "Company Detail" page of FIGURE 66;

FIGURE 69 is a "Contact Detail" page that is provided in response to a user selecting a contact name link in the "Contact List for a Company" page of FIGURE 68;

FIGURES 70a and 70b is a "Edit Contact" page that is provided in response to a user selecting an "Edit" link in the "Contact Detail" page of FIGURE 69;

FIGURES 71a and 71b is an "Add Contact" page that is provided in response to a user selecting an "Add User" link in the "User List for a Company" page of FIGURE 72;

FIGURE 72 is a "User List for a Company" page that is provided in response to a user selecting an "Users" link in the "Company Detail" page of FIGURE 66;

FIGURE 73 is a "Group List" page that is provided in response to a user selecting a "Groups" link in the "Administration" page of FIGURE 63;

FIGURE 74 is a "Group Detail" page that is provided in response to a user selecting a group detail link in the "Group List" page of FIGURE 73;

FIGURE 75 is an "Edit Group" page that is provided in response to a user selecting an "Edit" link in the "Group Detail" page of FIGURE 74;

FIGURE 76 is a "Code Tables" page that is provided in response to a user selecting a "Code Tables" link in the "Administration" page of FIGURE 63;

FIGURE 77 is a "Login List" page that is provided in response to a user selecting a "Logins" link in the "Administration" page of FIGURE 63; and

FIGURE 78 is a "Lock List" page that is provided in response to a user selecting a "Locks" link in the "Administration" page of FIGURE 63.

Detailed Description

While this invention is susceptible of embodiment in many different forms, there is shown in the drawings and will herein be described in detail a preferred embodiment of the invention with the understanding that the present disclosure is to be considered as an example of the principles of the invention and is not intended to limit the broad aspect of the invention to only the embodiments illustrated.

The present invention provides a system for keeping track of and managing projects in the development of graphic images for product packaging within the pre-press industry. The present invention also provides a system for managing assets used in the development of graphic images for product packaging.

One embodiment of the present invention can be described in the form of a system employing a communications server software 11, client system software 12, and a SQL database engine 13 each operating on computers 15, 16, 17 on a computer network 19, as shown in FIGURE 1. The communications server software of the preferred embodiment can be any server which is capable of an HTTP (hyper-text transfer protocol) connection, such as communications server software sold by Microsoft, Inc. or Netscape, Inc. The client software of the preferred embodiment can be any software capable of an HTTP connection. However, because the present embodiment incorporates frames and script code written in JavaScript, client software capable of displaying HTML (hypertext markup language) which incorporates frames and JavaScript is necessary, such as Microsoft Internet Explorer v4.0 and greater or Netscape Navigator v3.0 and greater.

The client computer 15 includes conventional input devices 14 and a display 18. The input devices 14 can include a conventional keyboard and a mouse for allowing a user to select items on the display 18. The display 18 can be a conventional cathode ray tube display or any other suitable device providing a user with a visual display.

The present embodiment allows for flexibility of setup options. The user of the present system could either operate and maintain the communications server and SQL database engine himself, or he could use the communications server and SQL database engine of a specialist who has the training and resources to maintain the system.

Additionally, because the computer code of the present embodiment resides on a communications server, the software is easily updated when product enhancements are created. This eliminates the need to update the software on each computer which runs the system. Also, due to the high level of availability of the client software, the system can be operated from a computer almost anywhere in the world without installing special client software, as long as the communications server is located on a wide area network, such as the Internet. However, if desired, the system can be stored and operated on a single computer platform. Moreover, the computer code can be loaded onto the platform from a removable transfer media such as a compact disk (CD) or the like.

The present invention preferably stores user data hierarchically. This means that instead of placing all of the data (images, attributes, event lists) into a single large container, the data is organized and stored in a tree-like structure whose complexity varies with the complexity of the products stored. Preferably, there are actually two simultaneous data hierarchies at work--a brand hierarchy and a project hierarchy--each of which offers a different view of the data. This structure not only provides an efficient method for organizing the data but also mimics the relationships found in the standard business cycle.

Turning to FIGURE 2, an illustration of a screen for providing a user interface is depicted. In an embodiment, all user selected screens are shown on the visual display 18 (FIGURE 1) and have the same basic structure within the present invention with each screen 20 including a search frame 21, a control frame 22, and a data frame 23. Accordingly, the frames 20, 21, 22 are displayed together on the display 18 by the client software 12. The words "screen" and "page" as used throughout this application shall be construed to referred to the frames 20, 21, 22.

The underlining of words in FIGURE 2 indicates that the word is a hyperlink. Hyperlinks may be selected using input devices 14 (FIGURE 1) such as a mouse or keyboard attached to a computer running the client software. In this disclosure, the terms 'click' and 'clicking' shall mean selecting a hyperlink with a mouse or keyboard. Upon clicking on a hyperlink, the user activates a functionality of the system.

The search frame 21 of screen 20 provides access to search functions that can be performed by clicking one of the buttons 24. The search frame 21 also provides application context by indicating the area in which a user is currently working. For example, the area may be indicated by highlighting a button.

The control frame 22 contains links 26 representing the operations to be performed. The control frame 22 is context-sensitive and thus changes based on the area of the application in which a user is working. At the top are links 26 that depend upon the contents of the data frame 23. Beneath these, and can be displayed in a different color, are links that apply to the application as a whole. These links are not context-sensitive and preferably are available at all times. As shown in FIGURE 2, the control frame 22 may include a thumbnails link 29 wherein, by clicking on the link, any thumbnail images associated selected objects are displayed in the data frame 23.

The data frame 23 is the main body of the page 20. The data frame 23 contains the data with which a user is currently working. Accordingly, the results of any performed operation is displayed in the data frame 23.

Preferably, every system user has an account through which he or she can log into and access the system application. A user account preferably contains information about the user including login ID, password, and email address. This information can be accessed by clicking a "User Info" button 49 in the control screen 22.

In an embodiment, users are allowed to modify at least some of their account information. For example, users can update their email address or phone number whenever this information changes. Preferably, the only user account values that a user

cannot change are login ID and company name. However, it is desired that an asset manage be allowed to change these values.

As will be appreciated by those skilled in the art, brand hierarchy is asset-oriented. Its purpose is to create a structure to logically organize the asset data associated with brands, products, jobs, and files as shown in FIGURE 3. In this example, the "Krunchy" brand 31 has two product lines 32a,32b: "Pretzels" and "Chips." Since there is one variety of "Pretzels," the hierarchy goes directly from brand 31 to product 32a to job 34a (i.e., "Job 206"). However, since there are three varieties of "Chips," the hierarchy goes from brand 31 to product 32b to sub-product 33a to job 34b (i.e., "Job 226"). A brand hierarchy 30 can have as many levels of brands 31, sub-brands, products 32, and sub-products 33 as needed to properly organize its contents.

Looking again at the example of FIGURE 3 and going further down the hierarchy 30, both the "Pretzels" product 32a and the "Potato" product 32b have jobs 34 associated with them. Each job 34 has a set of files 35 associated with it that are used when producing the job.

Project hierarchy is production-oriented. Its purpose is to create a structure that organizes the production data associated with projects, jobs, and events as shown in FIGURE 4. In this example, the "Winter Promo" project 37 contains two jobs 34b,34c, each of which contains a set of events 38 that must take place before the job is completed. The project also contains a set of events 39 that pertain to the project itself. Like the brand hierarchy 30, the project hierarchy 36 can be broken down into as many levels of organization as needed. Each project can contain several levels of sub-projects before specifying the required list of jobs and events.

As depicted in FIGURES 3 and 4, both the "Krunchy" brand hierarchy 30 and the "Winter Promo" hierarchy 36 contain "Job 226" (i.e., reference number 34b). This is because the two hierarchies intersect at the job level.

Jobs can be located from both an asset perspective and a production perspective. From an asset perspective, a job is a collection of required data elements--

the files. From a production perspective, a job contains a series of steps that must occur from start to finish--the events. Since jobs have this dual perspective they are included in both hierarchies and present a place for crossing over from one hierarchy to the other.

FIGURE 5 depicts the intersection at the job level between the hierarchies of FIGURES 3 and 4. "Job 226" (i.e., reference number 34b) is part of both the "Winter Promo" project 37 and the "Krunchy" brand 31. When viewing the events associated with "Job 226" 34b, a user observes the job from the project perspective; when viewing the files associated with the job, a user observes the job from the project perspective. The user can access the "Job 226" 34b by traveling down either hierarchy. And, the user can access both the events and the files associated with the job, regardless of which hierarchy the user travels down. This dual hierarchy, intersecting at the job level, is represented by the layout of the buttons in the Search frame 21 in the user interface 20 as explained in detail further herein.

As previously indicated above, the present invention employs three main functional areas: 1) brand or asset management, 2) project management, and 3) file exchange. In an embodiment the present invention also provides an administrative function for configuring system parameters and security.

Brand management allows a user to maintain and organize the files associated with particular brands and products. As indicated above, a brand is a product family. A brand may further contain more brands that are distinct brands within the product family. Eventually, however, a brand or sub-brand will contain at least one product. Within a particular product there may exist more products. Eventually, however, a product or sub-product will contain at least one job. Jobs contain files to be associated with a job and/or events to occur within the job. Files are the design elements of the packaging of a product. A file could be an image, font, document, or any other design element. Events are associated with the design process of a product package and will be further explained in conjunction with the project management function. Brands, products, jobs, files, and events are generally referred to as assets or objects. Because, for example, within a brand

there may exist different types of assets, such as a sub-brand or a product, the contents of an asset will sometimes be referred to generally as a sub-asset.

The brand management section organizes and displays four main types of information: 1) listings of sub-assets, 2) contact maintenance functions, 3) search brand assets functions, 4) asset maintenance functions. The brand management section also provides a simple link to events associated with a particular asset.

Preferably, there are three methods of locating information in the system database: browsing, navigating, and searching. The method selected depends upon both the type of data to be located and what the user already knows about that data.

The browsing method allows a user to proceed, level-by-level, through either the brand hierarchy or the project hierarchy to observe the contents of the hierarchy at each level. Browsing is conceptually similar to using a web browser. Just as one would pick a web site and then focus in on the information desired by clicking the appropriate links, browsing allows a user to pick a brand, then a product underneath it, then a job, and finally a file or event by clicking links to narrow the scope. Each time a user clicks a link to travel deeper into the hierarchy, the contents of the data frame 23 are replaced with new information. Browsing is context-focused and a desirably method to employ if a user determines it helpful to observe detailed descriptions of the objects and visual cues at the user travels down the hierarchy.

The navigating method allows a user to observe an overview of an entire brand hierarchy and then focus in on a particular level of objects. Navigating is conceptually similar to using the MACINTOSH FINDER or MICROSOFT WINDOWS EXPLORER. In either application, a user clicks the expansion icon next to any folder to view a list of its contents. Likewise, when navigating using the present invention, a user clicks on an icon to view the contents of brands, sub-brands, products, sub-product, and jobs in an outline format. Each time a user clicks, the outline is expanded while the previous outline contents remain in data frame 23. Navigating is relationship-focused and a desirable method to employ if a user does not need graphical cues or if a user wants

to understand the overall structure of the system database and observe how different objects relate to one another.

The searching method allows a user to locate an object or set of objects that satisfy a particular set of criteria. Searching is conceptually similar to using a web-based search tool. Just as a user might use a web search tool to locate "pages containing information about digital prepress," searching in the present invention allows a user to locate "all files associated with Brand X jobs that were completed in January of this year." Searching is criteria-focused and a desirably method to employ if a user does not remember where in the data hierarchy an object is stored or if the user wants to locate multiple objects that are not stored together in either data hierarchy.

As stated previously, browsing allows a user to locate an object by traveling down a particular branch of a brand or project hierarchy and examining its contents. At each level, a user can elect to focus further or to examine detail information about the objects. Each time a user requests new information the data frame is updated.

The following depicts illustrative steps for browsing the brand hierarchy in a database entitled - the "Whacky Foods" database. Of course a user's database will contain different information, and the following is for illustrative purposes to depict the capabilities of an embodiment of the browsing tool.

After logging onto the system, a top-level brand page 40 such as that show in FIGURE 6 is provided on the system display. At the top of the page 40 is the search frame 21. In addition to providing access to the search tools, the search frame 21 displays the application context, indicating which area of the data hierarchy a user is currently browsing. For example, the "Brands" button 25 is highlighted in FIGURE 6 since brands are being browsed.

At the top of the data frame 23, the page title 41 indicates the type of objects being browsed. For example, the "Brand List" is indicated in FIGURE 6. The illustrative "Whacky Foods" database used in this description is setup so the top-level brands are categories instead of actual trade name brands. This allows brand and product

information to be organized by the business sectors or product groupings already in place. For illustrative purposes, there are two top-level brands in the "Whacky Foods" database: "Consumer" and "Institutional."

Beneath the title 41 is the wayfinder 42. The wayfinder displays the levels of the hierarchy that form a path to the current object and allows a user to revert to any level by clicking a link. The wayfinder 42 is discussed in detail further herein.

Beneath the wayfinder 42 is the display record popup 43. The display record popup allows a user to display the next set of records if more records are retrieved than can be displayed on the screen at once. The number of records displayed is defined by a system setting and may vary for list, thumbnail, and contact sheet views. For example, if the list display setting is set to "5 records" and the database contained twenty brands, the display record popup initially reads "Display records 1 to 5 of 20 records found." Using the popup 43, a user can then display records 6 to 10, then 11 to 15, and so on.

At the bottom of the page is the data table 44. The data table lists all of the objects in the hierarchy at this level and provides columns 46 of information about each object. Clicking a data table column heading 47 sorts the table, alphabetically or numerically, by the column. The arrow 48 next to the column name indicates whether the column has been sorted in ascending "↑" or descending "↓"order.

The contents of the data table 44 varies depending on the type of object being browsed. For example, a brand displays one set of columns and a job displays a different set of columns. However, the object name 50, manager 51 and list 52 are columns common to many objects' data tables. The object name column 50 (in the case of FIGURE 6, the "Brand Name") contains links 54 denoting the names of the objects. Clicking on the object name link 54 displays the object's detail page, explained in detail further herein, which provides in-depth information about the object.

The manager column 51 contains links 56 denoting the names of people to contact in reference to the object. Clicking the manager link 56 displays a contact detail

page providing information about the manager, including a link that allows sending him or her an email message.

The list column 52 contains sub-asset hyperlinks 58 representing the sub-assets in the data hierarchy (in the case of FIGURE 6, sub-brands) that are beneath the current asset. Accordingly, clicking a list link 58 drills down to the next level in the hierarchy for the selected sub-asset.

Preferably, each list link 58 is a word signifying what type of sub-assets or object the asset contains, such as "sub-brand," "sub-product," or "product." Accordingly, by clicking on the list link 58, the data frame 23 is changed to show a screen depicting the assets or objects within the sub-asset clicked. For example, by clicking the list link 58 of FIGURE 6 for the brand "Consumer," the data frame 23 is changed to that depicted in FIGURE 7. The resulting page displays the associated name brands (i.e., sub-brands) beneath the Consumer brand comprising of, for example, "Cyber Snacks," "Krispy" cereal, and "Youbetcha" vegetables. It should be noted that the present invention facilitates data hierarchies having many levels of brands, sub-brands, products, and sub-products.

Traveling down further into the hierarchy, into the "Youbetcha" brand, is done by clicking the "Sub-Brands" list link 58 in the list column 52 in the bottom row of FIGURE 7. Clicking on the "Sub-Brands" list link 58 for the sub-brand "Youbetcha" results in the data frame 23 changing to depict the sub-assets (i.e, sub-brands) of "Youbetcha," as shown in FIGURE 8. As shown in the list column 52, "Youbetcha" has another level consisting of four sub-brands beneath it: Super Beans, Super Carrots, Super Corn, and Super Peas. Depending on which sub-brand is selected, clicking further down into the hierarchy will take a user to either another level of sub-brands or to a product level.

By clicking on the "Products" list link 58 for the sub-brand "Super Corn," next to the bottom row of table 44, the sub-assets (i.e., products) of "Super Corn" are shown in the data frame 23 as depicted in FIGURE 9. The title 41 is now shown as a

"Product List" and the data frame provides a user access to the top-level "Super Corn" products. Further, just like the "Brand List" pages before (i.e., FIGURES 6-8), the "Product List" page contains a data table 44 listing its contents - the available sub-products. There are four sub-products under the "Super Corn" brand, all of which contain further sub-product links or job links displayed in the List column 52 of a Product List page.

As shown in FIGURE 10, clicking the "Sub-Products" list link 58 of FIGURE 9 in the list column 52 of the "Mexican Super Corn" product displays its sub-products. This page list three sub-products under the Mexican Super Corn product. Furthermore, the products are getting more specific wherein they actually correspond to physical products that might be for sale. Since the products at this level are this specific, they no longer have sub-products beneath them. Instead, they have jobs.

Moreover, some products contain product templates in addition to jobs. These product templates are used to provide storage for the design templates associated with a product.

As shown in FIGURE 11, clicking the "Jobs" list link 58 associated with the "15 oz. Mexican Super Corn" product of FIGURE 10 displays a list of the jobs beneath the product. These three jobs are associated with the "15 oz. Mexican Super Corn product." Each job represents one iteration of the "15 oz. Mexican Super Corn" product label.

In FIGURE 11, the title 41 of the data frame 23 is appropriately changed to "Job List." Further, the data table 44 contains different types of information other than brands or products. Instead, the data table includes: job number, job name, job manager, job language, job due date, job end date, and job status.

As shown in FIGURE 11, the job "15 oz. Mexican Super Corn Baseline" contains two list links - one which indicates "events" and another which indicates "files." Thus, in the list column 52 a user can select a link to view either the files or the events associated with the job. Clicking a "Files" link 62 drills down one level further to the

bottom of the brand hierarchy and displays a File List page of FIGURE 12. Clicking an "Events" link 64 crosses over into the project hierarchy and displays an Event List page.

It should be noted that the intersection of the brand hierarchy and the project hierarchy is provided in FIGURE 11 via the "Files" list link 62 and the "Events" list link 64. Therefore, a user can switch between the two hierarchies at the job level.

As shown in FIGURE 12, clicking the "Files" list link 62 associated with the "15 oz. Mexican Super Corn Baseline" job of FIGURE 11 displays a list of files contained in the job. The title 41 of this page is "File List" and the page lists the files associated with the job and displays their associated software applications, file types, creation dates, and sizes. The job shown in FIGURE 12 has nine files associated with it, but only 4 are initially displayed in the data table. Using the display record popup 43, a user can view the remaining files in the list, first files 5 to 8, then finally file 9.

The first column 66 of the data table 44, Cart, provides a reference to the user's shopping cart comprising a storage area where a user can gather files. The icon 68 on the left of the Cart column 66 indicates whether the file is in or out of the shopping cart. The icon 70 on the right allows a user to move the file into or out of the shopping cart.

It should be noted that files are at the bottom of the data hierarchy. Since there are no objects or assets in the hierarchy that are beneath files, there is no List column in the data table 44 on a File List page of FIGURE 12.

As indicated previously, the wayfinder 42 provides for jumping to a different level of the hierarchy. The wayfinder 42 displays the brand hierarchy from the top of the database (designated by the Whacky Foods, Inc. link) down to the current location. Each level is displayed beneath the previous level, indented, and preceded by one of three icons such as a registered trademark symbol, a cylinder, or a file folder to designate a brand, a product, or a job, respectively.

Clicking a link in the wayfinder 42 jumps to a page representing the contents of the selected object. Clicking a link representing the parent level (displayed in bold type) displays the Detail page, explained in detail further herein, for the parent object.

Clicking the "Consumer" link in the wayfinder re-displays the brands contained in the Consumer category as shown in FIGURE 7 wherein another brand can be selected to continue browsing the brand hierarchy.

As shown above, the present invention allows a user to browse the brand hierarchy. A user simply selects the top level brand to examine, then travels deeper and deeper into the hierarchy by clicking links in the List columns of the displayed data tables. Clicking a brand, product, or job name, instead of a List column link, displays a Detail page, explained in detail further herein, at the selected level instead of drilling deeper into the hierarchy. The wayfinder can be used to back up to any level in the hierarchy.

When working in the project hierarchy, the browse tools operate in the same manner. A user can browse the project hierarchy by clicking the Project link in the Control frame. Once inside the project hierarchy, a user can select a top-level project and then drill down from project, to sub-project, to job, and finally to events by clicking links in the List column of the resulting data tables. As in the brand hierarchy, a user can examine a Detail page, explained in detail further herein, by clicking a project, sub-project, or job name link.

Navigating allows a user to examine the entire system brand hierarchy on one page, in an outline format. By expanding and contracting the outline, a user can examine the relationships among the various objects. In an embodiment, the navigation tool can be used to view the brand hierarchy only, and not the project hierarchy.

Once again an explanation regarding to navigate a brand hierarchy using the present invention is explained using the same illustrative "Whacky Foods" database. Although a user database will contain different information, this example illustrates capabilities of the system navigation tool.

Turning to FIGURE 6, clicking the "Navigate" button 74 in the Control frame 22 starts the navigation tool and displays the top-level brands as shown in FIGURE 13. At the top of the page is the system database name 76 (i.e., "Whacky Foods, Inc."). Beneath the database name 76 is a list of all objects 78 at the top of the brand hierarchy--the top-level brands. Every object 78 displayed is represented by a row containing three elements: the display or expansion icon 80, the object icon 82, and the object name 84.

The first element in each row is the display icon 80. If the icon 80 points to the right, the row is contracted and no levels beneath the object 78 are displayed. If the icon 80 points down, the row is expanded and one or more levels beneath the object 78 are displayed. If contracted, a user can click the icon 80 to display one level under the selected object 78. If expanded, a user can click the icon 80 to contract all levels under the selected object 78.

Following the display icon 80 is the object icon 82. The object icon is used to designate, at-a-glance, the type of object 78 the row represents wherein: a registered trademark symbol designates a brand, a cylinder designates a product, and a file folder symbol designates a job.

Following the object icon 82 is the object name 84, identifying the object. If the object name 84 is displayed as a hyperlink, then clicking the link displays the object's Detail page.

Clicking the display icon 80 preceding the "Consumer" brand displays the sub-brands under the Consumer category as shown in FIGURE 14. In this example, there are three brands displayed beneath the Consumer category: "Cyber Snacks," "Krispy" cereals, and "Youbetcha" vegetables.

Clicking the display icon 80 preceding the "Youbetcha" brand displays the sub-brands beneath it as shown in FIGURE 15. In this example, there are four sub-brands displayed beneath the "Youbetcha" brand: "Super Beans," "Super Carrots," "Super Corn," and "Super Peas."

Clicking the display icon 80 preceding the "Krispy" brand allows a user to examine that part of the brand hierarchy as well. As shown in FIGURE 16, the outline expands to reveal two sub-brands beneath the "Krispy" brand. The "Krispy Flakes" brand can be further expanded to reveal its contents because a display icon is provided. However, since the "Krispy Puffs" brand does not have any objects beneath it in the hierarchy, no display icon precedes it.

By expanding and contracting various objects, a user can examine more of the object hierarchy beneath the "Youbetcha" brand. Accordingly, navigating presents a different picture of the brand hierarchy than browsing does. On a single page a user can not only locate a particular object, but also compare objects. For example, as shown in FIGURE 17, the "Cheese Super Corn" product has one size but the "Mexican Super Corn" has three, and the "Mexican Super Corn" is the only product with a $0.20 coupon" job.

Contracting the display icon preceding the "Mexican Super Corn" product hides all levels beneath it as shown in FIGURE 18. As such, the contents of both the "15 oz. Mexican Super Corn" product and the "Mexican Super Corn" product are hidden.

As shown in FIGURE 19, expanding the "Mexican Super Corn" product again displays the three products beneath it. Notice that when the "Mexican Super Corn" product is again expanded, only one level beneath it is displayed, even though two levels were displayed before it was contracted. As such, is it preferred that contracting an object contracts all levels beneath it, but expanding an object displays only one level beneath it.

It should be noted that while navigating, the present invention preferably allows a user to jump into browse mode at any level of the data hierarchy by clicking an object's name link. Accordingly, doing so opens a Detail page for the selected object, explained in detail further herein.

The above demonstrates navigating the brand hierarchy. In summary, navigating is started by clicking the Navigate button in the Control frame. Once the

Navigate page is displayed, a user can expand or contract the object outline by clicking the display icons--to expand, to contract. Objects not preceded by a display icon have no objects beneath them in the hierarchy. Expanding the outline displays one additional level beneath the selected object. Contracting the outline hides all levels beneath the selected object.

Preferably, the first time a user clicks the Navigate button, the top-level brands are displayed in the data frame. If a user expands several objects, leaves the Navigate page without logging out, then returns to it, the system remembers the state of the data frame when a user last navigated, and displays it in that state. Therefore, if a user has expanded multiple objects, choosing to look at an object's Detail page does not mean that a user will have to re-expand everything when a user resume navigating.

Searching allows a user to locate an object or set of objects that meet a certain description. Specifying search criteria--a set of attributes and properties that characterize these objects, creates this description. A user can search for any type of object (brand, product, job, file, event, or project).

Searching provides a method to use when: a user does not remember where in the data hierarchy an object is stored, when a user knows where an object is stored and wants to zero in on it quickly, or when a user wants to locate multiple objects that are not stored together.

The following example shows how to search for a product in the above referenced illustrative system database--the "Whacky Foods" database. Although a user database will contain different information, the following illustrates capabilities of the system search tool.

Turning to FIGURE 6, clicking the "Products" button 90 in the Search frame 21 displays the Product Search form 92 depicted in FIGURE 20. The search form contains controls that allow a user to define a user search--both the criteria by which to search and the manner in which the results of the search are to be displayed.

The search criteria 94 includes a set of popups 96 and text boxes 98 with which a user can specify the characteristics of the objects to locate. Popups 96 are displayed when a selection must be made from a set of valid options. Text boxes 98 are displayed when any value can be entered.

The display criteria 100 include a set of popups 102 with which a user can indicate how to display the results of the search. A user can sort the results by various attributes in ascending or descending order and a user can display the results in various ways, depending on the type of object for which a user are searching. For example, products can be displayed as lists (similar to a Browse page data table) or as thumbnails (small graphic representations).

When a Search page is open, the "Search" button 104, "Advanced" button 106, and "Reset" button 108 are added to the Control frame 22. Clicking the "Search" button 104 executes a search using the displayed search criteria 94. Clicking the "Reset" button 108 replaces all modified criteria with their default values. Clicking the "Advanced" button 106 displays the "Advanced Search" page on which more complex search criteria can be defined. In an embodiment, if the Advanced Search page is displayed, the Control frame button 106 can read "Simple" instead of "Advanced."

Search criteria are defined by an attribute (value to be checked), an operator (type of comparison to make), and a limiting value (value the attribute is checked against). If a criterion is unnecessary, then a user can select the "(Any)" operator. This results in the entire criterion, including any value entered into the limiting value text box, to be ignored.

The search criteria 94 includes a "Product" criterion 110 that allows a user to search by product name or product number. As shown in FIGURE 21, a user can select either "Name" or "Number" from the first popup 96. Then, a user selects an operator from the second popup such as "(Any)," "Is," or "Begins With." Finally, a user enters a limiting value in the text box 98. Using these options a user can define various

types of product criteria, for example, "Product Name contains corn," or "Product Number begins with 3."

The search criteria 94 of FIGURE 20 also includes a "Product Parent" criterion 112 that allows a user to specify the type of parent objects the found objects must have. Preferably, only parent object types that are valid for the type of object a user are locating are displayed. As shown in FIGURE 22, a user can select the parent object type (i.e., Brand or Product) and attribute (i.e., Name or Number) from the first two popups 96. Then, a user selects an operator from the third popup 96 such as "(Any)," "Is," or "Begins With." Finally, a user enters a limiting value in the text box 98. Using these options a user can define various types of parent criteria, for example, "Parent Brand Name is Youbetcha," or "Parent Product Name contains snack."

The search criteria 94 of FIGURE 20 further includes a "Market" criterion 114 that allows a user to specify products from certain markets that should be included or excluded from a search. As shown in FIGURE 23, a user can select an operator from the first popup 96, either "Is" to include or "Is Not" to include. Then, a user selects a market from the second popup 96 such as "(Any)," "Northwest," "Southeast." Using these options a user can define various types of market criteria, for example, "Market Is Southeast," or "Market Is Not Northwest."

Turning back to FIGURE 20, the search criteria 94 also includes a "Note Text" criterion 116 that allows a user to search for products that have a specified word or phrase in their notes. Turning to FIGURE 24, a user can select an operator from the popup 96 such as "(Any)," "Begins With," or "Contains." Then, a user enters the text for which to search in the text box 98. Using these options a user can define various types of note text criteria, for example, "Note Text Contains modified" or "Note Text Ends with final."

The display criteria 110 of FIGURE 20 includes a "View" criterion 120. The "View" criterion has no effect in determining which objects are found. However, specifying a value for the "View" criterion 120 determines how the results are displayed

upon completion of the search. As shown in FIGURE 25, a user can select "List" from the popup 96 to display the search results as a simple list of object names. Further, a user can select "Thumbnails" from the popup 96 to include a graphical representations of the results.

Preferably, if a user is searching for file objects, the Contact Sheet value is added to the view criterion popup allowing the files a user located to be viewed as part of a contact sheet.

The display criteria 110 of FIGURE 20 also includes a "Sort By" criterion 122. As shown in FIGURE 26, the first popup 96 of the "Sort By" criterion 122 allows a user to select the value by which the results sorted such as "Product No.," "Product Name," "Manager," or "Status." The user can also select the sort order, either ascending or descending, from the second popup 96. Using these options a user can sort the results in various ways, for example, "Sort By Manager in ascending order," or "Sort By Product Number in descending order."

As shown in FIGURE 27, a user can specify criteria to find all products under the "Youbetcha" brand that contain the word "Mexican," returning the results listed alphabetically, by name. This is accomplished first, in the "Product" criterion 110, by a user selecting a "Name" from the attribute popup, selecting "Contains" from the operator popup 96, and typing "Mexican" in the text box. Next, in the "Product Parent" criterion 112, a user selects "Brand" and "Name" from the from the attribute popups, selects "Begins With" from the operator popup, and types "Youbetcha" in the text box. Further, in the "Sort By" criterion 122, a user selects "Product Name" and "Ascending" from the popups.

It should be noted that by selecting the "(Any)" operator in the "Market" and "Note Text" criteria, a user indicates that the products to be located can be in any market and have any text in their notes.

Once the criteria are defined, a search is perform by the user clicking the "Search" button 104 in the Control frame 22. The system searches the database for

products that match the criteria and then displays the results. The time needed to perform a search depends upon the complexity of the search criteria and the amount of information stored in the System database. The results of the search are displayed on the Product Search List page 124 as shown in FIGURE 28. If no objects match the search criteria, the page contains a message indicating that no objects were found. Otherwise, the page contains the resulting objects.

If more objects were found than can be displayed at once, the display record popup 43 can be used to view additional records. For example, if a user criteria matched 60 products, the page might say "Display records 1 to 10 of 60 records found." Using the popup 43, a user could display each subsequent set of ten products.

The data table 44 lists all of the objects found. Clicking a data table column heading 47 sorts the table by that column 46. The arrow 48 next to the column name indicates whether it is sorted in ascending or descending order.

Within the Product Search List, clicking a link in the "Product Name" column 126 displays the Product Detail page for the product, explained in detail further herein. Further, clicking a link in the Manager column 51 displays the Contact Detail page for the manager. Also, clicking a link in the List column 52 drills down one level into the brand hierarchy, displaying the objects beneath the product, in the case of FIGURE 28, the jobs. If a user is not happy with the results of a search--either too few or too many records were returned--a user can perform a new search by refining the search criteria, as described in the next step.

In an embodiment, the search criteria previously defined can be modified by clicking the "Products" button 90 in the search frame 21 to reopen the Product Search page of FIGURE 27. The Product Search page contains the criteria a user previously entered because the system preferably remembers the search criteria last entered for each type of object. Even if a user log out and log back in, the system displays a user previous criteria when a user reopens a Search page.

Modifying the search criteria allows a user to either narrow or widen the scope of a search. If a user previous search returned too many results, a user can narrow the search by adding more criteria--a user might specify Market criteria or further specify the product name (for example, Frozen Mexican). If a user previous search returned too few results, a user can expand the search by deleting or making existing criteria less specific--a user might remove the Product Parent criteria. If a user previous search returned the wrong set of results, a user can click the Reset button to clear the criteria and start over.

The above describes an embodiment for locating an object using the system search tool. In summary a user can start searching by clicking the button in the Search frame that corresponds to the type of object a user want to locate, for example, a project or a job. Accordingly, the Search page opens, displaying criteria that are appropriate for the type of object selected. On the Search page, a user defines the search criteria that distinguish the objects the user wants to find. Criteria are defined by specifying an attribute, an operator, and a limiting value. If a user does not want to search by a particular type of criterion, the user selects the "(Any)" operator and the criterion is ignored. A user can also define display criteria to control how the results are displayed. Once the criteria are defined, the user clicks the Search button in the Control frame to perform the search and display the results. If a user wants to refine the search just performed, the user clicks the same search button again. The system remembers the search criteria previously defined. The system also can include an Advanced Search function, containing more complex search criteria.

In an embodiment, the Advanced Search can be activated be clicking the "Advanced Search" button 106 wherein the search form 88 of FIGURES 29 and 30 is displayed for providing a comprehensive search of brand data. Through the advanced brand search form 88 all brand data, contact data, modification data, and display options can be searched to find brands which match the user-provided search criteria. Likewise, an advanced job search form 93 (FIGURES 31, 32, 33, and 34) which is provided for

searching of job information, job codes, contact information, and modification information.

As indicated previously, every system object--brands, products, jobs, files, projects, and events--can be viewed on a Detail page that provides in-depth information about the object. However, since the information stored for each type of object is different, the contents of the Detail page varies from object type to object type. A Detail page is displayed by clicking the link representing the object's name. The path a user takes to get to the link does not matter. In an embodiment, a user can display a Product Detail page by: browsing to a Product List page, then clicking the name of a product listed in the data table; expanding the navigation outline to display the desired product, then clicking the product's name link; searching for a particular type of product, then clicking the product's name link from the Search Result List page.

FIGURE 35 displays an embodiment of a Job Detail page 132 displaying extensive information about an object. The number, type, and contents of the fields on a Detail page vary with the type of object examined. Preferably, any value displayed as a link can be clicked to view more details. For example, on the displayed Job Detail page, clicking a user's name displays his or her Contact Detail page; clicking the Die link displays a Die Detail page. Clicking the [more...] link next to a Font name displays an expanded list of fonts. If a user has proper security permissions, then the user can modify the contents of the fields on an object's Detail page.

When a Detail page is displayed, buttons are added to the Control frame 22 that correspond to operations a user can perform on the object. The operations available, and therefore the buttons displayed, differ from object type to object type.

Once a user has finished examining the Detail page, the user can jump to another level of the data hierarchy using the wayfinder or click a button in the Control frame to perform a new operation.

With the proper security permission, a user can edit the fields of the Detail page. In an embodiment, the control frame 22 of the Detail page 132 can include an

"Edit" button 133. When an asset is displayed in the data frame 23, clicking the "Edit" button 133 causes the system to display an asset edit form 109 in the data frame 23 as shown in FIGURE 36. In the edit asset form 109, the asset number, asset name, asset type, asset status, and asset notes may be changed. After editing any or all of the asset data, a user can click the save changes hyperlink 118 for causing the changes to be saved. Although the asset shown in FIGURE 36 is a brand, it is to be understood that any asset may have its associated data edited in this manner.

The control frame of the Detail page can also include an asset attribute hyperlink for causing the system to display an attribute list screen 140 as shown in Figure 37. The attribute list screen 140 allows a user to create and edit attributes to be associated with an asset. As shown in Figure 37, the attribute list screen 140 includes information about associated attributes, such as the attributes' name and type. The attribute list screen 140 also provides an attribute detail hyperlink 142, an add attribute hyperlink 144, and a remove attribute hyperlink 146. By clicking the attribute detail hyperlink 142, the data form 23 is changed to display an attribute detail screen 148, as shown in FIGURE 38.

The attribute detail screen 148 information about the details of the attribute, such as: attribute name, attribute type, and attribute value. The attribute detail screen also includes an edit attribute hyperlink 150, an add attribute property hyperlink 152, and a remove attribute property hyperlink 154. By clicking on the edit attribute hyperlink 150, the data frame 23 is changed to an edit attribute screen 156, as shown in FIGURE 39. The edit attribute screen 156 shows information about the attribute, such as: attribute name, attribute type, and attribute sequence within the listed attributes which can be edited. This is accomplished through an attribute name field 158, an attribute sequence drop down list 160, and a save changes hyperlink 162. To change the attribute name and/or attribute sequence, the attribute name field 158 and/or attribute sequence drop down list 160 are changed and the save changes hyperlink clicked 162.

Referring back to FIGURE 38, by clicking the add property hyperlink 152, the data frame 23 is changed to display an add property screen 164, as shown in FIGURE 40. The add property screen 164 provides a property value field 166 and a submit hyperlink 168. By typing an additional value in the property value field 166 to be associated with the selected attribute and clicking the submit hyperlink 168, the value typed will be saved as a property of the attribute.

Referring back to FIGURE 38, by clicking the remove property hyperlink 154, the data frame 23 is changed to display a remove property screen 170, as shown in FIGURE 41. The remove property screen 170 provides a property list 172 associated with the particular attribute. Each property within the property list 172 has a property hyperlink 174. Each property hyperlink 174 is represented by the property name. By clicking on a property hyperlink 174, the property clicked on is disassociated from the attribute.

Referring to FIGURE 37, by clicking on the add attribute hyperlink 144, the data frame 23 is changed to display an add attribute screen 176 as shown in FIGURE 42. The add attribute screen 176 provides an attribute search form 178, a search hyperlink 180, a cancel hyperlink 182, and a new attribute hyperlink 184. By typing an attribute name that is already associated with another asset, but is not currently associated with present asset, into the attribute search form 178 and clicking the search hyperlink 180, the add attribute screen 176 is changed to include the attribute searched for in a table 181, if it exists within the system, as shown in FIGURE 43. By setting a display sequence 186 of the searched attribute and clicking on a searched attribute hyperlink 188 in Figure 43, the searched attribute will become associated with the particular asset. By clicking on the cancel hyperlink 182 of FIGURES 42 or 43, the data frame 23 is changed to display the attribute list screen 140 of FIGURE 37.

Alternatively, by clicking on a new attribute hyperlink 184 in FIGURES 42 or 43, the data frame 23 is changed to display a new attribute screen 190 as shown in FIGURE 44. The new attribute screen 190 contains an attribute name field 192, a

attribute type drop down list 194, and an add new attribute hyperlink 196. By typing a new attribute name in the attribute name field 192, selecting a new attribute type from the new attribute type drop down list 194, and clicking the add new attribute hyperlink 196, the new attribute typed appears in the add attribute screen 176 of Figure 43 in place of the searched attribute where it can be associated with the asset as described above. Possible attribute types are: value, list, and set. All attributes associated with a particular asset are then shown in the asset detail screen for the particular asset.

Referring to FIGURE 37, by clicking the remove attribute hyperlink 146, the data frame 23 is changed to display a remove attribute screen 198, as shown in FIGURE 45. The remove attribute screen 198 provides an attribute list 200 which displays information about the attributes associated with the particular asset, such as: the attribute name and the attribute type. Each attribute is associated with a remove attribute hyperlink 202 which is represented by the attribute name. By clicking the attribute hyperlink 202, the attribute name clicked on is disassociated with the particular asset.

Referring back to FIGURE 35, by clicking on the asset contact hyperlink 134, the data frame 23 is changed to show an asset contact list screen 204 as shown in FIGURE 46. The asset contact list screen 204 provides information about the manager in charge of a particular asset, such as contact type and contact name. The asset contact list screen 204 provides a edit contact hyperlink 206 and a contact detail hyperlink 208. By clicking the contact detail hyperlink 208, the data frame 23 is changed to display a contact detail screen. By clicking the edit contact hyperlink 206, the data frame 23 is changed to display an edit manager contact screen. It is contemplated that from the edit manager contact screen more than one contact can be associated with a particular asset, that new contacts can be input into the system, that current contacts can be edited, and that existing contacts not associated with the present asset can be associated, similar to previously explained in the case of attributes.

Referring back to FIGURE 35, by clicking on a asset file hyperlink 135, the data frame 23 is changed to display a sub-asset list window as shown in FIGURE 12.

In an embodiment, the control frame 22 can include an add brand screen which is appears and operates similarly to the edit brand screen 109 of FIGURE 36.

As indicated above, at the bottom of the data hierarchy are the files--objects representing specific digital assets, such as images, logos, or other documents. The shopping cart provides a user with a method of gathering these files together into an area.

When a user finds a file he or she wants, the user can put it in the shopping cart, then move on to other locations, accumulating additional files. In an embodiment, adding a file to a shopping cart does not move the file from its location in the system data hierarchy. Instead, it simply places a reference to the file in the shopping cart.

In an embodiment there are two types of shopping carts: the primary cart and named carts. Each user has a single primary shopping cart, but can create any number of named shopping carts. Preferably, files can be added to the primary shopping cart only. Named shopping carts are filled by loading in the contents of the primary shopping cart or another named cart.

In an embodiment, a user can put company logos in one shopping cart and product package dies in another. First, a user creates two named shopping carts: "Logos" and "Dies." Then accumulates the logo files in the primary shopping cart, loads the contents of the primary shopping cart into the "Logos" cart, and then empties out the primary cart. Next, the user accumulates the package die files in the empty primary cart, then load its new contents into the "Dies" cart. In an embodiment, a user can even create another named shopping cart, named "Logos and Dies," and load the contents of both named carts into it.

What a user ultimately does with the files accumulated in a shopping cart can be dependant upon the options set up during system installation and can vary from sit to site. For example, the system installation may allow a user to download the files or transfer them to another computer.

In depicting the operation of the shopping cart in the preferred embodiment, the illustrative system database--the "Whacky Foods" database, is unitized again.

Although a user's database will contain different information, these examples illustrate capabilities of the shopping carts.

Files stored in the system data hierarchy can be added to a user's primary shopping cart. Once in the primary cart, they can be loaded into a named shopping cart. To add files to the primary shopping cart, it is preferred that a user first locate the desired files by browsing, navigating, or searching.

As stated previously, the File List page of FIGURE 12 can be opened by using the "Browse" button 65 or search tool. The File List page contains files a user can add to the primary shopping cart. The first column 66 in the data table 44 is entitled "Cart." The Cart column 66 contains icons 68 that display which files are in the shopping cart and icons 70 that allow a user to move a file into or out of the shopping cart. The icon 68 on the left side of the column, the Status icon, indicates whether or not the file is currently in the primary shopping cart (the file is in the primary shopping cart; the file is not in the primary shopping cart). The icon 70 on the right side of the column, the Action icon, identifies that action will occur if the radio button 71 preceding the icon is selected (move the file into the primary shopping cart; move the file out of the primary shopping cart).

For example, the pinata.eps and the redpeps.eps files can be selected for inclusion in the primary shopping cart by clicking the radio buttons 71 preceding their Action icons 70 resulting in the data table 44 shown in FIGURE 47. If a user selects a file's Action icon and then changes his or her mind, then the user clicks the radio button 69 preceding its Status icon 68 to restore the file's original shopping cart state. Further, clicking the Reset button (FIGURE 12) in the Control frame, restores all files to their original state.

Preferably, clicking the radio button 71 preceding an Action icon 70 does not immediately move a file into or out of the shopping cart. Instead, the user should click the Save button (FIGURE 12) in the Control frame 22 for the action to take effect.

Therefore, a user can click back and forth on the Action and Status icons' radio buttons without making inadvertent changes to the shopping cart.

Turning to FIGURE 48, opening the primary shopping cart depicts which files are currently in the cart and allows a user to operate on the cart's contents. The "Contents of Primary Cart" page illustrated in FIGURE 48 is opened by clicking the Shopping Cart button 97 (FIGURE 12) in the Control frame. For each item in the cart, a thumbnail, a file name link, and shopping cart icons are displayed. The thumbnail provides a graphical preview of the file's contents. The file name link is displayed below the thumbnail. Clicking this link opens the File Detail page, allowing a user to examine detailed information about the file.

Below the file name links are the shopping cart icons. To move an item out of the shopping cart, a user clicks the radio button preceding its Action icon. To return the item to the shopping cart, a user clicks the radio button preceding its Status icon. Preferably, selecting the Action icon does not immediately move an item out of the shopping cart. Instead, it is desired that a user click the Save button in the Control frame for the action to take place.

At the bottom of the page is the files per row popup that can be used to change the number of files displayed in each row of the page. For example, in FIGURE 48, selecting "2" from the popup would display two files per row instead of four.

When a Contents of Primary Cart page is displayed, the following buttons can be added to the Control frame, if appropriate for the cart's's contents (for example, if the cart contains no items, the Empty button is not displayed): 1) a "Carts" button wherein clicking the button displays a page listing the shopping carts a user can access, including both the user's primary cart and named carts and, the user can view the contents of a listed cart by clicking the cart's name link; 2) a "Load" button wherein clicking the button allows a user to load the contents of any shopping cart into the displayed cart and preferably, loading adds the files in the selected source cart to the contents of the displayed cart; 3) a "Replace" button wherein clicking the button allows a user to replace

the contents of the displayed shopping cart with the contents of another cart and, unlike the load operation, replacing deletes the contents of the displayed cart before adding the new files; 4) an "Empty" button wherein clicking the button removes all the files from the displayed cart; 5) a "Remove All" button wherein clicking the button selects the Move Out action icon for every file in the displayed cart (even those that are scrolled off the page) marking them for removal, but, preferably, the files are not removed from the cart until the Save button is clicked; 6) a "Reset" button wherein clicking the button returns the shopping cart to the state it was in when a user first opened the page; and 8) a "Save" button wherein clicking the button permanently saves all changes made to the shopping cart's contents.

Preferably, when the Contents of a Named Cart page is displayed, the following buttons are added, if appropriate: 1) a "Delete" button wherein clicking the button deletes the named shopping cart, with a confirmation dialog box being displayed before the cart is deleted; and 2) an "Edit" button wherein clicking the button allows a user to change the name of the shopping cart to a unique name that is not assigned to another shopping cart.

As stated previously, named shopping carts are used to accumulate a set of files and organize them under a descriptive heading. The current invention allows a user to create and save multiple named shopping carts. The named shopping carts a user creates belongs exclusively to the user so that no other user can see the named shopping carts or access the contents. Further, a user's named shopping carts is persistent such that they remain in the shopping cart list after the user logs out and until then user explicitly deletes them.

In an embodiment, a user can create a named shopping cart in which files needed to redesign the Mexican Super Corn product can be collected. Turning to FIGURE 12, a user first clicks on the "Shopping Cart" button 97 wherein the contents of the primary shopping cart are displayed such as, for example, FIGURE 48. In the Control frame 22, the user clicks the "Carts" button wherein a list containing the names

of all the user's shopping carts is provided such as in FIGURE 49. The user then clicks the "Add Cart" button in the Control frame 22 to specify a name for the new shopping cart. As a result, the Add Cart display shown in FIGURE 50 is presented wherein a user can specify a name for the new shopping cart such as, for example, "MexCorn Redesign." Next, a user clicks the "Add Cart" button wherein the Cart List page is displayed again, but time the new shopping cart is listed as shown in FIGURE 51.

Once a user has created a named shopping cart, the user can add files to it. Preferably, however, a user cannot add files directly to a named shopping cart. Instead, a user first adds the desired files to the primary shopping cart and then loads the contents of the primary cart into the named cart. If desired, a user can empty the primary cart before starting to accumulate files to ensure that only the desired files are added.

Below provides an illustrative example regarding loading files from the primary shopping cart into the "MexCorn Redesign" shopping cart using an embodiment of the present invention. First, in the Control frame 22 of FIGURE 12 the user clicks the Shopping Cart button 97 wherein, as shown in FIGURE 48, the contents of the primary shopping cart are displayed. Here, the user can verify that only the desired files that should be loaded into the "MexCorn Redesign" cart are in the primary cart. If the primary cart contains files that should not be loaded, the user clicks the radio buttons preceding their Move Out Action icons and then clicks the "Save" button in the Control frame 22 to remove them from the cart.

Next, in the Control frame 22, the user clicks the "Carts" button wherein the page shown in FIGURE 51 is displayed for listing the user's shopping carts. In FIGURE 51, the "MexCorn Redesign" shopping cart has no files in it. Accordingly, a user opens the "MexCorn Redesign" shopping cart to add files to it.

The user then clicks the "MexCorn Redesign" link wherein the contents of "MexCorn Redesign" page is displayed as shown in FIGURE 52 with a message indicating that the "MexCorn Redesign" shopping cart is empty. The user then clicks the "Load" button in the control frame 22 wherein the page shown in FIGURE 53 is

displayed which allows a user to select the shopping cart containing the files for loading into the "MexCorn Redesign" shopping cart.

The user then clicks the "Primary Cart" link wherein the files from the primary shopping cart are loaded into the "MexCorn Redesign" shopping cart and the contents of the "MexCorn Redesign" cart are displayed as shown in FIGURE 54. Preferably, the files remain in the primary shopping cart even though they were loaded into another shopping cart. Although the above illustration loads files from the primary cart into the "MexCorn Redesign" cart, the present invention includes loading files from any shopping cart into any other cart.

As stated above, every system user has one primary shopping cart and can create additional named shopping carts to further organize the files he or she wants to accumulate. Files displayed on a File List page can only be added to the primary shopping cart. To place files into a named shopping cart, it is desired that the user add them to the primary shopping cart and then use the Load or Replace function to move them into a named cart.

Moreover, as stated above, when a shopping Cart page is displayed, buttons are added to the control frame 22 providing access to the operations a user can perform on a shopping cart. A particular system site can have custom shopping cart operations available--for example, operations that allow a user to transfer files from a shopping cart to another computer.

Files gathered within the user's named shopping cart could have operations performed upon them, such as changing the file size, file format, or even more advanced functions based on whether the file is a font, image, or other design element. Additionally, files gathered within the user's named shopping cart can be routed through a local area or wide area network in order to send the files to another user or another database.

As stated previously, the present invention provides project management functions that allow users to setup projects, to manage groups of jobs related to particular

projects, and to assign tasks, or events, that must be completed within a time frame in order to complete the project. A project is a specific campaign or promotion for a given brand. A job is a specific print run for a promotion.

Referring to FIGURE 6, by clicking on a "Projects" hyperlink 300 in the control frame 22, the data frame 23 is changed to display a project list screen 302, as shown in FIGURE 55. The project list screen 302 shown in FIGURE 55 displays information about projects for the manufacturer "Whacky Foods, Inc." as shown by the wayfinder 20. The project list screen 302 also provides a project detail hyperlink 304 and a job list hyperlink 306. Additionally, the control frame 22 includes an add project hyperlink 308. By clicking on the add project hyperlink 308, the data frame 23 is changed to display an add project screen 310, as shown in FIGURE 56. The add project screen 310 includes fields 312 for inputting project data such as: project number, project name, schedule date, actual start date, due date, actual end date, and notes; changing a status drop down list 314; and clicking a save changes hyperlink 316. By entering the project information into the forms 312, selecting a project status from the project status drop down list 314 and clicking the save changes hyperlink 316 new projects are created.

Referring back to FIGURE 55, by clicking the project detail hyperlink 304, the data frame 16 is changed to display a project detail screen 318, as shown in Figure 57. The project detail screen 318 includes the same information as the project list screen 302 about the particular project. The control frame 22 of the project detail screen 318 includes a sub-projects hyperlink 320, a jobs hyperlink 322, an events hyperlink 324, a contacts hyperlink 326, and a security hyperlink 328. By clicking the sub-projects hyperlink 320, the data frame 23 changes to display a new project detail screen 318 for any sub-projects. By clicking on the jobs hyperlink 322, the sub-asset list window of FIGURE 11 is displayed. By clicking on the events hyperlink 324, the data frame 23 changes to display an event list screen 336 associated with the project, as shown in FIGURE 58. The event list screen 336 is explained in greater detail below.

Referring back to FIGURE 57, by clicking on the edit hyperlink 326, the data frame 23 is changed to display an edit project screen 334, as shown in FIGURE 59. Within the edit project screen 334, all of the information about the project may edited and saved as previously explained according to other data types. By clicking on the contacts hyperlink 328, the data frame 23 changes to display a contact list screen 284, as shown in FIGURE 60. By clicking the security hyperlink 330, the data frame 23 is changed to display a security screen as will be explained below with respect to the administrative function.

Referring back to FIGURE 55, by clicking the jobs hyperlink 308, the data frame 23 is changed to display the sub-asset list window containing jobs, as shown in FIGURE 11.

Referring back to FIGURE 11, by clicking on a sub-asset hyperlink 64 which indicates events, the data frame 23 is changed to display an event list window 336, as shown in FIGURE 58. The events list window 336 contains relevant information about scheduled events within a job, such as: an event name, event contact, event due date, event end date, and event status. The event list window 336 also provides an event name hyperlink 338, an event contact hyperlink 340, and an add event hyperlink 342. By clicking on the event name hyperlink 338, the data frame 23 is changed to display an event detail screen 344, as shown in FIGURE 61. The event detail screen 344 provides an edit hyperlink 346 in the control frame 22, a contact hyperlink 348, and an email hyperlink 350.

In an embodiment administrative functions are provided. Turning to FIGURE 62, in order to provide security of the system from unauthorized access, there is provided a login screen 400 which provides a login name field 402, a password field 404, and a login hyperlink 406. In order to login, a user must type in a valid login name in the login name form 402, type in a valid password in the password form 404, and click the login hyperlink 406. If the login name and password are valid, the sub-asset list screen 40 of FIGURE 6 appears for the company with which the user is associated. By clicking on

the logout hyperlink 408 in the control frame 22 of FIGURE 6, the user is automatically logged out and returned to the login screen 400.

In an embodiment the screen 40 of FIGURE 6 can include and "Admin" hyperlink (such as reference number 407 in FIGURE 57) wherein, by clicking the link, the data frame 23 is changed to show an administration title screen 410, as shown in FIGURE 63. In the control frame 22 there are provided five administrative hyperlinks: a companies hyperlink 412, a groups hyperlink 414, a code tables hyperlink 416, a logins hyperlink 418, and a locks hyperlink 420. By clicking on the companies hyperlink 412, a company list screen 422 is displayed in the data frame 23, as shown in FIGURE 64. Additionally, an add company hyperlink 424 is added to the control frame 22. The company list screen 422 provides information about any companies that are associated with the present system. The company list screen 422 also provides a company name hyperlink 426, an email hyperlink 428, and a contacts hyperlink 429.

By clicking on the add companies hyperlink 424, the data frame 23 is changed to display an add company screen 430, as shown in FIGURE 65. The add company screen 430 contains forms 432 for company name, address, city, state, postal code, phone, fax, email, a country drop down list 434, a status drop down list 436, and a save changes hyperlink 438. After typing the relevant information into the fields 432, selecting the correct country and status from the drop down lists 434, 436, and clicking the save changes hyperlink 438, the new company information is added to the system.

By clicking the company name hyperlink 426 of FIGURE 64, the data frame 23 is changed to include a company detail screen 440 which contains relevant information about a client company, such as company name, company address, company phone, company fax, company status, as in FIGURE 66. The company detail screen 440 further includes an edit company hyperlink 442 and a company contacts hyperlink 444. By clicking the edit company hyperlink 442, the data frame 23 is changed to display an edit company screen 446, as shown in FIGURE 67. By changing the information in the

fields of the edit company screen 446 and clicking a save changes hyperlink 448, the information for the company is updated.

Referring to FIGURE 66, by clicking the company contacts hyperlink 444, the data frame 23 is changed to display a company contact list screen 450, as shown in FIGURE 68. The contact list screen 450 provides information about each contact associated with a company. The contact list screen 450 also includes a contact name hyperlink 452, a contact email hyperlink 454, and an add contact hyperlink 456. By clicking on the contact name hyperlink 452, the data frame 23 changes to display a contact detail screen 458, as shown in FIGURE 69. The contact detail screen 458 contains relevant information about the particular contact, such as contact name, contact company, contact address, contact email address, and contact status. The contact detail screen 458 includes a company hyperlink 460, an email hyperlink 462 and an edit contact detail hyperlink 464.

By clicking on the edit contact detail hyperlink, the data frame 23 is changed to display an edit contact detail screen 466, as shown in FIGURES 70a and 70b. The edit contact detail screen 466 allows contact detail to be edited and saved by clicking a save contact detail hyperlink 468.

Referring to FIGURE 69, by clicking on the company hyperlink 460, the data frame 23 is changed to display the company detail screen 440 of FIGURE 66. By clicking on the email hyperlink 462 of FIGURE 69, an email message can be sent to the contact.

Referring to FIGURE 68, by clicking on the add contact hyperlink 456, the data frame 23 is changed to display an add contact screen 470, as shown in FIGURES 71a and 71b. By changing the fields of the add contact screen 464 and clicking a save changes hyperlink 472.

Referring to FIGURE 64, by clicking on the email hyperlink 428, an email message can be sent to the company. By clicking on the contacts hyperlink 429, the data frame 23 is changed to display a contact list 450 of FIGURE 68.

Referring to FIGURE 66, by clicking on a users hyperlink 474, the data frame 23 is changed to display a users screen 476, as shown in FIGURE 72, which displays the current users for a company. The display users screen also provides an add user hyperlink 478. By clicking on the add user hyperlink 478, the data frame 23 is changed to display the add contact screen 470 of FIGURES 71a and 71b.

Referring to FIGURE 63, by clicking on a groups hyperlink 414, the data frame 23 is changed to display a groups list screen 480, as shown in FIGURE 73. The groups list screen 480 displays the current user groups, user group status, and number of users in the group. The groups list screen 480 also contains a group detail hyperlink 482, a group number hyperlink 484, and an add group hyperlink 486. By clicking on the group detail hyperlink, the data frame 23 is changed to display a group detail screen 488, as shown in FIGURE 74. The group detail screen 488 shows the details shown in group lists screen 480. The group detail screen 488 further includes an edit group hyperlink 490. By clicking the edit group hyperlink 490, the data frame 23 is changed to display an edit group screen 492, as shown in FIGURE 75. By editing the fields on the edit group screen 492 and clicking on a save changes hyperlink 494, the group information is changed. By associating users with groups, the rights for a particular user to perform certain functions with the brand management, project management, and file exchange functions can be edited. The rights of certain users to only view, rather than edit or add new data can be restricted.

Referring to FIGURE 63, by clicking a code tables hyperlink 498, the data frame 23 is changed to show a code tables screen 500, as shown in FIGURE 76. The code tables screen 500 is used to provide customization of the present embodiment for each particular user of the system.

Referring to FIGURE 63, by clicking a logins hyperlink 502, the data frame 23 is changed to show a logins screen 506, as shown in FIGURE 77. The logins screen 506 displays the list of users currently logged into the system.

Referring to FIGURE 63, by clicking a locks hyperlink 504, the data frame 23 is changed to show a locks screen 508, as shown in FIGURE 78. The locks screen 508 displays the list of records currently locked on the system.

While this specific embodiment has been illustrated and described, numerous modifications come to mind without significantly departing from the spirit of the invention and the scope of protection is only limited by the scope of the accompanying Claims.

CLAIMS



WE CLAIM:

1. A system for managing the development of graphical package depictions comprising:

a brand portion for accepting and holding an identification of one or more brands, the brand portion being on a first relational level;

a product portion for accepting and holding an identification of one or more products, the product portion being on a second relational level which is directly related to the first relational level; and,

a job portion for accepting and holding an identification of one or more jobs, the job portion being on a third relational level which is directly related to the second relational level.

2. The system of claim 1 wherein the first relational level is on a one to many relationship with the second relational level.

3. The system of claims 1 or 2 wherein the second relational level is on a one to many relationship with the third relational level.

4. The system of claim 1 further including a brand editor, a brand searcher, a company identification portion, a company editor, a brand contact identification portion, a brand market portion, a brand file portion, a brand property portion, a brand attribute portion.

5. The system of claim 1 further comprising a product editor, a product searcher, a product contact identification portion, a contained product portion, a product file portion, a product property portion, a product attribute portion.

6. The system of claim 1 further comprising a job editor, a job searcher, a job contact identification portion, a job property portion, a job market portion, an job event portion, a job file portion.



7. The system of claim 1 wherein the brand portion further comprises a sub-brand portion having a sub-brand editor, a sub-brand searcher, a sub-brand contact identification portion, a sub-brand market portion, a sub-brand file portion, a sub-brand property portion, a sub-brand attribute portion, the brand portion storing at least one brand therein, and the sub-brand portion having at least one sub-brand portion therein, and wherein at least one brand has at least one sub-brand related thereto.

8. The system of claim 1 wherein the product portion further comprises a sub-product portion having a sub-product editor, a sub-product searcher, a sub-product contact identification portion, a contained sub-product portion, a sub-product file portion, a sub-product property portion, a sub-product attribute portion,
the product portion storing at least one product therein, and the sub-product portion having at least one sub-product portion therein, and wherein at least one product has at least one sub-product related thereto.

9. The system of claim 1 wherein the first relational level being directly related to the second relational level directly links the brand portion to the product portion, respectively, and wherein at least one product is linked to at least one brand with the product so linked to the brand is accessible from the brand portion.

10. The system of claim 1 wherein the second relational level being directly related to the third relational level directly links the product portion to the job portion, respectively, and wherein at least one job is linked to at least one product with the job so linked to the product is accessible from the product portion.

11. The system of claim 1 wherein the brand searcher can search for a brand by at least one property value of the brand property portion.

12. The system of claim 1 wherein the product searcher can search for a product by at least one property value of the product property portion.

13. The system of claim 1 wherein the job searcher can search for a job by at least one property value of the job property portion.



14. The system of claim 1 wherein a property value of the brand property portion can be used to search for a file within the brand file portion.

15. The system of claim 1 wherein a property value of the product property portion can be used to search for a file within the product file portion.

16. The system of claim 1 wherein a property value of the job property portion can be used to search for a file within the job file portion.

17. The system of claim 1 further comprising:

a project portion having a project searcher, a project editor, a project contact identification portion, a project property portion, a project event portion, a project job portion, the project portion being directly related to the job portion; and,

an event portion having an event searcher, an event editor, and an event property portion; the event portion being directly related to the job portion.

18. The system of claim 17 wherein at least one job is linked to at least one project, and wherein the job so linked to the project is accessible from the project portion

19. The system of claim 17 wherein a job is linked to at least one event, and wherein the job so linked to the event is accessible from the event portion

20. The system of claim 17 wherein the project searcher can search for a project by at least one property value of the project property portion.

21. The system of claim 17 wherein the event searcher can search for an event by at least one property value of the event property portion.

22. The system of claim 1 further comprising:

a collector for collecting one or more files selected from one or more of the brand file portion, the product file portion, and the job file portion, into a collected file portion.

23. The system of claim 22 wherein the collector can remove one or more of the files from the collected file portion.

24. The system of claim 1 further comprising:



a brand associator for associating at least one file to a brand entered into the brand portion.

25. The system of claim 1 further comprising:

a product associator for associating at least one file to a product entered into the product portion.

26. The system of claim 1 further comprising:

a job associator for associating at least one file to a job entered into the job portion.

27. A system for managing the development of graphical package depictions comprising:

a brand portion having a brand editor, a brand searcher, a company identification portion, a company editor, a brand contact identification portion, a brand file portion, the brand portion being on a first relational level;

a product portion having a product editor, a product searcher, a product file portion, the product portion being on a second relational level which is directly related to the first relational level; and,

a job portion having a job editor, a job searcher, a job file portion, the job portion being on a third relational level which is directly related to the second relational level.

28. The system of claim 27 wherein the brand portion further comprises a brand market portion for accepting and holding an identification of one or more markets for one or more brands.

29. The system of claim 27 wherein the brand portion further comprises a brand property portion for accepting and holding an identification of one or more properties for one or more brands.

30. The system of claim 27 wherein the brand portion further comprises a brand attribute portion for accepting and holding an identification of one or more attributes for one or more brands.



31. The system of claim 27 wherein the product portion further comprises a product contact identification portion for accepting and holding an identification of one or more contacts for one or more products.

32. The system of claim 27 wherein the product portion further comprises a contained product portion for accepting and holding an identification of one or more contained products for one or more products.

33. The system of claim 27 wherein the product portion further comprises a product property portion for accepting and holding an identification of one or more properties for one or more products.

34. The system of claim 27 wherein the product portion further comprises a product attribute portion for accepting and holding an identification of one or more attributes for one or more products.

35. The system of claim 27 wherein the job portion further comprises a job contact identification portion for accepting and holding an identification of one or more contacts for one or more jobs.

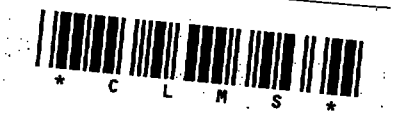
36. The system of claim 27 wherein the job portion further comprises a job property portion for accepting and holding an identification of one or more properties for one or more jobs.

37. The system of claim 27 wherein the job portion further comprises a job market portion for accepting and holding an identification of one or more markets for one or more jobs.

38. The system of claim 27 wherein the job portion further comprises a job event portion for accepting and holding an identification of one or more events for one or more jobs.

39. A system for resource and project management comprising:

at least one program stored on a compact removable transfer media and readable by the system, the program having:



code for creating a multilevel asset hierarchy having a plurality of objects including jobs and files;

code for creating a multilevel project hierarchy having a plurality of objects including the jobs in the asset hierarchy and events;

code for creating a visual display of the objects within the asset hierarchy and the project hierarchy;

code for allowing editing of the objects within the asset hierarchy and the project hierarchy; and

code for creating a visual display of the jobs within both the asset hierarchy and the project hierarchy.

40. The system of claim 39 further including code for allowing a user to browse the asset hierarchy by visually observing the objects therein on a level-by-level basis.

41. The system of claim 39 further including code for allowing a user to browse the project hierarchy by visually observing the objects therein on a level-by-level basis.

42. The system of claim 39 further including code for allowing a user to navigate the asset hierarchy by visually observing an overview of the asset hierarchy.

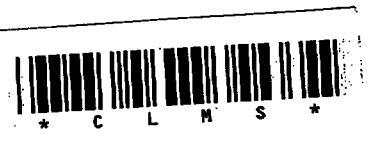
43. The system of claim 41 further including code for allowing a user to focus on a particular level of the objects.

44. The system of claim 39 further including code for allowing a user to navigate the project hierarchy by visually observing an overview of the project hierarchy.

45. The system of claim 44 further including code for allowing a user to focus on a particular level of the objects.

46. The system of claim 39 further including code for allowing a user to search the asset hierarchy to locate an object that satisfies a particular set of user defined criteria.

47. The system of claim 39 further including code for allowing a user to search the project hierarchy to locate an object that satisfies a particular set of user defined criteria.



48. The system of claim 39 further including code for providing a shopping cart that allows a user to collect at least one of the files.

49. The system of claim 48 further including code for allowing a user to perform operations on files collected in the shopping cart.

50. The system of claim 49 further including code for allowing a user to route files collected in the shopping cart to another database.

51. The system of claim 39 further including code for allowing a user to edit objects within the asset hierarchy.

52. The system of claim 39 further including code for allowing a user to edit objects within the project hierarchy.

53. A method for resource and project management comprising the steps of:

creating a multilevel asset hierarchy having a plurality of objects including jobs and files;

storing said asset hierarchy in a system memory;

creating a multilevel project hierarchy having a plurality of objects including the jobs in the asset hierarchy and events;

storing said project hierarchy in the system memory;

creating and providing a visual display of the objects within the asset hierarchy and the project hierarchy;

editing the objects within the asset hierarchy and the project hierarchy; and

creating and providing a visual display of the jobs within both the asset hierarchy and the project hierarchy.

54. The method of claim 53 further including the step of allowing a user to visually observe the objects within the asset hierarchy on a level-by-level basis.

55. The method of claim 53 further including the step of allowing a user to visually observe the objects within the project hierarchy on a level-by-level basis.

56. The method of claim 53 further including the step of allowing a user to visual observe an overview of the asset hierarchy.



57. The method of claim 56 further including the step of allowing a user to focus on a particular level of the objects.

58. The method of claim 53 further including the step of allowing a user to visual observe an overview of the project hierarchy.

59. The method of claim 58 further including the step of allowing a user to focus on a particular level of the objects.

60. The method of claim 53 further including the step of allowing a user to search the asset hierarchy to locate an object that satisfies a particular set of user defined criteria.

61. The method of claim 53 further including the step of allowing a user to search the project hierarchy to locate an object that satisfies a particular set of user defined criteria.

62. The method of claim 53 further including the step of providing a shopping cart that allows a user to collect at least one of the files.

63. The method of claim 62 further including the step of allowing a user to perform operations on at least one of the files collected in the shopping cart.

64. The method of claim 62 further including the step of allowing a user to route at least one of the files collected in the shopping cart to another database.

65. The method of claim 53 further including the step of allowing a user to edit objects within the asset hierarchy.

66. The method of claim 53 further including the step of allowing a user to edit objects within the project hierarchy.

67. A system for resource and project management comprising:

a multilevel asset hierarchy stored in a system memory and having a plurality of objects including jobs and files;

a multilevel project hierarchy stored in the system memory and having a plurality of objects including the jobs in the asset hierarchy and events;

program code for creating a visual display of the objects within the asset hierarchy and the project hierarchy; and



program code for creating a visual display of the jobs within both the asset hierarchy and the project hierarchy.

68. The system of claim 67 further including code for allowing editing of the objects within the asset hierarchy and the project hierarchy.

69. The system of claim 67 further including code for allowing a user to visually observe the objects within the asset hierarchy on a level-by-level basis.

70. The system of claim 67 further including code for allowing a user to visually observe the objects within the project hierarchy on a level-by-level basis.

71. The system of claim 67 further including code for allowing a user to visually observe an overview of the asset hierarchy.

72. The system of claim 71 further including code for allowing a user to focus on a particular level of the objects.

73. The system of claim 67 further including code for allowing a user to visually observe an overview of the project hierarchy.

74. The system of claim 73 further including code for allowing a user to focus on a particular level of the objects.

75. The system of claim 67 further including code for allowing a user to locate an object in the asset hierarchy that satisfies a particular set of user defined criteria.

76. The system of claim 67 further including code for allowing a user to locate an object in the project hierarchy that satisfies a particular set of user defined criteria.

77. The system of claim 67 further including a shopping cart that allows a user to collect at least one of the files.

78. The system of claim 77 further including code for allowing a user to perform operations on files collected in the shopping cart.

79. The system of claim 77 further including code for allowing a user to route files collected in the shopping cart to another database.

80. The system of claim 67 further including code for allowing a user to edit objects within the asset hierarchy.



81. The system of claim 67 further including code for allowing a user to edit objects within the project hierarchy.

82. A method for resource and project management comprising the steps of:

storing a multilevel asset hierarchy in a system memory, the asset hierarchy having a plurality of objects including jobs and files;

storing a multilevel project hierarchy in the system memory, the project hierarchy having a plurality of objects including the jobs in the asset hierarchy and events;

creating and displaying a graphic representation of the objects within the asset hierarchy and the project hierarchy; and

creating and displaying a graphic representation of the jobs within both the asset hierarchy and the project hierarchy.

83. The method of claim 82 further including the step of editing the objects within the asset hierarchy.

84. The method of claim 82 further including the step of editing the objects within the project hierarchy.

85. The method of claim 82 further including the step of allowing a user to visually observe a graphic representation of the objects within the asset hierarchy on a level-by-level basis.

86. The method of claim 82 further including the step of allowing a user to visually observe a graphic representation of the objects within the project hierarchy on a level-by-level basis.

87. The method of claim 82 further including the step of allowing a user to visually observe a graphic representation of an overview of the asset hierarchy.

88. The method of claim 87 further including the step of allowing a user to focus on a particular level of the objects.

89. The method of claim 87 further including the step of allowing a user to visually observe a graphic representation of an overview of the project hierarchy.



90. The method of claim 89 further including the step of allowing a user to focus on a particular level of the objects.

91. The method of claim 82 further including the step of allowing a user to locate an object in the asset hierarchy that satisfies a particular set of user defined criteria.

92. The method of claim 82 further including the step of allowing a user to locate an object in the project hierarchy that satisfies a particular set of user defined criteria.

93. The method of claim 82 further including the step of providing a shopping cart that allows a user to collect at least one of the files.

94. The method of claim 93 further including the step of allowing a user to perform operations on files collected in the shopping cart.

95. The method of claim 94 further including the step of allowing a user to route files collected in the shopping cart to another database.

96. The method of claim 82 further including the step of allowing a user to edit objects within the asset hierarchy.

97. The method of claim 82 further including the step of allowing a user to edit objects within the project hierarchy.