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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 December 2000 (14.12.2000)

PCT

(10) International Publication Number
WO 00/75812 A1

(51) International Patent Classification⁷: G06F 17/30

(21) International Application Number: PCT/US00/14665

(22) International Filing Date: 26 May 2000 (26.05.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/137,302	3 June 1999 (03.06.1999)	US
09/378,891	24 August 1999 (24.08.1999)	US

(71) Applicant *(for all designated States except US)*: **AGENT-WARE INC.** [US/US]; 1221 Brickell Avenue, Suite 1780, Miami, FL 33131-3259 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **OTTOLENGHI, Leslie, M.** [US/US]; 1210 Oglethorpe Drive, Atlanta, GA 30319 (US). **MARTIN, Glenn, D.** [US/US]; 2358 Howell Mill Road, Atlanta, GA 30318 (US).

(74) Agents: **DIPPERT, William, H.** et al.; Cowan, Liebowitz & Latman, P.C., 1133 Avenue of the Americas, New York, NY 10036-6799 (US).

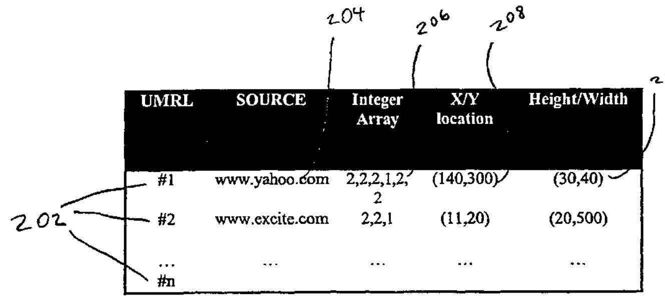
(81) Designated States *(national)*: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG,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

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

Published:
— *With international search report.*

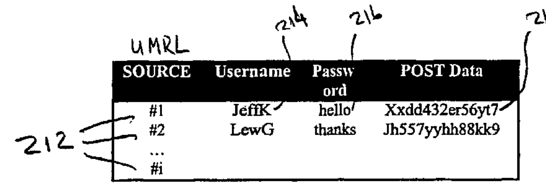
For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: PERSONALIZED METABROWSER


204
206
208
210
202

UMRL	SOURCE	Integer Array	X/Y location	Height/Width
#1	www.yahoo.com	2,2,2,1,2, 2	(140,300)	(30,40)
#2	www.excite.com	2,2,1	(11,20)	(20,500)
...	...	...	...	...
#n				

(b)


UMRL
214
216
218
212

SOURCE	Username	Passw ord	POST Data
#1	JeffK	hello	Xxdd432er56yt7
#2	LewG	thanks	Jh557yyhh88kk9
...			
#i			

(c)

(57) **Abstract:** The inventive application dynamically configures content data segments selected from numerous Internet data resources (202), such as documents (218), based on user defined preferences. The resulting presentation template data file formates a "real time" customized display (212) that gathers and aggregates selected content data segments into a single viewing area. The inventive presentation template does not store the actual data of the stored segments. Contrarily, what is being saved are the addresses of the parent document (202) as well as the coordinates of the actual data segments within their parent document (202). This customized presentation template may be shared with other users via e-mail, storage medium, or proxy server. It may be displayed by a custom-built or any off-the shelf commercially available Internet browser (204) on any Internet connected device, e.g., PC, Set, Set Top Box, personal digital assistant, etc.

PERSONALIZED METABROWSER

CROSS-REFERENCE TO RELATED APPLICATIONS

The following patent claims the benefit of a Provisional U.S. Patent Application Serial No. 60/137,302, entitled "Information for Personalized Browsing/Intelligent Agent Software, which was filed on June 3, 1999 and is incorporated herein by reference.

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to the presentation of data retrieved from the Internet, and more specifically to building a presentation template for presenting various data segments of multiple Internet documents dynamically on a display screen.

Description of the Background of the Invention

The onset of Internet technology has made it possible for users of computing devices connected to the Internet to access vast and ever-increasing sources of information. However, despite the wide availability and volume of such information, the means of bringing diverse elements of such information together in real-time for the purpose of displaying them on a single presentation page or screen is still evolving.

Commercially available programs designed for retrieving and displaying Internet documents are called browsers. Examples include Microsoft Internet Explorer™ and Netscape Navigator™. Browsers download and display Internet documents, which may be found in hyper-text markup language (HTML) or embedded extensible markup language (XML) format, one document at a time. These documents may be retrieved from Internet-connected devices, called Internet sites, which are assigned discrete Uniform Resource Locator (URL) addresses.

Where a single browser is employed, two separate documents may be viewed by requesting, retrieving, and viewing a first document, followed by the request, retrieval, and viewing of the second document. However, the first retrieved document will be lost from view when the second is viewed, although BACK and FORWARD browser navigation commands may allow fast re-presentation of a previously retrieved (and now lost to view) document. This problem can be somewhat alleviated by using more than one copy of a browser. Where multiple copies of a browser are employed, multiple separate documents may be viewed by requesting each copy of the browser being used to retrieve and display a separate document. In the windowing environment of most current operating systems, e.g., WindowsNT™, multiple documents retrieved by individual browsers may be viewed simultaneously, side by side, or in whatever manner the user chooses to arrange the viewable windows.

To facilitate solutions to problems of simultaneous retrieval of a plurality of Internet documents, intelligent agent software may also be used. Intelligent agent software, shown in Figure 1, is used to perform a wide variety of activities on the Internet, including an assortment of computing tasks such as searching, evaluating, reconfiguring, and filtering of documents. Intelligent agent software may act autonomously on behalf of users, and may sense the state of their computing environment, adapt to changes, and perform intended activities according to predefined and learned parameters.

When multi-tasking intelligent agent software is used to download Internet documents, it may automatically load a plurality of predetermined documents from a variety of Internet sites to the user's desktop at one time. This multi-tasking ability allows intelligent agent software to create a "personal portal" on the user's device, which may facilitate finding, qualifying, comparing, and procuring products and services online.

The one issue that intelligent agent software does not address or resolve, is the arrangement and presentation of the aggregated content to the user on a single display page. Products currently available to search and display multiple Internet documents include applications such as CatchTheWeb, a description of which may be found at http://www.catchtheweb.com/. CatchTheWeb's site describes the application as follows:

> "… an integrated development environment for managing your page archive. You can view the pages, re-organize their order, remove them, change their titles, and revise your notes about them."

Although programs such as CatchTheWeb may solve the problem of collecting related information, the end product of CatchTheWeb is a list of Internet documents (pages) which can only be viewed one document at a time. The comprehensive presentation of data of interest filtered and collected from more than one page and displayed on a single page display is not addressed by this product.

Presently, a need exists to customize the display of retrieved Internet documents. Customized displays may include a whole document, a part of the document, or just one data segment presented in a document. A single data segment may be static or it may be dynamically modified on a regular or irregular basis by its host Internet site. For example, a vacation resort may have an HTML/XML document that describes the resort, its room availability and its prices. Another related HTML/XML document may describe local amenities, i.e., restaurants, hotels, theaters, and recreation facilities near the hotel. A different Internet document may include daily weather reports for each city in the state where the target vacation resort is located, including water and air temperatures updated hourly.

Using today's browsing technology, an Internet user is required to retrieve and view each of these documents separately: the resort document, then the entertainment document, followed by the weather document, although not necessarily in that order. And, as noted above, viewing one document may preclude viewing another at the same time. Additionally, users interested only in the resort's weather would be required to retrieve and display the entire HTML/XML document containing that information, an inconvenient process when only one element of the page on which such information appears is desired.

Therefore, what is needed is a mechanism to retrieve multiple HTML/XML documents located in a variety of separate URL locations, select the information from them in which the user is interested, and display that information all at once on a pre-formatted presentation document, according to instructions found in a presentation template, and all on a single page. Furthermore, the size (in bits) of the presentation template must be minimal so that the aggregated content may be displayed on Internet connected devices of various capacities.

SUMMARY OF THE INVENTION

The present invention allows a user of an Internet connected device to build presentation templates for the display of data segments from multiple documents. These documents may be retrieved by browsers or intelligent agent software.

All requested documents are gathered and parsed into nodes containing data segments. These documents are then displayed showing selectable data segments separated by lines. The user peruses and selects wanted data segments. A display re-presentation is then built by arranging selected data segments in appropriate windows of the display re-presentation.

Display re-presentations may also be constructed from a prefabricated set of display re-presentations and multi-media presentation metaphors. To complete the formation, the following data for all selected data segments may be stored in the presentation template:

1. the address of the document, e.g., the URL;
2. the location of each data segment within its parent document, as described by counting "begin" and "end" tags from the start of the document;
3. the size of the display re-presentation and the arrangement location of each data segment within the display re-presentation; and
4. attribute information for each data segment.

After completion, the presentation template may comprise instructions for making Internet requests for target Internet documents, intelligently selecting data segments from these Internet documents, and presenting these data segments in an organized fashion to a viewer.

Upon being employed by a presentation segment of the inventive software, at a user's request or on a scheduled basis, the presentation template may provide instructions to retrieve specific documents, translate exact content locations of these documents, and display these contents in specified formats. The user pre-configured presentation template allows for real-time retrieval and selection of multiple data segments from multiple Internet sites and other sources, locally or via the Internet. The presentation template may be used for execution by the inventive application or may be executed by any off-the-shelf browser.

BRIEF DESCRIPTION OF DRAWINGS

The foregoing objects and advantages of the present invention may be more readily understood by one skilled in the art with reference being had to the following detailed description of a preferred embodiment thereof, taken in conjunction with the accompanying drawings wherein like elements are designated by identical reference numerals throughout the several views, and in which:

Figure 1a is a pictorial representation of the environment in which the inventive software creates a "personal portal" for accessing a plurality of documents.

Figure 1b is a diagram of a computer system used for implementing the present invention.

Figure 2 is a sample retrieved document.

Figure 3 represents the data of the document shown in Figure 2.

Figure 4 is a flowchart of an inventive process for retrieving and parsing documents and for building a presentation template.

Figure 5 is a flowchart of an inventive process for parsing individual retrieved documents.

Figure 6a shows the document data of Figure 3 after it has been parsed by the parsing process of Figure 4.

Figure 6b is an illustration of an object tree constructed in accordance with the document 20 as shown in Figure 2.

Figures 7a-d are samples of retrieved document of Figure 2, displaying separated, selectable data segments.

Figure 8a is an user interface view combining component programs of the invention and a document display.

Figure 8b is an user interface view for building the presentation template.

Figure 9a is a flowchart of an inventive process for building the inventive presentation template.

Figures 9b and c represent a data file or a blueprint for storing pointers to data segments.

Figure 10a is a diagram of a computer system used for displaying the presentation template of the present invention.

Figure 10b is a flowchart of an inventive process for displaying the inventive presentation template.

Figure 11 is a flowchart of an inventive process for translating and displaying retrieved selected Internet documents according to the inventive presentation template.

Figure 12 is a flowchart of an inventive error checking process.

Figure 13 is a display of rearranged data segments from the document of Figure 8a.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT OF THE INVENTION

Referring now to the drawings and in particular to Figure 1a, there is shown a network system 10, which comprises a computer system 12 upon which the software of this invention is executed, a plurality of Internet-connected devices or websites 14a-g, -a plurality of databases 16a and b, and a network 11 for interconnecting the computer system 12 to each of the websites 14 and the databases 16. In one illustrative embodiment, the present invention may be implemented over a network 11, such as the Internet or intranet. Alternatively, a network may not be necessary, where the data may be retrieved from a storage device.

Where the network 11 takes the form of the Internet, the computer system 12 may be engineered to work with existing products built using interactive Internet (web) development tools such as a common gateway interface (CGI) 18 using Perl scripts, and an active server agent 18, which works with documents written in Java and Java/Visual languages. Through the use of these tools, the present invention may simultaneously access and retrieve data from the plurality of Internet or web sites 14a-g, as well as the Internet connected databases 16a and b.

Referring now to Figure 1b, the details of the computer system 12 are shown. In one illustrative embodiment of this invention, the computer system 12 comprises a bus 1, which is connected directly to each of a central processing unit (CPU) 2, a memory 3, a video interface 4, an input/output (I/O) interface 6, and a communications interface 8. The common bus 1 is connected by the I/O interface 6 to a storage device 7, which may illustratively take the form of memory gates, disks, diskettes, compact disks(CD), digital video disks (DVD), etc. The video interface 4 couples a display 5 to the common bus 1. The communications interface 8, e.g., a modem, is coupled to an internet connection 9, e.g., an Internet Service Provider (ISP), which in turn is connected to the network 11, whereby a data path is provided between the network 11 and the computer system 12 and, in particular, its common bus 1.

The computer system 12, as shown in Figures 1 a and b, operates in the context of this invention to transmit requests via the network 11 to selected websites 14 to access and download therefrom data, which represent a document 20 as illustrated in Figure 2. The document 20 is transferred to the computer system 12, where it is processed one data element at a time and displayed on the display 5 as the document 20 shown in Figure 2. The downloaded document data as processed corresponds to the text shown in Figure 2, i.e., "Our rates and prices:" as well as the rest of the illustrated text. In addition, the document data also includes a collection of embedded codes that explain the meaning or desired formatting for the marked text. The embedded codes include instructions or commands, which control what the document will look like and/or where it will be positioned within the document 20. Though the markup language illustratively shown herein is the well-known Hyper Text and Extensible Markup Languages (HTML or XML), it is appreciated that there are other such delimited formats, e.g., "Acrobat" or "WordPerfect", which may be employed in this invention. It is appreciated that only the marked up text, but not the markup language itself, will appear in the displayed document 20 of Figure 2.

The document data transmitted over the network 11 from a selected website 14 is comprised of the data text shown in Figure 2 and the embedded codes for imparting the arrangement of that text as shown in Figure 2. Physical arrangement of the document data is re-arranged in Figure 3 to illustrate how the embedded codes delimit data containers, and how those data containers are nested at different levels within the document data.

As will be explained below, the data containers are assigned various numbers. As also shown in Figure 3, each of the plurality of data containers comprises the content that is identified by the numeral 24, and a pair of embedded codes, known in computer terminology as "tags" or "tokens" 22 and 26. These tags 22 and 26 are disposed on either side of their data content 24 and are respectively known as "begin" tags 22 and "end" tags 36, respectively. The begin tags 22 and the end tags 26 are both constructed of closed brackets "< >", but differ from each other in that only the end tags 26 include a "/". In one embodiment of this invention, the data container is identified by the numeral 32a and enclosed by a rectangularly shaped block to readily recognize the data container 32a. The begin tag 22 "<H1>" and the end tag 26 "</H1>" are disposed in the data container 32a directly on either side of the data text 24, "Our rates and prices". Contrast this configuration of data container 32a with that of container 32b, where the begin tag 22 <TABLE> and the end tag 24 </TABLE> are surrounded by the rectangularly shaped box 34b; the begin tag 22 and the end tag 26 are displaced from each other and serve to delimit the data content 24 contained in the box 32d. Though the data content 24 has been described in terms of text and shown in Figure 2 as such, it is appreciated that such content may comprise video data, images and/or links to other websites 14 that are connected to the network 11.

The boxes shown in Figure 3 illustrate the nesting of the data containers with respect to each other and at a plurality of levels. Figure 3 shows an illustrative embodiment of this invention in which the nesting of the data containers occurs at 6 levels. The container segments 32a and 32b, which are identified by the rectangular boxes identified by these numerals, are the data containers at the highest level of the nested structure shown in Figure 3; this highest level is also known as level 1. There are no containers nested within the data container 32a. However, there are a number of data containers nested within the container 32b; in particular, there are 3 data containers 34a, 34b and 34c, which are all level 2 containers and are nested within the level 1 container

32b. Further, there are 2 level 3 data containers 40a and 40b which are nested within the level 2 data container 34b. In turn, there is 1 level 4 container 42, which is nested within the level 3 container 40 b. Still further, there are 3 level 5 containers 44a, 44b and 44c, which are nested within the level container 42. Finally in this illustrative embodiment, there are 2 level 6 containers 46a and 46b, which are nested within the level 5 container 44a.

As will be explained below, the document 20 (Figure 2), which is downloaded from one of the websites 14 (Figure 1a), is subdivided or parsed into a plurality of data segments, which are identified by the numerals 132a and 132b as shown in Figure 7a. The parsing routine parses the nested data elements 32a-b, 34a-c, 42, 44a-c, etc., by identifying and counting the number of begin tags 22 and end tags 26 within the document 20 to determine the relative location of each of the data containers within the document page 20. This identify and count operation may be performed on the entire document 20 until all nested data containers 32a-b, 34a-c, 42, 44a-c, etc. within that document 20 have been identified and located.

The begin tag 22 and end 26 tag for each data container may also be marked with other types of codes known as content attributes that determine for example the graphics, the manner of text display, the structure of the text as a table, etc.

Container relationships similar to those described above may also be found in most existing documents created by programs like Microsoft Word™, Microsoft Excel™, Microsoft Powerpoint™, Lotus Notes™, WordPerfect™, Adobe Acrobat™, etc., as well as existing database programs. As long as the document data are delimited according to some defined rules, the present invention may be used to subdivide or parse the document 20 (Figure 2) into typically smaller data structures which are termed herein data segments. Capitalizing on the fact that interactive Internet software services produce documents organized in the container fashion, e.g., in HTML/XML format, the present invention may interpret the organization of data elements in these documents and may reorganize data elements dynamically at a graphical interface level for viewing.

Figure 4 shows the steps of the information retrieving and parsing operations of this invention. At step 60, the URL addresses for the websites 14 (Figure 1a), where document 20 (Figure 2) of potential interest may be found, are retrieved to the computer system 12 (Figure 1a). The addresses may be provided manually one at a time or automatically in a canned file. Additionally, since some computer systems 12 (Fig. 1a) may be located behind Internet firewalls, an HTTP proxy may be used. For this reason, the inventive application may have the ability to configure such a proxy, so that all Internet activity may proceed through that proxy.

A determination is made at step 62, whether all of the needed data have been found. If retrieved documents 20 (Figure 2) do not contain the needed data segments, the process returns to step 60 for further document retrieval. If the documents 20 of interest have been found, the parsing of the retrieved documents commences at step 64 as shown in Figure 4. The detailed steps of parsing will be explained below with respect to Figure 5. Basically, step 60 parses or subdivides the selected document 20 (Figure 2) into a plurality data segments 132a-b (Figure 7a). The number of the data segments may be increased as needed by the user (Figures 7b, 7c and 7d).

After selecting the desired degree of granularity, the user in step 66 selects which data segments of the displayed document 20 (Figure 2) contain the data of interest and then saves a pointer to that segment so that its data may be retrieved later. In step 68, the user indicates whether there are further data segments to be identified either from the same website or perhaps another. If further data segments are to be identified, the process returns to step 60 and the document selection and parsing in steps 60, 62 and 64 are repeated until all of the desired data segments have been identified. If all the necessary data segments have been identified in step 68, the user saves the pointers or addresses to each of the selected data segments.

The detailed steps of the parsing process 64 (Figure 4) will now be explained in detail with respect to Figure 5. Initially at step 70, data representing the document 20 (Figure 2) as retrieved from its website 14 (Figure 1a) is downloaded to the computer system 12 (Figure 1a). The document data, as explained above, is a sequence of data elements, some of which are data content and others are embedded codes, and is shown in Figure 3 in a manner to illustrate the nesting structure of the data containers. At step 72, each data element is examined and the begin tags 22 and the end tags 26 are saved, while all other data content elements are discarded. The output of step 73 is illustrated in Figure 6a to show the sequence of the begin tags 22 and the end tags 26, where all data content 24 have been removed.

Each tag of this output is examined in Step 74 to determine whether the tag is a begin tag 22 and, if so, a node A and a corresponding branch of an object tree 89 is constructed as shown in Figure 6b for each begin tag 22 that is so identified. The object tree 89 has a single root node that is identified as A1, where the building process begins. Next, step 78 extracts the data type from the begin tag 22 and, in step 80, applies a label to the node A that identifies the data type of that begin tag 22, i.e., whether it is an image, text, etc. tag. Thereafter the process returns to step 73 to examine the next data element.

If step 74 determines that the tag is an end tag 26, i.e., it is not a begin tag 22, step 86 determines whether the detected end tag 26 is the final end tag in the document 20 (Figure 2). As shown in Figure 3, the final end tag 26 is the </TR> tag at the bottom of the Figure 3, which marks the end of the document data that represents the page document 20 (Figure 2). If such a final end tag 26 is identified, step 88 terminates the construction of the object tree 89 (Figure 6b). If step 86 decides that the end tag 26 is not the final end tag, step 82 traverses the object tree 89 one level or branch back towards the root node A1. There is only one path from any node A to the root node A1 for the tree constructing process to return to the root node A1. After traversing back up the tree 89, the process returns to step 73 to examine the next data element. This parsing process 64 is repeated through steps 71 - 88, until each data element of the document data has been examined and step 88 has identified the final end tag 26. The final structure of the tree 89 that corresponds to the document 20 (Figure 2) is shown in Figure 6b.

Referring now to Figures 6a and b, the object tree constructing process 64 begins at the root node A1 and initially detects the first begin tag <H1> 22 of the data container 32a, whereby the first branch connecting the root node A1 to node A2 is added to the tree 89. Next, the process

64 detects the closing node </H1> 26 of the data container 32a and moves back up the tree 89 from node A2 backwards to the root node A1. Constructing is carried out such that when a begin node 22 is detected, the process moves away from the root node A1 and backwards toward the root node A1 when an end tag 26 is detected. The process 64 next detects the first begin tag <TABLE> 22 of the next data container 32b and constructs a branch leading to a node A3. The begin node <TR> 22 of the data container 34a is detected next, whereby the branch to node A4 is constructed. Next, the first opening node <TH align = left> 22 of the nested data container 36a is detected, and the branch to node A7 is constructed. Next, the closing tag </TH> 26 of the data container 36a is detected and the constructing process moves back toward the root node A1 to node A4. Next the begin tag <TD> 22 of the data container 36b is detected, whereby a branch to node A8 is added. Next, the end tag </TD> 26 of the data container 36b is detected, whereby the process moves backward to node A4. Next, the end tag <TR> 26 of the data container 34a is detected, and the process moves backward to the node A3.

The tree building process 64 continues in this manner until the process has constructed all of the nodes A1 - A19 and the process 64 has returned to the root node A1, which occurs when the final return node </TABLE> 26 of the document 20 has been detected in step 88 of the process 64. As seen in Figure 6a, each data container may be identified by that node A, which the process 64 constructed upon detection of the first begin tag examined in that data container. The structure of the object tree 89 identifies each connection between adjacent data containers and the level of that data container in the nesting structure. As will be explained below the relative position of the first begin tag 22 of the data container 32b marks the beginning of that data container and the structure of the nested data containers as indicated by the object tree 89 dictate the placement of its end tag 26 and thus the end of the data container 32a. Likewise, the beginning tag 22 of the remaining data containers mark the beginning and end of their respective data containers as identified by each of the nodes A2 - A19 (Figure 6b). As will be explained below, the begin tags 22 will be used to mark the beginning and end of each data segment of a document 20, whereby the document 20 may be parsed or subdivided.

An alternative to this comprehensive approach to parsing may be a simple content stripping routine. Such a routine may simply find a data content element 24 (Figure 3) included between its begin tag 22 (Figure 3) and end tag 26 and strip that data content. In other words, a copy of document data without its data content but with all the tags 22, 26 (Figure 3) intact may be sufficient for the parsing process 64 (Figures 4 and 5) to examine and successfully parse. This approach although acceptable is not as thorough and comprehensive as the one described in conjunction with Figures 5, 6a and 6b, where additional information is collected and used by sequentially reading in and processing each element of document data.

Selection Of Data Segments

When all of the data containers 32a-b, 34a-c, 36a-b, 38a-b, 40a-b, 42, 44a-c, 46a-b, 48a-b, 50a-b (Figure 6) of the document 20 (Figure 2) are identified and parsed, the document 20 (Figure 7a-c) may be graphically displayed with each of these nested data containers shown upon the display 5, whereby the user may identify specific data of the downloaded document data and the size or granularity of the data segment encompassing the selected document data. The user may now point, click, and drag any or all of selectable data segments shown on the display 5 (Figure 1) to build a presentation template. After choosing all of the needed data segments from all of the parsed documents 20, the process configures a target presentation template.

The document 20 shown in Figure 7a may be displayed on the display 5 to the user with each of its data segment 132a-b visually separated by lines 54a and b. These lines separate and delineate the document 20 into data segments 132a-b, i.e., line 54b delineates the data segment 132a from the data segment 132b. The lines 54a and 54b are generated in accordance with the respective position of the first begin tag <H1> 22 of the data container 32a and the first begin tag <TABLE> 22 of the date container 32b. It is also appreciated that the data segments 132a and 132b display the same data content as their related data containers 32a and 32b, respectively. Further as seen in Figure 6a, the data segments 132a and 132b display data from the same level of their data containers 32a and 32b, namely level 1. The presentation of the document 20 may be scrollable. Moreover, additional icons 56a may be superimposed by the display 5 on a particular data segment to indicate, for example, that the data segment 132b may be parsed or subdivided into further data segments. This is achieved purposefully so as not to inundate the user with the granularity of a display which may be too small and unnecessary. If the user desires to increase the granularity for example of data segment 132b, the user merely clicks on the icon 56a to cause the document 20 to be further divided, as shown in Figure 7b, into data segments 134a, 134b and 134c, which display content from the level 2 containers 34a, 34b and 34c. Similarly, the icon 56a may be again clicked to further parse the document 20 as shown in Figure 7c to further include the data segments 136a-b, 138a-b, and 140a-b, which display data from the level 3 containers 36a-b, 38a-b and 40a-b. If the icon 56a is again clicked, the document 20 will be further subdivided to include as shown in Figure 7d segments 146a-b, 148a-b and 150a-b, which display data from the level 6 data containers 46a, 46b and 46c.

Additionally, where document data elements are found to use embedded logic, frames, and pointers to other documents, then those items may be activated, loaded and parsed as if they all came from the same document data. This may be achieved recursively until there is no more data to process.

For further discussion please consider, as shown in Figure 8a, document 90 having data segments 92 separated by lines 54. The document 90 is presented as a window in the inventive user interface 94, which may be shown on display 5 (Figure 1). By a selection of a menu 96 of icons 98, a user may control which operations are called into service. For example, the user may click on any of an icon 100 to initiate the retrieving and parsing operations 60 and 64 described above, an icon 102 to initiate the compiling and building functions as will be described, and an icon 104 to initiate the distributing and opening of completed presentation templates as will be described, and an icon 106 to call the displaying and translating operations as will be described. All component parts of the invention may be included within the user interface 94, individually or in combination.

Presentation Template Builder

After the compiling/building 102 icon is activated and all of the retrieved documents 20 have been parsed, each user desired data segment 92 may be selected with a mouse or a keyboard. Selecting the data segment places that data segment and its related pointers, such as the address of the parent document 20 and a list of the begin tags 22 and tags 26 (Figure 6) from the start of the document data 20 (Figure 3) to the first begin tag 22 of the data segment comprising the data container; these pointers are stored in the builder file and may be displayed in a builder window 110 (Figure 8b).

The builder window 110 may be displayed within the user interface 94. That window 110 comprises data segments 92a selected from the document 90 (Figure 8a) and the segment 92b selected from another document, not shown. Once data segments are displayed in the builder window 110, they may be moved to form a presentation template.

By activating the compile/build icon 102, the parsing step 64 (Figure 2) is initiated. A user may pre-select the size of a presentation template array 112. Depending on how many data segments 92a-b are expected to be displayed within the formed presentation template. For example if the resultant presentation template display will have four items, the appropriate size of the presentation template array 112 may be two boxes by two boxes. The presentation template array 112 shown in Figure 8b is four by four boxes and is able to accommodate 16 data segments. The individual boxes of the presentation template array 112 may dynamically change shape and size to accommodate the required number of data segments 92a-b to be displayed therein. To build a presentation template, the user may simply click on a selected one of the data segments 92a-b and drag and drop the selected segment into a desired presentation template array box 114. The letter identifying the object 116 will appear in the presentation template array box 114.

Figure 9a shows the progression of steps in the builder process of the invention. After all of the desired data objects are selected, step 120 selects the coordinate of each data segment's coordinates from the object tree 89 generated by the parsing process 64 (Figure 4). The data segment's coordinates comprise a target website 14, e.g., the URL address, and the path to the desired data segment in the object tree 89. The path comprises all begin tags 22 and end tags 26 starting from the first begin tag 22 in the document 20 to the begin tag of the selected data segment. Before this address information is saved at step 128, it is stored in a holding area in step 122, and the user ascertains that no rearrangement of the data segments is necessary at step 124. Otherwise data segments are rearranged at step 126 and the user acceptance is again sought at step 124.

Pointers to data segments are saved in one illustrative embodiment of this invention as a set or collection of records called Uniform Micro Resource Locators (UMRL). As shown in Figure 9b, each pointer or UMRL 202 may contain elements listed below.

1) An address 204, such as the URL address for the website 14 (Figure 1a) where the document 20 (Figure 20) may be found.

2) An array or a string of integers 206, each integer referring to a unique container at each level in the object tree, described above in conjunction with Figure 6b. In the object tree shown in Figure 6b, containers are labeled at each branch from left to right with integers 1 A2, A4, A7, A9, A11, A14, A16, A18, 2 A3, A5, A8, A10, A12, A15, A17, A19, and 3 A6. Higher integers may be used where more container branches exist. The container A19 in the lower

right hand corner of the object tree 89 may be addressed with the set of integers (2 A3, 2 A5, 2 A10, 1 A13, 2 A15, and 2 A19).

3) Screen display coordinates 208 (Figure 9b) for displaying the data segment. In one embodiment of the present invention these coordinates may be specified with X, Y coordinates, for example a screen column number for the X axis coordinate and a screen row number for the Y axis coordinate. In a different embodiment of the present invention these coordinates may be specified with the location and size of the data segment as it may appear on a computer screen, these coordinate having been defined when the data segment was positioned in the presentation template array 112 (Figure 8b), where the location may be represented by coordinates (1,1) of the cell 114 (Figure 8b) within the array 112 (Figure 8b).

4) The height and width attributes 210 (Figure 9b) of the data segment, attributes may be these retrieved during parsing of the document 20 (Figure 2) or user defined.

Optional authentication information may be necessary for accessing password protected documents requiring user authentication. As shown in Figure 9c, if an authentication method is used at a particular document associated with an UMRL 212, then information comprising of a username 214 and password 216 pair is saved as supplementary data fields. If the authentication method used requires a POST request, a command that may be given to a computer system so that it returns a particular input form to be filled out by the requestor, then information required to submit the POST request to authenticate the user at the host 218 may also be stored.

This completes the description of the construction of the presentation template data file. The resulting data file may now be delivered by any means available, i.e., a diskette, e-mail, Internet download, etc. to any intended user and loaded into the user's device. To execute the presentation template data file a user's device 220, as shown in Figure 10a, may comprise a bus 221 which is connected directly to each of a CPU 222, a memory 223, a video interface 226, and a communications interface 228. No additional storage, e.g., a disk storage 7 (Figure 1b), is required; therefore the invention may be utilized on devices such as digital phones, palm top devices, personal digital assistant (PDA) devices, and digital television's set top boxes, among others. Proxy services may be utilized for smaller digital appliances to facilitate the implementation of the invention.

Presentation Template Display

Figure 10b shows the process for displaying the presentation template formed according to this invention. At step 130, the presentation template or blueprint is loaded and individual data segments are extracted. At step 132, all information associated with each data segment is fed into the translator routine until all data segments are translated. Then at step 134, a dynamic view designed with the help of the builder section of the invention, where the presentation template array was constructed, is filled in with all of the found and retrieved data segments. The filled-in presentation template document is then displayed to the user, after which the process terminates at step 136.

The detailed steps of the translator process 132 are shown at Figure 11. There at step 152, the URL address of the website 14 from which the document 20 and data segment's relative location in the parsed tree 89 is accepted. At step 154, the data segment's parent document 20 is retrieved, followed by a determination by an error routine at step 156 of whether the retrieved data segment is valid, non-existent, or whether different unrelated information now occupies the selected data segment's location in the retrieved parent document 20. If it is determined that the retrieved data segment is not the intended data, an error message is generated at step 158 and the erroneous data segment will not be displayed. If step 160 determines that the data segment is valid, then step 160 retrieves its data in accordance with saved data segment attributes.

The error checking step 156 may be performed for each data segment individually, or the validity of all data segments of a loaded presentation template may be examined at the same time. The routine shown in Figure 12 may load data segments' addresses into a buffer or a set of registers at step 162. An iterative loop may be initialized at step 164, and each data segment may be processed one at a time until all of the data segments have been examined and the process returns to routine 132 (Figure 11) at step 174. At step 166 a determination is made as to whether the requested document 20 still exists at a given website 14, which matches the URL saved by the presentation template. If it does not, the requested data segment is no longer valid. In such a case, step 168 generates an error message and the process returns to the translator routine 132 (Figure 11).

On the other hand if the requested document is found, the document is read in at step 170 and its contents are parsed in the same manner described above in connection with Figure 5 at step 172. Furthermore at step 172, the newly calculated object tree 89 and those trees saved in the presentation template are compared. If the tree structures match, the data segments are accepted as valid, and the process returns to the top of the loop at step 164 to continue checking addresses until all data segments are tested. If however the tree structures 89 do not match, the data segment is no longer deemed valid. In such case, step 168 generates an error message and the process returns to routine 132 (Figure 11). In a further embodiment of this invention, it is contemplated that the comparison software may be able to detect whether a newly generated tree that differs from the stored tree may nevertheless by acceptable dependent on the nature and degree of the differences of these tree structures.

When the display process of Figure 10b is started, a list of available presentation templates may be displayed to the user, who may be asked to chose from among these presentation templates. The name of each presentation template may be the same name that was given when the presentation template was created and saved. Clicking on any one presentation template may cause the inventive application to load the presentation template into memory and to begin retrieving and parsing these documents. After all of the documents referenced in the presentation template are retrieved and parsed, selected data segments from each of these documents may be copied and displayed at coordinates and with attributes described in the presentation template. The displayed content itself is not saved in the presentation template of this invention, rather the data content is retrieved from the websites on as needed basis which allows for any content updates to be reflected.

Figure 13 shows a presentation template display 180 which is a display of rearranged data found in the document 90 (Figure 8a).

Time Slicing

Another embodiment of the present invention may include a time slicing feature. This feature may be used with data segments as described above, as well as with video broadcast content. With time slicing, a plurality of data segments 92a-b (Figure 8b) are entered into each presentation template box 114 (Figure 8b). Utilizing simple programming techniques, each entered data segment may be displayed for a certain time duration in a round robin or any other order according to the user's preferences. Time slicing for video broadcast content may be performed by substituting begin tags with start time and end tags with end times and by displaying the video segments accordingly.

Uses

The present invention may be used to customize and to individualize presentations and functionality of information services available through the public networks and subscription based network services such as cable television. Specifically, these applications may include empowering stock brokers, real estate agents, travel agents, and other info-media based businesses to build presentation templates for their customers throughout the customer care continuum, i.e., the find-it, qualify-it, compare-it, shop-it, buy-it, deliver-it and service-it stages of the purchasing process.

Consumer applications may allow consumers to build virtual compilations of information such as the presentation templates described above to share with other consumers thus creating a community aspect of information sharing. An example of this may be enabling users to aggregate a virtual CD by assembling links and song titles of MP3 files on the Internet. The chosen song titles are stored in the presentation template together with a metaphoric CD player. Such a template may be sent to a receiving party, who may then view the aggregated song links and then download and play the intended audio segments from their Internet-connected devices or offload the presentation template to a MP3 player for later playback.

Presentation template files may contain username/password pairs for securing presentation templates from unauthorized use. These user names and passwords should be saved in the presentation templates in an encrypted format. When the presentation template is saved, the user may be asked for a password. This password may be used to generate the Triple Data Encryption Standard (DES) key that is used for encryption and must be remembered for the time when presentation templates are reloaded in the future. If the presentation template is sent to another user, then the password must also be provided to that person. The user may also have the option of saving an unencrypted presentation template so that no password may be required.

Additionally, some Internet and non-Internet documents may be secured with a user name/password pair. To address such a documents either a hyper text transport protocol (http) authentication or a custom mechanism that utilizes Point of Sales Terminal (POST) requests from the user's browser may be used. In either situation, the user must provide needed user names and passwords to be stored with the presentation template.

While the invention has been particularly shown and described with respect to illustrative and preferred embodiments thereof, it will be understood by those skilled in the art that the foregoing and other changes in form and details may be made therein without departing from the spirit and scope of the invention that should be limited only by the scope of the appended claims.

CLAIMS

Having thus described our invention, what we claim as new, and desire to secure by Letters Patent is:

1. A method for creating a template comprising coordinates for displaying on a display of a user computer one or more segments from one or more documents, said documents comprising said segments, the method comprising steps of:

a) retrieving said one or more documents, each of said documents comprising data;

b) parsing said data into a one or more containers, each of said containers corresponding to each of said segments;

c) displaying said documents in a form to mark said one or more segments according to a beginning and an end of a corresponding container;

d) selecting from among said marked segments to form a display view comprised of said selected marked segments; and

e) placing coordinates of said corresponding containers of said selected marked segments into cells of an array representation of said display view, to create said template.

2. The method of claim 1, wherein said one or more documents are retrieved according to a location of said one or more documents.

3. The method of claim 2, wherein said one or more documents are retrieved according to a keyword associated with

said one or more documents.

4. The method of claim 3, wherein many of said one or more documents are simultaneously retrieved.

5. The method of claim 4 wherein if said containers comprise embedded logic, frames, and pointers to further documents of said one or more documents, then said embedded logic, frames, and said further documents are retrieved and parsed as part of said containers.

6. The method of claim 5 wherein, if said containers are comprised of nested data, more of said marked segments will be displayed at step (c).

7. The method of claim 6, wherein a number of said cells of said array representation is defined by a user to accommodate a number of said selected marked segments.

8. The method of claim 7, wherein a shape of said array representation is defined by said user to accommodate a position of said selected marked segments relative to said screen view.

9. The method of claim 8, wherein each of said number of cells of said array representation will dynamically change shape and size to accommodate attributes of said selected marked segments.

10. The method of claim 9, wherein said coordinates comprise said location and a first distance from a beginning of one of said one or more documents to a beginning of said container.

11. The method of claim 10, wherein step (d) further comprises a step of compiling a list of desired containers by entering selecting said marked segments.

12. The method of claim 11, wherein coordinates of more than one selected marked segments are placed into said each of said cells of said array representation.

13. The method of claim 12, wherein if more than one container is entered into each of said cells, each of said entered data container may be displayed according to user preferences.

14. The method of claim 13, wherein if said selected marked segment is video, said segment is displayed in accordance with predefined start and end time parameters.

15. The method of claim 14, wherein said network is the Internet.

16. The method of claim 15, wherein said location is an address of one or more computer devices connected to a network, said one or more computer devices storing said documents.

17. The method of claim 16, further comprising a user interface for controlling said steps.

18. The method of claim 17, further comprising steps of:

f) saving said template including said array representation; and

g) distributing said presentation template to at least one user.

19. A method for displaying on a computer device display a display view comprised of one or more segments from one or more documents, the method comprising steps of:

a) restoring from a template an array representation of said display view and coordinates of said one or more segments;

b) retrieving each of said one or more documents for each said restored location;

c) parsing each of said one or more documents to recapture each said one or more segments for each said first distance; and

d) displaying said display view.

20. The method of claim 19, wherein said coordinates comprise a location of said one or more documents and a first distance from a beginning of one of said one or more documents to a beginning of said one or more segments.

21. The method of claim 20, wherein said location is an address of one or more computer devices connected to a network, said one or more computer devices storing said documents.

22. The method of claim 21, wherein said one or more segments are positioned in cells of said array representation, said cells representing display screen positions.

23. The method of claim 22, wherein step (c) further comprises a step of parsing each of said retrieved documents and forming a second distance from a beginning of one of said one or more documents to a beginning of said one or more segments

24. The method of claim 23, wherein if said location is invalid each said segment is not displayed.

25. The method of claim 24, wherein if said first distance does not match said second distance, said segment is not displayed.

26. A method of constructing a data file that stores a plurality of pointers, each pointer related to a data segment which is taken from document data derived from a corresponding one of at least one data source, said document data comprising an ordered sequence of data content elements and data delimiting elements, a data container comprising at least one data content element and begin and end delimiting elements disposed respectively before and after said one data content element in said document data, and a data document comprising one or more of said data segments, said data segment comprising one or more data containers, each of said pointers including an address of its data source and a coordinate locating the relative position of its data segment in said data document, said method comprising the steps of:

a) selecting and downloading at least one of said data documents from its one data source;

b) examining each of said data elements of said downloaded data segment, said examining step comprising the sub-steps of:

1) identifying said begin delimiting elements of said document data;

2) selecting at least one of said data segments from said down loaded data document;

3) identifying said begin delimiting element of a first one of said data containers in said document data of said selected data segment, and

4) defining at least in part said coordinate of said selected data segment in terms of said identified first begin delimiting element; and

c) inserting in said data file said defined coordinate of said selected data segment and said address of the one data source from which said downloaded document and its selected data segment are taken.

27. The data file constructing method as claimed in claim 26, wherein first and second data containers are embedded into said document data in a nested relationship, and said begin and end delimiting elements of said first container define respectively its starting point and its ending point, said method comprising the step of using said end delimiting element of said first data container as the end point of said selected

data segment, said identifying sub-step 3) identifies said begin delimiting element of said first container, whereby said corresponding end delimiting element of said first container encloses said first and second containers and said selected segment includes said first and second data containers.

28. The data file constructing method of claim 27, wherein said selected data segment comprises first and second data containers, said first data container being disposed at a first level and said second data container being disposed at a second level lower than said first level.

29. The data file constructing method as claimed in claim 28, wherein said identifying step 3) identifies said starting delimiting element of said second data container in said document data, whereby a corresponding second data segment is selected, said second data segment consisting of said second data container.

30. A method of constructing a data file that stores at least one pointer, said pointer identifying a data segment which comprises document data of a document retrieved from a selected one of a plurality of data sources, said document data comprising an ordered sequence of data content elements and data delimiting elements, a plurality of data containers, each

comprising at least one data content element and begin and end delimiting elements disposed respectively before and after said one data content element, and a data document comprising one or more of said data segments, said data segment comprising one or more data containers, said pointer including an address of its selected data source and a coordinate locating the relative position of said one data segment in said data document, said method comprising of the steps of:

a) selecting and retrieving at least one data document from its one data source;

b) parsing said document data of said retrieved data document into a plurality of said data containers;

c) displaying said document;

d) superimposing on said document at least one display line dividing said document into at least said one data segment according to said begin delimiting element of a first occurring data container in said data one segment;

e) defining in part said coordinate of said one data segment in terms of said begin delimiting element of said first occurring data container; and

f) inserting in said data file said defined coordinate of said one data segment and said address of the selected one data source from which said downloaded document and its said one data segment are taken.

31. The method of constructing a data file as claimed in claim 30, wherein there are first and second data segments, each of said first and second data segments respectively comprising a data container, and said method comprises the step of selecting one of said plurality of data segments, said superimposing step d) superimposing first and second lines to divide said document into first and second data segment according to said begin delimiting element of said first occurring data container of each of said first and second data segment.

32. The method of using a data file comprising a memory for storing a pointer that identifies a data source from which document data representing a document is retrieved and a data segment of said document, said pointer comprising an ID of said data source and a coordinate specifying the relative position of said data segment within its document, said document data comprising an ordered sequence of data content elements and data delimiting elements, a plurality of data containers, each comprising at least one data content element and begin and end delimiting elements disposed respectively before and after said one data content element, said document data comprising one or more of said data segments, said data segment comprising one or more data containers, said method comprising the steps of:

a) distributing said data file to at least one user;

b) employing said ID and said pointer to access and retrieve from that data source corresponding to said ID and that data segment identified by said coordinate; and

c) displaying at least one current data content element of said retrieved data segment.

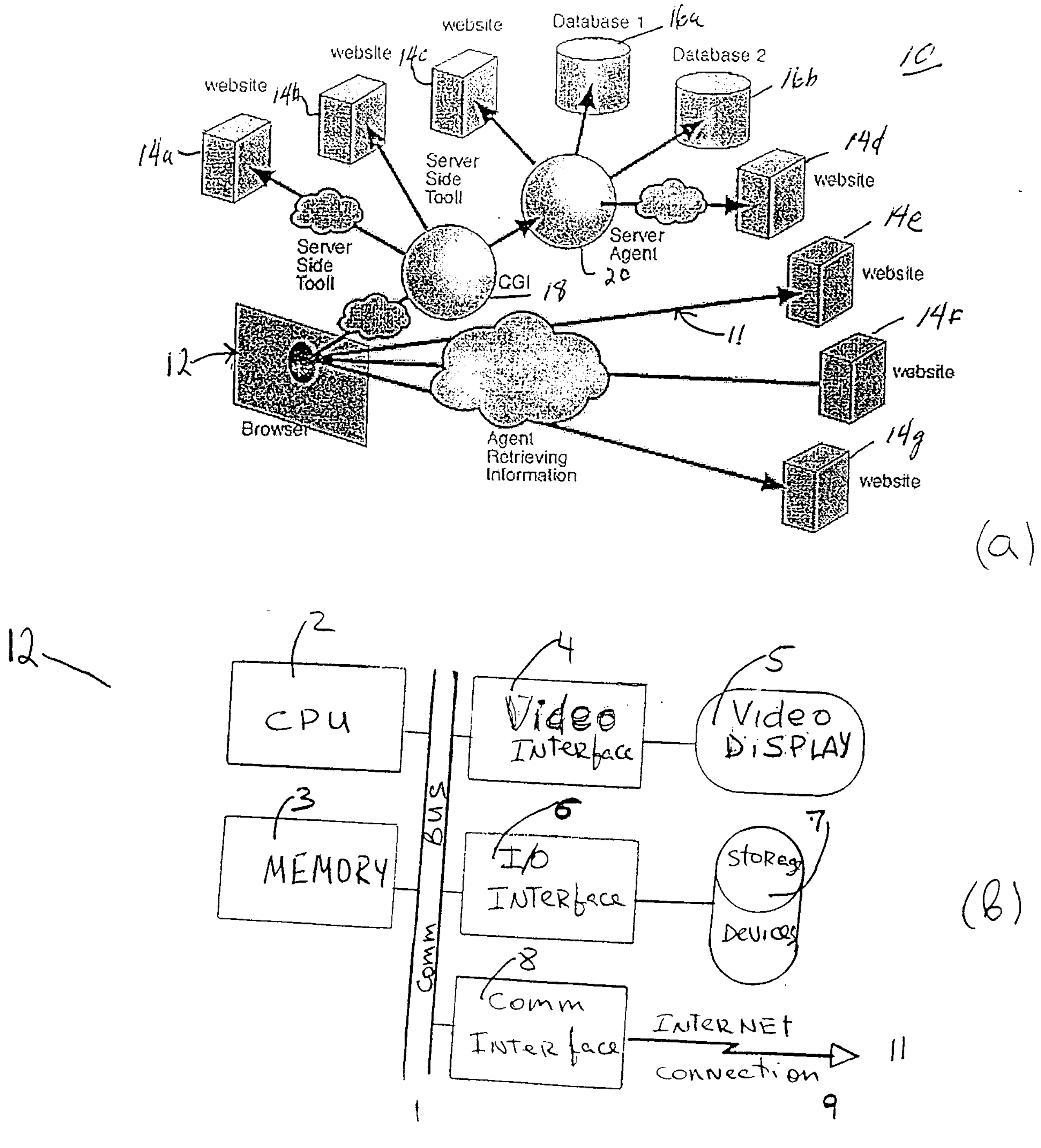
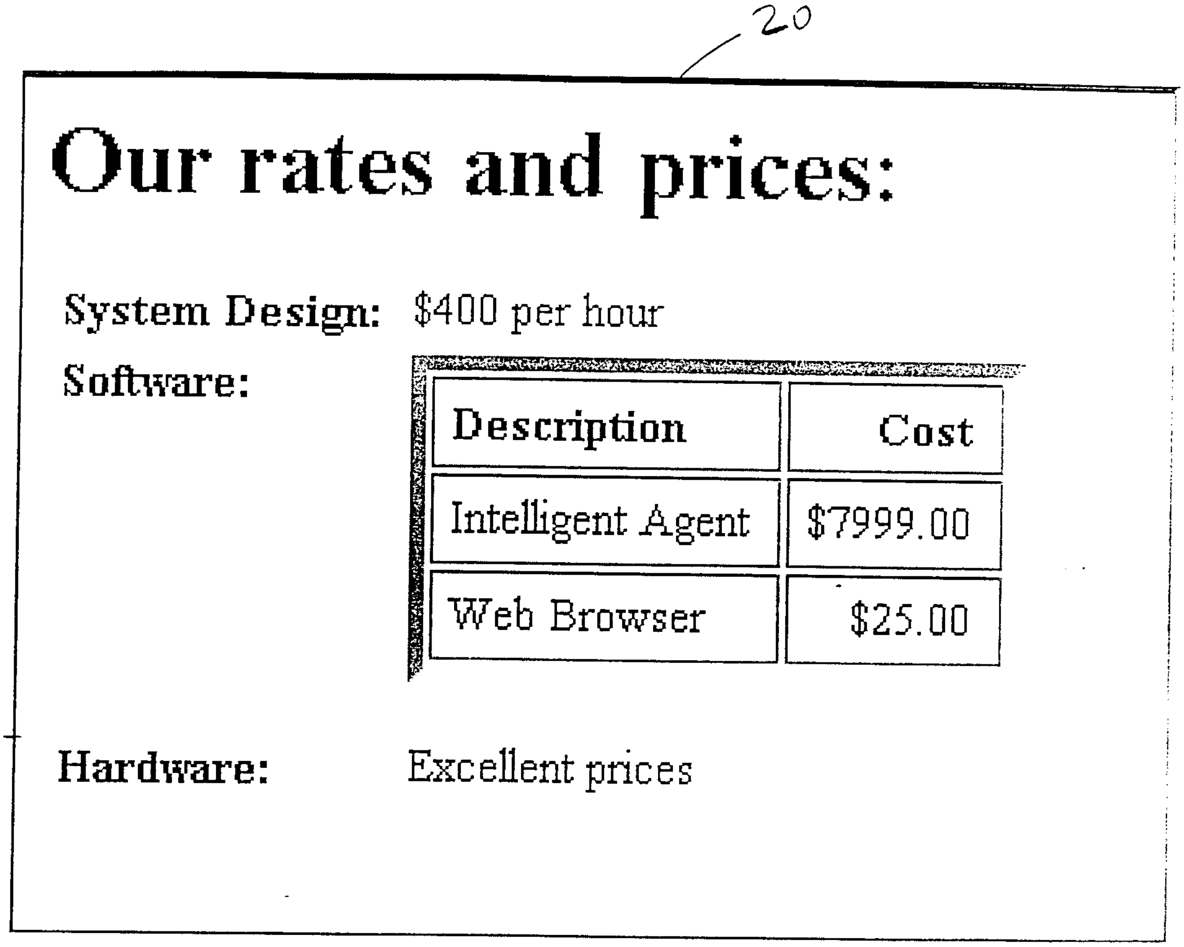

website
Database 1
16a
website
14c
Database 2
16b
10
website
14b
14a
Server
Side
Tooll
14d
website
14e
Server
Side
Tooll
Server
Agent
20
website
CGI
18
11
14f
12
website
Browser
Agent
Retrieving
Information
14g
website
(a)
12
2
CPU
4
Video
Interface
5
Video
DISPLAY
3
MEMORY
Comm BUS
6
I/O
INTERface
7
Storage
Devices
(B)
8
Comm
InteRface
INTERNET
connection
11
1
9

Figure 1



Our rates and prices:

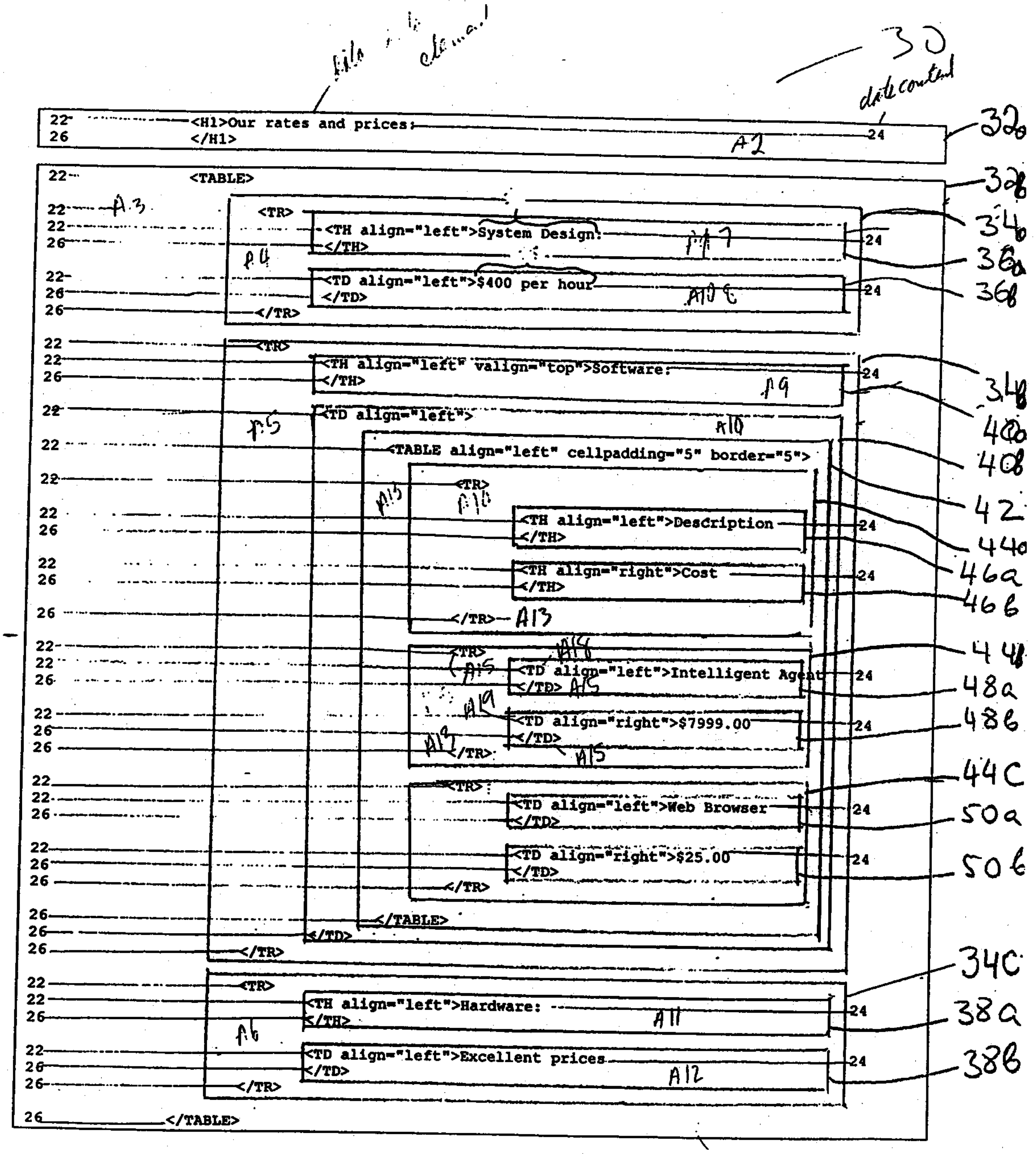
System Design: $400 per hour

Software:

Description	**Cost**
Intelligent Agent	$7999.00
Web Browser	$25.00

Hardware: Excellent prices

Figure 2: Sample Web document



```
<H1>Our rates and prices:
</H1>
<TABLE>
<TR>
<TH align="left">System Design:
</TH>
<TD align="left">$400 per hour
</TD>
</TR>
<TR>
<TH align="left" valign="top">Software:
</TH>
<TD align="left">
<TABLE align="left" cellpadding="5" border="5">
<TR>
<TH align="left">Description
</TH>
<TH align="right">Cost
</TH>
</TR>
<TR>
<TD align="left">Intelligent Agent
</TD>
<TD align="right">$7999.00
</TD>
</TR>
<TR>
<TD align="left">Web Browser
</TD>
<TD align="right">$25.00
</TD>
</TR>
</TABLE>
</TD>
</TR>
<TR>
<TH align="left">Hardware:
</TH>
<TD align="left">Excellent prices
</TD>
</TR>
</TABLE>
```

Figure 3

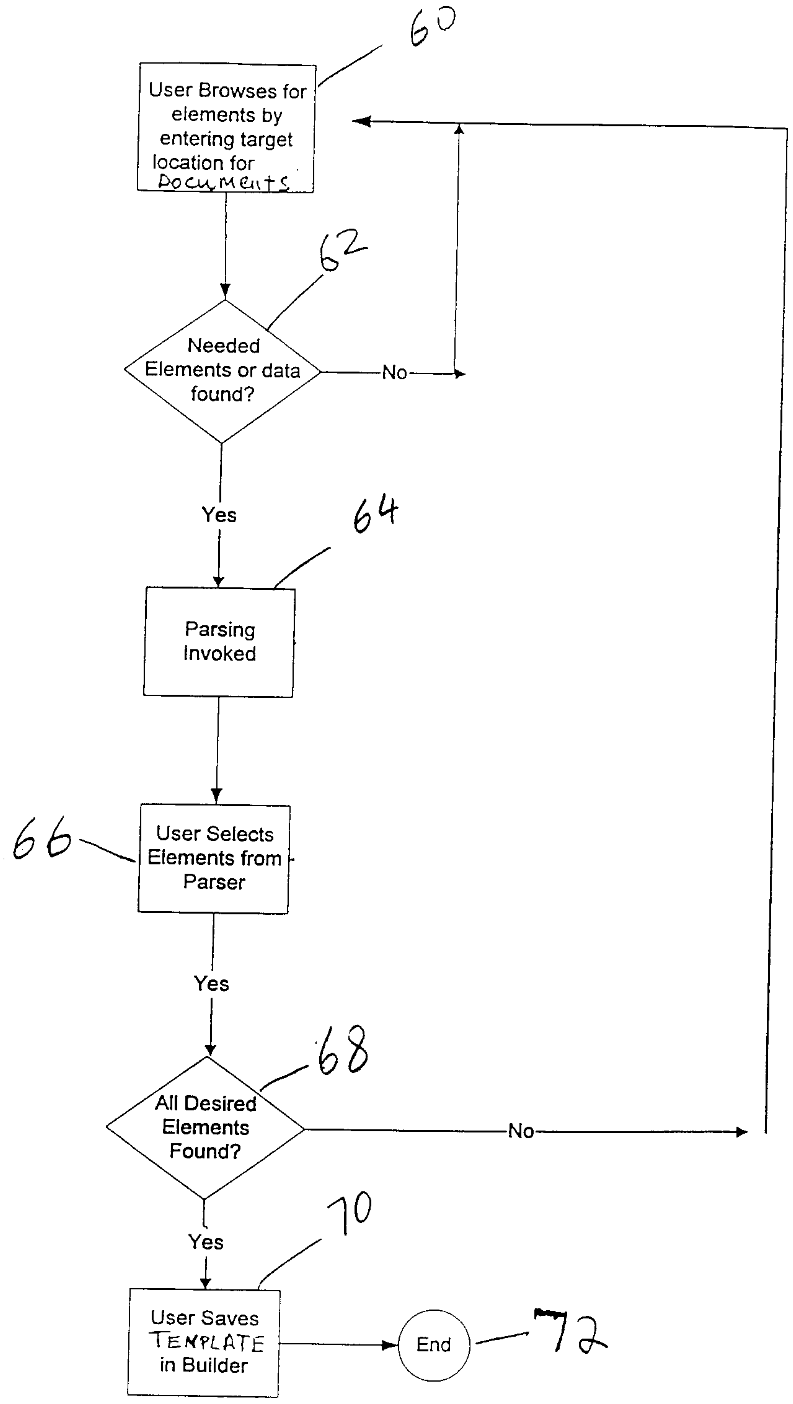

60
User Browses for elements by entering target location for Documents
62
Needed Elements or data found?
No
Yes
64
Parsing Invoked
66
User Selects Elements from Parser
Yes
68
All Desired Elements Found?
No
Yes
70
User Saves TEMPLATE in Builder
End
72

Figure 4

Parsing Flowchart

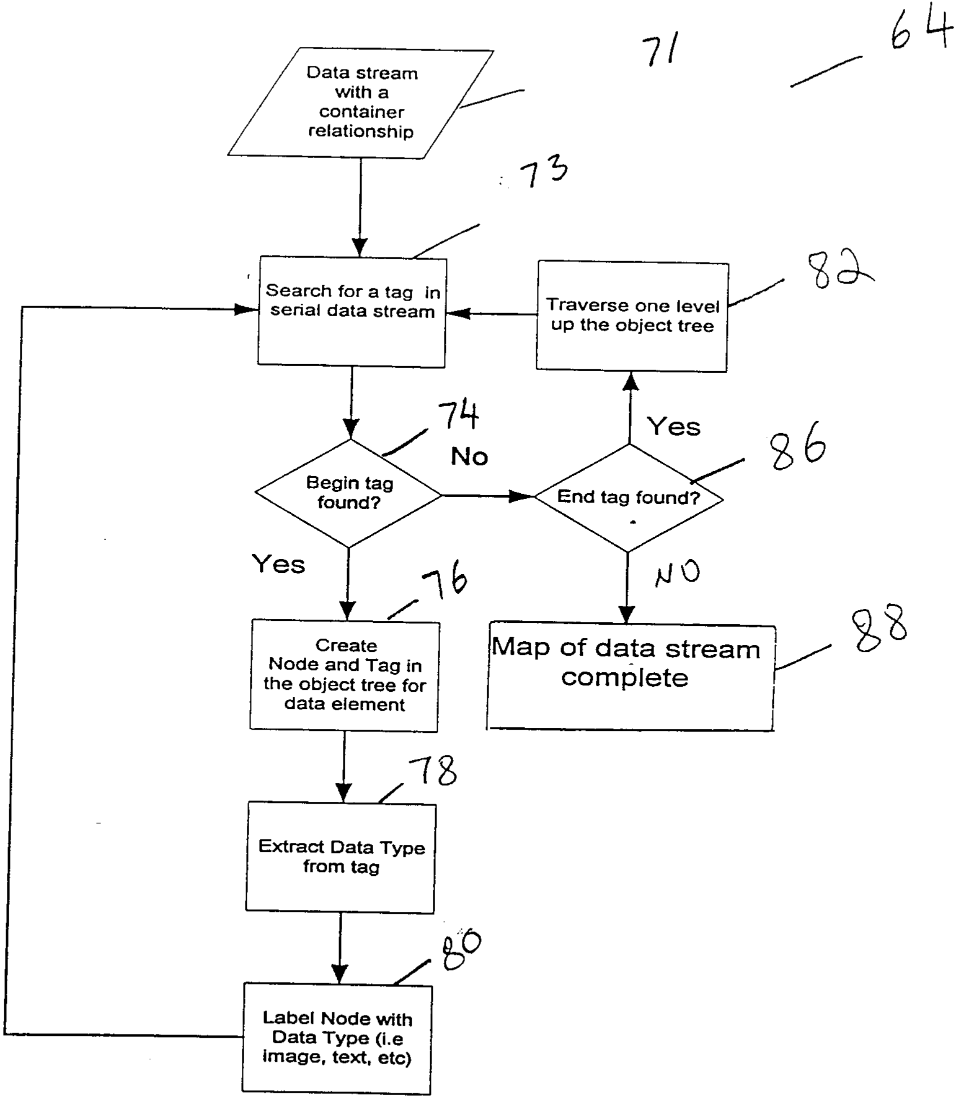

64
71
Data stream with a container relationship
73
Search for a tag in serial data stream
Traverse one level up the object tree
82
74
Begin tag found?
No
End tag found?
Yes
86
Yes
NO
76
Create Node and Tag in the object tree for data element
Map of data stream complete
88
78
Extract Data Type from tag
80
Label Node with Data Type (i.e image, text, etc)

Figure 5

6/20

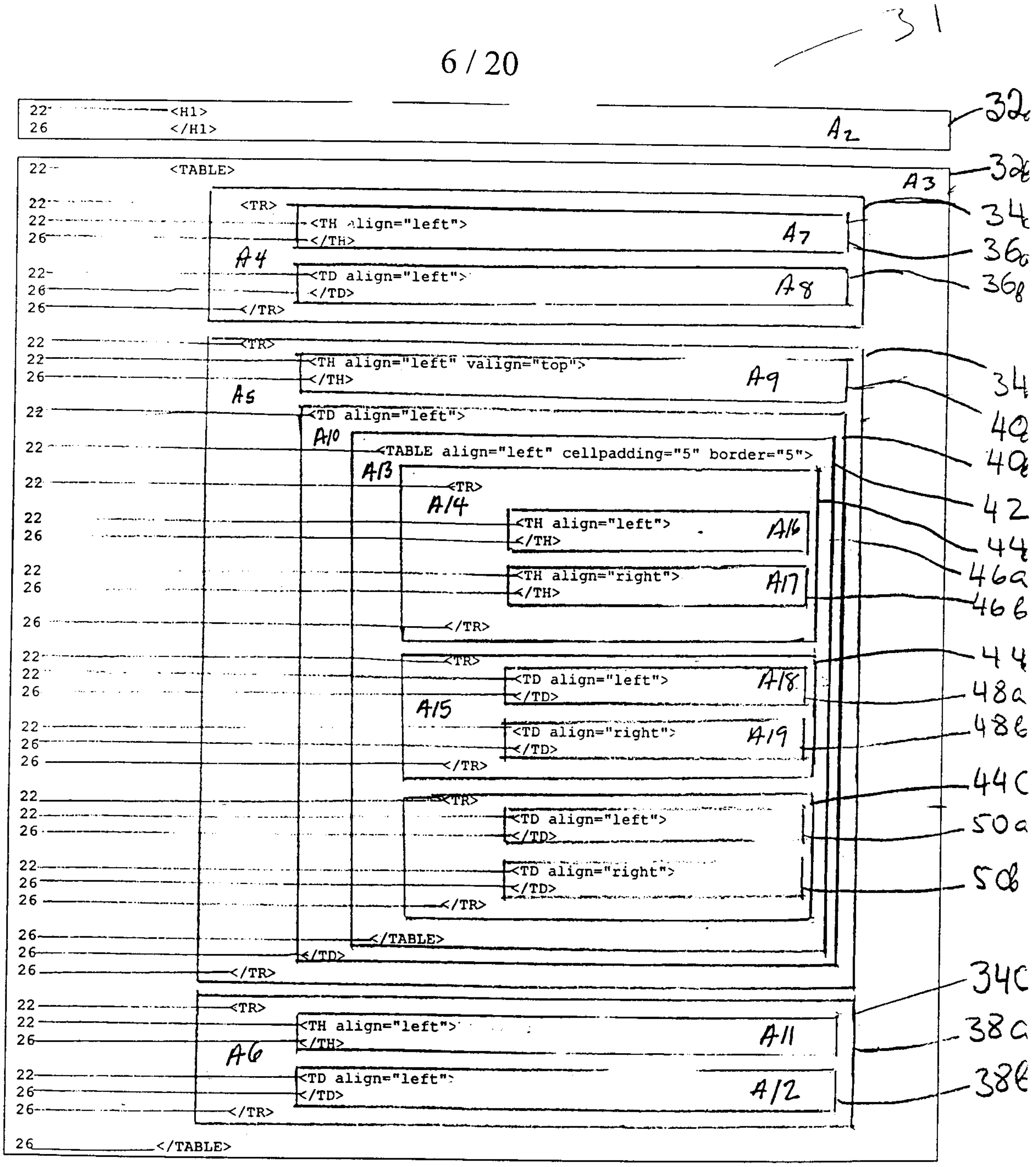


Figure 6A

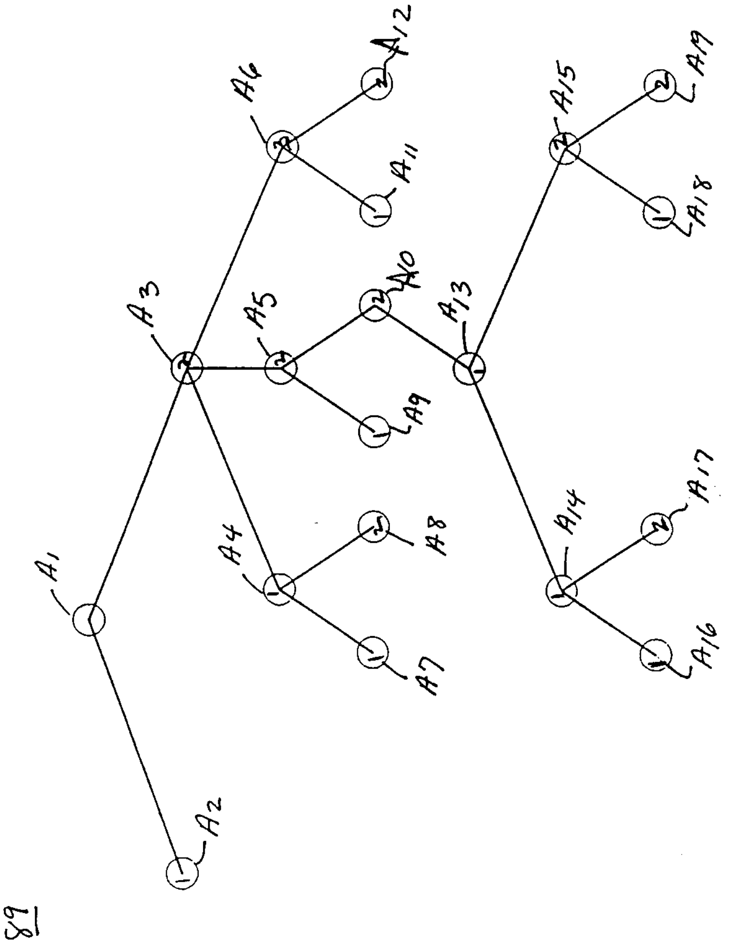
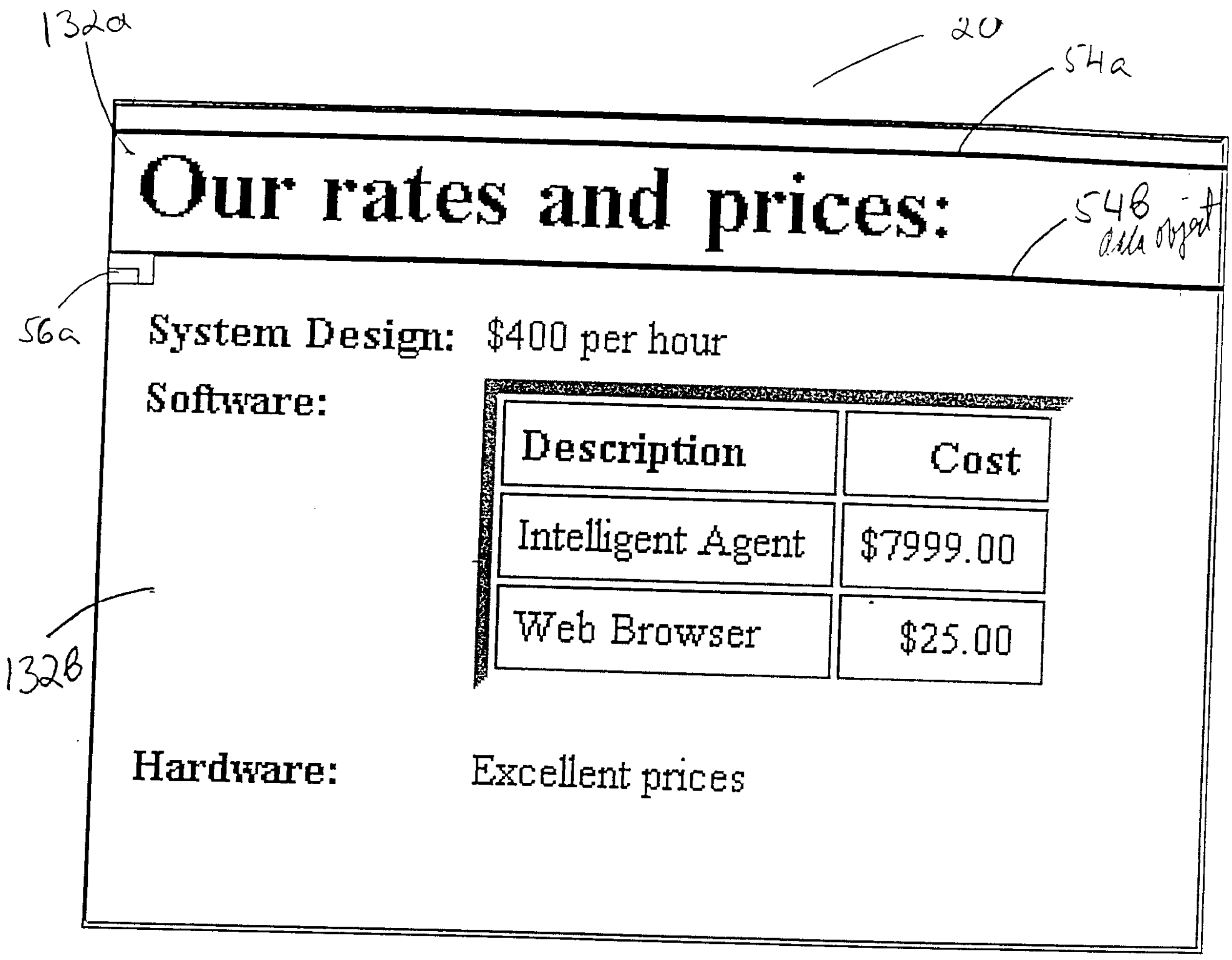

68
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
A17
A18
A19
1
2

FIG. 6B



Our rates and prices:

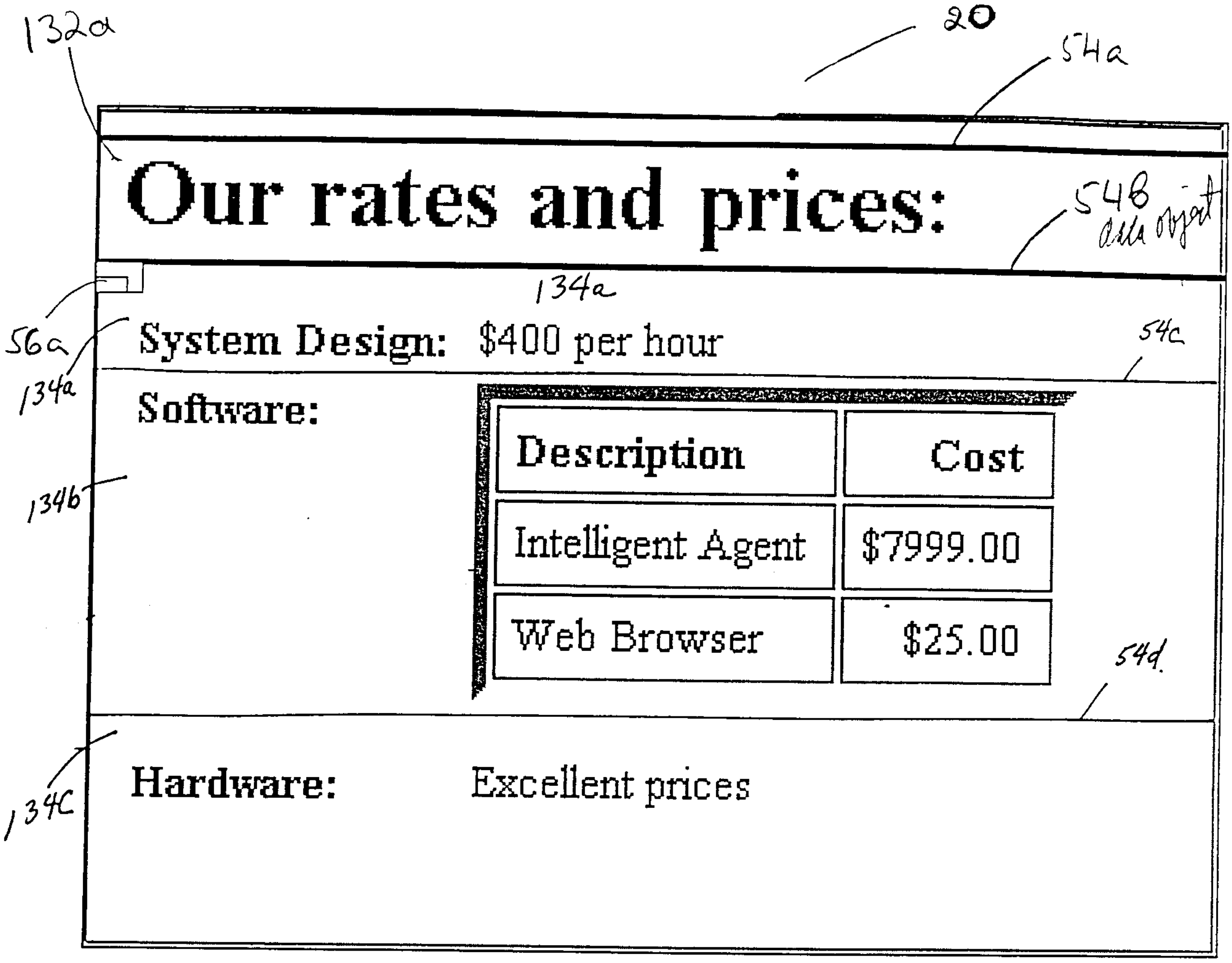
System Design: $400 per hour

Software:

Description	Cost
Intelligent Agent	$7999.00
Web Browser	$25.00

Hardware: Excellent prices

Figure 7a: Sample HTML



Our rates and prices:

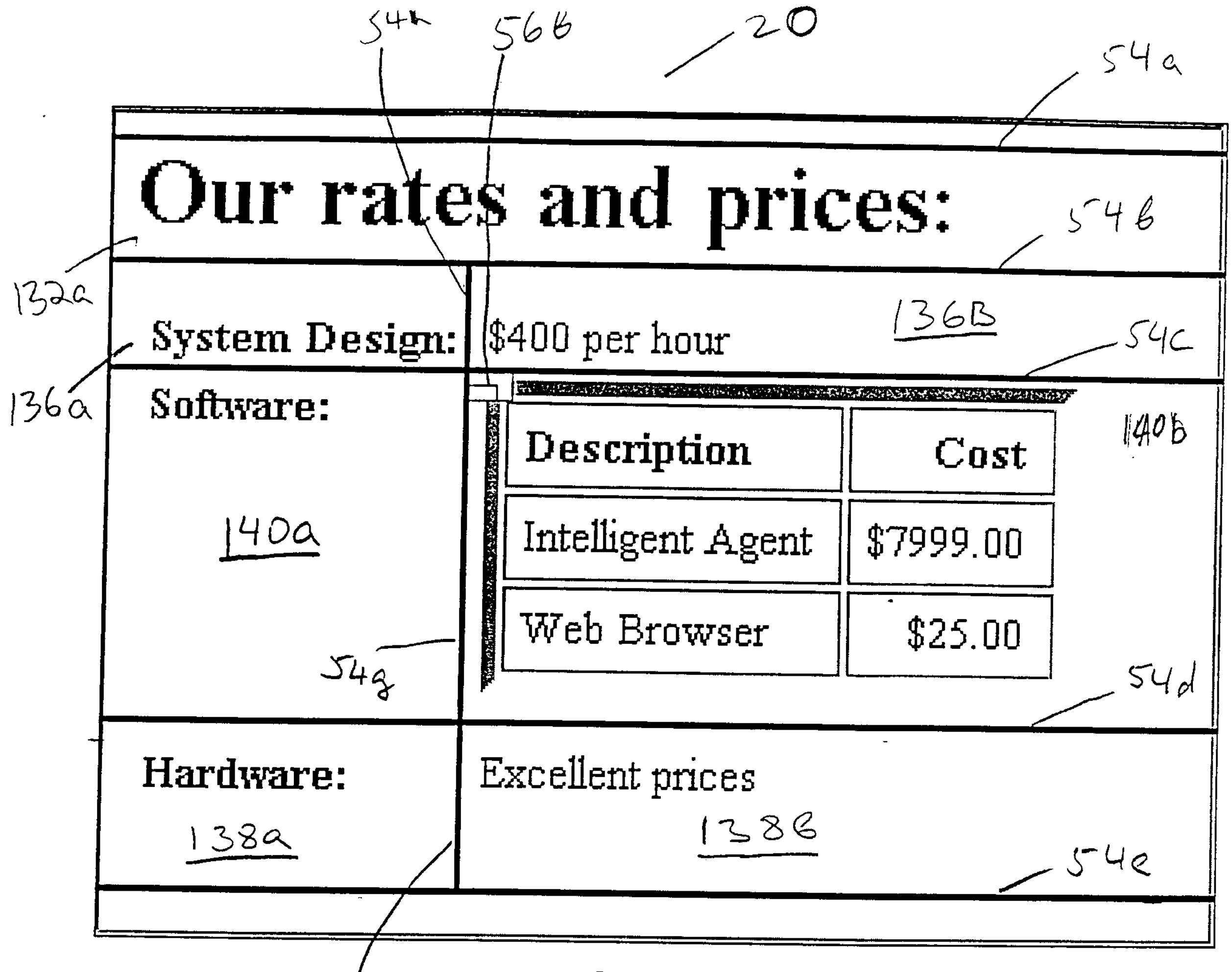
System Design: $400 per hour

Software:

Description	Cost
Intelligent Agent	$7999.00
Web Browser	$25.00

Hardware: Excellent prices

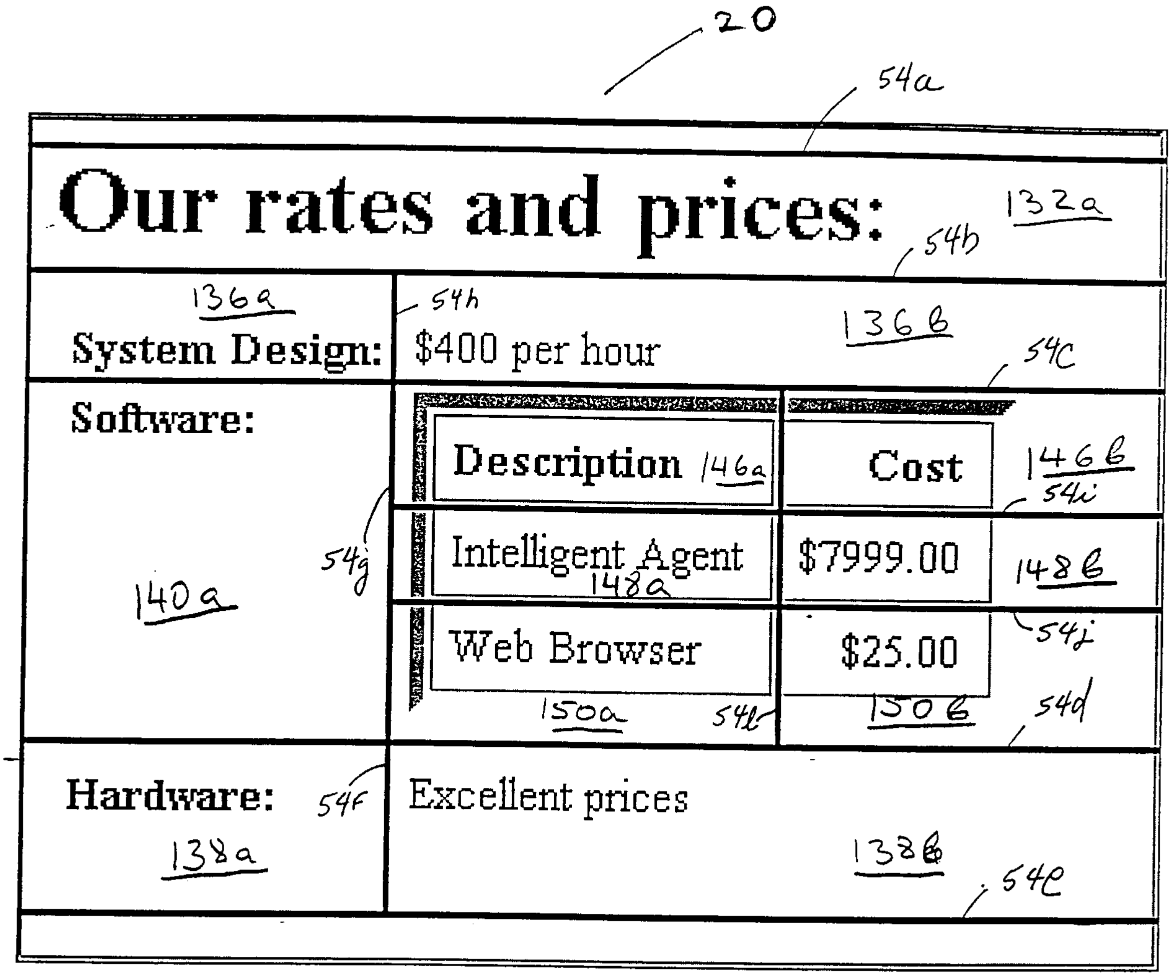
Figure 7: Sample HTML
b



Our rates and prices:

<table>
<tr><td>System Design:</td><td>$400 per hour</td></tr>
<tr><td>Software:</td><td><table><tr><td>Description</td><td>Cost</td></tr><tr><td>Intelligent Agent</td><td>$7999.00</td></tr><tr><td>Web Browser</td><td>$25.00</td></tr></table></td></tr>
<tr><td>Hardware:</td><td>Excellent prices</td></tr>
</table>

Figure 7 Sample HTML

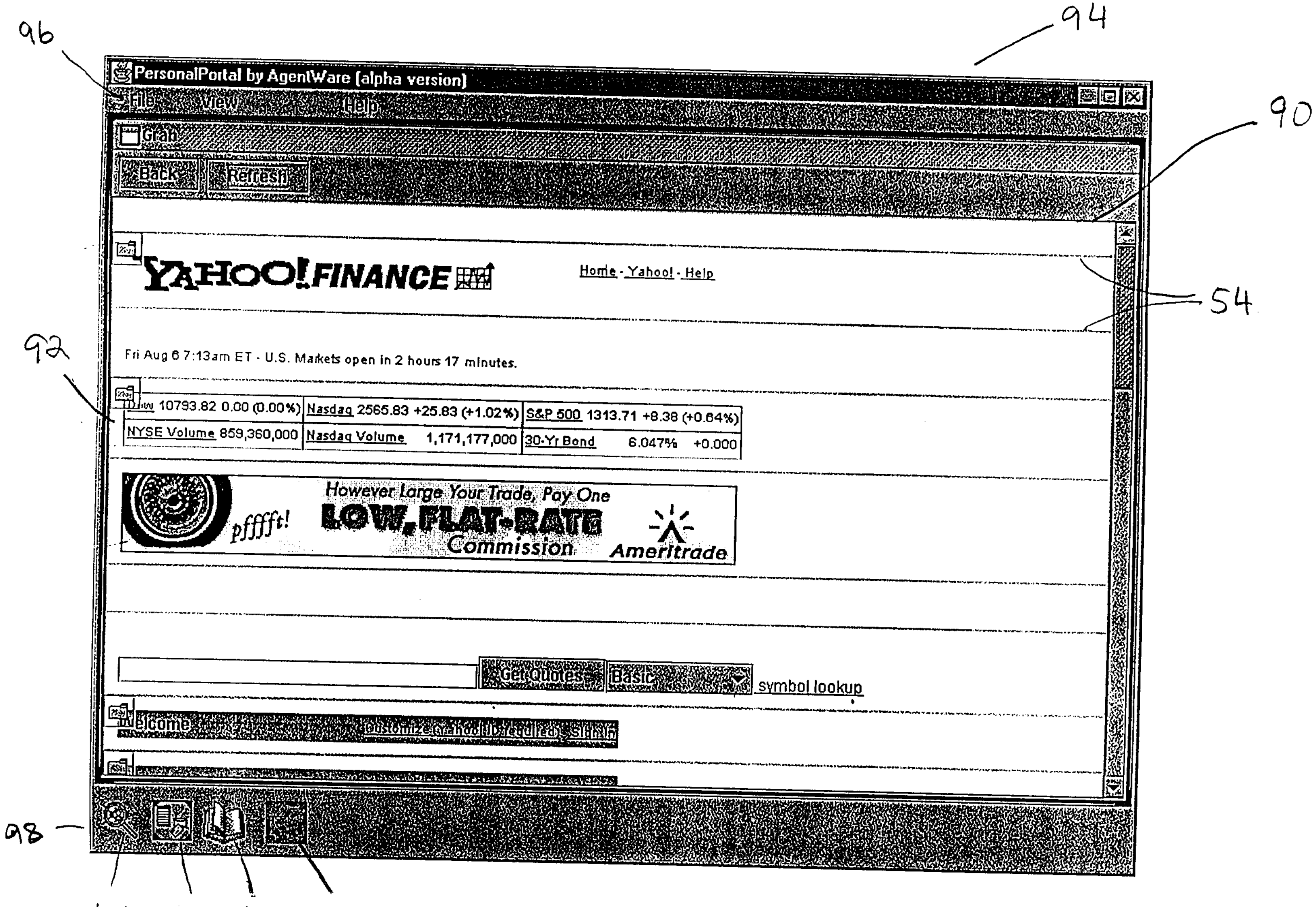


Our rates and prices:

System Design:	$400 per hour
Software:	Description / Cost Intelligent Agent / $7999.00 Web Browser / $25.00
Hardware:	Excellent prices

Description	Cost
Intelligent Agent	$7999.00
Web Browser	$25.00

Figure 7d Sample HTML

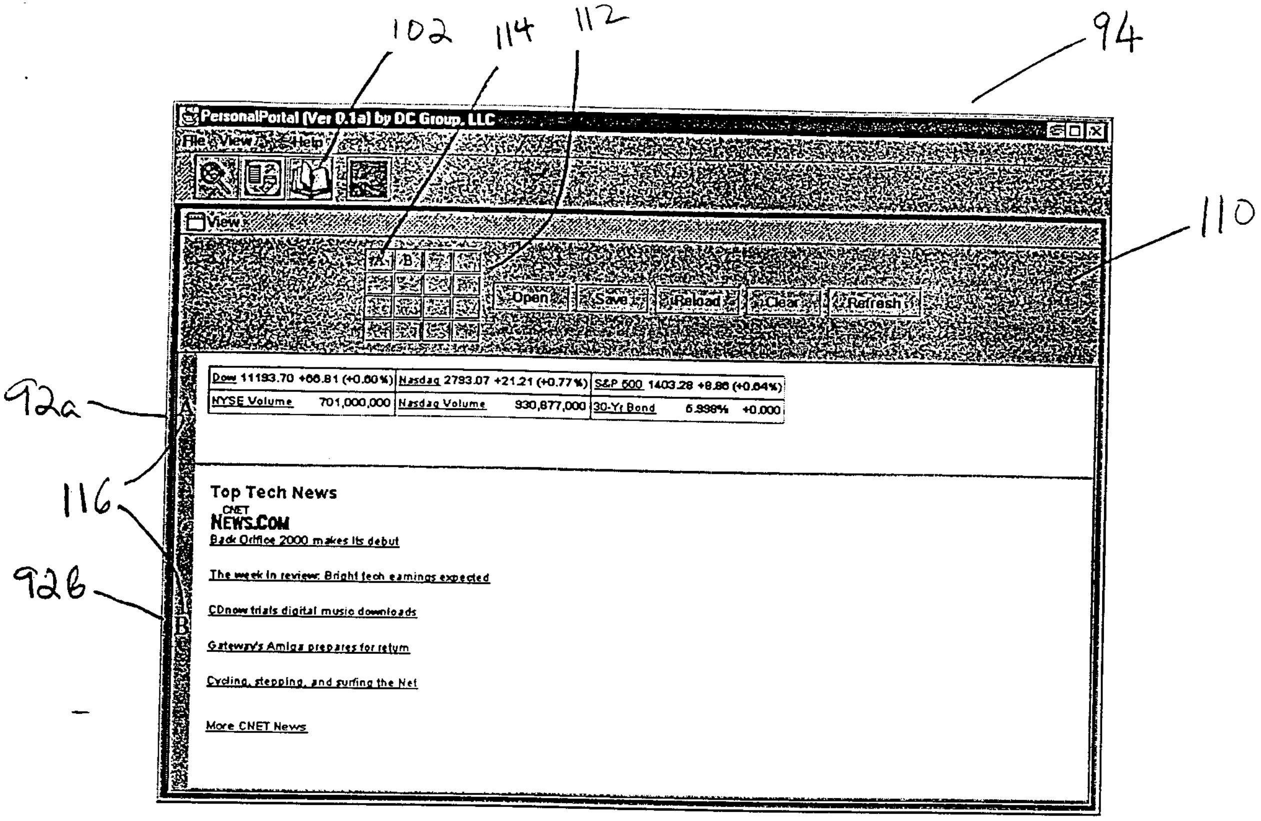


Fri Aug 6 7:13am ET - U.S. Markets open in 2 hours 17 minutes.

Dow 10793.82 0.00 (0.00%)	Nasdaq 2565.83 +25.83 (+1.02%)	S&P 500 1313.71 +8.38 (+0.64%)
NYSE Volume 859,360,000	Nasdaq Volume 1,171,177,000	30-Yr Bond 6.047% +0.000

Welcome Customize (Yahoo! ID required) - Sign in

FIGURE 8a



Dow 11193.70 +66.81 (+0.60%)	Nasdaq 2793.07 +21.21 (+0.77%)	S&P 500 1403.28 +8.86 (+0.64%)
NYSE Volume 701,000,000	Nasdaq Volume 930,877,000	30-Yr Bond 6.998% +0.000

Top Tech News

CNET NEWS.COM

Back Office 2000 makes its debut

The week in review: Bright tech earnings expected

CDnow trials digital music downloads

Gateway's Amiga prepares for return

Cycling, stepping, and surfing the Net

More CNET News

Above is a screenshot of the builder, where components are labeled "A,B,C..." and then placed in a 4X4 grid above.

Figure 8b

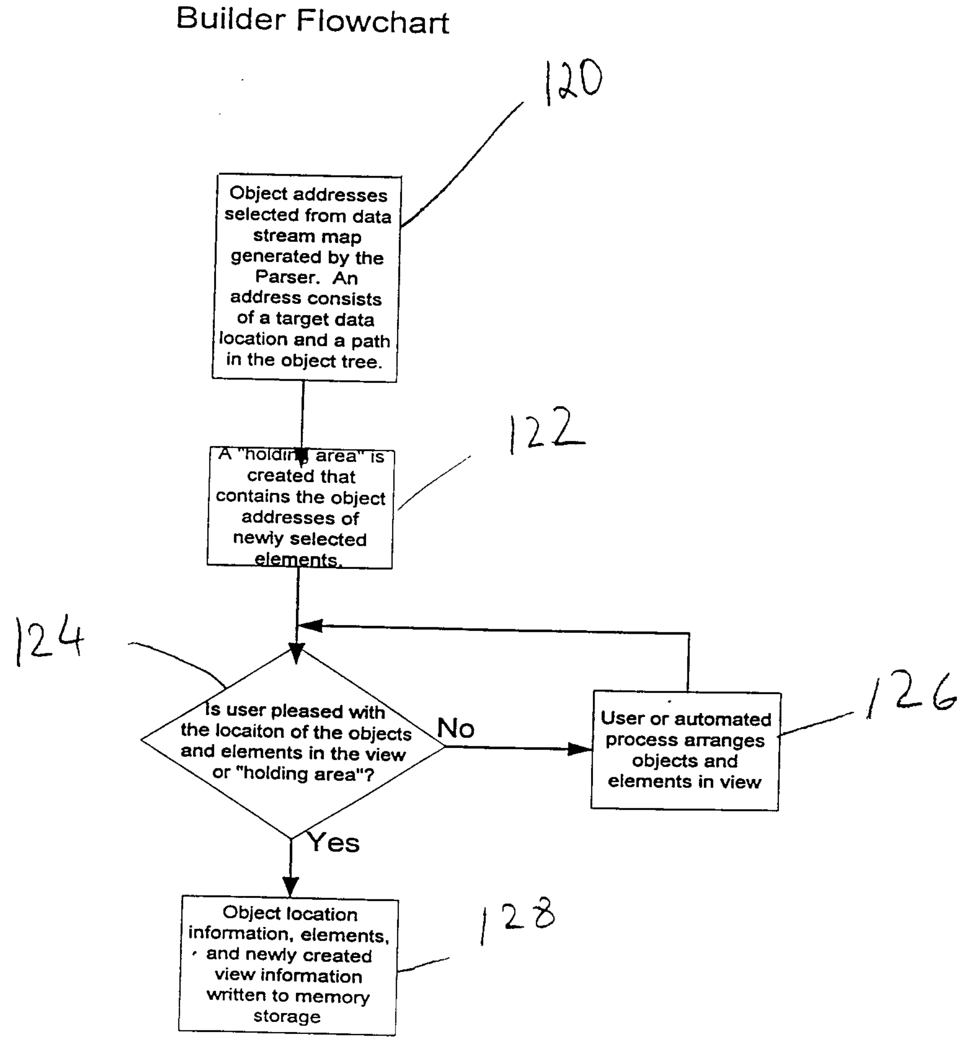
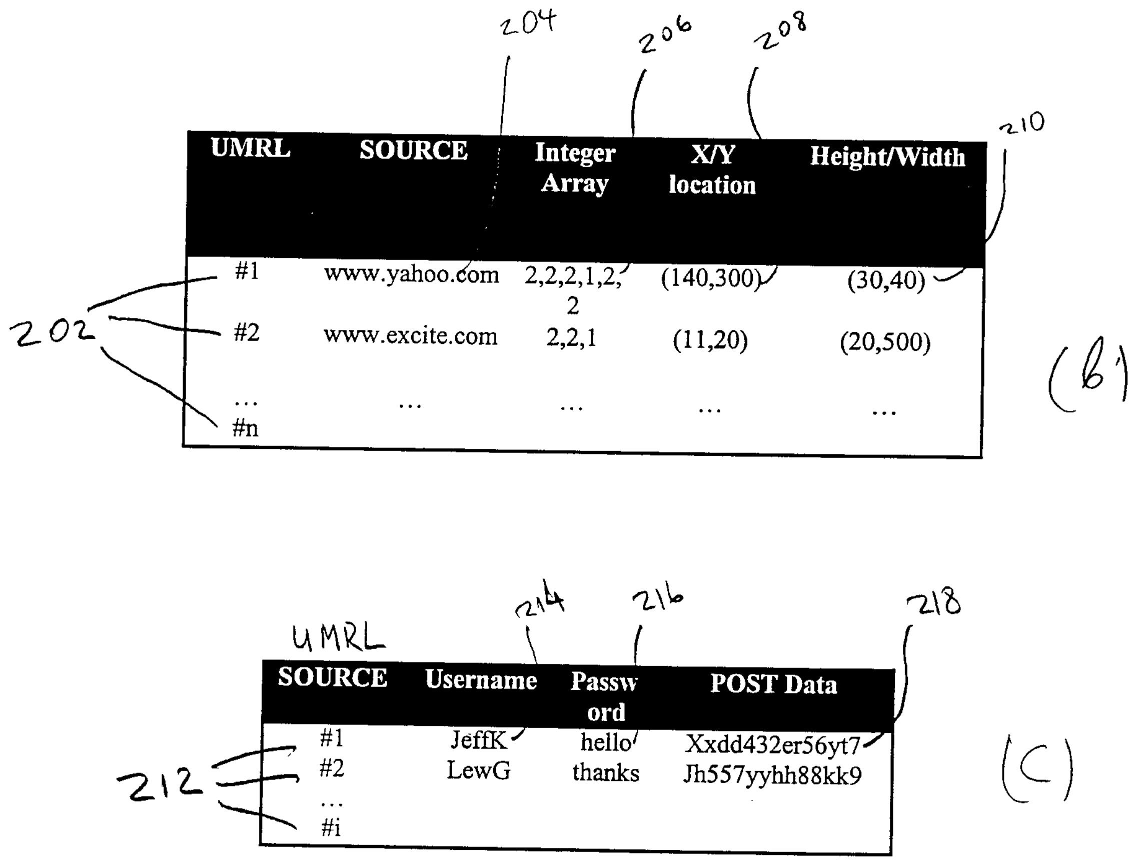

Builder Flowchart
120
Object addresses selected from data stream map generated by the Parser. An address consists of a target data location and a path in the object tree.
122
A "holding area" is created that contains the object addresses of newly selected elements.
124
Is user pleased with the locaiton of the objects and elements in the view or "holding area"?
No
User or automated process arranges objects and elements in view
126
Yes
Object location information, elements, and newly created view information written to memory storage
128

Figure 9a



204
206
208
210
202
(b)
UMRL
214
216
218
212
(c)

UMRL	SOURCE	Integer Array	X/Y location	Height/Width
#1	www.yahoo.com	2,2,2,1,2,2	(140,300)	(30,40)
#2	www.excite.com	2,2,1	(11,20)	(20,500)
…	…	…	…	…
#n				

SOURCE	Username	Password	POST Data
#1	JeffK	hello	Xxdd432er56yt7
#2	LewG	thanks	Jh557yyhh88kk9
…			
#i			

Figure 9

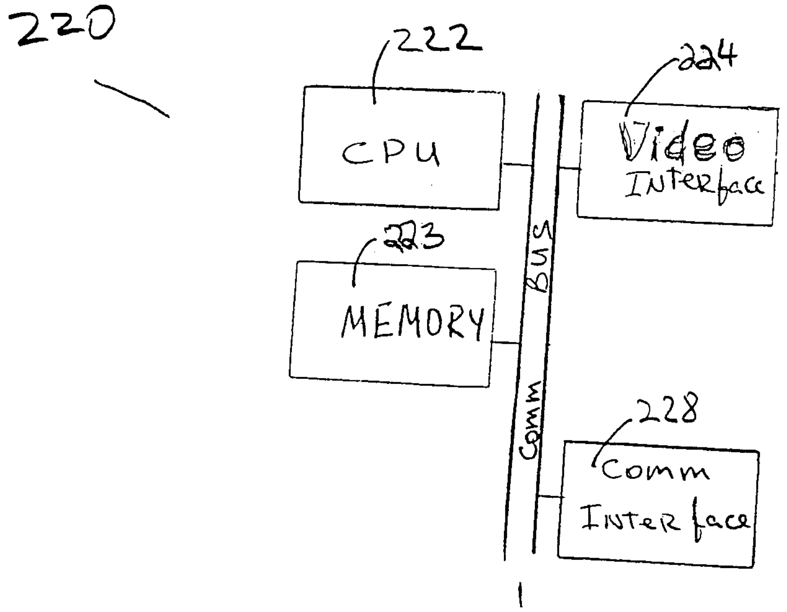

220
222
CPU
224
Video Interface
223
MEMORY
COMM BUS
228
Comm Interface

Figure 10a

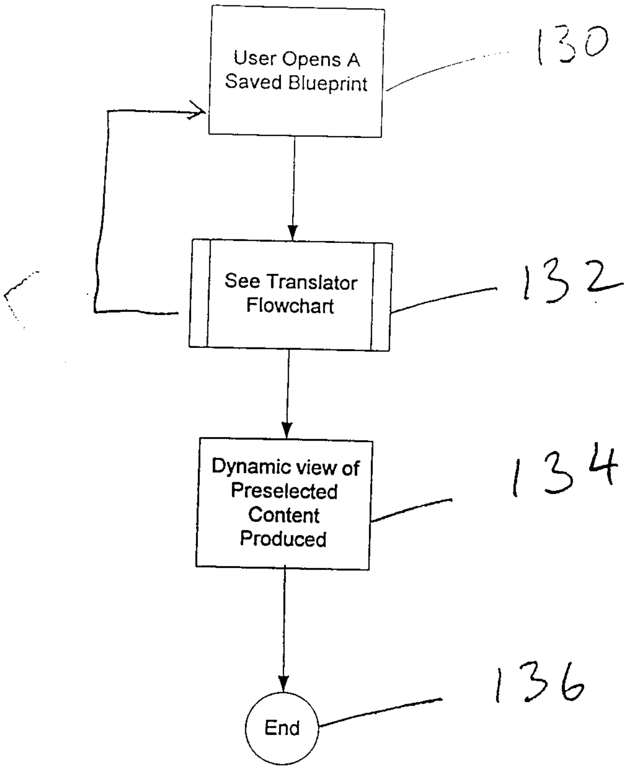

User Opens A Saved Blueprint
130
See Translator Flowchart
132
Dynamic view of Preselected Content Produced
134
End
136

Figure 10b

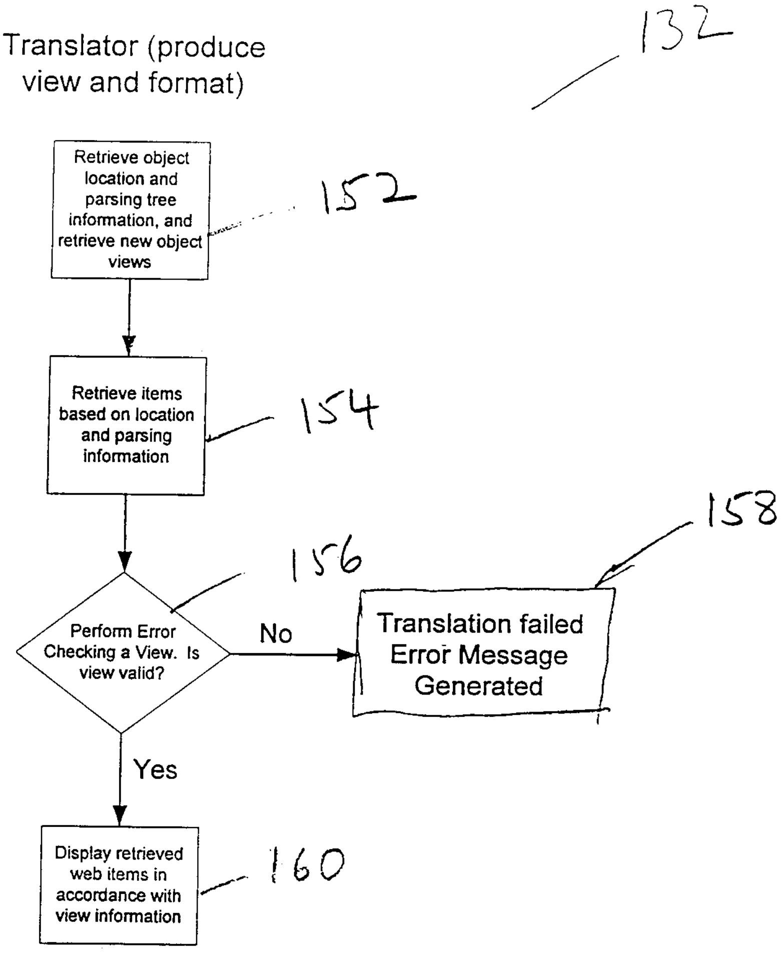

Translator (produce view and format)
132
Retrieve object location and parsing tree information, and retrieve new object views
152
Retrieve items based on location and parsing information
154
Perform Error Checking a View. Is view valid?
156
No
Translation failed Error Message Generated
158
Yes
Display retrieved web items in accordance with view information
160

Figure 11

Error Checking a View

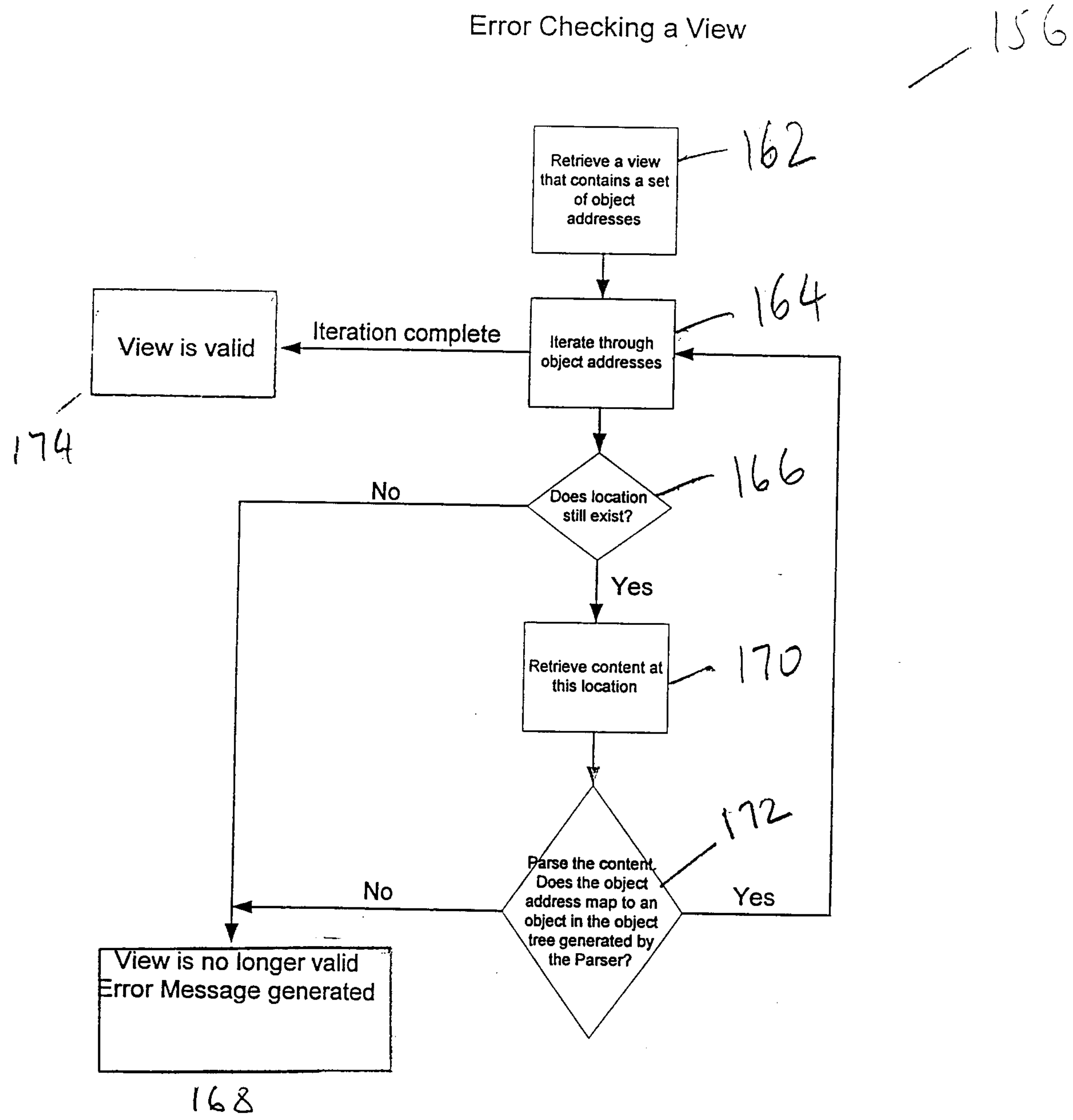
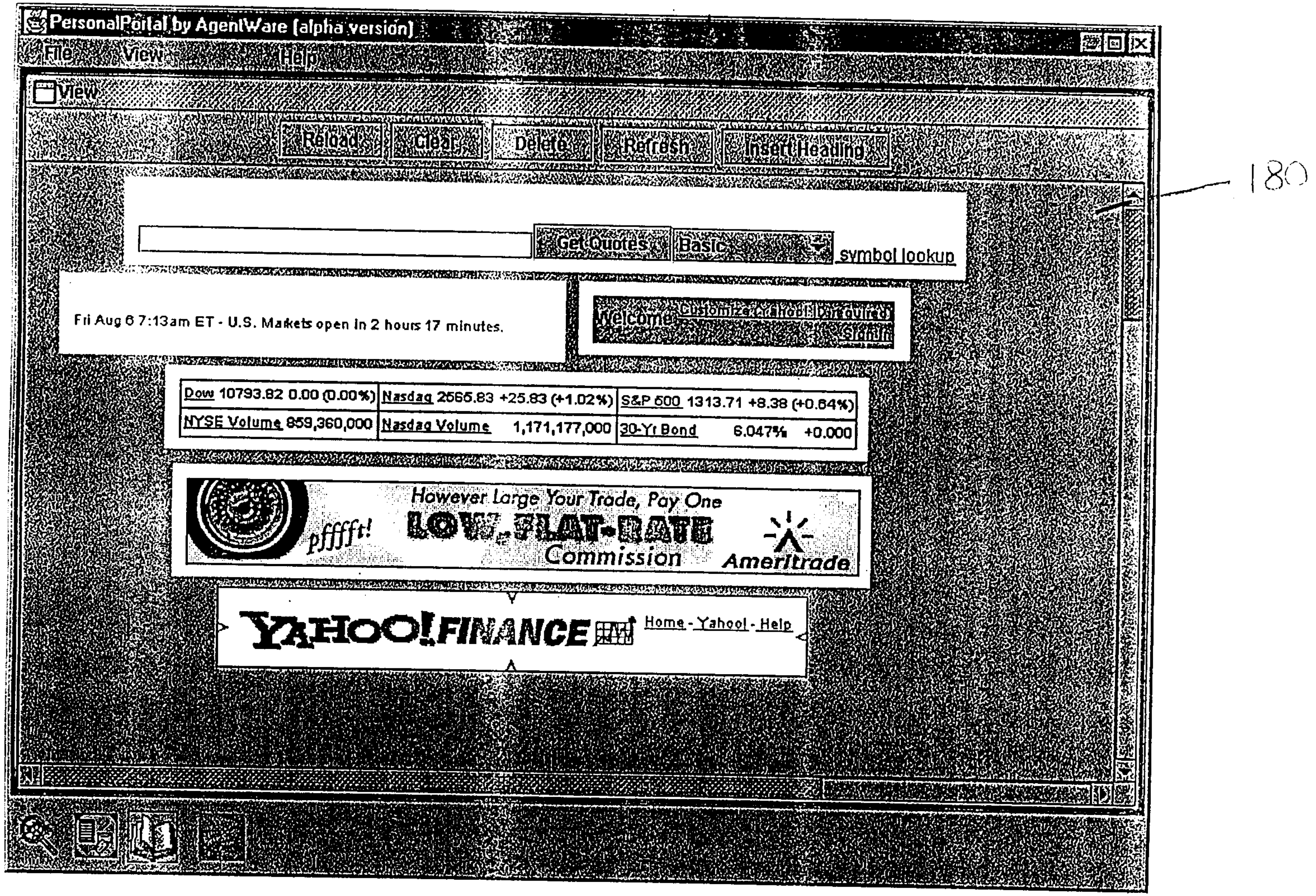

156
Retrieve a view that contains a set of object addresses
162
Iterate through object addresses
164
Iteration complete
View is valid
174
Does location still exist?
166
No
Yes
Retrieve content at this location
170
Parse the content. Does the object address map to an object in the object tree generated by the Parser?
172
No
Yes
View is no longer valid
Error Message generated
168

Figure 12



Fri Aug 6 7:13am ET - U.S. Markets open in 2 hours 17 minutes.

Dow 10793.82 0.00 (0.00%)	Nasdaq 2565.83 +25.83 (+1.02%)	S&P 500 1313.71 +8.38 (+0.64%)
NYSE Volume 859,360,000	Nasdaq Volume 1,171,177,000	30-Yr Bond 6.047% +0.000

Figure 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/14665

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 17/30
US CL :Please See Extra Sheet.
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 707/102, 103, 500, 501, 514, 516, 517, 526, 542; 709/206, 219; 345/334, 335, 348, 356.

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

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

Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,903,902 A (ORR et al) 11 May 1999, col. 2, lines 52-62, col. 7, lines 33-52, col. 10, lines 29-57, col. 16, lines 1-21, col. 22, lines 33-67, col. 23, lines 1-67, and col. 29, lines 7-19.	1-32
Y	US 5,877,966 A (MORRIS et al) 02 March 1999, col. 3, lines 15-33, col. 5, lines 5-12, col. 6, lines 39-67, col. 7, lines 1-67, col. 8, lines 1-67, col. 9, lines 1-67, col. 10, lines 1-42 and lines 57-67, col. 12, lines 50-67, col. 13, lines 1-33, col. 15, lines 3-8 and lines 30-54.	1-32

☒ Further documents are listed in the continuation of Box C. □ See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

02 AUGUST 2000

Date of mailing of the international search report

23 AUG 2000

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer
KIM VU James R. Matthews
Telephone No. (703) 305-4393

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/14665

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,907,704 A (GUDMUNDSON et al) 25 May 1999, col. 8, lines 7-67, col. 9, lines 1-13, col. 15, lines 9-18, col. 19, lines 3-5, col. 37, lines 40-54, col. 54, lines 43-53, and col. 64, lines 18-33.	1-32

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/14665

A. CLASSIFICATION OF SUBJECT MATTER:
US CL :

707/102, 103, 500, 501, 514, 516, 517, 526, 542; 709/206, 219; 345/334, 335, 348, 356.

B. FIELDS SEARCHED
Electronic data bases consulted (Name of data base and where practicable terms used):

WEST
Search terms; template creation, documents, parsing, intelligent agent software, multi-tasking or multi-user, HTML, XML, data segments, nodes, pages, multimedia, object tree.