PTO/SB/05 (08-08)
Approved for use through 06/30/2010. OMB 0651-0032
U.S. Patent and Trademark Office. U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

# UTILITY PATENT APPLICATION TRANSMITTAL

*(Only for new nonprovisional applications under 37 CFR 1.53(b))*

| | |
|---|---|
| *Attorney Docket No.* | EQUI0016D2 |
| *First Inventor* | Sean BARGER |
| *Title* | Automated Media Delivery System |
| *Express Mail Label No.* | Submitted Electronically Via EFS-We |

## APPLICATION ELEMENTS

*See MPEP chapter 600 concerning utility patent application contents.*

*ADDRESS TO:*
**Commissioner for Patents**
**P.O. Box 1450**
**Alexandria VA 22313-1450**

1. □ **Fee Transmittal Form** (e.g., PTO/SB/17)
2. ☑ **Applicant claims small entity status.** See 37 CFR 1.27.
3. ☑ **Specification** [*Total Pages* 40 ]
   Both the claims and abstract must start on a new page
   *(For information on the preferred arrangement, see MPEP 608.01(a))*
4. ☑ **Drawing(s)** (*35 U.S.C. 113*) [*Total Sheets* 23 ]
5. **Oath or Declaration** [*Total Sheets* 3 ]
   a. ☑ Newly executed (original or copy)
   b. □ A copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional with Box 18 completed)*
      i. □ DELETION OF INVENTOR(S)
      Signed statement attached deleting inventor(s) name in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
6. □ **Application Data Sheet.** See 37 CFR 1.76
7. □ **CD-ROM or CD-R** in duplicate, large table or Computer Program *(Appendix)*
   □ Landscape Table on CD
8. **Nucleotide and/or Amino Acid Sequence Submission** *(if applicable, items a. – c. are required)*
   a. □ Computer Readable Form (CRF)
   b. Specification Sequence Listing on:
      i. □ CD-ROM or CD-R (2 copies); or
      ii. □ Paper
   c. □ Statements verifying identity of above copies

## ACCOMPANYING APPLICATION PARTS

9. □ **Assignment Papers** (cover sheet & document(s))
   Name of Assignee____________________________
10. □ **37 CFR 3.73(b) Statement** *(when there is an assignee)* ☑ **Power of Attorney**
11. □ **English Translation Document** *(if applicable)*
12. ☑ **Information Disclosure Statement** (PTO/SB/08 or PTO-1449)
    □ Copies of citations attached
13. □ **Preliminary Amendment**
14. □ **Return Receipt Postcard** (MPEP 503) (*Should be specifically itemized)*
15. □ **Certified Copy of Priority Document(s)** *(if foreign priority is claimed)*
16. □ **Nonpublication Request** under 35 U.S.C. 122(b)(2)(B)(i). Applicant must attach form PTO/SB/35 or equivalent.
17. □ Other:____________________________

18. If a CONTINUING APPLICATION, *check appropriate box, and supply the requisite information below and in the first sentence of the specification following the title, or in an Application Data Sheet under 37 CFR 1.76:*

□ Continuation ☑ Divisional □ Continuation-in-part (CIP) of prior application No.: 12/173,747...............

*Prior application information:* *Examiner* Charles C. Murphy *Art Unit:* 2441

## 19. CORRESPONDENCE ADDRESS

☑ The address associated with Customer Number: 22862 **OR** □ Correspondence address below

| | | | | | |
|---|---|---|---|---|---|
| Name | | | | | |
| Address | | | | | |
| City | | State | | Zip Code | |
| Country | | Telephone | | Email | |

| | | | |
|---|---|---|---|
| Signature | | Date | February 26, 2010 |
| Name (Print/Type) | Michael A. Glenn | Registration No. (Attorney/Agent) | 30,176 |

This collection of information is required by 37 CFR 1.53(b). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

*If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.*

# Automated Media Delivery System

## BACKGROUND OF THE INVENTION

### CROSS REFERENCE TO RELATED APPLICATIONS

This application is a Divisional of U.S. Serial No. 12/173,747, filed July 15, 2008, which is a Divisional of U.S. Serial No. 11/269,916, filed November 7, 2005, which is a Continuation-in-Part of U.S. Serial No. 09/929,904, filed August 14, 2001, now U.S. Patent No. 6,964,009 granted on November 8, 2005, which is a Continuation of U.S. Serial No. 09/425,326, filed October 21, 1999, now U.S. Patent No. 6,792,575, granted on September 14, 2004, each of which is hereby incorporated in its entirety by this reference thereto.

### TECHNICAL FIELD

The invention relates to software systems. More particularly, the invention relates to an Internet server-based software system that provides delivery of automated graphics and other media to Web sites for access by an end user or consumer.

### DESCRIPTION OF THE PRIOR ART

Most Web sites today are primarily handmade. From the guy publishing a simple online technology newsletter from his home, to the Fortune 1000 company's multi-tiered site with hundreds of pages of text, images, and animations, the Web developer and each of his HTML-coding and graphics-producing coworkers toil page by page and image by image. Thousands of established online companies employ hundreds of highly-skilled workers just to produce and maintain their Web sites. After all, the Web is now a major selling vehicle and marketing medium for many of these companies. The Web has even sprouted service industries such as, for example, public companies with multi-billion dollar valuations created just to consult and produce Web sites for others.

Most Web developers who use established WYSIWYG tools in the industry still must produce each page on their Web site one by one. The same rate applies to preparing and placing images, animations, and other visual assets. Each page represents its own set of issues ranging from whether to use GIF, JPEG, or PNG file formats, to finding the optimum bit depth for each image to ensure the fastest downloading through the different browsers of the consumer. The bottlenecked state of the customer's workflow to produce graphics for Web pages can be described as follows:

**Current Workflow for Creating Web Graphics**

- Original Artwork/Asset Creation
  - Use third-party point products
- Asset Editing
  - Scale/reduce/slice
- Asset Format Conversion
  - JPEG/GIF/PNG
- Asset Staging
  - Place in Web file system
  - Edit HTML
- Create/Modify HTML for particular page
- Store HTML on Web server
- View final pages
- Repeat process for each version of each graphic on each page

Estimated time

- Two hours per page times the number of pages

Also, from a user's perspective, the current state of the art is to offer the consumer zooming and panning capabilities so that by clicking on an image the consumer can view more closely or from a different angle. On the horizon are pages with three-dimensional imagery that enable a user to move around a page that can look more like a room than a brochure. While interesting, these features are merely incremental improvements to a consumer's surfing experience.

D. C. A. Bulterman, *Models, Media, and Motion: Using the Web to Support Multimedia Documents,* Proceedings of 1997 International Conference on Multimedia Modeling, Singapore, 17-20 Nov. 1997 discloses "an effort underway by members of industry, research centers and user groups to define a standard document format that can be used in conjunction with time-based transport protocols over the Internet and intranets to support rich multimedia presentations. The paper outlines the goals of the W3C's Synchronized Multimedia working group and presents an initial description of the first version of the proposed multimedia document model and format."

*Text and Graphics on UMI's ProQuest Direct: The Best (yet) of both Worlds,* Online, vol. 21, no. 2, pp. 73-7, March- April 1997 discloses an information system that offers "periodical and newspaper content covering a wide range of business, news, and professional topics... letting the user search both text and graphics and build the product to suit. Articles can be retrieved in varying levels of detail: citation, abstracts, full text, and text with graphics. Images come in two flavors: Page Image, a virtual photocopy, and Text+Graphics, in which graphics are stored separately from the text and are manipulable as discrete items. ...[The system] comes in two versions: Windows and Web."

John Mills Dudley, *Network-Based Classified Information Systems*, AU-A-53031/98 (27/08/98) discloses a "system for automatically creating databases containing industry, service, product and subject classification data, contact data, geographic location data (CCG-data) and links to web pages from HTML, XML, or SGML encoded web pages posted on computer networks such as Internets or Intranets....The... databases may be searched for references (URLs) to web pages by use of enquiries which reference one or more of the items of the CCG-data. Alternatively, enquiries referencing the CCG-data in the databases may supply contact data without web page references. Data duplication and coordination is reduced by including in the web page CCG-data display controls which are used by web browsers to format for display the same data that is used to automatically update the databases."

Cordell *et al*, *Automatic Data Display Formatting with A Networking Application*, U.S. Patent No. 5,845,084 (Dec. 1, 1998) discloses a placeholder image mechanism. "When a data request is made, the data transfer rate is monitored. When the receive data transfer rate is slow, and the data contains an embedded graphical image of unknown dimensions, a small placeholder image is automatically displayed for the user instead of the actual data. The small placeholder image holds a place on a display device for the data or the embedded graphical image until the data or embedded graphical image is received. When embedded graphical image is received, the placeholder image is removed, and the display device is reformatted to display the embedded graphical image."

Jonathon R. T. Lewis, *System For Substituting Tags For Non-Editable Data Sets In Hypertext Documents And Updating Web Files Containing Links Between Data Sets Corresponding To Changes Made To The Tags*, U.S. Patent No. 5,355,472 (Oct. 11, 1994) discloses a "hypertext data processing system wherein data sets participating in the hypertext document may be edited, the data processing system inserting tags into the data sets at locations corresponding to the hypertext links to create a file which is editable by an editor and the data processing system removing the tags, generating a revised data set and updating the link information after the editing process. Its main purpose is to preserve the linking hierarchy that may get lost when the individual data sets get modified."

Wistendahl *et al*, *System for Mapping Hot Spots in Media Content Interactive Digital Media Program*, U.S. Patent No. 5,708,845 (Jan. 13, 1998) discloses a "system for allowing media content to be used in an interactive digital media (IDM) program [that] has Frame Data for the media content and object mapping data (N Data) representing the frame addresses and display location coordinates for objects appearing in the media content. The N Data are maintained separately from the Frame Data for the media content, so that the media content can be kept intact without embedded codes and can be played back on any system. The IDM program has established linkages connecting the objects mapped by the N Data to other functions to be performed in conjunction with display of the media content. Selection of an object appearing in the media content with a pointer results in initiation of the interactive function. A broad base of existing non-interactive media content, such as movies, videos, advertising, and television programming can be converted to interactive digital media use. An authoring system for creating IDM programs has an object outlining tool and an object motion tracking tool for facilitating the generation of N Data. In a data storage disk, the Frame Data and the N Data are stored on separate sectors. In a network system, the object mapping data and IDM program are downloaded to a subscriber terminal and used in conjunction with presentation of the media content."

Rogers *et al*, *Method for Fulfilling Requests of A Web Browser*, U.S. Patent No. 5,701,451 (Dec. 23, 1997) and Lagarde *et al*, *Method for Distributed Task Fulfillment of Web Browser Requests*, U.S. Patent No. 5,710,918 (Jan. 20, 1998) disclose essentially "improvements which achieve a means for accepting Web client requests for information, obtaining data from one or more databases which may be located on multiple platforms at different physical locations on an Internet or on the Internet, processing that data into meaningful information, and presenting that information to the Web client in a text or graphics display at a location specified by the request."

Tyan *et al*, *HTML Generator*, European Patent Application No. EP 0843276 (May 20, 1998) discloses "generating an HTML file based on an input bitmap image, and is particularly directed to automatic generation of an HTML file, based on a scanned-in document image, with the HTML file in turn being used to generate a Web page that accurately reproduces the layout of the original input bitmap image."

TrueSpectra has a patent pending for the technology employed in its two products, IrisAccelerate and IrisTransactive. These products are designed for zooming and panning and simple image transformations and conversions, respectively. They support 10 file formats and allow developers to add new file formats via their SDK. They do not require the use of Flashpix for images. However, their documentation points out that performance is dependent on the Flashpix format. The system would be very slow if a non-Flashpix format was used.

TrueSpectra allows the image quality and compression to be set for JPEGs only. The compression setting is set on the server and all images are delivered at the same setting.

TrueSpectra has a simple caching mechanism. Images in the cache can be cleared out automatically at certain times and it does not have any dependency features for image propagation. The Web server needs to be brought down in order to update any original assets.

TrueSpectra does not require plug-ins to operate features such as zooming/panning or compositing. The alternative to plug-ins is using their Javascript or active server page technology. These technologies are used by many Web sites to provide interactivity, but not all Web browsers work correctly with these technologies.

TrueSpectra relies on Flashpix as its native file format and does not support media types such as multi-GIFs and sound formats. Flashpix files are typically larger than most file formats. Access to files is faster for zooming and panning, but appears to be quite slow.

The key to IrisTransactive is the compositing subsystem. It requires three things to build a shopping solution using image composition.

1) The original images must be created. It is suggested that the image be converted to Flashpix for better performance.

2) All of the individual images must be described in XML using the image composer program. The program allows the editor to specify anchor points, layer attributes, and layer names. The resulting file is between 5k and 50k.

3) The Web designer must place HTML referring to the XML in the Web site. By specifying parameters to the XML, the Web designer can turn on or off layers.

The herein above process for compositing images enables Web designers to create shopping sites. However, a lot of overhead is the result. The XML documents add 5k-50k to a Web site. The compositing commands that are embedded in the HTML are difficult to understand. And, because the compositing feature requires several steps to implement, it is not suitable for every image on a Web site. The process seems to be designed for the specific purpose of shopping.

MediaBin(TM) is limited to activities behind the firewall automating only the "post-creative busywork." In addition, MediaBin requires the use of an application server to function through a web interface. Thus images may not be directly added to any existing web page.

Macromedia's Generator operates by embedding variables in their proprietary Flash format. Therefore the actual imaging operations are somewhat limited and cannot be controlled directly from a web page request.

MGI Software sells point solutions that require end-users to download a viewer to process a proprietary image format.

PictureIQ offers a server-side image-processing appliance that provides a limited set of Photoshop functionalities. This appliance runs on the web-page server, processes information embedded in the web page, and rewrites the web page with image data.

The disclosed prior art fail to provide systems and methodologies that result in a quantum leap in the *speed* with which they can modify and add images, video, and sound to sites, in the *volume* of data they can publish internally and externally, and in the *quality* of the output. The development of such an automated media delivery system would constitute a major technological advance.

It would be advantageous to empower an end user with flexibility and control by providing *interactive* page capabilities.

It would be advantageous from an end user's perspective to generate Web pages that contain *active* graphics. For example, clicking on a Corvette image will cause a simple menu to pop up suggesting alternative colors and sizes in which to see the car. Clicking on portions of the image, such as a fender, can call up a close-in view of the fender.

It would be advantageous to provide an automated graphics delivery system that becomes part of the *Web site infrastructure* and operates as part of the *Web page transaction* and that thereby provides a less expensive and less time-consuming process.

It would be advantageous to provide a system for automated processing and delivery of media (images, video, and sound) to a Web server whereby it eliminates the laborious post-production and conversion work that must be done before a media asset can be delivered on a Web server.

It would be advantageous to create a dynamic Web site, wherein images are generated on demand from original assets, wherein only the original assets need to be updated, and wherein updated changes propagate throughout the site.

It would be advantageous to provide a system that generates media based on current Web server traffic thereby optimizing throughput of the media through the Web server.

It would be advantageous to provide a system that generates media that is optimized for the Web client, wherein client connection speed determines optimum quality and file size.

It would be advantageous to provide a system that generates media, whereby the media is automatically uploaded.

It would be advantageous to provide a system that automatically caches generated media so identical requests can be handled without regeneration of images.

It would be advantageous to provide a system that resides behind the Web server, thereby eliminating security issues.

It would be advantageous to provide a system wherein the client browser does not require a plug-in.

It would be advantageous to provide a system wherein the system does not require any changes to a Web server.

It would be advantageous to provide a system wherein the system manages the Web server media cache.

It would be advantageous to provide a system wherein the Web media is generated only if requested by a client browser.

It would be advantageous for a system to reduce the need for a Web author to create different versions of a Web site, the system automatically handling image content.

It would be advantageous to provide dynamic imaging capabilities, have a more complete set of image processing functionality, and be controlled directly through an image URL.

It would be advantageous to provide an end-to-end solution requiring only a standard browser that is completely controllable using the proprietary tags contained within a simple image link in the web page.

It would be advantageous to run an image application as a separate server controlled directly by single image requests to that server, such that any web server, even one that is only sending static HTML can access imaging features.

## SUMMARY OF THE INVENTION

An automatic graphics delivery system that operates in parallel with an existing Web site infrastructure is provided. The system streamlines the post-production process by automating the production of media through content generation procedures controlled by proprietary tags placed within URLs embedded within Web documents. The author simply places the original media in the system, and adds proprietary tags to the URLs for accessing that media. The system automatically processes the URL encoded tags and automatically produces derivative media for the web site from the original media.

The system takes as input the client connection, server traffic, content generation procedures, and proprietary tags placed within the URL to generate optimized media for the client. The need for the Web author to create different versions of a Web site is reduced because the image content of the site is automatically handled by the system. In addition, generated media is cached such that further requests for the same media require little overhead.

Because the invention takes the original media, content generation procedures, and proprietary URL tags as inputs for generating the Web media, it is possible to modify any of these inputs and have the system automatically update the media on the associated Web pages.

## BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic diagram showing the placement of the system within a current Web infrastructure according to the invention;

Fig. 2 is a schematic diagram showing how a typical Web site delivers an HTML document and its graphics to a Web browser according to the prior art;

Fig. 3 is a schematic diagram showing delivery of an HTML document and media to a Web browser according to the invention;

Fig. 4 is a schematic diagram showing the components involved in Web site administration according to the prior art;

Fig. 5 is a schematic diagram showing the components of the system involved in Web site administration according to the invention;

Fig. 6 is a simple overview showing the components of the system according to the invention;

Fig. 7 is a schematic diagram showing the process flow of a proprietary enabled page delivered to a Web browser according to the invention;

Fig. 8 is a flow chart showing an authoring process according to the invention;

Fig. 9 is a flow chart showing an HTML parsing process according to the invention;

Fig. 10 is a flow chart showing a media creation process according to the invention;

Fig. 11 is a screen shot showing an administration tool according to the invention;

Fig. 12 displays a structure of a database record used for the system according to the invention;

Fig. 13 shows original media to be processed according to the invention;

Fig. 14 shows a portion on an HTML document with a proprietary tag according to the invention;

Fig. 15 shows an HTML document and an HTML document source according to the invention;

Fig. 16 shows a generated GIF image according to the invention;

Fig. 17 is a schematic diagram of an image system within a typical Web infrastructure according to the invention;

Fig. 18 is a schematic diagram showing delivery of an HTML document and original media according to the invention;

Fig. 19 is a schematic diagram showing components of Web site administration according to a preferred embodiment of the invention;

Fig. 20 is a simple overview showing components of the image system according to a preferred embodiment of the invention;

Fig. 21 is a schematic diagram showing process flow of a proprietary enabled page delivered to a Web browser according to a preferred embodiment of the invention;

Fig. 22 shows a flowchart of a content generation procedure according to a preferred embodiment of the invention; and

Fig. 23 is a flow chart showing an authoring process according to a preferred embodiment of the invention.

## DETAILED DESCRIPTION OF THE INVENTION

An automatic graphics delivery system that operates in parallel with an existing Web site infrastructure is provided. The system streamlines the post-production process by automating the production of media through content generation procedures controlled by proprietary tags placed within URLs embedded within Web documents. The author simply places the original media in the system, and adds proprietary tags to the URLs for accessing that media. The system automatically processes the URL encoded tags and automatically produces derivative media for the web site from the original media.

The system takes as input the client connection, server traffic, content generation procedures, and proprietary tags placed within the URL to generate optimized media for the client. The need for the Web author to create different versions of a Web site is reduced because the image content of the site is automatically handled by the system. In addition, the generated media is cached so that further requests for the same media require little overhead.

Because the invention takes the original media, content generation procedures, and proprietary URL tags as inputs for generating the Web media, it is possible to modify any of these inputs and have the system automatically update the media on the associated Web pages.

A detailed description of such automatic media delivery system operating in parallel with existing Web site infrastructure is found below in the section under the heading as such.

Fig. 1 is a schematic diagram showing the placement of the system within a current Web infrastructure according to a preferred embodiment of the invention. The system 100 is attached to a Web server 110, which is connected to multiple client browsers 120(a-d) via the Internet 130.

Fig. 2 is a schematic diagram showing how a typical Web site delivers an HTML document and its graphics to a Web browser according to the prior art. An original media 200 is passed to post-production systems 210, wherein the media 200 is manipulated by hand and prepared for the Web. The result is a Web media 220. The Web media 220 and an associated HTML document 230 referring to the media 220 by media tags are input to a Web server 110 for a Web browser 120 to view via the Internet 130.

Fig. 3 is a schematic diagram showing delivery of an HTML document and media to a Web browser according to a preferred embodiment of the invention. An original media 200 and an HTML document embedded with proprietary media tags 300 are input into the system 100. The system 100 generates a Web-safe media 220 and a modified HTML document 230 that refers to the Web media, and automatically loads them onto the Web server 110 for view by a Web browser 120 via the Internet 160.

Fig. 4 is a schematic diagram showing components involved in Web site administration according to the prior art. Original media assets 400 are original images, video, or sound that have not been prepared for the Web. Web sites usually need to manage the placement of media on the network for easy retrieval by Web designers. Post-production systems 410 vary from Web site to Web site. Post-production systems 410 are usually custom procedures that Web designers use to convert an original media, such as an image, to one that can be displayed on the Web. Post-production systems 410 also upload finished images to Web image systems. Web images 420 are Web versions of the original images. Web images 420 are ready for retrieval by the Web server 110 to be delivered to a Web browser 120. Any image to be modified or updated must pass through the herein above three components before it can be delivered to the Web browser 120. HTML pages 460 have references to Web images 420.

Fig. 5 is a schematic diagram showing the components involved in Web site administration according to a preferred embodiment of the invention. Web site administration is simplified using the claimed invention. Asset management, automatic image manipulation, automatic image conversion, automatic image upload, and automatic disk management 500 are provided by the claimed invention.

Fig. 6 is a simple overview showing the components of the system according to a preferred embodiment of the invention. HTML with proprietary tags 300 is the original HTML document that is embedded with proprietary tags which describe how the images are to be manipulated for the Web. Java servlet engine 600 is a third-party product that allows the system 100 to interface with the Web server 110 and execute Java servlet code. The Web server 110 is third-party software that delivers Web pages to a Browser 120. The Browser 120 views Web pages that are sent from the Web server 110. Modified HTML with system created images 230 are a final result of the system. Modified HTML 230 is a standard HTML document without proprietary embedded tags and with standard Web graphics.

**The System.**

A preferred embodiment of the system 100 is provided.

HTML parsing subsystem 610 parses through an HTML document and searches for proprietary tags. If it finds a proprietary tag it hands it to a media caching subsystem 620 for further processing. The media caching subsystem 620 returns a standard HTML tag. The HTML parsing subsystem 610 then replaces the proprietary tag it found with the returned tag. The parsing subsystem 610 then continues searching for a next proprietary tag, repeating the process herein above. The process is finished when no more proprietary tags can be found.

The media caching subsystem 620 determines if an image has been created for the requested proprietary tag. If the image has already been created and the files that built that image have not been modified, the media caching subsystem 620 returns an HTML tag that refers to a previously-generated image. If the image has not been created, the media caching subsystem 620 hands the HTML tag to a media creation subsystem 630. The media creation subsystem 630 returns an image to the media caching subsystem 620. The media caching subsystem 620 adds the created image and the HTML tag to a media cache database 640.

The media cache database 640 contains references to the created images 645. In a preferred embodiment, the references are the script used to create the image, the names of the images used to create the image, the dates of those files, and the HTML that represents the created image. The media caching subsystem 620 performs lookups in this database to determine if the image has been created. If the image has not been created the media caching subsystem 620 calls upon the media creation subsystem 630 to create the image and then store the results in the media cache database 640.

The media creation subsystem 630 takes a proprietary tag from the media caching subsystem 620 and generates an image. The image is generated by deciphering the tag and handing it to the media processing engine 650. After the image is created, the media creation subsystem returns the name of the newly created image to the media caching subsystem 620.

The media processing engine 650 interprets the proprietary tag and generates the image. The media processing engine 650 looks up images in a media repository to obtain the location of the original file.

The media repository 660 contains original images 665 used in the system 100.

Fig. 7 is a schematic diagram showing the process flow of a proprietary enabled page delivered to a Web browser according to a preferred embodiment of the invention. An original media 200 is created. The media 200 is placed into the system 100 in the media repository 660. Similarly, an HTML document with proprietary tags 300 is created and placed on a Web server 110. A user requests a Web page from a Web browser 120. The Web server 110 passes the requested page to an HTML parser 610. The HTML parser 610 parses HTML looking for media tags. The parser 610 looks up media tags in a media tags database 640. If the media tag is found, then the system 100 produces a modified HTML document 230. Otherwise, the media creation subsystem 630 uses the media tag to generate a Web media 220. The generated Web media 220 is placed in a media cache subsystem 620. The proprietary media tag is converted by a converter 700 to a standard HTML tag that refers to the generated media 220 in cache. The media tag and the HTML equivalent are stored in the media tags database 640. Media tags are replaced by standard HTML equivalent to provide a modified HTML document 230. The modified HTML document 230 is delivered to the Web server 110. The Web server 100 delivers the modified HTML document 230 to the browser 120 via the Internet for a user to view.

Fig. 8 is a flow chart showing an authoring process according to a preferred embodiment of the invention. The process starts (800) when a user adds an original graphic to the system (810). The user then creates an HTML document that contains proprietary media tags (820). The user then places the HTML document on a Web server (830) and ends the authoring process (840).

Fig. 9 is a flow chart showing an HTML parsing process according to a preferred embodiment of the invention. The process starts (900) when a consumer requests a

Web page (910). A Web server hands the request of the Web page to the system (920). The system parses the Web page (930). The system looks for a media tag (940). If found, the system retrieves the HTML equivalent of the media tag (950) and replaces the media tag with the HTML equivalent tag (960). The system continues parsing the Web page for tags (970) by returning to step (940). When no more tags are found, the system delivers the modified Web page to the Web server (980) and therein ends the process (990).

Fig. 10 is a flow chart showing a media creation process according to a preferred embodiment of the invention. The process starts (1000) when the system requests an HTML equivalent to a proprietary media tag (1010). The Media tag is combined with bandwidth information (1020). The subsystem checks if the media tag already exists in the media tag database (1030). If it does, the subsystem checks if any of the original assets used to create the media have been changed (1040). If not, then the subsystem retrieves the HTML equivalent tag from the database (1050) and returns the HTML equivalent tag to the requesting system (1060). If any of the original assets used to create the media have been changed (1040), then the subsystem removes the media tag entry from the media database (1070) and creates the media using the media tag (1080). The subsystem then stores the media in a media cache (1090). The subsystem generates the HTML referring to the generated media (1100) and places the media tag and the HTML equivalent in the media tag database (1110). The HTML equivalent is returned to the requesting system (1060) and the process stops (1120).

The differences between using HTML and the proprietary tags disclosed herein are noted. HTML allows Web designers to create Web page layouts. HTML offers some control of the images. HTML allows the Web designer to set the height and width of an image. However, all of the other image operations disclosed herein are supported by the claimed invention and are not supported by HTML.

Table A herein below provides the claimed proprietary tags according to a preferred embodiment of the invention. The use of the term "freeride" refers to an internal code name for the invention.

Table A

### Tags

**Generate image**

<freerideimage> mediascript </freerideimage>

Generate a standard Web image.

**Generate thumbnail image linked to full image**

<freerideimagethumbnail> mediascript <xs=size ys=size /freerideimagethumbnail>

Generate a thumbnail of specified size and link it to the full size version.

**Generate zoom and pan image**

<freerideimagezoom> mediascript </freerideimagezoom>

Generate a zoomable/panable image.

**Security**

<freerideimagesecure> </freerideimagesecure>

Specifies that all images found between these tags are secured images and the system will determine access before generating.

Table B herein below provides the claimed script commands according to a preferred embodiment of the invention. Additional commands may be added as needed.

Table B

### Media processing script commands

**Add Noise**

Noise_AddNoise( [amount=<value 1..999>] [gaussian] [grayscale] )

This command adds noise to the image.

**Adjust HSB**

AdjustHsb([hue @ <value ±255>]  [saturation @ <value ±255>]  [brightness @ <value ±255>])

This command allows the HSB of an image to be altered. This can be applied to images of all supported bit-depths.

### Adjust RGB

AdjustRgb( [brightness @ <value ±255>] [contrast @ <value ±255>] [red @ <value ±255>] [green @ <value ±255>] [blue @ <value ±255>] [noclip @ <true, false>] [invert @ <true, false>] )

This command allows the contrast, brightness, and color balance of an image to be altered.

### Blur

Blur( radius @ <value 0..30>)

This command applies a simple blur filter on the image.

### Blur Convolve

Blur_Blur()

This command commands perform a simple 3x3 convolution for blurring.

### Blur Convolve More

Blur_MoreBlur()

This command commands perform a stronger 3x3 convolution for blurring.

### Blur Gaussian

Blur_GaussianBlur( [radius=<value 0.1..250>] )

This command applies a Gaussian blur to the image.

### Blur Motion

Blur_MotionBlur( [distance=<value 1..250>] [angle=<degrees>] )

This command applies motion blurring to the image using the specified distance and angle.

### Brush Composite

Composite( source @ {<User-Defined Media Object name>} [x @ <pixel>] [y @ <pixel>] [onto] [opacity @ <value 0..255>] [color @ <color in hexadecimal>] [colorize @ <true, false>] [saturation @ <value 0..255>] )

This command composites the specified “brush” (foreground) image onto the current “target” (background) image.

### Colorize

Colorize( color @ <color in hexadecimal> [saturation @ <value 0..255>] )

This command changes the hue of the pixels in the image to the specified color.

### Convert

Convert( rtype @ <bit-depth> {dither @ <value 0..10>] )

This command converts the image to the specified type/bit-depth.

### Convolve

Convolve( Filter @ <filtername> )

This command applies a basic convolution filter to the image. In a user interface driven system, the filters could be stored in files and edited/created by the user.

### Crop/Resize Canvas

Crop( [xs @ {<pixels>, <percentage + "%">}] [ys @ {<pixels>, <percentage + "%">}] [xo @ <left pixel>]
[yo @ <top pixel>] [padcolor @ <color in hexadecimal>] [padindex @ <value 0..255>] )

This command crops the media to a specified size.

### Discard

Discard()

This command removes the designated Media Object from memory.

### Drop Shadow

DropShadow( [dx @ <pixels>] [dy @ <pixels>] [color @ <color in hexadecimal>] [opacity @ <value 0..255>] [blur @ <value 0..30>] [enlarge @ <true, false>])

This command adds a drop shadow to the image based on its alpha channel.

### Equal

Equal( source @ {<User-Defined Media Object name>})

This command compares the current media with the one specified. If the media are different in any way, an error value is returned.

### Equalize

Equalize( [brightness @ <-1, 0..20>] [saturation @ <-1, 0..20>])

This command equalizes the relevant components of the media. Equalization takes the used range of a component and expands it to fill the available range.

### Export Channel

ExportGun( Channel @ <channelname> )

This command exports a single channel of the source as a grayscale image.

### Find Edges

Stylize_FindEdges( [threshold=<value 0..255>] [grayscale] [mono] [invert] )

This command finds the edges of the image based on the specified threshold value.

### Fix Alpha

FixAlpha()

This command adjusts the RGB components of an image relative to its alpha channel.

### Flip

Flip( <horizontal, vertical> @ <true, false> )

This command flips the media vertically or horizontally.

### Frame Add

FrameAdd( Source @ <filename> )

This command adds the given frame(s) to the specified Media Object.

### Glow/Halo

Glow( Size @ <value 0..30> [halo @ <value 0..size>] [color @ <color in hexadecimal>] [opacity @ <value 0..255>] [blur @ <value 0..30>] [enlarge @ <true, false>] )

This command produces a glow or halo around the image based on the image's alpha.

### High Pass

Other_HighPass( [radius=<value 0.1..250>] )

This command replaces each pixel with the difference between the original pixel and a Gaussian blurred version of the image.

### Import Channel

ImportGun( channel @ <channel name> source @ {<User-Defined Media Object name>} [rtype @ <bit-depth>])

This command imports the specified source image (treated as a grayscale) and replaces the selected channel in the original.

### Load

Load( Name @ <filename> [type @ <typename>] [transform @ <true, false>] )

This command loads a media from the specified file.

### Maximum

Other_Maximum( [radius=<value 1..10>] )

This command scans the area specified by the radius surrounding each pixel, and then replaces the pixel with the brightest pixel found.

### Minimum

Other_Minimum( [radius=<value 1..10>] )

This command scans the area specified by the radius surrounding each pixel, and then replaces the pixel with the darkest pixel found.

### Normalize

Normalize( [clip @ <value 0..20>] )

This command expands the volume of the sample to the maximum possible.

### Pixellate Mosaic

Pixellate_Mosaic( [size=<value 2..64>] )

This command converts the image to squares of the specified size, where each square contains the average color for that part of the image.

**Pixellate Fragment**

Pixellate_Fragment( [radius=<value 1..16>] )

This command produces four copies of the image displaced in each direction (up, down, left, right) by the specified radius distance and then averages them together.

**Quad Warp**

QuadWarp( [tlx=<position>] [tly=<position>] [trx=<position>] [try=<position>] [blx=<position>] [bly=<position>] [brx=<position>] [bry=<position>] [smooth] )

This command takes the corners of the source image and moves them to the specified locations, producing a warped effect on the image.

**Reduce to Palette**

Reduce( [colors @ <num colors>] [netscape @ <true, false>] [b&w @ <true, false>] [dither @ <value 0..10>] [dithertop @ <value 0..10>] [notbackcolor] [pad @ <true, false>] )

This command applies a specified or generated palette to the image.

**Rotate**

Rotate( Angle @ <value 0..359> [smooth @ <true, false>] [enlarge @ <true, false>] [xs @ <pixels>]
[ys @ <pixels>] )

This command rotates the media by the specified angle in degrees.

**Rotate 3D**

Rotate3d( [anglex @ <angle ±89>] [angley @ <angle ±89>] [distance @ <value>] )

This command rotates the image in 3D about either the x-axis or y-axis.

**Save**

Save([type @ <image-type>])

This command saves a media to the specified file.

**Scale**

Scale( [xs @ {<pixels>, <percentage + "%">}] [ys @ {<pixels>, <percentage + "%">}] [constrain @ <true, false>] [alg @ {"fast", "smooth", "outline"}] [x1 @ <pixels>] [y1 @ <pixels>] [x2 @ <pixels>] [y2 @ <pixels>] )

This command scales the image to the specified size.

**Select**

Selection( [source @ <User-Defined media Object>}] [remove @ <true, false>] [invert @ <true, false>]
[backcolor] [color=<color>] [index=<value>] [opacity @ <value 0..255>] )

This command manages the selected region for the current Media Object.

**Set Color**

SetColor( [backcolor @ <color in hexadecimal>] [forecolor @ <color in hexadecimal>] [backindex @ <value 0..255>] [foreindex @ <value 0..255>] [transparency @ ("on","off")] )

This command allows the background color, foreground color, and transparency state of an image to be set.

### Set Resolution

```
SetResolution( [dpi @ <value>] [xdpi @ <value>] [ydpi @ <value>] )
```

This command changes the DPI of the image in memory.

### Sharpen

```
Sharpen_Sharpen()
```

This command sharpens the image by enhancing the high-frequency component of the image.

### Sharpen More

```
Sharpen_SharpenMore()
```

This command sharpens the image by enhancing the high-frequency component of the image, but is stronger than the standard sharpening.

### Stylize Diffuse

```
Stylize_Diffuse( [radius=<value 0..>] [lighten] [darken] )
```

This command diffuses the image by randomizing the pixels within a given pixel radius.

### Stylize Embose

```
Stylize_Emboss( [height=<value 1..10>] [angle=<degrees>] [amount=<percentage 1..500>])
```

This command converts the image to an embossed version.

### Text Drawing

```
DrawText( Text @ <string> Font @ <font file>  [size @ <value>]
[color @ <color in hexadecimal>] [smooth @ <true, false>] [<left, right, top, bottom> @ <true, false>]
[x @ <pixel>] [y @ <pixel>] [wrap @ <pixel-width>] [justify @ {left,center,right}] [angle @ <angle>])
```

This command composites the specified text string onto the image.

### Text Making

```
MakeText( text @ <string> font @ <font file> [path @ <path to font directory>] [size @ <value 1..4095>]
[color @ <color in hexadecimal>] [smooth @ <true, false>] [wrap @ <pixel-width>]
[justify @ {left,center,right}] [angle @ <angle>] )
```

This command creates a new image that includes only the specified text.

### Trace Contour

```
Stylize_TraceContour( [level=<value 0..255>] [upper] [invert] )
```

This command traces the contour of the image at the specified level (for each gun).

**Unsharpen Mask**

```
Sharpen_UnsharpMask( [amount=<percentage 1..500>] [radius=<value 0.1..250>]
[threshold=<value 0..255>] )
```

This command enhances the edges and detail of an image by exaggerating differences between the image and a gaussian blurred version of the same image.

**Zoom**

```
Zoom( [xs @ <pixels>] [ys @ <pixels>] [scale @ <value>] [x @ <left pixel>] [y @ <top pixel>] )
```

This command zooms in on a specified portion of the media and fits it to the specified size. This constitutes a crop followed by a scale.

Table C herein below provides a list of features provided by a preferred embodiment of the invention. It is noted that the list of features included in Table C is by no means complete. In other embodiments, the list of features is expanded or reduced as needed.

Table C -System Feature List

- Reads and writes various file formats:
  BMP, GIF, JPG, PNG, TIF, PICT, TGA, PSD, FPX;
- Supports many image processing operations;
- Dynamically creates Web images from original assets;
- Dynamically creates thumbnail images;
- Dynamically creates images that can be panned and zoomed without browser plug-ins or special file formats;
- Automatically propagates changes of original assets throughout a Web site;
- Uses an intelligent caching mechanism:
  Clean up image cache on demand;
  Eliminates orphaned image files; and
  Optimizes Web server cache by providing most recent images;

- Renders TrueType fonts on the server instead of browser;
- Uses intelligent scaling of line drawings;
- Allows Web designers to manipulate images with proprietary tags;
- Preserves original image assets;
- Optimizes Web server traffic by adjusting the bandwidth of graphics;
- Optimizes images for client connection speed;
- Allows clients to specify the quality of images on a Web site; and
- Allows Web designers to dynamically create images by manipulating proprietary tags in their applications (server or client side).

Fig. 11 is a screen shot showing an administration tool according to a preferred embodiment of the invention. Specifically, Fig. 11 shows an administration page that contains cached images of generated scripts. The use of the term "freeride" refers to an internal code name for the invention.

Fig. 12 displays a structure of a database record used for the system according to a preferred embodiment of the invention. A Script Table 1200 has 5 columns, Media Script 1210, HTML Equivalent 1220, Bandwidth 1230, Generated File 1240, and Dependency List 1250. A Dependency Table 1260 has two columns, File Name 1270 and Modification Date 1280.

**Snowboard Store Example.**

Background.

The snowboard store highlights several features of the claimed system. The snowboard store is an imaginary store that allows a user to configure his or her snowboard. The store consists of five logos, five board colors, and four boards. The consumer clicks on the buttons to change the snowboard represented in the middle of the screen. When the consumer has configured the snowboard they the snowboard can be purchased by selecting a buy button.

Prior Art Method.

To create the snowboard site today, the Web designer must render all possible combinations of the board. The number of combinations is five logos x five board colors x four boards = 100. The designer also must render all the buttons. The creation process is very tedious and involves a lot of production work. Typically, most Web sites do not even attempt such an endeavor. Also, other issues must be addressed, such as, for example, updating the Web site and scripting. For example, updating a single logo involves updating a minimum of 20 images.

The prior art method sustains a graphic intensive site that requires management of at least 100 images. Updates to the Web site are time-consuming and prone to human error.

The Claimed Method.

A preferred embodiment of the method scripts the image creation process in HTML to create a dynamic Web site. There is no need to create over 100 images. The claimed system generates images on demand. The Web site only needs to create original assets. The scripting process involves writing the proprietary scripts. In the current example herein, scripting buttons is very simple. Once one button is created, simply copy and paste the HTML to create another button or many buttons. Only the name of the image to be overlaid on the button must be changed. The Webmaster then creates a simple program that reads what object a user has clicked on and generates a proprietary tag. The tag is then sent to the claimed system to generate a center image.

The claimed method allows the creation of all 100 combinations automatically. When the Web site receives an updated image, only the original image needs to be updated. Any change to the original image automatically propagates throughout the system. The Web site is easier to manage. Testing of the Web site is easier because there is no need to test all 100 combinations. A small subset of combinations will guarantee adequate coverage.

**Processing of an Image Tag Example (Fig. 13-16).**

Fig. 13 shows two original images 1300 and 1310 to be processed according to a preferred embodiment of the invention.

Fig. 14 shows a portion on an HTML document with a proprietary tag 1400, <freerideimage></freerideimage> according to a preferred embodiment of the invention. The use of the term “freeride” refers to an internal code name for the invention.

Fig. 15 shows an HTML document 1500 as viewed in a browser and an HTML document source 1510, according to a preferred embodiment of the invention. The use of the term “freeride” refers to an internal code name for the invention.

Fig. 16 shows a generated GIF image 1600 according to a preferred embodiment of the invention.

**Automatic media delivery system operating in parallel with existing Web site infrastructure.**

It should be noted that the words, media, graphics, and images are used herein interchangeably.

An automatic graphics delivery system that operates in parallel with an existing Web site infrastructure is provided. The system streamlines the post-production process by automating the production of media through content generation procedures controlled by proprietary tags placed within URLs embedded within Web documents. The author simply places the original media in the system, and adds proprietary tags to the URLs for accessing that media. The system automatically processes the URL encoded tags and automatically produces derivative media for the web site from the original media.

The system takes as input the client connection, server traffic, content generation procedures, and proprietary tags placed within the URL to generate optimized media for the client. The need for the Web author to create different versions of a Web site is reduced because the image content of the site is automatically handled by the system. In addition, the generated media is cached so that further requests for the same media require little overhead.

Because the invention takes the original media, content generation procedures, and proprietary URL tags as inputs for generating the Web media, it is possible to modify any of these inputs and have the system automatically update the media on the associated Web pages.

A preferred embodiment of the invention is described with reference to Fig. 17. Fig. 17 is a schematic diagram of an image system within a typical Web infrastructure according to the invention. The image system 100 is placed in parallel to an existing Web server 110. The image system 100 may be on-site or off-site to the Web infrastructure. Multiple client browsers 120a-120d communicate with both the Web server 110 and the image server 100 via the Internet 130.

The delivery of an HTML document and media according to a preferred embodiment of the invention is described with reference to Fig. 18. Resource locators (URLs) are placed within HTML documents 301 accessible to the Web server 110. These URLs direct browsers to generate requests for media to the system 100. The system processes such URLs by interpreting the proprietary tags, executing the indicated image generation procedures on the original media 200, and returning derivative Web-safe media to client browsers 120a-120d via the Internet 130. Additionally, such generated media is cached on the image server 100 and, therefore need not be regenerated for subsequent requests.

Web site administration according to another preferred embodiment of the invention is described with reference to Fig. 19. Fig. 19 is a schematic diagram showing the components of Web site administration according to a preferred embodiment of the invention, whereby Web site administration is simplified. The preferred embodiment provides, but is not limited to the following services: asset management, automatic image manipulation, automatic image conversion, automatic image upload, automatic image customization based on browser characteristics, automatic disk management, automatic control of proxy caching, and image delivery 501.

Fig. 20 is a simple overview showing components of the system according to a preferred embodiment of the invention. HTML pages with proprietary URL tags 301 describe how referenced media therein is to be manipulated for Web. Browsers 120 send such tags to the image system 100 as media requests. A server 2000 within the image system 100 receives the media requests, decodes the URL tags, and retrieves any media that already exists in the media caching system 2010. Non-existent media is subsequently generated by a media creation system 2020 using original media 2050 stored in a media repository 2040 and using content generation procedures 2030.

The Image System.

Following is a detailed description of the preferred embodiment of the invention with reference to Fig. 21 below.

The system receives a request for media through a URL containing proprietary tags for controlling image generation. The system parses this URL to determine the content generation procedure to execute, input to the content generation procedure, post-processing directives for, for example, zoom/pan/slice, browser properties, and any cache control directives. Such data is handed to a media caching subsystem that returns the requested image if found. If the image is not found, the information is handed to the media generation subsystem that executes the specified content generation procedure to produce a derivative image. The media generation subsystem returns one or more images to the media caching system for subsequent reuse.

The media caching subsystem is a mechanism for associating final or intermediate derivative media with the procedure, input, and user characteristics used to generate said media, specified through proprietary tags within the requested URL. This system may be implemented using a database, file system, or any other mechanism having capability to track such associations.

The media generation subsystem executes a primary content generation procedure to produce a derivative image whose identifier is provided to the media caching subsystem. This derivative image is composed of one or more original images acquired from the media repository. This media is then passed to the dynamic image content system, if necessary, to generate a subsequent derivative media suitably modified for the needs of zooming, panning, or slice. The resulting media is passed to the user profile system where it is again modified to account for any specific user browser characteristics specified using the proprietary URL tags. This media is then returned to the browser, along with any cache control directives encoded within the URL, and its identifier is passed to the media caching system for subsequent retrieval.

The dynamic content system operates on intermediate derivative images to generate image subsets or scalings used by Web site designers to implement zooming in on an image, panning across an image, slicing an image into parts, and the like for special Web page effects. The input to this system is cached by the media caching system such that the intermediate image need not be regenerated.

The user profile system operates on the final image about to be returned to the browser and may modify the image to account for individual needs of Web site users. The designer of a site is able to implement freely custom post-processing of images to meet the specific needs of their clients.

Fig. 21 is a schematic diagram showing the process flow of a proprietary enabled page delivered to a Web browser according to a preferred embodiment of the invention. Original media 200 is created and placed into the system 100 in a media repository 2040. A content generation procedure 2140 is created with instructions on how the media is to be transformed to create the desired Web page content. An HTML page 301 is created for the Web site comprising the system 100, the page containing one or more URLs directing a browser 120 to request the specified content generation procedure 2140 from the system 100 using input parameters specified with proprietary tags encoded within the URL. The browser 120 requests the Web page 301 from the Web site 110. Upon receipt of the page 301, the browser contacts the system 100 requesting media specified in the URL. The system parses the URL 2100 to determine the content generation procedure 2140 to execute, any corresponding input parameters to be used by such procedure, any dynamic content processing 2150 to be performed by dynamic media procedures, any user profile information 2160 to be used to modify the resulting image, and any cache control HTTP headers 2190 the site instructs to accompany the resulting image.

The parser generates a unique primary lookup key 2110 for the specified resulting media. If the key corresponds to an existing generated media 2180, such media is returned immediately to the browser 120 through a media cache 2120, and the transaction is complete. Otherwise, a media generation occurs. In the case of media generation requiring dynamic content processing, a unique secondary lookup key corresponding to intermediate media is generated 2130. If intermediate media 2170 corresponding to this key is found, such media is passed directly to the dynamic media content system 2150 having dynamic media procedures, wherein appropriate action is taken to generate the required derivative from the intermediate media data. A unique key is generated for the derivative 2130 and passed to the media caching system 2120. If the media caching system finds no such intermediate image, such intermediate image is generated according to instructions specified by the content generation procedure, cached by the media cache system 2120 as a secondary cached media 2170, and passed to the dynamic media system 2150. Again, appropriate action is taken to generate the required derivative from the intermediate image data.

The resulting image after any dynamic media processing is complete, is checked to ensure that the image is in a valid Web image format. If not, the image is automatically converted into a valid format.

The final media is passed to a user profile system 2160 wherein browser characteristics specified through proprietary tags within the URL are inspected, and appropriate modification to the media is performed, based on such characteristics. The resulting image is handed to the media cache system 2120 for caching and returned to the browser 120.

Fig. 22 shows a flowchart of the content generation procedure according to a preferred embodiment of the invention. A URL containing proprietary tags (2200) is parsed (2210) to determine the content generation procedure to execute, any dynamic modifications to the media, user profile characteristics, and proxy-cache control. A unique final lookup key is generated for the media (2220) and the media cache is checked (2230). If the indicated media exists, control passes to proxy-cache control (2290) and the media is delivered to the browser (2295). Otherwise, dynamic media system tags are separated from content generation control tags (2240) and a unique intermediate image lookup key is generated (2250). The cache is then checked for such intermediate media (2261). If such intermediate media is found, it is used directly for dynamic processing, if required. Otherwise, content is generated (2262) and cached (2263), and the result is evaluated for dynamic processing (2270). If dynamic processing is required, the media is operated upon by the dynamic content generator (2271), otherwise it is evaluated for valid content type (2272). If the content type is invalid, the media is automatically converted to a valid type (2273). The resulting image is then customized by the user profiling system (2280) for specified browser or client attributes. Finally, any cache-control directives specified are attached to the response (2290) and the media is delivered to the browser (2295).

Fig. 23 is a flow chart showing an authoring process according to a preferred embodiment of the invention. The process starts (2300) when a user adds an original graphic or other media (2310) to the system. The author then creates a content generation procedure (2320) containing instruction on how the original media should be processed to generate the desired Web page content. The user then creates an HTML document (2330) that refers to that image by using a URL pointing to a content generation procedure on the image server. The system ends (2340). The authoring subsystem assists the Web site designer with choosing parameters and with designing the content generation procedure such that the desired Web site graphic is obtained.

It should be appreciated that differences exist between specifying an image with a URL and requesting an image using a content creation process that interprets proprietary parameters encoded within a URL. That is, URLs allow Web site designers to load specific graphic images into a Web page. In contrast and according to the invention, URLs containing proprietary content creation tags initiate a process whereby graphic images for a site are automatically produced.

Table D below is a list of example proprietary URL tags used for content generation within the system according to the preferred embodiment of the invention. Additional tags may be added to the system as necessary.

Table D – Tags

f=function

Names the content creation procedure used to generate all or part of the desired graphic.

args=arguments

Supplies page dependent parameters used to control the content creation procedure from within the Web page.

cr=crop rectangle

Specifies that portion of the image generated by the content generation procedure to be returned to the browser.

st=slice table

Specifies a rectangular grid to be placed over the image produced by the content generation procedure, each portion of which can be returned to the browser.

sp= slice position

Specifies that portion of the slice table grid placed over the image generated by the content creation procedure to be returned to the browser.

is=image size parameter

Specifies scale factors to be applied to any portion of an image generated by any combination of a content generation procedure, arguments, crop rectangles, slice tables, and slice positions.

p=user profile string

Specifies a user profile identifier used to modify the final image prior to returning the image to the browser, thus allowing clients to modify the image returned to the browser to account for individual browsing conditions.

c=cache control

Specifies a proxy-cache control string to accompany the returned image within an HTTP header.

Table E below is a list of example supported content creation commands according to a preferred embodiment of the invention. Additional commands may be added as necessary.

Table E – Content Creation Commands

Adjust HSB

Allows the HSB of an image to be altered.

Adjust RGB

Allows the contrast, brightness, and color balance of an image to be altered.

Colorize

Alters the hue of the pixels in the image to that of the specified ***color***.

Brush Composite

Composites the specified brush image onto the current target image.

Convert

Converts the rasters to the specified type/bit-depth.

Crop

Crops the media to the specified size.

Dropshadow

Adds a drop shadow to the image, based on the alpha-channel.

Equalize

Performs an equalization on the relevant components of the media.

FixAlpha

Adjusts the RGB components of an image relative to its alpha-channel.

Flip

Flips the media vertically or horizontally.

Glow

Produces a glow or halo around the image.

Load

Loads in a media from the specified file.

Normalize

Similar to equalize, but for audio.

Reduce

Reduces the image to a specified palette.

Rotate

Rotates the media clockwise by the specified ***angle*** in degrees.

Save

Saves the media to the specified file.

Scale

Scales the media to the specified size.

SetColor

Allows the background color, foreground color, and transparency state of the media to be set.

Text Drawing

Composites the specified text onto the image.

Text Making

This command, instead of compositing text onto the target, creates a new image that just encloses the text.

Zoom

Zooms in on a specified portion of the media, and fits it to the specified size. Effectively this constitutes a crop followed by a scale.

Table F below lists comprises some, and is not limited to all major features of a preferred embodiment of the invention. Additional features may be added as necessary.

Table F - System Features

Reads and writes various file formats;

Supports many image processing operations;

Dynamically creates Web images from original assets;

Dynamically creates thumbnail images;

Dynamically and efficiently creates images that can be panned, zoomed, or sliced from original assets without Browser plugins;

Automatically propagates changes in original assets throughout the Web site;

Uses an intelligent caching mechanism for both final and intermediate graphics, comprising:

Clean up cache on demand;

Eliminates orphaned Web files; and

Optimizes Web server cache by providing most recent images;

Renders True-Type fonts on server instead of browser;

Uses intelligent scaling of line drawings;

Allows Web designers to manipulate images using a combination of content generation procedures and proprietary URL tags;

Preserves original image assets;

Optimizes Web server traffic by adjusting the bandwidth of graphics;

Optimizes images for client connection speed;

Allows clients to specify the quality of images on a Web site;

Allow site-specific customized image optimizations for a variety of purposes; and

Allows Web designers to dynamically create images by manipulating proprietary URL tags in applications.

Accordingly, although the invention has been described in detail with reference to a particular preferred embodiment, persons possessing ordinary skill in the art to which this invention pertains will appreciate that various modifications and enhancements may be made without departing from the spirit and scope of the claims that follow.

## CLAIMS

1. A computer program product comprising one or more computer-readable media having thereon computer-executable instructions that, when executed by one or more processors of a computing system, causes the computing system to perform a method for dynamically transcoding media content to be presented on a client presentation system, the method comprising:

an act of receiving a request for media content to be delivered to a client presentation system;

after receiving the request, an act of identifying transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system;

an act of transcoding the requested media content in accordance with the identified transcoding parameters;an act of causing the transcoded media content to be delivered to the client presentation system; and

an act of caching the transcoded media content.

2. A computer program product in accordance with claim 1, wherein the one or more computer-readable media are physical memory and/or storage media.

3. A computer program product in accordance with claim 2, wherein the act of receiving a request for media content is performed by receiving the request over the Internet.

4. A computer program product in accordance with claim 2, wherein the client presentation system is a first client presentation system, the method further comprising:

an act of receiving a request for the media content to be delivered to a second client presentation system;

an act of determining that the transcoding parameters to be applied to the requested media content prior to delivery to the second client presentation system are the same as the transcoding parameters that were applied to the requested media content prior to delivery to the first client presentation system; and an act of delivering the cached transcoded media content, rather than separately transcoding the requested media content again for the second client presentation system.

5. A computer program product in accordance with claim 2, wherein the requested media content is video media.

6. A computer program product in accordance with claim 2, wherein the identified transcoding parameters includes at least video size parameters.

7. A computer program product in accordance with claim 2, wherein the method further comprises:

prior to the act of transcoding, an act of determining that a transcoded representation of the requested media content transcoded using the identified transcoding parameters does not already exist.

8. A computer program product in accordance with claim 2, wherein the act of identifying transcoding parameters comprises an act of selecting a transcoding profile from a set of pre-existing transcoding profiles.

9. A computer program product in accordance with claim 2, wherein the act of identifying transcoding parameters comprises:

an act of reading one or more transcoding parameters from the request; and

an act of identifying the one or more transcoding parameters from the request as being the identified transcoding parameters to be applied.

10. A computer program product in accordance with claim 2, wherein the act of act of transcoding the requested media content in accordance with the identified transcoding parameters comprises:

an act of selecting a pre-existing base transcoded version of the requested media content that has been transcoded in accordance with only a portion of the identified transcoding parameters; and an act of incrementally performing further transcoding of the pre-existing base transcoded version in accordance with a remaining portion of the identified transcoding parameters.

11. A method for accessing dynamically transcoding media content, the method comprising:

an act of receiving a request for media content to be delivered to a client presentation system for media content;

after receiving the request, an act of identifying transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system;

an act of determining that the transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system are the same as transcoding parameters that are being applied to the requested media content prior to delivery to another client presentation system; and

an act of delivering the transcoded media content to both client presentation systems concurrently.

# Automated Media Delivery System

## ABSTRACT

An automatic graphics delivery system that operates in parallel with an existing Web site infrastructure is provided. The system streamlines the post-production process by automating the production of media through content generation procedures controlled by proprietary tags placed by an author within URLs embedded within Web documents.

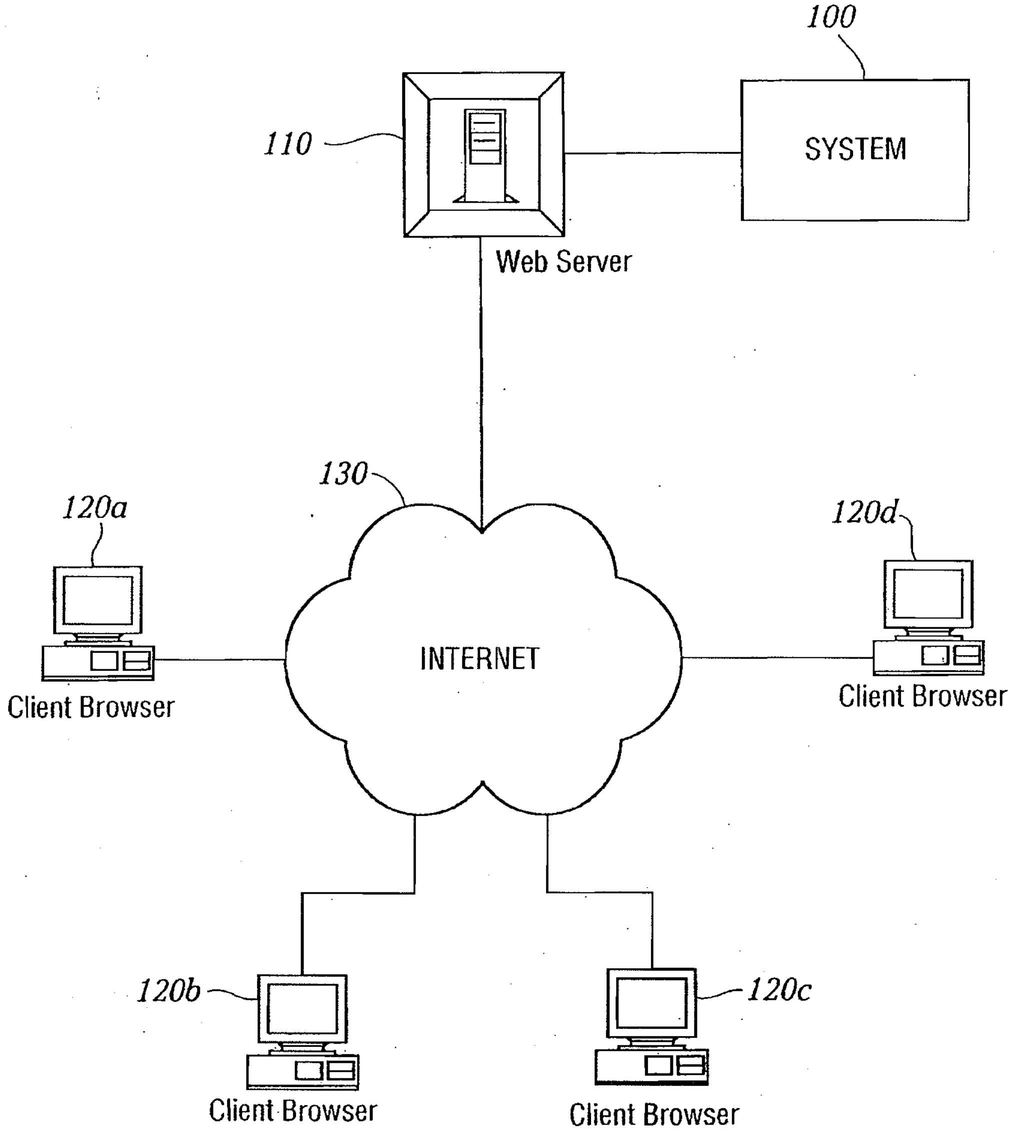

100
SYSTEM
110
Web Server
130
INTERNET
120a
Client Browser
120d
Client Browser
120b
Client Browser
120c
Client Browser

FIG. 1

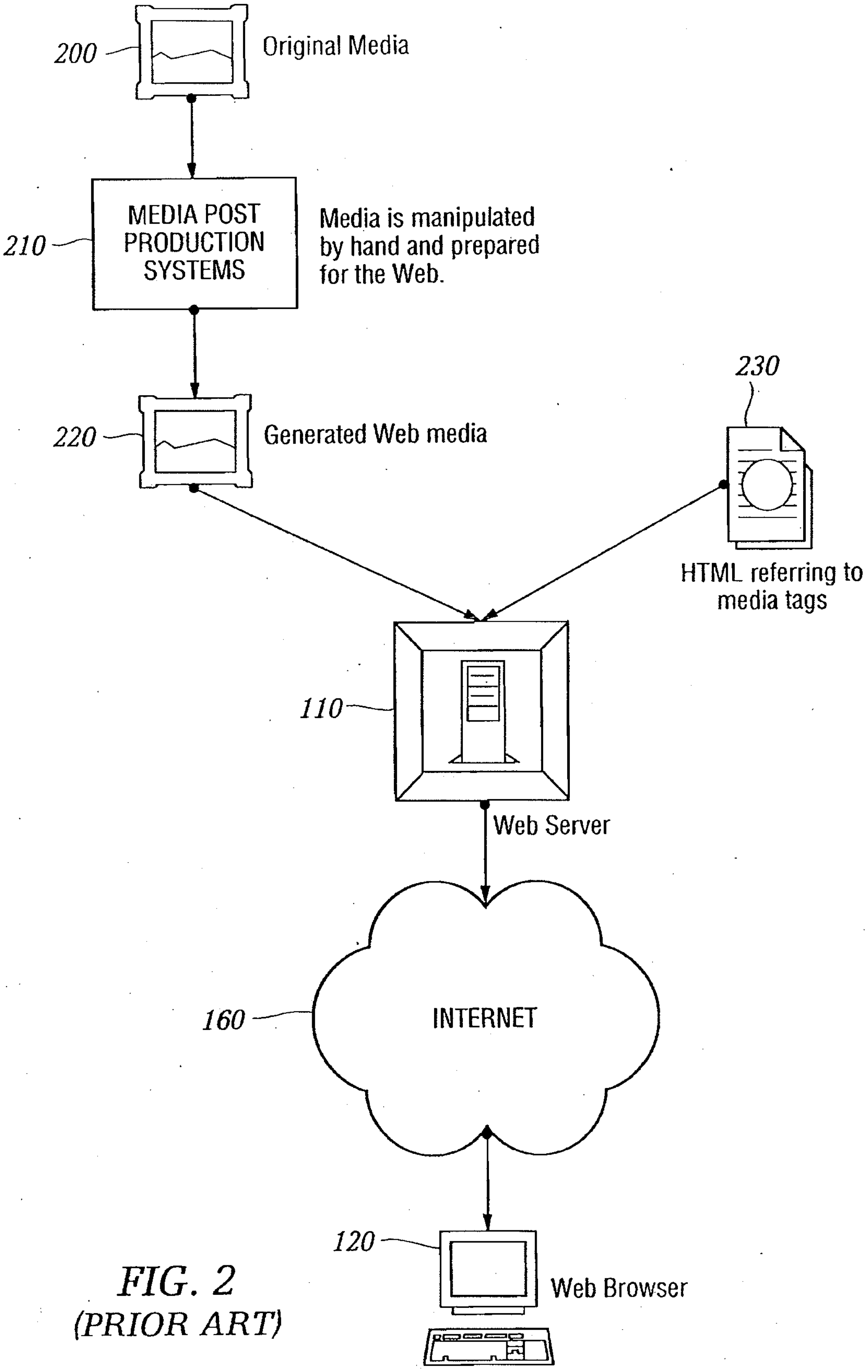

200
Original Media
210
MEDIA POST PRODUCTION SYSTEMS
Media is manipulated by hand and prepared for the Web.
220
Generated Web media
230
HTML referring to media tags
110
Web Server
160
INTERNET
120
Web Browser

FIG. 2
(PRIOR ART)

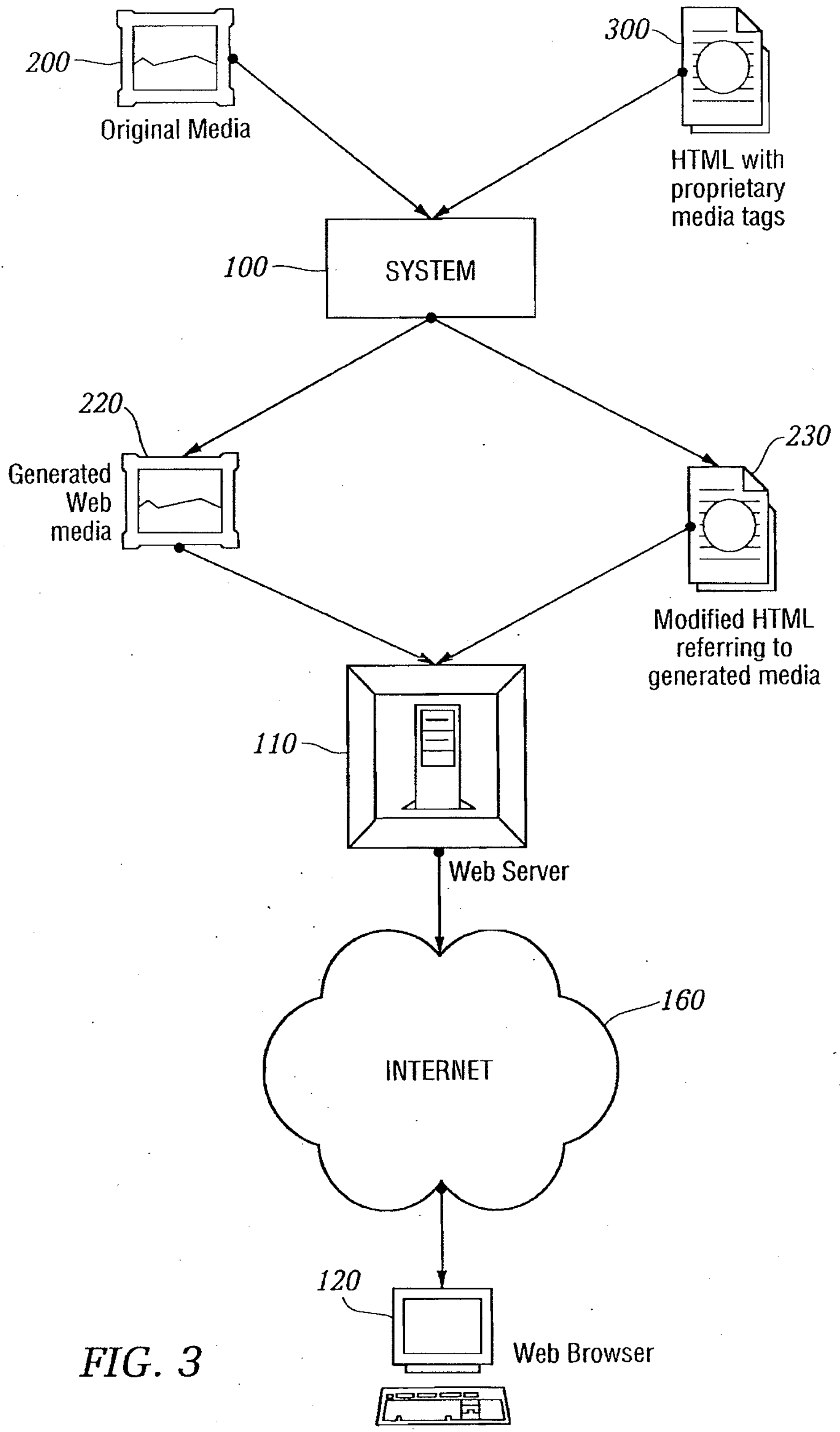

200
Original Media
300
HTML with proprietary media tags
100
SYSTEM
220
Generated Web media
230
Modified HTML referring to generated media
110
Web Server
160
INTERNET
120
Web Browser

FIG. 3

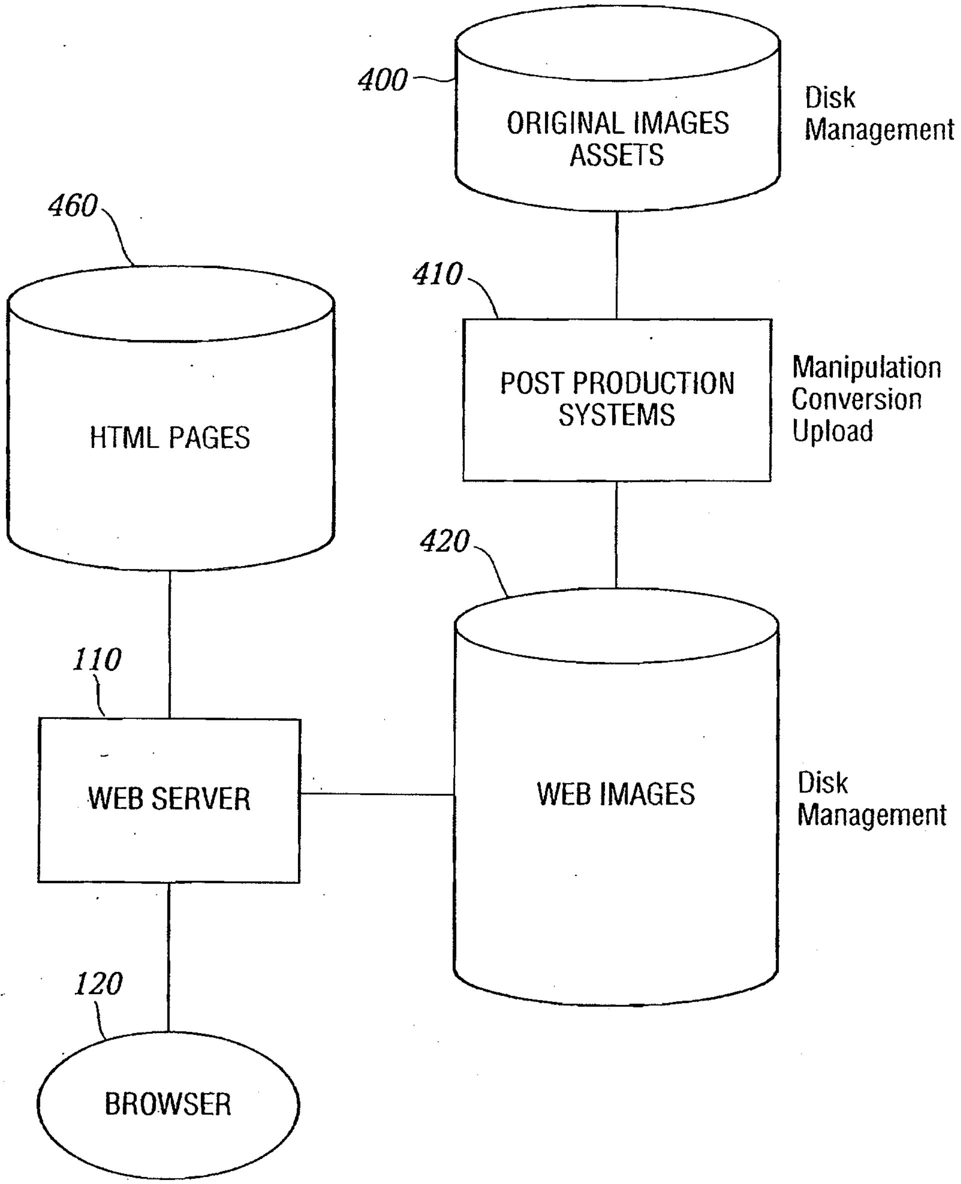


*FIG. 4*
*(PRIOR ART)*

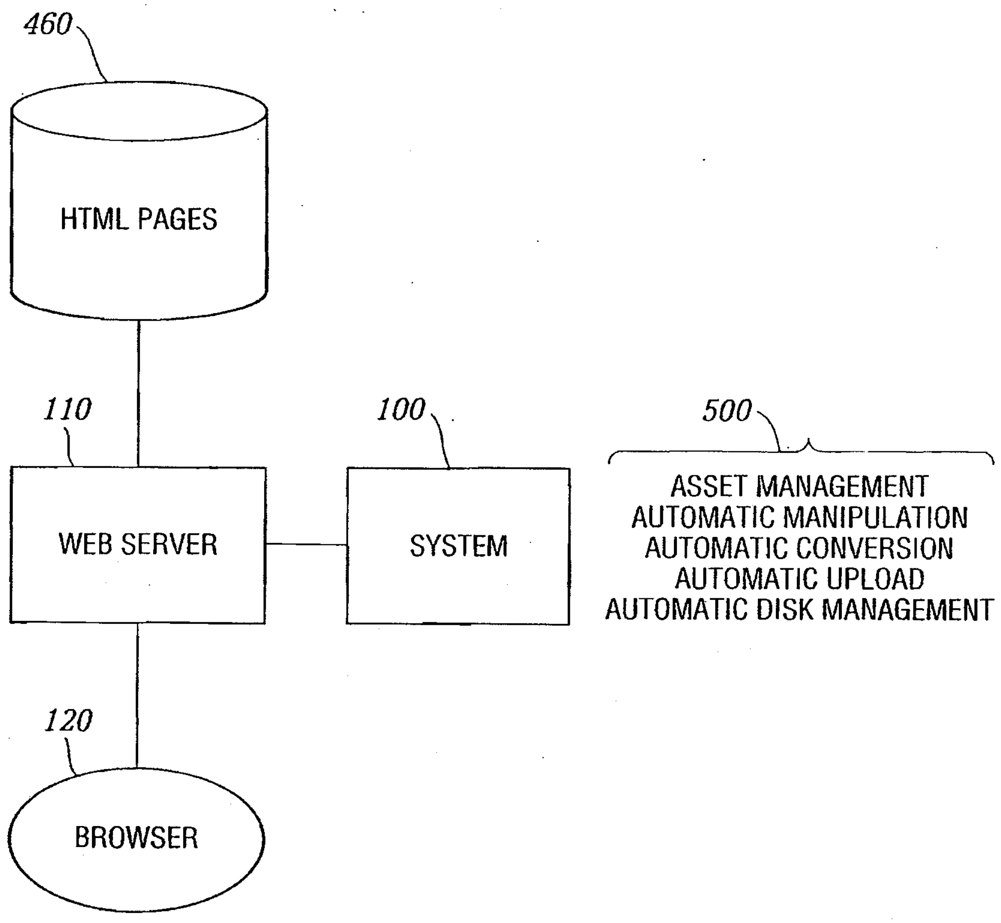

460
HTML PAGES
110
WEB SERVER
100
SYSTEM
500
ASSET MANAGEMENT
AUTOMATIC MANIPULATION
AUTOMATIC CONVERSION
AUTOMATIC UPLOAD
AUTOMATIC DISK MANAGEMENT
120
BROWSER

FIG. 5

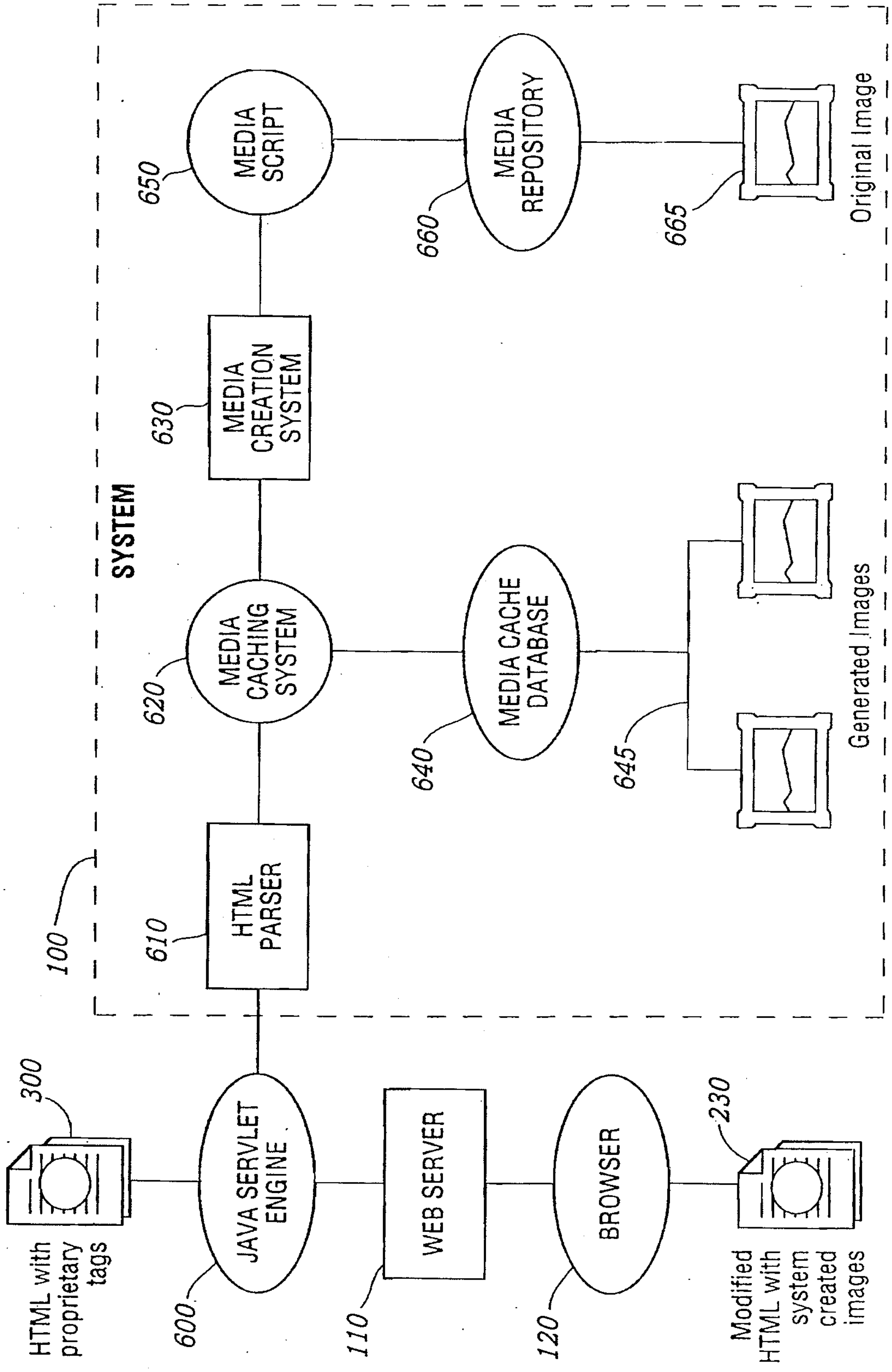

HTML with proprietary tags
300
600
JAVA SERVLET ENGINE
110
WEB SERVER
120
BROWSER
230
Modified HTML with system created images
100
SYSTEM
610
HTML PARSER
620
MEDIA CACHING SYSTEM
630
MEDIA CREATION SYSTEM
650
MEDIA SCRIPT
640
MEDIA CACHE DATABASE
645
Generated Images
660
MEDIA REPOSITORY
665
Original Image

FIG. 6

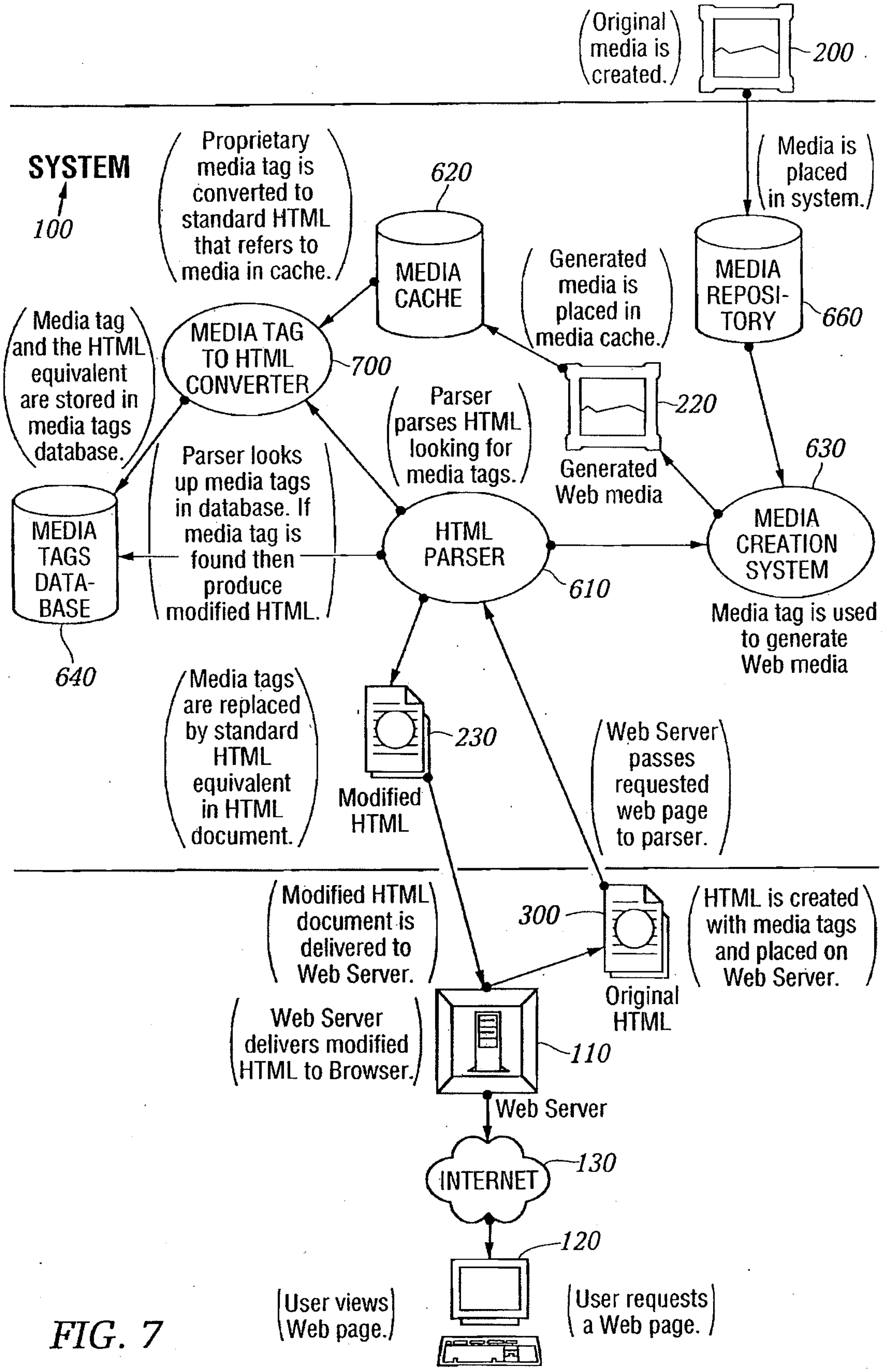

Original media is created.
200
SYSTEM
100
Proprietary media tag is converted to standard HTML that refers to media in cache.
620
MEDIA CACHE
Generated media is placed in media cache.
Media is placed in system.
MEDIA REPOSITORY
660
Media tag and the HTML equivalent are stored in media tags database.
MEDIA TAG TO HTML CONVERTER
700
Parser parses HTML looking for media tags.
220
Generated Web media
630
Parser looks up media tags in database. If media tag is found then produce modified HTML.
MEDIA TAGS DATA-BASE
640
HTML PARSER
610
MEDIA CREATION SYSTEM
Media tag is used to generate Web media
Media tags are replaced by standard HTML equivalent in HTML document.
230
Modified HTML
Web Server passes requested web page to parser.
Modified HTML document is delivered to Web Server.
300
HTML is created with media tags and placed on Web Server.
Original HTML
Web Server delivers modified HTML to Browser.
110
Web Server
INTERNET
130
120
User views Web page.
User requests a Web page.

FIG. 7

AUTHORING FLOWCHART

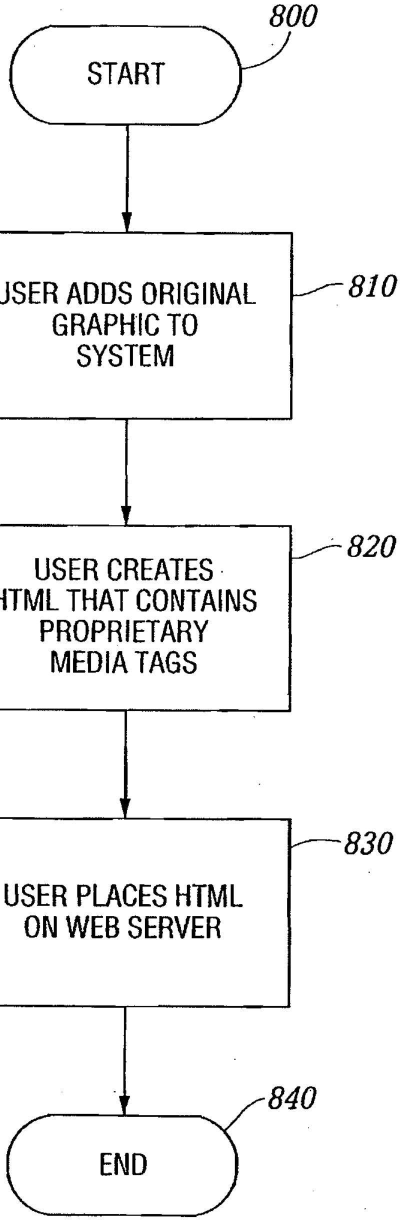

800
START
USER ADDS ORIGINAL GRAPHIC TO SYSTEM
810
USER CREATES HTML THAT CONTAINS PROPRIETARY MEDIA TAGS
820
USER PLACES HTML ON WEB SERVER
830
840
END

*FIG. 8*

HTML PARSING FLOWCHART

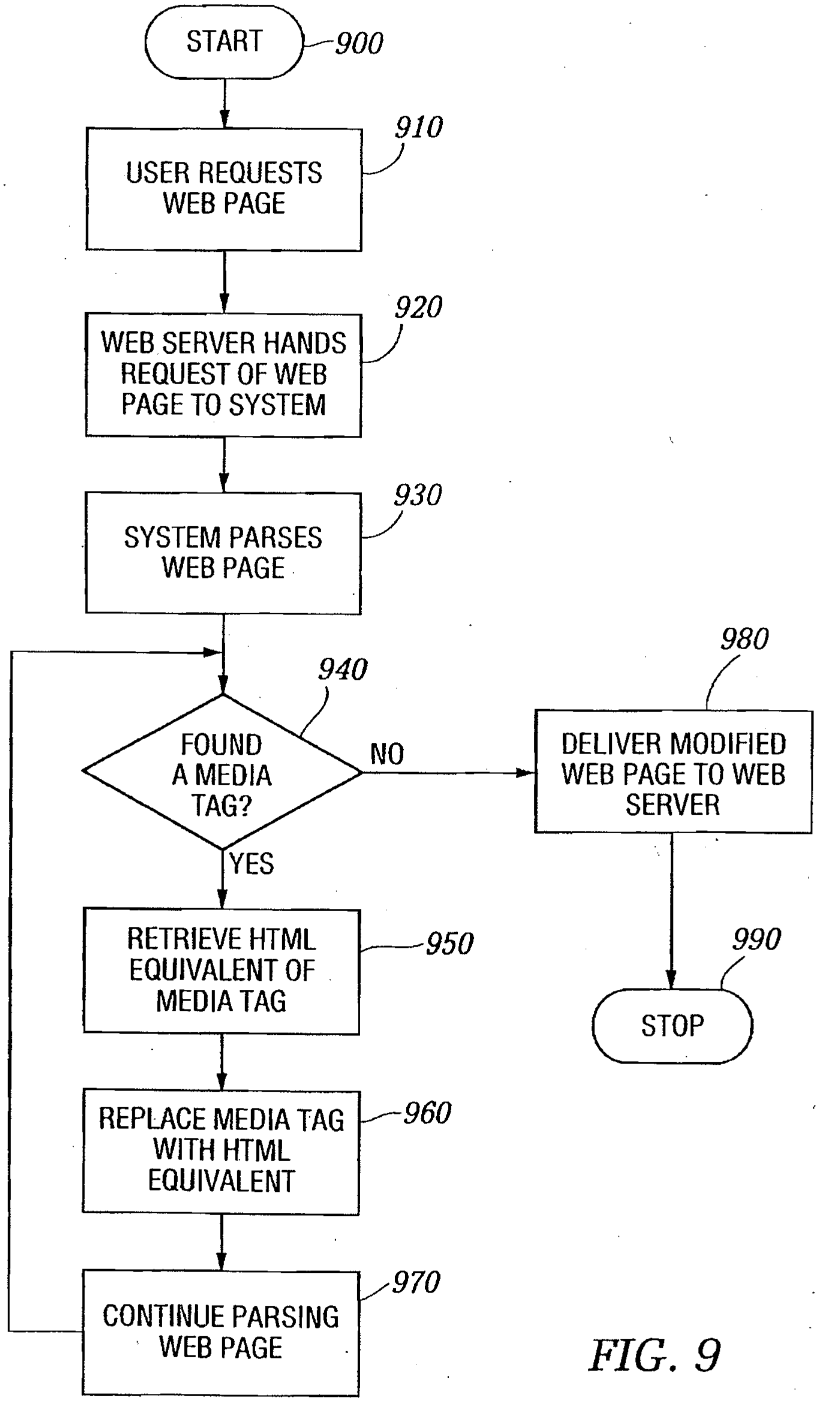

START
900
USER REQUESTS WEB PAGE
910
WEB SERVER HANDS REQUEST OF WEB PAGE TO SYSTEM
920
SYSTEM PARSES WEB PAGE
930
FOUND A MEDIA TAG?
940
NO
YES
DELIVER MODIFIED WEB PAGE TO WEB SERVER
980
RETRIEVE HTML EQUIVALENT OF MEDIA TAG
950
STOP
990
REPLACE MEDIA TAG WITH HTML EQUIVALENT
960
CONTINUE PARSING WEB PAGE
970

*FIG. 9*

MEDIA CREATION FLOWCHART

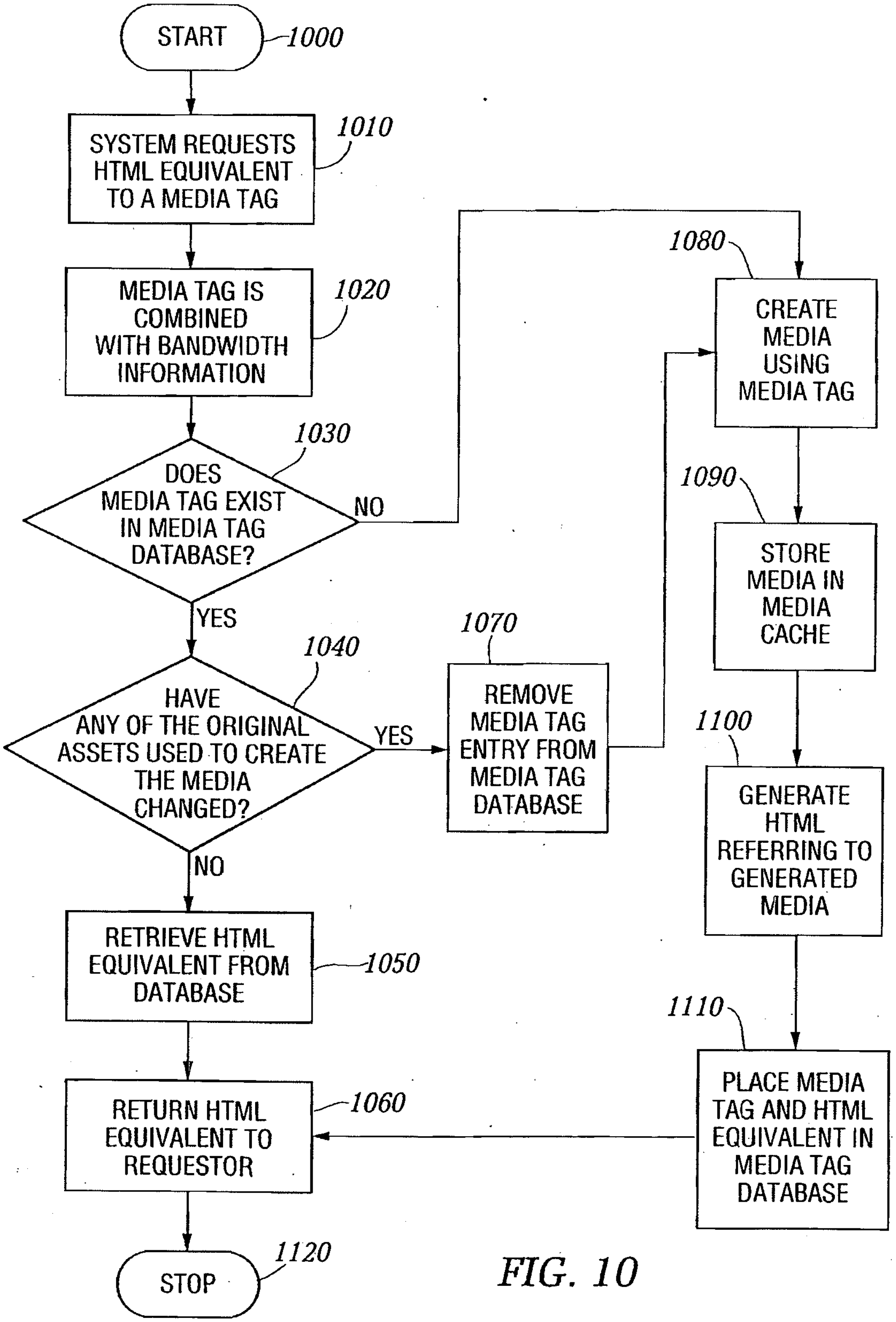
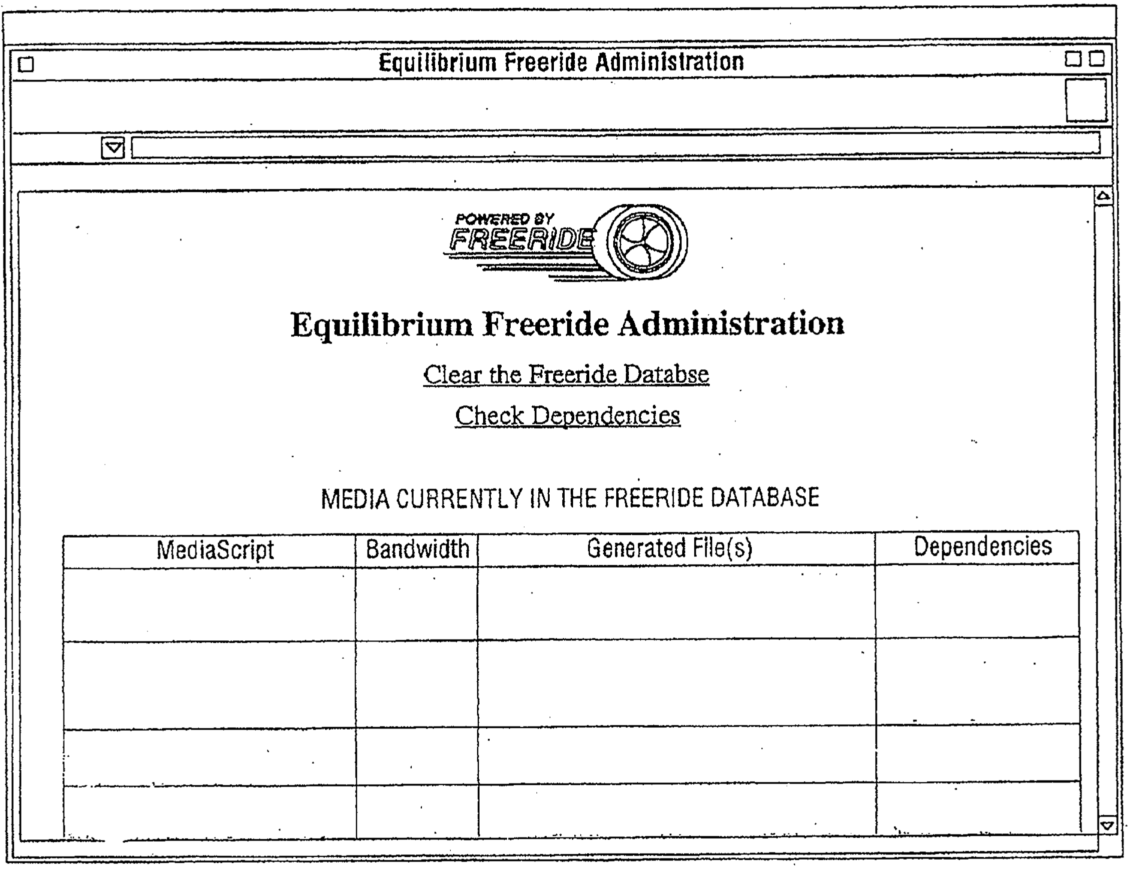

START
1000
SYSTEM REQUESTS HTML EQUIVALENT TO A MEDIA TAG
1010
MEDIA TAG IS COMBINED WITH BANDWIDTH INFORMATION
1020
DOES MEDIA TAG EXIST IN MEDIA TAG DATABASE?
1030
NO
YES
HAVE ANY OF THE ORIGINAL ASSETS USED TO CREATE THE MEDIA CHANGED?
1040
YES
NO
REMOVE MEDIA TAG ENTRY FROM MEDIA TAG DATABASE
1070
CREATE MEDIA USING MEDIA TAG
1080
STORE MEDIA IN MEDIA CACHE
1090
GENERATE HTML REFERRING TO GENERATED MEDIA
1100
PLACE MEDIA TAG AND HTML EQUIVALENT IN MEDIA TAG DATABASE
1110
RETRIEVE HTML EQUIVALENT FROM DATABASE
1050
RETURN HTML EQUIVALENT TO REQUESTOR
1060
STOP
1120

FIG. 10



Equilibrium Freeride Administration

## Equilibrium Freeride Administration

Clear the Freeride Databse

Check Dependencies

MEDIA CURRENTLY IN THE FREERIDE DATABASE

| MediaScript | Bandwidth | Generated File(s) | Dependencies |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |

FIG. 11

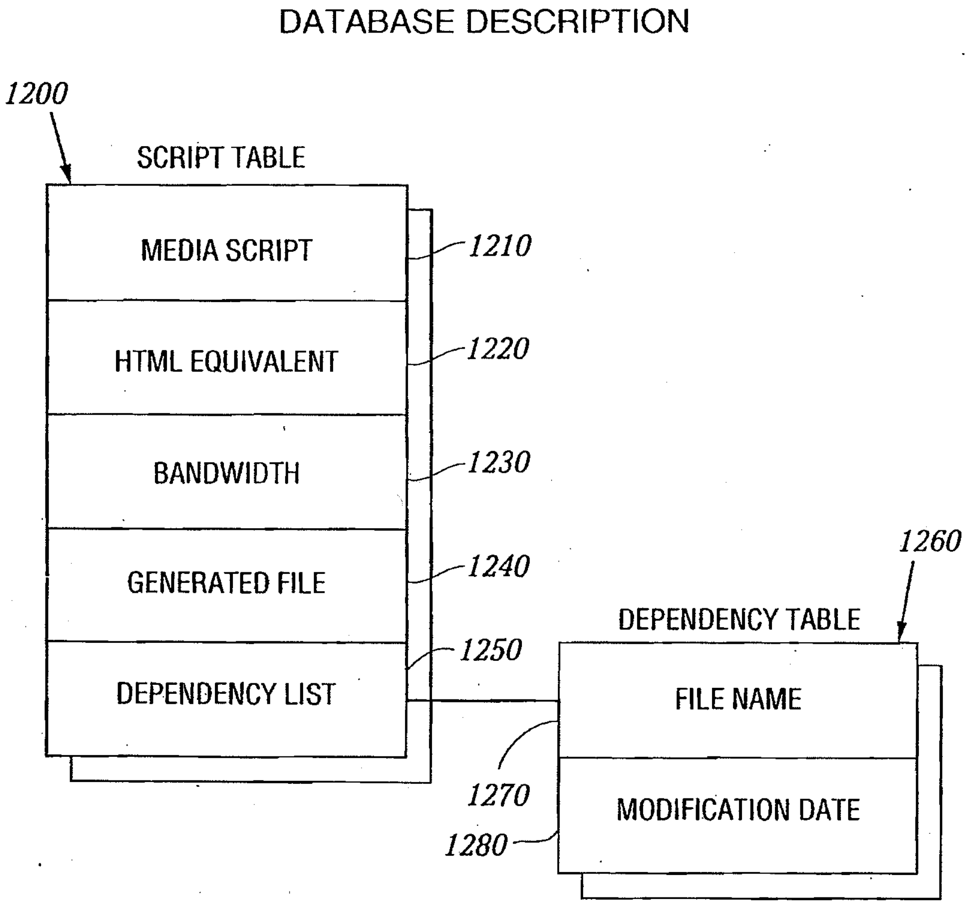

DATABASE DESCRIPTION
1200
SCRIPT TABLE
MEDIA SCRIPT
1210
HTML EQUIVALENT
1220
BANDWIDTH
1230
GENERATED FILE
1240
DEPENDENCY LIST
1250
1260
DEPENDENCY TABLE
FILE NAME
1270
MODIFICATION DATE
1280

FIG. 12

ORIGINAL IMAGES

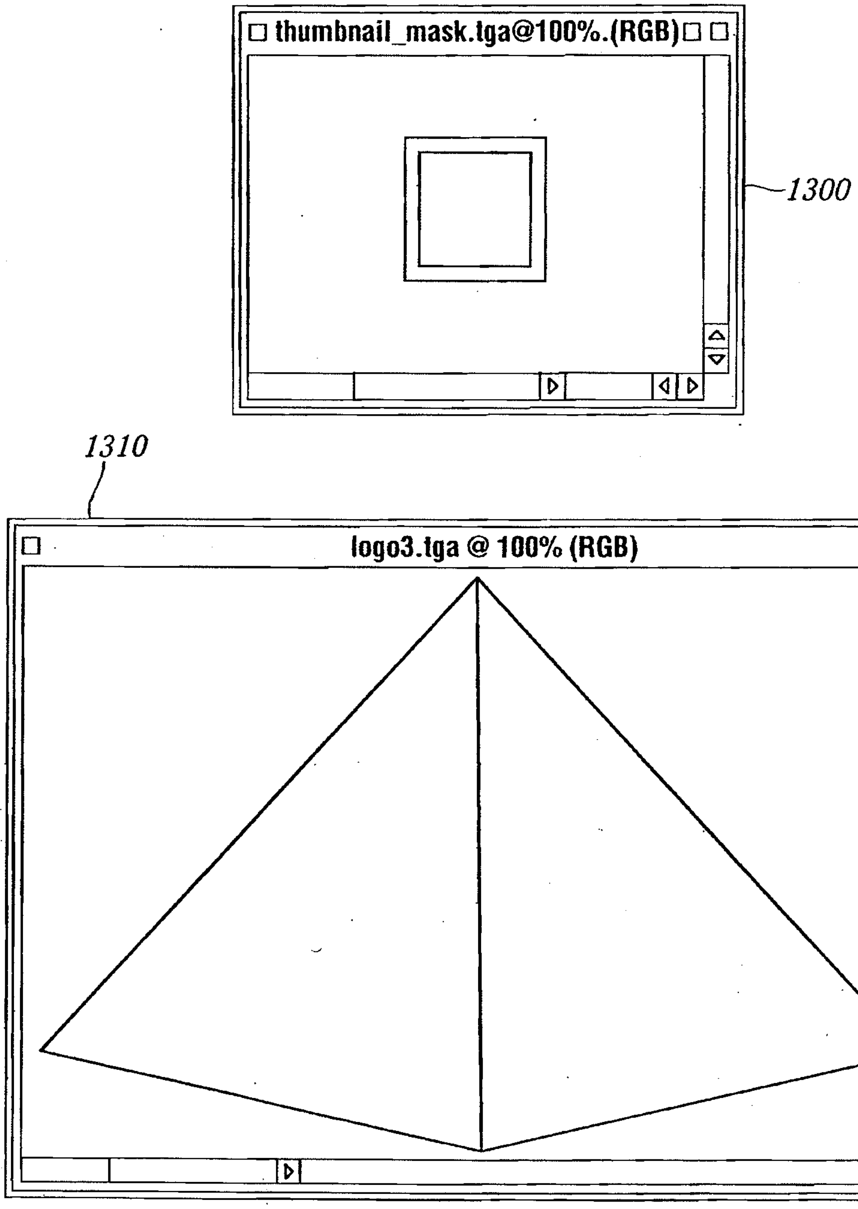

□ thumbnail_mask.tga@100%.(RGB) □ □
1300
1310
□ logo3.tga @ 100% (RGB) □ □

*FIG.13*

HTML DOCUMENT WITH PROPRIETARY TAG

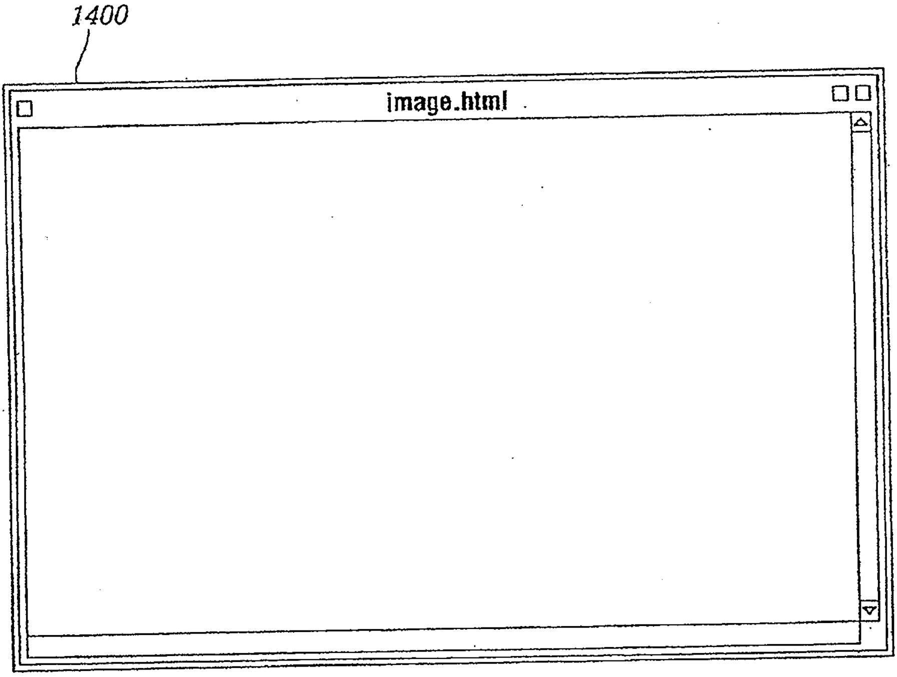


FIG.14

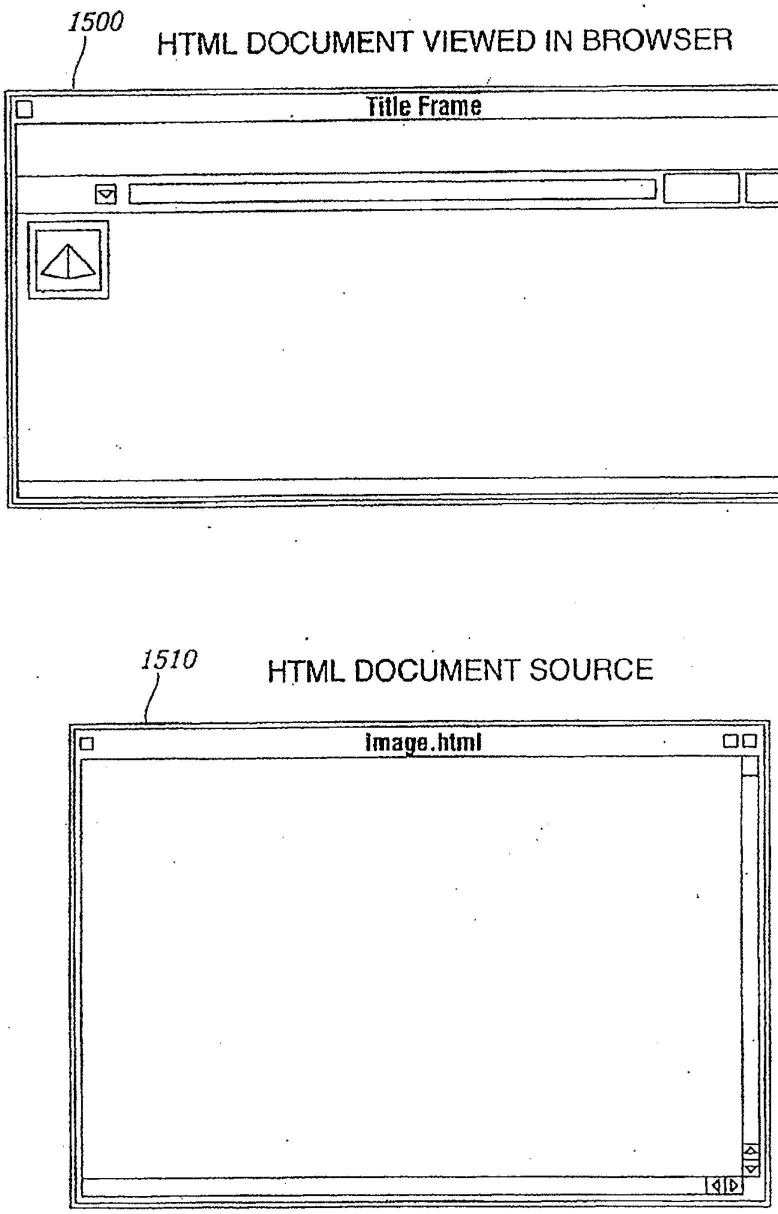

1500
HTML DOCUMENT VIEWED IN BROWSER
Title Frame
1510
HTML DOCUMENT SOURCE
image.html

*FIG.15*

GENERATED GIF IMAGE

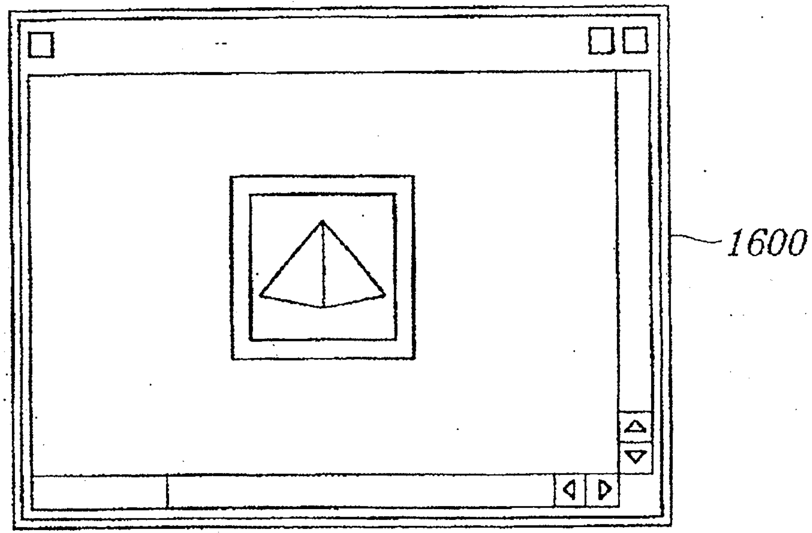


FIG.16

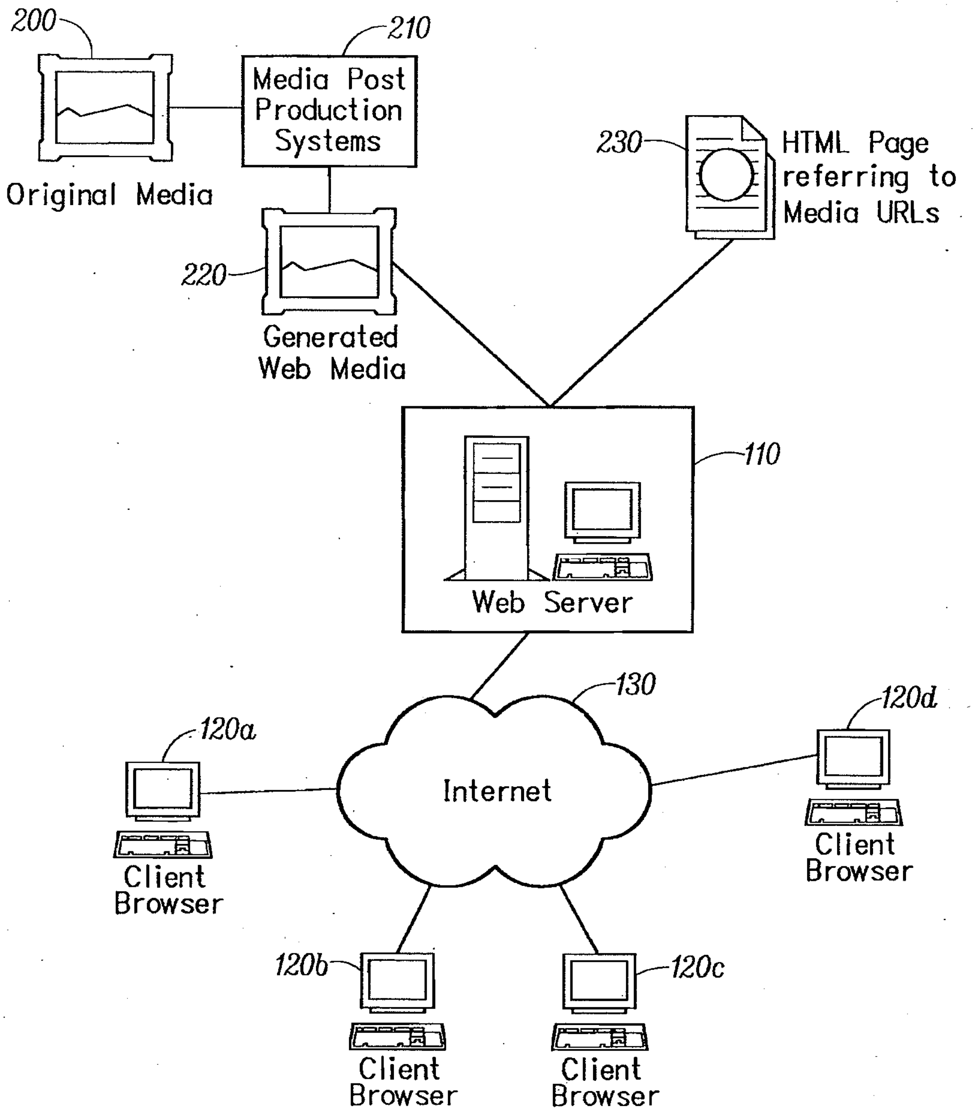

200
210
Media Post Production Systems
Original Media
230
HTML Page referring to Media URLs
220
Generated Web Media
110
Web Server
130
Internet
120a
Client Browser
120d
Client Browser
120b
Client Browser
120c
Client Browser

*FIG. 17*

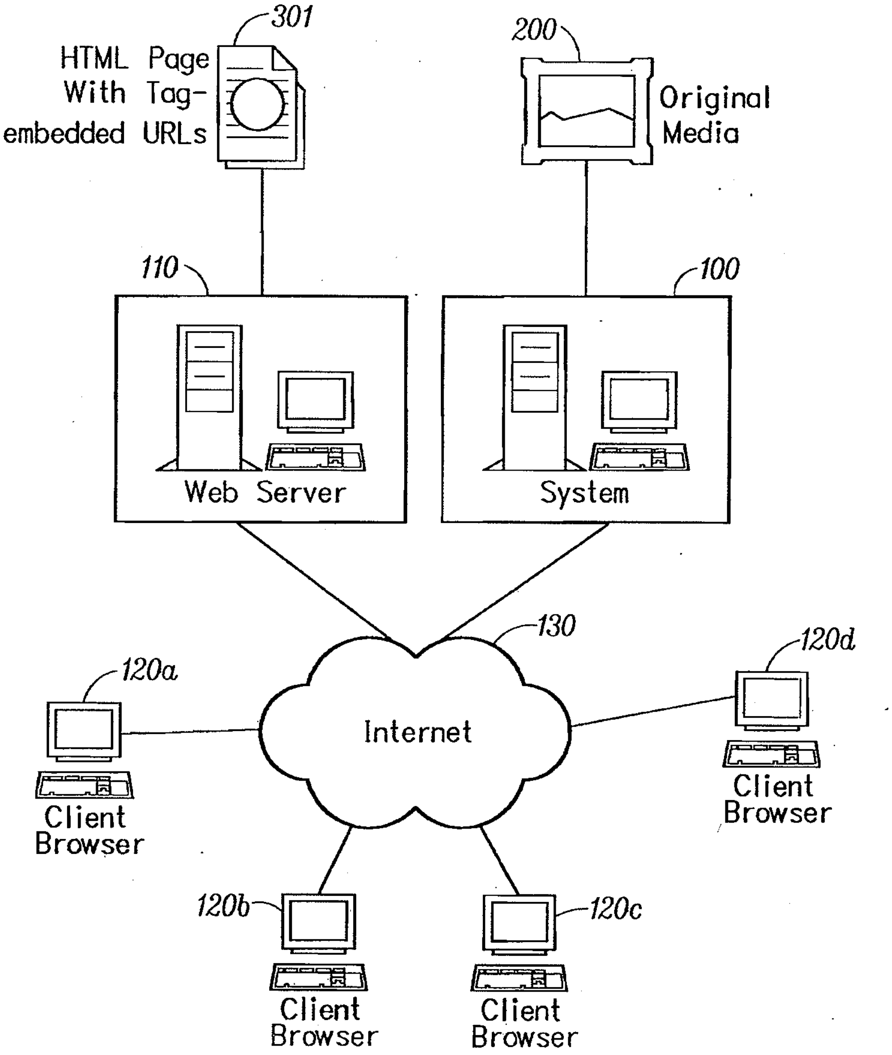

301
HTML Page With Tag-embedded URLs
200
Original Media
110
Web Server
100
System
130
Internet
120a
Client Browser
120b
Client Browser
120c
Client Browser
120d
Client Browser

FIG. 18

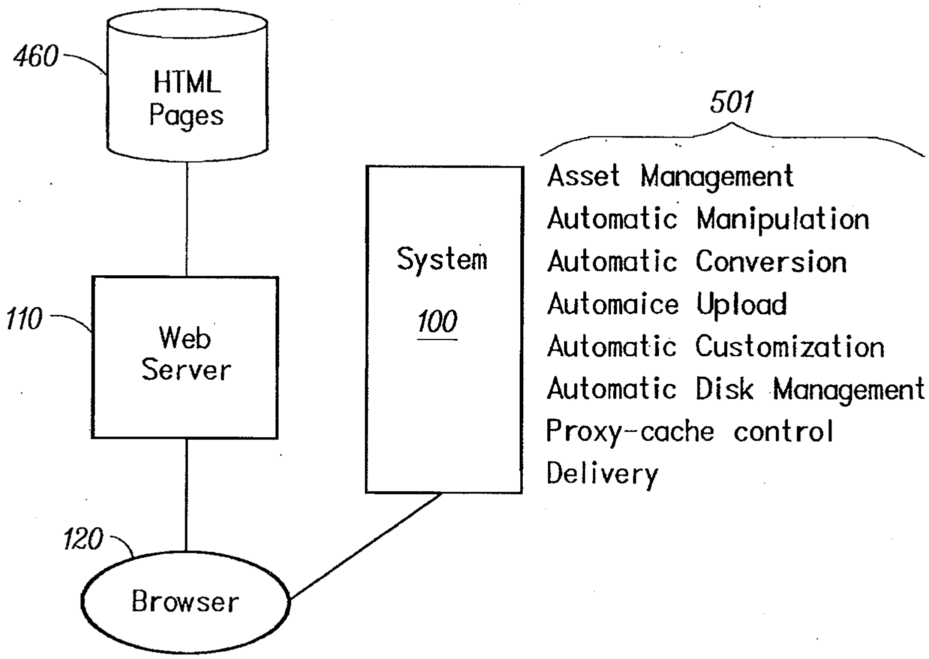

460
HTML Pages
110
Web Server
120
Browser
System
100
501
Asset Management
Automatic Manipulation
Automatic Conversion
Automaice Upload
Automatic Customization
Automatic Disk Management
Proxy-cache control
Delivery

*FIG. 19*

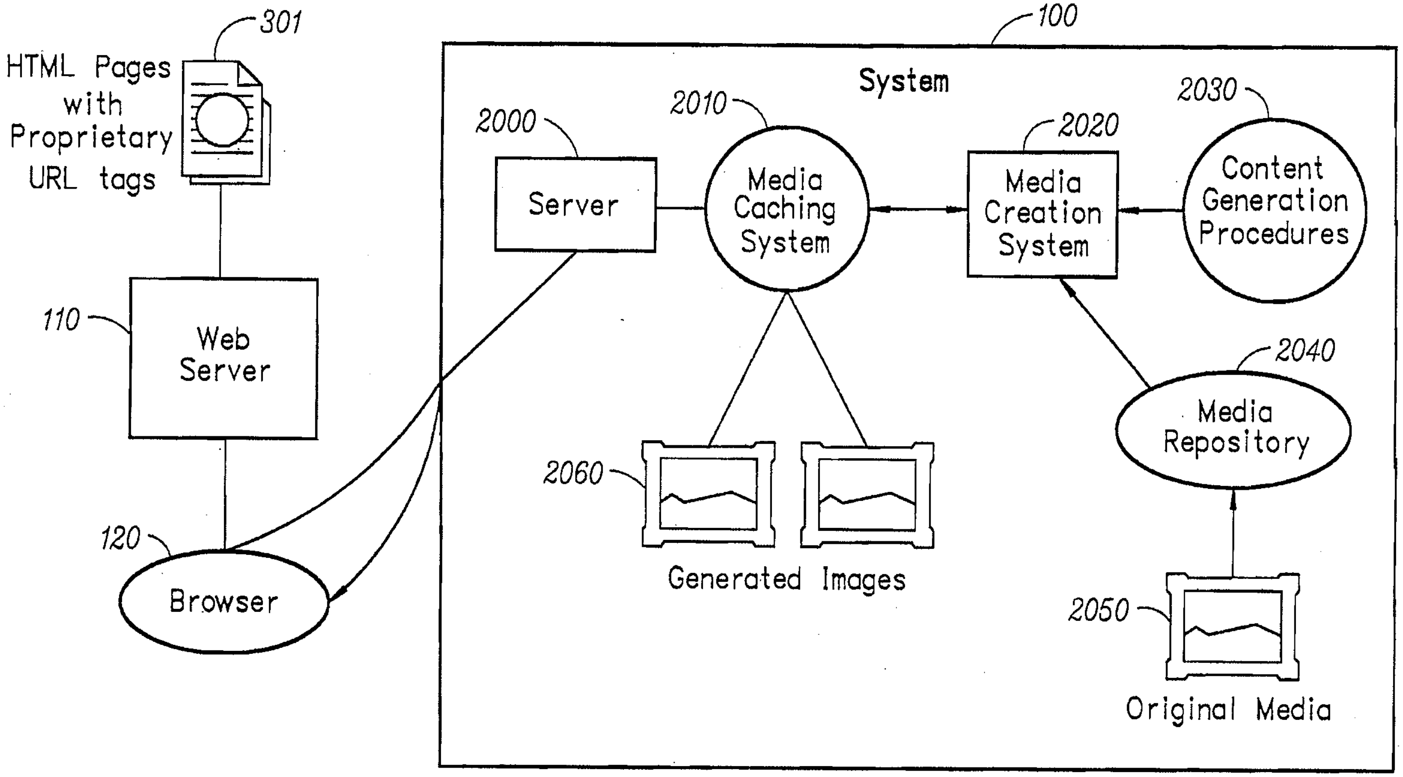

301
HTML Pages with Proprietary URL tags
110
Web Server
120
Browser
100
System
2000
Server
2010
Media Caching System
2020
Media Creation System
2030
Content Generation Procedures
2040
Media Repository
2050
Original Media
2060
Generated Images

FIG. 20

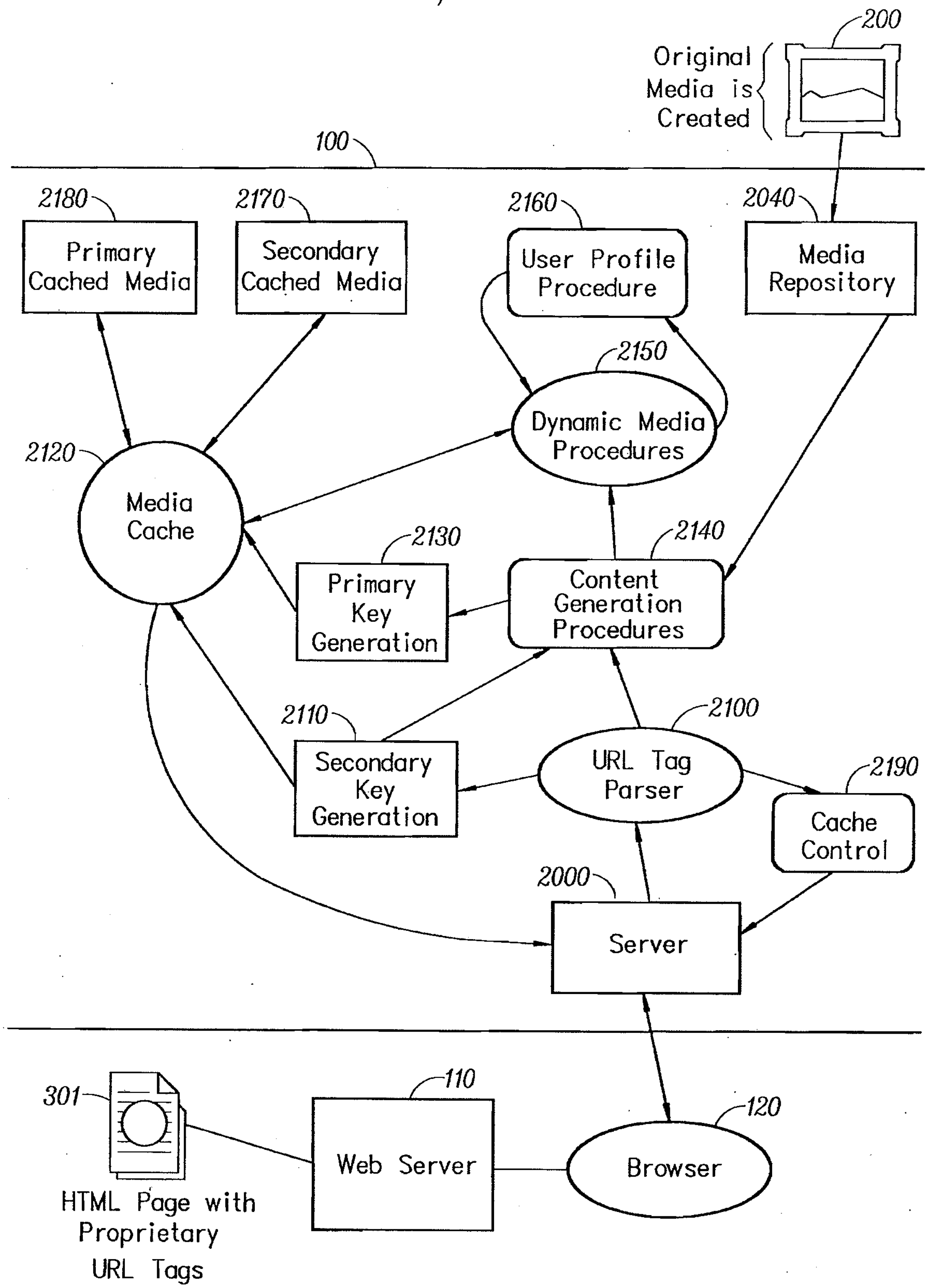

200
Original Media is Created
100
2180
Primary Cached Media
2170
Secondary Cached Media
2160
User Profile Procedure
2040
Media Repository
2150
Dynamic Media Procedures
2120
Media Cache
2130
Primary Key Generation
2140
Content Generation Procedures
2110
Secondary Key Generation
2100
URL Tag Parser
2190
Cache Control
2000
Server
301
110
Web Server
120
Browser
HTML Page with Proprietary URL Tags

FIG. 21

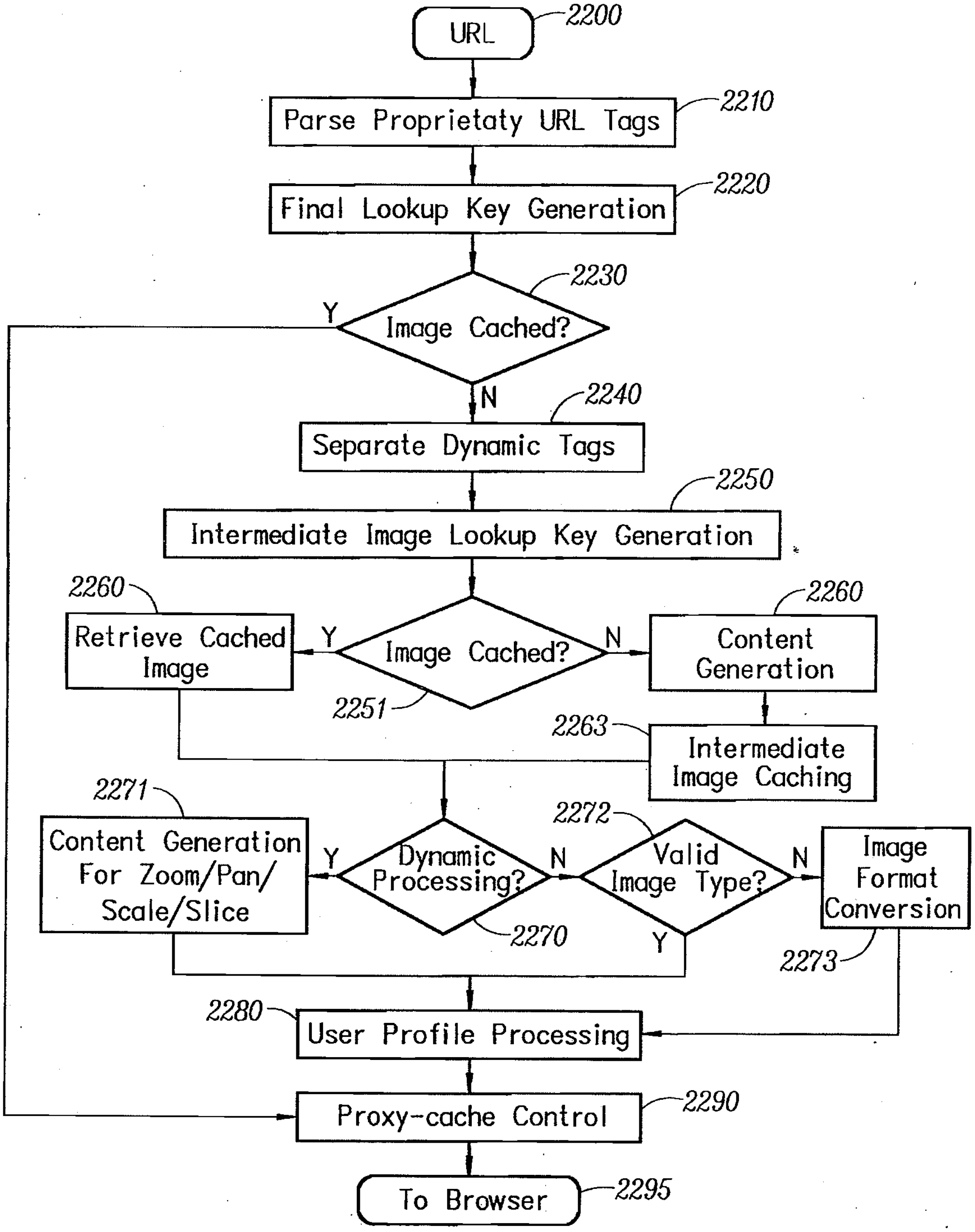

URL
2200
Parse Proprietaty URL Tags
2210
Final Lookup Key Generation
2220
Image Cached?
2230
Y
N
Separate Dynamic Tags
2240
Intermediate Image Lookup Key Generation
2250
2260
Retrieve Cached Image
Y
Image Cached?
N
2251
2260
Content Generation
2263
Intermediate Image Caching
2271
Content Generation For Zoom/Pan/ Scale/Slice
Y
Dynamic Processing?
N
2270
2272
Valid Image Type?
N
Y
Image Format Conversion
2273
2280
User Profile Processing
Proxy-cache Control
2290
To Browser
2295

FIG. 22

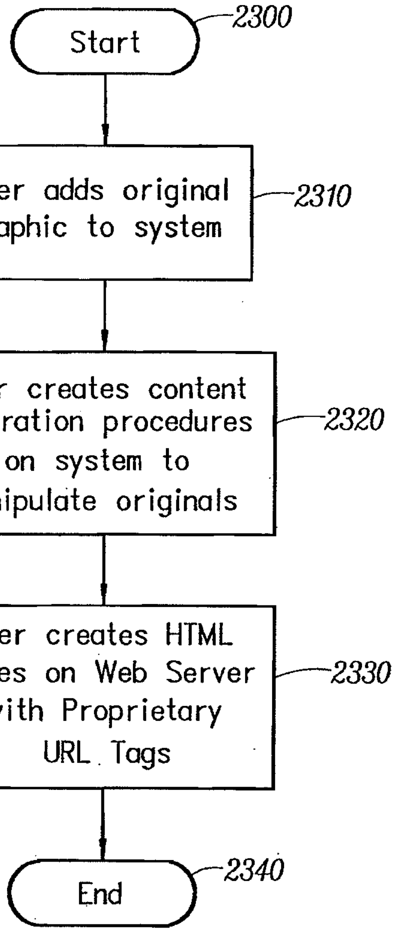

Start
2300
User adds original graphic to system
2310
User creates content generation procedures on system to manipulate originals
2320
User creates HTML pages on Web Server with Proprietary URL Tags
2330
End
2340

*FIG. 23*

# DECLARATION FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

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

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

## Automated Media Delivery System

the specification of which (check one) _X_ is attached hereto, or __ was filed on ____________
as Application Serial No. __________ and was amended on ____________ (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

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

---

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

| | Priority Claimed | |
|---|---|---|
| Prior Foreign Application(s) | Yes | No |
| ________________________ | ______ | ______ |
| Number Country Day/Month/Year Filed | | |

---

**POWER OF ATTORNEY**: As a named inventor, I hereby appoint the attorney(s) and/or agent(s) associated with Customer Number **22862** to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

**DIRECT CORRESPONDENCE TO**: the address associated with Customer Number **22862**.

---

I hereby claim the benefit under Title 35, United States code, Section 119(e)/120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulation, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

| | | |
|---|---|---|
| 12/173,747 | 7/15/2008 | Pending |
| Application Ser. No. | Filing Date | Status: Patented, Pending, Abandoned |
| 11/269,916 | 11/7/2005 | Abandoned |
| Application Ser. No. | Filing Date | Status: Patented, Pending, Abandoned |
| 09/929,904 | 8/14/2001 | Patented |
| Application Ser. No. | Filing Date | Status: Patented, Pending, Abandoned |
| 09/425,326 | 10/21/1999 | Patented |
| Application Ser. No. | Filing Date | Status: Patented, Pending, Abandoned |

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

Full name of First Inventor: Sean Barger
Inventor's signature ______ Date: 1/29/10
Residence 222 Marguerite Avenue, Mill Valley, CA 94941
Mailing Address same
Citizenship United States of America

Full name of Second Inventor: Steve Johnson
Inventor's signature ______ Date: 1/29/10
Residence 2 Hill Street, Mill Valley, CA 94941
Mailing Address same
Citizenship United States of America

Full name of Third Inventor: Matt Butler
Inventor's signature M Butler Date: 2/3/2010
Residence 13405 SW Park Way, Beaverton, OR 97005
Mailing Address same
Citizenship United States of America

Full name of Fourth Inventor: Jerry Destremps
Inventor's signature Date: 2/9/2010
Residence 450 Sherwood Dr. Apt. 104, Sausalito, CA 94965
Mailing Address same
Citizenship United States of America

Full name of Fifth Inventor: David Pochron
Inventor's signature David Pochron Date: 2/13/10
Residence 1514 Hillside Road, Cambridge, WI 53523
Mailing Address same
Citizenship United States of America

Full name of Sixth Inventor: Trent Brown
Inventor's signature Date: 1/29/10
Residence 4 Alpine Terrace, San Anselmo, CA 94960
Mailing Address same
Citizenship United States of America

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

| | | |
|---|---|---|
| First Named Inventor | : | Sean BARGER |
| Serial No. | : | Application Submitted Herewith |
| Filed | : | February 22, 2010 |
| Art Unit | : | Unassigned |
| Confirmation Number | : | Application Submitted Herewith |
| Examiner | : | Unassigned |
| Title | : | Automated Media Delivery System |
| Attorney Docket No. | : | EQUI0016D2 |

February 26, 2010

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

## INFORMATION DISCLOSURE STATEMENT

Examiner:

This Information Disclosure Statement is submitted under 37 CFR 1.97(b).

The Commissioner is authorized to charge any additional fees or credit any overpayment to Deposit Account No. 07-1445 (Order No. EQUI0016D2).

Applicant(s) submit herewith PTO Form 1449 (Modified) -- Information Disclosure Citation together with copies of patents, publications or other information of which applicant(s) are aware, which applicant(s) believe(s) may be material to the examination of this application and for which there may be a duty to disclose in accordance with 37 CFR 1.56.

Each of the Claims currently pending in this Application is copied from United States patent publication no. 2009/0254672 to Zhang (hereinafter referred to as "Zhang").

37 C.F.R. § 1.607 and MPEP § 2001.06(d) require an Applicant to identify the patent and the numbers of the patent claims of the copied patent application. The Applicants identify how the Claims are mapped to Zhang as follows:

Claim 1 corresponds to Claim 1 of Zhang;
Claim 2 corresponds to Claim 2 of Zhang;
Claim 3 corresponds to Claim 3 of Zhang;
Claim 4 corresponds to Claim 4 of Zhang;
Claim 5 corresponds to Claim 8 of Zhang;
Claim 6 corresponds to Claim 9 of Zhang;
Claim 7 corresponds to Claim 10 of Zhang;
Claim 8 corresponds to Claim 11 of Zhang;
Claim 9 corresponds to Claim 12 of Zhang;
Claim 10 corresponds to Claim 13 of Zhang; and
Claim 11 corresponds to Claim 14 of Zhang.

It is requested that the information disclosed herein be made of record in this application.

Respectfully Submitted,

Michael A. Glenn
Reg. No. 30,176

Customer No. 22862

| | |
|---|---|
| **Form 1449 (Modified)** | **Serial No.: Application Submitted Herewith** |
| | **Atty. Docket No.: EQUI0016D2** |
| **Information Disclosure** | **Applicant: Sean BARGER, et al.** |
| **Statement By Applicant** | **Art Unit: Application Submitted Herewith** |
| | **Confirmation No.: Application Submitted Herewith** |
| (Use Several Sheets if Necessary) | **Filing Date:** |

### U.S. Patent Documents

| Examiner Initials | No. | Patent No. | Issue Date | Patentee | Filing Date |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

### Published U.S, Patent Application

| Examiner Initials | No. | Publication No. | Publication Date | Applicant |
|---|---|---|---|---|
| | 1 | 2009/0254672 | 10/08/2009 | Zhang |
| | | | | |
| | | | | |

### Foreign Patent or Published Foreign Patent Application

| Examiner Initials | No. | Document No. | Publication Date | Applicant |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

### Non-Patent Literature Documents

| Examiner Initials | No. | Author, Title, Date, Place (e.g. Journal) of Publication |
|---|---|---|
| | | |
| | | |
| | | |
| | | |

Examiner's Signature:______________________________ Date:_______________

Examiner: Initial citation considered. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

## Electronic Patent Application Fee Transmittal

| | |
|---|---|
| **Application Number:** | |
| **Filing Date:** | |
| **Title of Invention:** | Automated Media Delivery System |
| **First Named Inventor/Applicant Name:** | Sean BARGER |
| **Filer:** | Michael Glenn/Christine Ortt |
| **Attorney Docket Number:** | EQUI0016D2 |

Filed as Small Entity

### Utility under 35 USC 111(a) Filing Fees

| Description | Fee Code | Quantity | Amount | Sub-Total in USD($) |
|---|---|---|---|---|
| **Basic Filing:** | | | | |
| Utility filing Fee (Electronic filing) | 4011 | 1 | 82 | 82 |
| Utility Search Fee | 2111 | 1 | 270 | 270 |
| Utility Examination Fee | 2311 | 1 | 110 | 110 |
| **Pages:** | | | | |
| **Claims:** | | | | |
| **Miscellaneous-Filing:** | | | | |
| **Petition:** | | | | |
| **Patent-Appeals-and-Interference:** | | | | |

| Description | Fee Code | Quantity | Amount | Sub-Total in USD($) |
|---|---|---|---|---|
| **Post-Allowance-and-Post-Issuance:** | | | | |
| **Extension-of-Time:** | | | | |
| **Miscellaneous:** | | | | |
| | | | **Total in USD ($)** | **462** |

# Electronic Acknowledgement Receipt

| | |
|---|---|
| **EFS ID:** | 7098191 |
| **Application Number:** | 12713637 |
| **International Application Number:** | |
| **Confirmation Number:** | 5103 |
| **Title of Invention:** | Automated Media Delivery System |
| **First Named Inventor/Applicant Name:** | Sean BARGER |
| **Customer Number:** | 22862 |
| **Filer:** | Michael Glenn/Christine Ortt |
| **Filer Authorized By:** | Michael Glenn |
| **Attorney Docket Number:** | EQUI0016D2 |
| **Receipt Date:** | 26-FEB-2010 |
| **Filing Date:** | |
| **Time Stamp:** | 15:01:31 |
| **Application Type:** | Utility under 35 USC 111(a) |

## Payment information:

| | |
|---|---|
| Submitted with Payment | yes |
| Payment Type | Deposit Account |
| Payment was successfully received in RAM | $462 |
| RAM confirmation Number | 1345 |
| Deposit Account | 071445 |
| Authorized User | |

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

## File Listing:

| Document Number | Document Description | File Name | File Size(Bytes)/ Message Digest | Multi Part /.zip | Pages (if appl.) |
|---|---|---|---|---|---|
| 1 | | 2010-2-26_Application_EQUI0016D2.pdf | 2404204<br>9c2a97f78de767c81c56f1608f357924f61b0bf6 | yes | 70 |

**Multipart Description/PDF files in .zip description**

| Document Description | Start | End |
|---|---|---|
| Transmittal of New Application | 1 | 1 |
| Specification | 2 | 37 |
| Claims | 38 | 40 |
| Abstract | 41 | 41 |
| Drawings-only black and white line drawings | 42 | 64 |
| Oath or Declaration filed | 65 | 67 |
| Transmittal Letter | 68 | 69 |
| Information Disclosure Statement (IDS) Filed (SB/08) | 70 | 70 |

**Warnings:**

**Information:**

| Document Number | Document Description | File Name | File Size(Bytes)/ Message Digest | Multi Part /.zip | Pages (if appl.) |
|---|---|---|---|---|---|
| 2 | Fee Worksheet (PTO-875) | fee-info.pdf | 32849<br>5c2bc724af4fabfd8551f2efdc5fd8c32d1b99e3 | no | 2 |

**Warnings:**

**Information:**

| | |
|---|---|
| **Total Files Size (in bytes):** | 2437053 |

**This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.**

**New Applications Under 35 U.S.C. 111**
**If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.**

**National Stage of an International Application under 35 U.S.C. 371**
**If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.**

**New International Application Filed with the USPTO as a Receiving Office**
**If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.**

PTO/SB/06 (12-04)

**Date:** 02/26/10

Approved for use through 7/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

| **PATENT APPLICATION FEE DETERMINATION RECORD**<br>Substitute for Form PTO-875 | Application or Docket Number<br>12/713,637 |
|---|---|

## APPLICATION AS FILED – PART I

| | (Column 1) | (Column 2) | SMALL ENTITY | | OR | OTHER THAN SMALL ENTITY | |
|---|---|---|---|---|---|---|---|
| FOR | NUMBER FILED | NUMBER EXTRA | RATE ($) | FEE ($) | | RATE ($) | FEE ($) |
| BASIC FEE<br>(37 CFR 1.16(a), (b), or (c)) | N/A | N/A | N/A | 82 | | N/A | |
| SEARCH FEE<br>(37 CFR 1.16(k), (i), or (m)) | N/A | N/A | N/A | 270 | | N/A | |
| EXAMINATION FEE<br>(37 CFR 1.16(o), (p), or (q)) | N/A | N/A | N/A | 110 | | N/A | |
| TOTAL CLAIMS<br>(37 CFR 1.16(i)) | 11 minus 20 = | | x$26 | | OR | x$52 | |
| INDEPENDENT CLAIMS<br>(37 CFR 1.16(h)) | 2 minus 3 = | * | x$110 | | | x$220 | |
| APPLICATION SIZE FEE<br>(37 CFR 1.16(s)) | If the specification and drawings exceed 100 sheets of paper, the application size fee due is $260 ($130 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR | | | | | | |
| MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j)) | | | 195 | | | 390 | |
| | | | TOTAL | 462 | | TOTAL | |

* If the difference in column 1 is less than zero, enter "0" in column 2.

## APPLICATION AS AMENDED – PART II

| | | (Column 1) | | (Column 2) | (Column 3) | SMALL ENTITY | | OR | OTHER THAN SMALL ENTITY | |
|---|---|---|---|---|---|---|---|---|---|---|
| AMENDMENT A | | CLAIMS REMAINING AFTER AMENDMENT | | HIGHEST NUMBER PREVIOUSLY PAID FOR | PRESENT EXTRA | RATE ($) | ADDI-TIONAL FEE ($) | | RATE ($) | ADDI-TIONAL FEE ($) |
| | Total<br>(37 CFR 1.16(i)) | * | Minus | ** | = | X = | | OR | X = | |
| | Independent<br>(37 CFR 1.16(h)) | * | Minus | *** | = | X = | | OR | X = | |
| | Application Size Fee (37 CFR 1.16(s)) | | | | | | | | | |
| | FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j)) | | | | | N/A | | OR | N/A | |
| | | | | | | TOTAL ADD'T FEE | | OR | TOTAL ADD'T FEE | |

| | | (Column 1) | | (Column 2) | (Column 3) | | | OR | | |
|---|---|---|---|---|---|---|---|---|---|---|
| AMENDMENT B | | CLAIMS REMAINING AFTER AMENDMENT | | HIGHEST NUMBER PREVIOUSLY PAID FOR | PRESENT EXTRA | RATE ($) | ADDI-TIONAL FEE ($) | | RATE ($) | ADDI-TIONAL FEE ($) |
| | Total<br>(37 CFR 1.16(i)) | * | Minus | ** | = | X = | | OR | X = | |
| | Independent<br>(37 CFR 1.16(h)) | * | Minus | *** | = | X = | | OR | X = | |
| | Application Size Fee (37 CFR 1.16(s)) | | | | | | | | | |
| | FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j)) | | | | | N/A | | OR | N/A | |
| | | | | | | TOTAL ADD'T FEE | | OR | TOTAL ADD'T FEE | |

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Paten and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

*If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.*

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NUMBER | FILING or 371(c) DATE | GRP ART UNIT | FIL FEE REC'D | ATTY.DOCKET.NO | TOT CLAIMS | IND CLAIMS |
|---|---|---|---|---|---|---|
| 12/713,637 | 02/26/2010 | 2447 | 462 | EQUI0016D2 | 11 | 2 |

**CONFIRMATION NO. 5103**

**FILING RECEIPT**

*OC000000040522439*

22862
GLENN PATENT GROUP
3475 EDISON WAY, SUITE L
MENLO PARK, CA 94025

Date Mailed: 03/12/2010

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

**Applicant(s)**

Sean BARGER, Mill Valley, CA;
Steve Johnson, Mill Valley, CA;
Matt Butler, Beaverton, OR;
Jerry Destremps, Sausalito, CA;
David Pochron, Cambridge, WI;
Trent Brown, San Anselmo, CA;

**Power of Attorney:** The patent practitioners associated with Customer Number 22862

**Domestic Priority data as claimed by applicant**

This application is a DIV of 12/173,747 07/15/2008
which is a DIV of 11/269,916 11/07/2005 ABN
which is a CIP of 09/929,904 08/14/2001 PAT 6,964,009
which is a CON of 09/425,326 10/21/1999 PAT 6,792,575

**Foreign Applications**

**If Required, Foreign Filing License Granted:** 03/09/2010

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 12/713,637**

**Projected Publication Date:** 06/17/2010

**Non-Publication Request:** No

**Early Publication Request:** No
**** SMALL ENTITY ****

**Title**

Automated Media Delivery System

**Preliminary Class**

709

## PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

## LICENSE FOR FOREIGN FILING UNDER

## Title 35, United States Code, Section 184

## Title 37, Code of Federal Regulations, 5.11 & 5.15

**GRANTED**

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

**NOT GRANTED**

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NUMBER | FILING OR 371(C) DATE | FIRST NAMED APPLICANT | ATTY. DOCKET NO./TITLE |
|---|---|---|---|
| 12/713,637 | 02/26/2010 | Sean BARGER | EQUI0016D2 |

**CONFIRMATION NO. 5103**

**PUBLICATION NOTICE**

*OC000000042145456*

22862
GLENN PATENT GROUP
3475 EDISON WAY, SUITE L
MENLO PARK, CA 94025

**Title:**Automated Media Delivery System

**Publication No.**US-2010-0153495-A1
**Publication Date:**06/17/2010

## NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO's publically available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently http://www.uspto.gov/patft/.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Office of Public Records. The Office of Public Records can be reached by telephone at (703) 308-9726 or (800) 972-6382, by facsimile at (703) 305-8759, by mail addressed to the United States Patent and Trademark Office, Office of Public Records, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently http://pair.uspto.gov/. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Office of Data Managment, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 12/713,637 | 02/26/2010 | Sean BARGER | EQUI0016D2 | 5103 |

22862 7590 10/19/2010

GLENN PATENT GROUP
3475 EDISON WAY, SUITE L
MENLO PARK, CA 94025

| EXAMINER | |
|---|---|
| MURPHY, CHARLES C | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2455 | |

| NOTIFICATION DATE | DELIVERY MODE |
|---|---|
| 10/19/2010 | ELECTRONIC |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

eptomatters@glenn-law.com

| | | | |
|---|---|---|---|
| ***Office Action Summary*** | **Application No.**<br>12/713,637 | **Applicant(s)**<br>BARGER ET AL. | |
| | **Examiner**<br>CHARLES MURPHY | **Art Unit**<br>2455 | |

***-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --***

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☐ Responsive to communication(s) filed on _26 February 2010_.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

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

**Disposition of Claims**

4)☒ Claim(s) _1-11_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-11_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _26 February 2010_ is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

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

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _02/26/2010_.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

## DETAILED ACTION

This communication is in response to application file on 02/26/2010 in which claims 1-11 are presented for examination.

### Information Disclosure Statement

The information disclosure statement (IDS) submitted was filed on 02/26/2010. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the examiner

### Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claim 1-11 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

As per claim 1, the claim limitation recites "computer program product". However, the usage of the phrase "computer program product" is broad enough to include both "non-transitory" and "transitory" (moving electrons, etc) media. The specification does not clearly limit the utilization of a non-transitory computer readable medium and, thus does not constitute functional descriptive material. Therefore, when the broadest reasonable interpretation of a claim covers a signal per se, the claim must be rejected under 35 U.S.C. § 101 as covering non-statutory subject matter. See *In re Nuijten,* 500 F.3d

1346, 1356-57 (Fed. Cir. 2007) (transitory embodiments are not directed to statutory subject matter).

The USPTO recognizes that applicants may have claims directed to computer readable media that cover signals *per se*, which the USPTO must reject under 35 U.S.C. § 101 as covering both non-statutory subject matter and statutory subject matter. In an effort to assist the patent community in overcoming a rejection or potential rejection under 35 U.S.C. § 101 in this situation, the USPTO suggests the following approach. A claim drawn to such a computer readable medium that covers both ***transitory*** and ***non-transitory*** embodiments may be amended to narrow the claim to cover only statutory embodiments to avoid a rejection under 35 U.S.C. § 101 by adding the limitation "***non-transitory***" to the claim. *Cf. Animals - Patentability*, 1077 *Off. Gaz. Pat. Office* 24 (April 21, 1987) (suggesting that applicants add the limitation "non-human" to a claim covering a multi-cellular organism to avoid a rejection under 35 U.S.C. § 101). Such an amendment would typically not raise the issue of new matter, even when the specification is silent because the broadest reasonable interpretation relies on the ordinary and customary meaning that includes signals *per se*. The limited situations in which such an amendment could raise issues of new matter occur, for example, when the specification does not support a non-transitory embodiment because a signal *per se* is the only viable embodiment such that the amended claim is impermissibly broadened beyond the supporting disclosure. *See, e.g., Gentry Gallery, Inc. v. Berkline Corp.*, 134 F.3d 1473 (Fed. Cir. 1998).

The dependent claims included in the statement of rejection but not specifically addressed in the body of the rejection have inherited the deficiencies of their parent claim and have not resolved the deficiencies. Therefore, they are rejected based on the same rationale as applied to their parent claims above.

### *Claim Rejections - 35 USC § 102*

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

1. **Claim 1 is rejected under 35 U.S.C. 102(e) as being anticipated by Bhagwat et al. US Patent 6,563,517 (hereinafter referred to as Bhagwat)**

2. Regarding claim 1, **Bhagwat** teaches an computer program product when executed causes a computer system to perform a method comprising:

   an act of receiving a request for media content to be delivered to a client presentation system;*(**Bhagwat** teaches receiving a request for media to be sent to a client)**(Bhagwat, col. 3, lines 9-15)*** *(**Bhagwat** also teaches using a computer readable medium to implement the invention)**(Bhagwat, col. 14, lines 1-10)***

   after receiving the request, an act of identifying transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system; *(**Bhagwat** teaches using various parameters transcode media before it is delivered to a requesting client)**(Bhagwat, col. 3, lines 9-20; col. 5, lines 39-45)***

an act of transcoding the requested media content in accordance with the identified transcoding parameters; *(**Bhagwat** teaches transcoding the requested media based on the transcoding parameters)**(Bhagwat, col. 3, lines 9-20; col. 5, lines 39-45; col 6, lines 45-50)***

an act of causing the transcoded media content to be delivered to the client presentation system; and *(**Bhagwat** teaches delivering the requested transcoded content to the requesting client))**(Bhagwat, col. 3, lines 9-20; col. 5, lines 39-45; col 6, lines 45-50)***

an act of caching the transcoded media content. *(**Bhagwat** teaches caching a transcoded media file)**(Bhagwat, col. 5, lines 15-25 and 50-57; col 6, lines 45-50)***

## ***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

**3. Claim 2-9 rejected under 35 U.S.C. 103(a) as being unpatentable over Bhagwat et al. US Patent 6,563,517 (hereinafter referred to as Bhagwat)**

**4.** Regarding claim 2, **Bhagwat** further teaches the computer program product of claim 1, wherein:

> the one or more computer-readable media are physical memory and/or storage media. *(**Bhagwat** teaches using a computer readable medium to implement the invention. It would be obvious to one of ordinary skill in the art that a computer readable medium could comprises either a physical memory, such as RAM, or storage media, such a CD or DVD disc)**(Bhagwat, col. 14, lines 1-10)***

**5.** Regarding claim 3, **Bhagwat** further teaches the computer program product of claim 2, wherein:

> the act of receiving a request for media content is performed by receiving the request over the Internet. *(**Bhagwat** teaches a client computer and a*

*web server communicating with each other over the internet)* ***(Bhagwat, col. 3, lines 9-15)***

**6.** Regarding claim 4, **Bhagwat** further teaches the computer program product of claim 2, wherein:

an act of receiving a request for the media content to be delivered to a second client presentation system; *(**Bhagwat** teaches a client computer and a web server communicating with each other over the internet)* ***(Bhagwat, col. 3, lines 9-15)***

an act of determining that the transcoding parameters to be applied to the requested media content prior to delivery to the second client presentation system are the same as the transcoding parameters that were applied to the requested media content prior to delivery to the first client presentation system; and*(**Bhagwat** teaches first checking a cache after a request is recieved by a client for a media object. If the media object transcoded with the required parameters is cached then the cached media object is provided to the requesting client.)****(Bhagwat, col. 5, lines 50-67)***

**7.** Regarding claim 5, **Bhagwat** further teaches the computer program product of claim 2, wherein:

the requested media content is video media. *(**Bhagwat** teaches a client computer with the ability to request video content)* ***(Bhagwat, col. 6, lines 19-25)***

**8.** Regarding claim 6, **Bhagwat** further teaches the computer program product of claim 2, wherein:

the identified transcoding parameters includes at least video size parameters. *(**Bhagwat** teaches various transcoding parameters including the resolutions of a media object.)****(Bhagwat, col. 5, lines 15-25; col. 6, lines 5-15)***

**9.** Regarding claim 7, **Bhagwat** further teaches the computer program product of claim 2, comprising:

prior to the act of transcoding, an act of determining that a transcoded representation of the requested media content transcoded using the identified transcoding parameters does not already exist. *(**Bhagwat** teaches checking if a request media file is cached at the requested resolution or transcoding level. These parameters are what is used to transcode the media file.)****(Bhagwat, col. 5, lines 50-60)***

**10.** Regarding claim 8, **Bhagwat** further teaches the computer program product of claim 2, wherein:

the act of identifying transcoding parameters comprises an act of selecting a transcoding profile from a set of pre-existing transcoding profiles. *(**Bhagwat** teaches choosing a set of transcoding parameters based on user's preferences.)**(Bhagwat, col. 13, lines30-40)***

**11.** Regarding claim 9, **Bhagwat** further teaches the computer program product of claim 2, comprising:

an act of reading one or more transcoding parameters from the request; and *(**Bhagwat** teaches a client computer sending a request to a server for a requested media file. The server then, using a user's preferences, transcodes the media file and transmits it back to the requesting client.)**(Bhagwat, col. 13, lines 30-40; )***

an act of identifying the one or more transcoding parameters from the request as being the identified transcoding parameters to be applied. *(**Bhagwat** teaches choosing a set of transcoding parameters based on user's preferences.)**(Bhagwat, col. 13, lines30-40)***

### ***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

**12. Claim 11 rejected under 35 U.S.C. 103(a) as being unpatentable over Bhagwat et al. US Patent 6,563,517 (hereinafter referred to as Bhagwat) in view of Garofalakis et al. US Patent 5,845,279 (hereinafter referred to as Garofalakis)**

**13.** Regarding claim 11, **Bhagwat** teaches a method comprising:

an act of receiving a request for media content to be delivered to a client presentation system for media content;*(**Bhagwat** teaches receiving a request for media to be sent to a client)**(Bhagwat, col. 3, lines 9-15)*** *(**Bhagwat** also teaches using a computer readable medium to implement the invention)**(Bhagwat, col. 14, lines 1-10)***

after receiving the request, an act of identifying transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system; *(**Bhagwat** teaches using various parameters transcode media before it is delivered to a requesting client)**(Bhagwat, col. 3, lines 9-20; col. 5, lines 39-45)*** an act of determining that the transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system are the same as transcoding parameters that are being applied to the requested media content prior to delivery to another client presentation system; and*(**Bhagwat** teaches checking if a request media file is cached at the requested resolution or transcoding level. These parameters are what is used to transcode the media file.)**(Bhagwat, col. 5, lines 50-60)***

**Bhagwat** does not teach:

an act of delivering the transcoded media content to both client presentation systems concurrently.

In an analogous art, **Garfalakis** teaches:

an act of delivering the transcoded media content to both client presentation systems concurrently.*(**Garofalakis** teaches the ability to deliver media concurrently to multiple systems)**(Garofalakis, col. 2, lines 50-55)***

At the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teachings of **Bhagwat** and **Garfalakis** before him or her, to modify **Bhagwat** to include the teachings of **Garfalakis** because this would create a system that would schedule and transmit various media segments of different popularity and lengths at different times. (**Garfalakis, col. 2, lines 60-63)**

**14. Claim 10 rejected under 35 U.S.C. 103(a) as being unpatentable over Bhagwat et al. US Patent 6,563,517 (hereinafter referred to as Bhagwat) in view of Krishnaswamy et al. US Patent 6,909,708 (hereinafter referred to as Krishnaswamy)**

**15.** Regarding claim 10, **Bhagwat** further teaches the computer program product of claim 2, comprising:

an act of incrementally performing further transcoding of the pre-existing base transcoded version in accordance with a remaining portion of the identified transcoding parameters. *(**Bhagwat** teaches incrementally transcoding a file)**(Bhagwat, col. 12, lines 14-16)***

**Bhagwat** does not teach:

an act of selecting a pre-existing base transcoded version of the requested media content that has been transcoded in accordance with only a portion of the identified transcoding parameters; and

In an analogous art, **Krishnaswamy** teaches:

an act of selecting a pre-existing base transcoded version of the requested media content that has been transcoded in accordance with only a portion of the identified transcoding parameters; and *(**Krishnaswamy** teaches to use a base file that has already been transcoded, and convert the base file into various formats.)**(Krishnaswamy, col. 135, lines 25-35)***

At the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teachings of **Bhagwat** and **Krishnaswamy** before him or her, to modify **Bhagwat** to include the teachings of **Krishnaswamy** because this would create a system where users can manage and control a network while a network operator can maintain network quality and routing selection (**Krishnaswamy, col. 2, lines 60-63)**

### ***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHARLES MURPHY whose telephone number is (571)270-5444. The examiner can normally be reached on 8AM - 5PM Monday - Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Saleh Najjar can be reached on 571-272-4006. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/CHARLES MURPHY/
Examiner, Art Unit 2441
10/14/2010

/saleh najjar/
Supervisory Patent Examiner, Art Unit 2455

|  |  |  |  |
|---|---|---|---|
| ***Notice of References Cited*** | Application/Control No.<br>12/713,637 | Applicant(s)/Patent Under Reexamination<br>BARGER ET AL. | |
| | Examiner<br>CHARLES MURPHY | Art Unit<br>2455 | Page 1 of 1 |

**U.S. PATENT DOCUMENTS**

| * | | Document Number<br>Country Code-Number-Kind Code | Date<br>MM-YYYY | Name | Classification |
|---|---|---|---|---|---|
| * | A | US-5,845,279 | 12-1998 | Garofalakis et al. | 1/1 |
| * | B | US-6,563,517 | 05-2003 | Bhagwat et al. | 715/735 |
| * | C | US-6,909,708 | 06-2005 | Krishnaswamy et al. | 370/352 |
| | D | US- | | | |
| | E | US- | | | |
| | F | US- | | | |
| | G | US- | | | |
| | H | US- | | | |
| | I | US- | | | |
| | J | US- | | | |
| | K | US- | | | |
| | L | US- | | | |
| | M | US- | | | |

**FOREIGN PATENT DOCUMENTS**

| * | | Document Number<br>Country Code-Number-Kind Code | Date<br>MM-YYYY | Country | Name | Classification |
|---|---|---|---|---|---|---|
| | N | | | | | |
| | O | | | | | |
| | P | | | | | |
| | Q | | | | | |
| | R | | | | | |
| | S | | | | | |
| | T | | | | | |

**NON-PATENT DOCUMENTS**

| * | | Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages) |
|---|---|---|
| | U | |
| | V | |
| | W | |
| | X | |

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

| | | |
|---|---|---|
| ***Index of Claims*** | **Application/Control No.**<br>12713637 | **Applicant(s)/Patent Under Reexamination**<br>BARGER ET AL. |
| | **Examiner**<br>CHARLES MURPHY | **Art Unit**<br>2455 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ✓ | **Rejected** | - | **Cancelled** | N | **Non-Elected** | A | **Appeal** |
| = | **Allowed** | ÷ | **Restricted** | I | **Interference** | O | **Objected** |

□ **Claims renumbered in the same order as presented by applicant** □ **CPA** □ **T.D.** □ **R.1.47**

| **CLAIM** | | **DATE** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Final | Original | 10/07/2010 | | | | | | | | |
| | 1 | ✓ | | | | | | | | |
| | 2 | ✓ | | | | | | | | |
| | 3 | ✓ | | | | | | | | |
| | 4 | ✓ | | | | | | | | |
| | 5 | ✓ | | | | | | | | |
| | 6 | ✓ | | | | | | | | |
| | 7 | ✓ | | | | | | | | |
| | 8 | ✓ | | | | | | | | |
| | 9 | ✓ | | | | | | | | |
| | 10 | ✓ | | | | | | | | |
| | 11 | ✓ | | | | | | | | |

|  |  |  |
|---|---|---|
| ***Search Notes*** | **Application/Control No.**<br>12713637 | **Applicant(s)/Patent Under Reexamination**<br>BARGER ET AL. |
|  | **Examiner**<br>CHARLES MURPHY | **Art Unit**<br>2455 |

## SEARCHED

| Class | Subclass | Date | Examiner |
|---|---|---|---|
| 709 | 263 234 231 226 219 217 204 203 200 | 10/7/2010 | Charles Murphy |
| 370 | 552 468 431 353 352 330 280 241 | 10/7/2010 | Charles Murphy |

## SEARCH NOTES

| Search Notes | Date | Examiner |
|---|---|---|
| Searched East | 10/7/2010 | Charles Murphy |

## INTERFERENCE SEARCH

| Class | Subclass | Date | Examiner |
|---|---|---|---|
|  |  |  |  |

/CHARLES MURPHY/
Examiner.Art Unit 2455

| | |
|---|---|
| **Form 1449 (Modified)** | **Serial No.: Application Submitted Herewith** |
| | **Atty. Docket No.: EQUI0016D2** |
| **Information Disclosure** | **Applicant: Sean BARGER, et al.** |
| **Statement By Applicant** | **Art Unit: Application Submitted Herewith** |
| | **Confirmation No.: Application Submitted Herewith** |
| (Use Several Sheets if Necessary) | **Filing Date:** |

**U.S. Patent Documents**

| Examiner Initials | No. | Patent No. | Issue Date | Patentee | Filing Date |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

**Published U.S, Patent Application**

| Examiner Initials | No. | Publication No. | Publication Date | Applicant |
|---|---|---|---|---|
| /C.M./ | 1 | 2009/0254672 | 10/08/2009 | Zhang |
| | | | | |
| | | | | |

**Foreign Patent or Published Foreign Patent Application**

| Examiner Initials | No. | Document No. | Publication Date | Applicant |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

**Non-Patent Literature Documents**

| Examiner Initials | No. | Author, Title, Date, Place (e.g. Journal) of Publication |
|---|---|---|
| | | |
| | | |
| | | |
| | | |

Examiner's Signature: /Charles Murphy/ Date: 10/07/2010

Examiner: Initial citation considered. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

## EAST Search History

## EAST Search History (Prior Art)

| Ref # | Hits | Search Query | DBs | Default Operator | Plurals | Time Stamp |
|---|---|---|---|---|---|---|
| L1 | 34 | (transcod$4) same (content or media or file) near4 (concurrent$3) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 17:40 |
| L2 | 0 | (transcod$4) same (content or media or file) near4 (concurrent$3) @ad<"19991021" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 17:42 |
| L3 | 0 | (transcod$4) same (content or media or file) near4 (concurrent$3) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 17:42 |
| L4 | 15 | (transcod$4) (content or media or file) near4 (concurrent$3) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 17:42 |
| L5 | 10 | (transcod$4) (content or media or file) same (deliver$3 or send$3 or transmit$3) same (concurrent$3 or simultan $4) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 17:43 |
| L6 | 0 | (transcod$4 same parameter) (content or media or file) same (deliver$3 or send$3 or transmit$3) same (concurrent$3 or simultan $4) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:18 |
| L7 | 2709 | (content or media or file) same (deliver$3 or send $3 or transmit$3) same (concurrent$3 or simultan $4) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:25 |

| | | | | | | |
|---|---|---|---|---|---|---|
| L8 | 763 | (content or media or file) near5 (deliver$3 or send $3 or transmit$3) same (concurrent$3 or simultan $4) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:25 |
| L9 | 184 | (content or media or file) near5 (deliver$3 or send $3 or transmit$3) near5 (concurrent$3 or simultan $4) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:25 |
| L10 | 41 | 9 (transcod$3 or convert) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:26 |
| L11 | 1 | 9 (transcod$3) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:26 |
| L12 | 25 | 9 (media.ti.) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:27 |
| L13 | 3 | (content or media or file) near5 (transcod$3 or convert$3) same (deliver $3 or send$3 or transmit $3) near5 (concurrent$3 or simultan$4) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:30 |
| L14 | 157 | (content or media or file) near5 (transcod$3 or convert$3) (deliver$3 or send$3 or transmit$3) near5 (concurrent$3 or simultan$4) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:30 |
| L15 | 6 | 14 (media.ti.) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:30 |

| | | | | | | |
|---|---|---|---|---|---|---|
| L16 | 22 | (content or media or file or audio or video or movie) near5 (transcod$3 or convert$3) same (deliver$3 or send$3 or transmit$3) near5 (concurrent$3 or simultan $4) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:31 |
| L17 | 1 | ("5845279").PN. | US-PGPUB; USPAT; USOCR | OR | OFF | 2010/10/07 18:35 |
| L18 | 0 | 17 concurent$3 | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:35 |
| L19 | 0 | 17 concurent$4 | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:35 |
| L20 | 1 | 17 concurrent$4 | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:35 |
| S1 | 1 | ("20090254672").PN. | US-PGPUB; USPAT; USOCR | OR | OFF | 2010/10/07 14:44 |
| S2 | 74 | ("5088052" \| "5355472" \| "5442771" \| "5530852" \| "5701451" \| "5708845" \| "5710918" \| "5737619" \| "5745908" \| "5758110" \| "5761555" \| "5761655" \| "5793964" \| "5819261" \| "5822436" \| "5845084" \| "5845299" \| "5860068" \| "5860073" \| "5861881" \| "5862325" \| "5864337" \| "5870552" \| "5880740" \| "5884337" \| "5890170" \| "5890170" \| "5895476" \| "5895476" \| "5895477" \| "5903892" \| "5937160" \| "5943680" \| "5943680" \| "5956737" \| "5956737" \| "6009436" \| "6456305" \| "6484149" \| "6563517" \| "6591280" \| "6591280" \| "6623529" \| "6623529"). PN. | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 14:48 |

| | | | | | | |
|---|---|---|---|---|---|---|
| S3 | 2 | S2 (transcod$4) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 14:49 |
| S4 | 74 | ("5088052" \| "5355472" \| "5442771" \| "5530852" \| "5701451" \| "5708845" \| "5710918" \| "5737619" \| "5745908" \| "5758110" \| "5761555" \| "5761655" \| "5793964" \| "5819261" \| "5822436" \| "5845084" \| "5845299" \| "5860068" \| "5860073" \| "5861881" \| "5862325" \| "5864337" \| "5870552" \| "5880740" \| "5884337" \| "5890170" \| "5890170" \| "5895476" \| "5895476" \| "5895477" \| "5903892" \| "5937160" \| "5943680" \| "5943680" \| "5956737" \| "5956737" \| "6009436" \| "6456305" \| "6484149" \| "6563517" \| "6591280" \| "6591280" \| "6623529" \| "6623529").PN. | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 14:51 |
| S5 | 1 | S4 (transcod$4 same cache) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 14:51 |

**EAST Search History (Interference)**

<This search history is empty>

**10/7/2010 7:01:44 PM**
**C:\ Documents and Settings\ cmurphy1\ My Documents\ EAST\ Workspaces\ 12713637.wsp**

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

# BIB DATA SHEET

**CONFIRMATION NO. 5103**

| SERIAL NUMBER | FILING or 371(c) DATE | CLASS | GROUP ART UNIT | ATTORNEY DOCKET NO. |
|---|---|---|---|---|
| 12/713,637 | 02/26/2010<br>**RULE** | 709 | 2441 | EQUI0016D2 |

**APPLICANTS**

Sean BARGER, Mill Valley, CA;
Steve Johnson, Mill Valley, CA;
Matt Butler, Beaverton, OR;
Jerry Destremps, Sausalito, CA;
David Pochron, Cambridge, WI;
Trent Brown, San Anselmo, CA;

** **CONTINUING DATA** *************************

This application is a DIV of 12/173,747 07/15/2008
which is a DIV of 11/269,916 11/07/2005 ABN
which is a CIP of 09/929,904 08/14/2001 PAT 6,964,009
which is a CON of 09/425,326 10/21/1999 PAT 6,792,575

** **FOREIGN APPLICATIONS** *************************

** **IF REQUIRED, FOREIGN FILING LICENSE GRANTED** ** ** SMALL ENTITY **
03/09/2010

| | | STATE OR COUNTRY | SHEETS DRAWINGS | TOTAL CLAIMS | INDEPENDENT CLAIMS |
|---|---|---|---|---|---|
| Foreign Priority claimed ☐ Yes ☑ No<br>35 USC 119(a-d) conditions met ☐ Yes ☑ No<br>Verified and Acknowledged /CHARLES C MURPHY/<br>Examiner's Signature | ☐ Met after Allowance<br>Initials | CA | 23 | 11 | 2 |

**ADDRESS**

GLENN PATENT GROUP
3475 EDISON WAY, SUITE L
MENLO PARK, CA 94025
UNITED STATES

**TITLE**

Automated Media Delivery System

| | | |
|---|---|---|
| **FILING FEE RECEIVED**<br>462 | FEES: Authority has been given in Paper No.____________ to charge/credit DEPOSIT ACCOUNT No.____________ for following: | ☐ All Fees<br>☐ 1.16 Fees (Filing)<br>☐ 1.17 Fees (Processing Ext. of time)<br>☐ 1.18 Fees (Issue)<br>☐ Other ________________<br>☐ Credit |

## EAST Search History

## EAST Search History (Prior Art)

| Ref # | Hits | Search Query | DBs | Default Operator | Plurals | Time Stamp |
|---|---|---|---|---|---|---|
| L1 | 7 | (transcod$3) same (fly) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/09/23 09:46 |
| L2 | 2595 | (transcod$3 or convert$3) same (fly) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/09/23 09:46 |
| L3 | 680 | (transcod$3 or convert$3) near5 (fly) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/09/23 09:47 |
| L4 | 680 | (transcod$3 or convert$3) near5 (fly) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/09/23 09:47 |
| L5 | 183 | 4 (transcod$3 or convert$3).ti. | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/09/23 09:47 |
| L6 | 2 | 5 (multimedia or multi!media or audio or video or image) same (transcod$3) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/09/23 09:48 |

| | | | | | | |
|---|---|---|---|---|---|---|
| S1 | 66 | US-5088052-$.DID. OR US-5355472-$.DID. OR US-5530852-$.DID. OR US-5701451-$.DID. OR US-5708845-$.DID. OR US-5710918-$.DID. OR US-5737619-$.DID. OR US-5745908-$.DID. OR US-5758110-$.DID. OR US-5761655-$.DID. OR US-5793964-$.DID. OR US-5819261-$.DID. OR US-5822436-$.DID. OR US-5845084-$.DID. OR US-5845299-$.DID. OR US-5860068-$.DID. OR US-5860073-$.DID. OR US-5861881-$.DID. OR US-5862325-$.DID. OR US-5864337-$.DID. OR US-5870552-$.DID. OR US-5880740-$.DID. OR US-5890170-$.DID. OR US-5895476-$.DID. OR US-5895477-$.DID. OR US-5903892-$.DID. OR US-5937160-$.DID. OR US-5943680-$.DID. OR US-5956737-$.DID. OR US-6009436-$.DID. OR US-6456305-$.DID. OR US-6563517-$.DID. OR US-6591280-$.DID. OR US-6623529-$.DID. | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/02/02 14:10 |
| S2 | 4 | US-5442771-$.DID. OR US-6484149-$.DID. | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/02/02 14:11 |

| | | | | | | |
|---|---|---|---|---|---|---|
| S3 | 70 | S1 or S2 | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/02/02 14:42 |
| S4 | 0 | S3 (transcod$3 same concurrent or simultaneous$3 same (alternate or different) near5 format) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/02/02 14:43 |
| S5 | 0 | S3 (transcod$3 same (alternate or different) near5 format) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/02/02 14:43 |
| S6 | 2 | S3 (transcod$3) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/02/02 14:44 |
| S8 | 53 | ((transcod$3) same (format) same (analog or wav)) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/04/01 15:27 |
| S9 | 1 | ("6909708").PN. | US-PGPUB; USPAT; USOCR | OR | OFF | 2010/04/06 14:25 |
| S10 | 1 | S9 client | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/04/06 15:44 |
| S11 | 1 | S9 computer | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/04/06 15:48 |

| | | | | | | |
|---|---|---|---|---|---|---|
| S12 | 0 | S9 computer same prefered | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/04/06 15:51 |
| S13 | 0 | S9 computer same preffered | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/04/06 15:51 |
| S14 | 0 | S9 computer same preferred | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/04/06 15:51 |
| S15 | 0 | S9 computer same perferred | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/04/06 15:52 |
| S16 | 0 | S9 computer same embodiement | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/04/06 15:52 |
| S17 | 0 | S9 computer with preferred | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/04/06 15:52 |
| S18 | 0 | S9 computer and preferred | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/04/06 15:52 |
| S19 | 155 | ((transcod$3) same (format) same (analog or wav)) @ay<"2002" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/04/06 16:47 |

| | | | | | | |
|---|---|---|---|---|---|---|
| S20 | 224 | (transcod$3 or convert) same (format) same (on adj the adj fly or simultan$4 or concurrent$3) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/09/22 16:00 |
| S21 | 17 | (transcod$3 or convert) same (format) near5 (on adj the adj fly or simultan$4 or concurrent$3) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/09/22 16:01 |
| S22 | 6 | S20 (transcod$3 or convert).ti. | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/09/22 16:04 |

**9/23/2010 9:58:04 AM**
**C:\ Documents and Settings\ cmurphy1\ My Documents\ EAST\ Workspaces\ 12173747.wsp**

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

| | | |
|---|---|---|
| First Named Inventor | : | Sean Barger |
| Serial No. | : | 12/713,637 |
| Filed | : | February 26, 2010 |
| Art Unit | : | 2455 |
| Confirmation Number | : | 5103 |
| Examiner | : | Charles C. Murphy |
| Title | : | Automated Media Delivery System |
| Attorney Docket No. | : | EQUI0016D2 |

January 13, 2011

Commissioner for Patents
Mail Stop Amendment
P.O. Box 1450
Alexandria, VA 22313-1450

## RESPONSE TO OFFICE ACTION

Applicant submits this Response to the Office Action dated October 19, 2010 in connection with the above-identified patent application.

A **Listing of Claims** begins on Page 2 of this paper, and

**Remarks** begin on Page 6 of this paper.

The Commissioner is authorized to charge any fees that may be due and to credit any overpayments to Deposit Account 50-5066 (Order No. EQUI0016D2). Applicant considers this document to be filed in a timely manner.

## LISTING OF CLAIMS

1. (Currently Amended) A non-transitory computer program product for causing a computing system to ~~comprising one or more computer-readable media having thereon computer-executable instructions that, when executed by one or more processors of a computing system, causes the computing system to perform a method for~~ dynamically ~~transcoding~~ transcode media content to be presented on a client presentation system, the computer program product ~~method~~ comprising:

one or more computer-readable physical and/or memory storage media, the one or more computer-readable memory media having stored thereon computer-executable instructions that, when executed by one or more processors of the computing system, cause the computing system to perform the following:

an act of receiving a request for media content to be delivered to a client presentation system, wherein the requested media content has a limited number of base transcoding profiles associated therewith, each base transcoding profile corresponding to a cached version of the requested media content;

after receiving the request, an act of identifying transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system;

an act of transcoding the requested media content in accordance with the identified transcoding parameters, such that the identified transcoding parameters are used to perform additional incremental transcoding on top of the base transcoding profile;

an act of causing the transcoded media content to be delivered to the client presentation system; and

an act of caching the transcoded media content.

2. (Cancelled).

3. (Currently Amended) A computer program product in accordance with claim [[2]] 1, wherein the act of receiving a request for media content is performed by receiving the request over the Internet.

4. (Currently Amended) A computer program product in accordance with claim [[2]] 1, wherein the client presentation system is a first client presentation system, the method further comprising:

an act of receiving a request for the media content to be delivered to a second client presentation system;

an act of determining that the transcoding parameters to be applied to the requested media content prior to delivery to the second client presentation system are the same as the transcoding parameters that were applied to the requested media content prior to delivery to the first client presentation system; and

an act of delivering the cached transcoded media content, rather than separately transcoding the requested media content again for the second client presentation system.

5. (Currently Amended) A computer program product in accordance with claim [[2]] 1, wherein the requested media content is video media.

6. (Currently Amended) A computer program product in accordance with claim [[2]] 1, wherein the identified transcoding parameters includes at least video size parameters.

7. (Currently Amended) A computer program product in accordance with claim [[2]] 1, wherein the method further comprises:

prior to the act of transcoding, an act of determining that a transcoded representation of the requested media content transcoded using the identified transcoding parameters does not already exist.

8. (Currently Amended) A computer program product in accordance with claim [[2]] 1, wherein the act of identifying transcoding parameters comprises an act of selecting a transcoding profile from a set of pre-existing transcoding profiles.

9. (Currently Amended) A computer program product in accordance with claim [[2]] 1, wherein the act of identifying transcoding parameters comprises:

an act of reading one or more transcoding parameters from the request; and

an act of identifying the one or more transcoding parameters from the request as being the identified transcoding parameters to be applied.

10. (Currently Amended) A computer program product in accordance with claim [[2]] 1, wherein the act of act of transcoding the requested media content in accordance with the identified transcoding parameters comprises:

an act of selecting a pre-existing base transcoded version of the requested media content that has been transcoded in accordance with only a portion of the identified transcoding parameters; and

an act of incrementally performing further transcoding of the pre-existing base transcoded version in accordance with a remaining portion of the identified transcoding parameters.

11. (Currently Amended) A method for accessing dynamically transcoding media content, the method comprising:

an act of receiving a request for media content to be delivered to a client presentation system for media content, wherein the requested media content has a limited number of base transcoding profiles associated therewith, each base transcoding profile corresponding to a cached version of the requested media content;

after receiving the request, an act of identifying transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system;

an act of determining that the transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system are the same as transcoding parameters that are being applied to the requested media content prior to delivery to another client presentation system; [[and]]

an act of transcoding the requested media content in accordance with the identified transcoding parameters, such that the identified transcoding parameters are used to perform additional incremental transcoding on top of the base transcoding profile; and

an act of delivering the transcoded media content to both client presentation systems concurrently.

## **REMARKS**

Applicants respectfully request further examination and reconsideration in view of the above amendments and arguments set forth fully below.

### **Claim Rejections under 35 U.S.C. § 101**

Within the Office Action, Claims 1-11 were rejected under 35 USC 101 for allegedly being directed to nonstatutory subject matter. Applicants amend the claims herein, thereby rendering the rejection moot.

### **Claim Rejections under 35 U.S.C. § 102(e)**

Also within the Office Action, Claim 1 was rejected under 35 U.S.C. § 102(e) as being anticipated by United States Patent No. 6,563,517 to Bhagwat (hereinafter referred to as "Bhagwat").

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631 (Fed. Cir. 1987).

The Applicants respectfully traverse this rejection because each and every element set forth in Claim 1 is not found in Bhagwat, neither expressly nor inherently described. Specifically, Bhagwat does not describe transcoding of content based on transcoding parameters included in a request for content, wherein requested media that has associated with it a limited set of transcoding profiles, each corresponding to a limited number of base transcoding profiles, and performing transcoding by performing additional incremental transcoding on top of the base transcoding, nor does the Examiner suggest that he does.

For at least this reason, Claim 1 is not anticipated by Bhagwat.

**Claim Rejections under 35 U.S.C. § 103**

Also within the Office Action, Claims 2-9 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Bhagwat.

To establish a *prima facie* case of obviousness of a claimed invention, all the claimed features must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). The Applicants respectfully traverse this rejection because Bhagwat does not disclose all of the limitations of Claims 2-9.

Specifically Bhagwat does not teach or suggest transcoding of content based on transcoding parameters included in a request for content, wherein requested media that has associated with it a limited set of transcoding profiles, each corresponding to a limited number of base transcoding profiles, and performing transcoding by performing additional incremental transcoding on top of the base transcoding, nor does the Examiner suggest that he does.

For at least this reason, Claims 2-9 are not rendered obvious by Bhagwat.

Also within the Office Action, Claim 11 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Bhagwat in view of United States Patent No. 5,845,279 to Garofalakis et al. (hereinafter referred to as "Garofalakis"). The Applicants respectfully traverse this rejection neither Bhagwat nor Garofalakis, neither alone nor in combination, disclose all of the limitations of Claim 11.

Specifically neither Bhagwat nor Garofalakis teach or suggest transcoding of content based on transcoding parameters included in a request for content, wherein requested media that has associated with it a limited set of transcoding profiles, each corresponding to a limited number of base transcoding profiles, and performing transcoding by performing additional incremental transcoding on top of the base transcoding, nor does the Examiner suggest that he does.

For at least this reason, Claim 11 is not rendered obvious by a hypothetical combination of Bhagwat and Garfalakis.

Also within the Office Action, Claim 10 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Bhagwat in view of United States Patent No. 6,909,708 to Krishnaswamy et al. (hereinafter referred to as "Krishnaswamy"). The Applicants respectfully traverse this rejection neither Bhagwat nor Krishnaswamy, neither alone nor in combination, disclose all of the limitations of Claim 10.

Specifically neither Bhagwat nor Krishnaswamy teach or suggest transcoding of content based on transcoding parameters included in a request for content, wherein requested media that has associated with it a limited set of transcoding profiles, each corresponding to a limited number of base transcoding profiles, and performing transcoding by performing additional incremental transcoding on top of the base transcoding, nor does the Examiner suggest that he does.

For at least this reason, Claim 10 is not rendered obvious by a hypothetical combination of Bhagwat and Krishnaswamy.

For the record, the Applicant respectfully traverses any and all factual assertions in the file that are not supported by documentary evidence. Such include assertions based on findings of inherency, assertions based on Official Notice, and any other assertions of what is well known or commonly known in the prior art.

Additionally, the foregoing amendments are made solely in the interest of expediency, in recognition of the Office policy of compact prosecution. They do not indicate agreement by Applicant with the Office's position, nor do they reflect intent to forsake Claim scope. In fact, Applicant expressly reserves the right to pursue patent protection of a scope it is reasonably entitled to in future submissions to the Office.

## CONCLUSION

Applicant respectfully posits that the pending claims have been distinguished from the art of record, and that all objections to and rejections of the claims have been overcome. Accordingly, Applicant respectfully requests allowance. Should the Examiner deem it helpful he is encouraged to contact Applicant's attorney at (650) 474-8400.

Respectfully submitted,

Michael A. Glenn
Reg. No. 30,176

Customer No. 22862

# Electronic Acknowledgement Receipt

| | |
|---|---|
| **EFS ID:** | 9227192 |
| **Application Number:** | 12713637 |
| **International Application Number:** | |
| **Confirmation Number:** | 5103 |
| **Title of Invention:** | Automated Media Delivery System |
| **First Named Inventor/Applicant Name:** | Sean BARGER |
| **Customer Number:** | 22862 |
| **Filer:** | Michael Glenn/Jessica Pallach |
| **Filer Authorized By:** | Michael Glenn |
| **Attorney Docket Number:** | EQUI0016D2 |
| **Receipt Date:** | 13-JAN-2011 |
| **Filing Date:** | 26-FEB-2010 |
| **Time Stamp:** | 16:36:01 |
| **Application Type:** | Utility under 35 USC 111(a) |

## Payment information:

| | |
|---|---|
| Submitted with Payment | no |

## File Listing:

| Document Number | Document Description | File Name | File Size(Bytes)/ Message Digest | Multi Part /.zip | Pages (if appl.) |
|---|---|---|---|---|---|
| 1 | | Response-EQUI0016D2--efiled011311.pdf | 421364<br>a4d9d25095b867f5a45895390d3798e7b8b818e0 | yes | 11 |

| | **Multipart Description/PDF files in .zip description** | | |
|---|---|---|---|
| | **Document Description** | **Start** | **End** |
| | Miscellaneous Incoming Letter | 1 | 1 |
| | Amendment/Req. Reconsideration-After Non-Final Reject | 2 | 2 |
| | Claims | 3 | 6 |
| | Applicant Arguments/Remarks Made in an Amendment | 7 | 11 |
| **Warnings:** | | | |
| **Information:** | | | |
| | **Total Files Size (in bytes):** | 421364 | |

**This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.**

**New Applications Under 35 U.S.C. 111**
**If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.**

**National Stage of an International Application under 35 U.S.C. 371**
**If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.**

**New International Application Filed with the USPTO as a Receiving Office**
**If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.**

| | | |
|---|---|---|
| **TRANSMITTAL FORM**<br>*(to be used for all correspondence after initial filing)* | Application Number | 12/713,637 |
| | Filing Date | February 26, 2010 |
| | First Named Inventor | Sean Barger |
| | Art Unit | 2455 |
| | Examiner Name | Charles Murphy |
| Total number of pages including cover sheet. | Attorney Docket Number | EQUI0016D2 |

**ENCLOSURES** *(Check all that apply)*

□ Fee Transmittal Form

□ Fee Attached

☑ Amendment/Reply

□ After Final

□ Affidavits/declaration(s)

□ Extension of Time Request

□ Express Abandonment Request

□ Information Disclosure Statement

□ Certified Copy of Priority Document(s)

□ Reply to Missing Parts/ Incomplete Application

□ Reply to Missing Parts under 37 CFR 1.52 or 1.53

□ Drawing(s)

□ Licensing-related Papers

□ Petition

□ Petition to Convert to a Provisional Application

□ Power of Attorney, Revocation Change of Correspondence Address

□ Terminal Disclaimer

□ Request for Refund

□ CD, Number of CD(s) ____________

□ Landscape Table on CD

□ After Allowance Communication to TC

□ Appeal Communication to Board of Appeals and Interferences

□ Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**

□ Proprietary Information

□ Status Letter

□ Other Enclosure(s) (please Identify below):

Remarks

**SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT**

| | | | |
|---|---|---|---|
| Firm Name | GLENN PATENT GROUP, Customer No. 22862 | | |
| Signature | | | |
| Printed name | Michael A. Glenn | | |
| Date | January 13, 2011 | Reg. No. | 30,176 |

## CERTIFICATE OF ELECTRONIC FILING

I hereby certify that this correspondence is being electronically transmitted to the USPTO via EFS-Web on the date shown below.

| | | | |
|---|---|---|---|
| Signature | | | |
| Typed or printed name | Jessica Pallach | Date | January 13, 2011 |

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

*If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.*

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

| | | | |
|---|---|---|---|
| **PATENT APPLICATION FEE DETERMINATION RECORD**<br>Substitute for Form PTO-875 | Application or Docket Number<br>12/713,637 | Filing Date<br>02/26/2010 | □ To be Mailed |

APPLICATION AS FILED – PART I

| | (Column 1) | (Column 2) | SMALL ENTITY ☒ | | OR | OTHER THAN SMALL ENTITY | |
|---|---|---|---|---|---|---|---|
| FOR | NUMBER FILED | NUMBER EXTRA | RATE ($) | FEE ($) | | RATE ($) | FEE ($) |
| □ BASIC FEE (37 CFR 1.16(a), (b), or (c)) | N/A | N/A | N/A | | | N/A | |
| □ SEARCH FEE (37 CFR 1.16(k), (i), or (m)) | N/A | N/A | N/A | | | N/A | |
| □ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q)) | N/A | N/A | N/A | | | N/A | |
| TOTAL CLAIMS (37 CFR 1.16(i)) | minus 20 = | * | X $ = | | OR | X $ = | |
| INDEPENDENT CLAIMS (37 CFR 1.16(h)) | minus 3 = | * | X $ = | | | X $ = | |
| □APPLICATION SIZE FEE (37 CFR 1.16(s)) | If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s). | | | | | | |
| □ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j)) | | | | | | | |
| | | | TOTAL | | | TOTAL | |

* If the difference in column 1 is less than zero, enter "0" in column 2.

APPLICATION AS AMENDED – PART II

| | | (Column 1) | | (Column 2) | (Column 3) | SMALL ENTITY | | OR | OTHER THAN SMALL ENTITY | |
|---|---|---|---|---|---|---|---|---|---|---|
| AMENDMENT | **01/13/2011** | CLAIMS REMAINING AFTER AMENDMENT | | HIGHEST NUMBER PREVIOUSLY PAID FOR | PRESENT EXTRA | RATE ($) | ADDITIONAL FEE ($) | | RATE ($) | ADDITIONAL FEE ($) |
| | Total (37 CFR 1.16(i)) | * 10 | Minus | ** 20 | = 0 | X $26 = | 0 | OR | X $ = | |
| | Independent (37 CFR 1.16(h)) | * 2 | Minus | ***3 | = 0 | X $110 = | 0 | OR | X $ = | |
| | □ Application Size Fee (37 CFR 1.16(s)) | | | | | | | | | |
| | □ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j)) | | | | | | | OR | | |
| | | | | | | TOTAL ADD'L FEE | **0** | OR | TOTAL ADD'L FEE | |

| | | (Column 1) | | (Column 2) | (Column 3) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| AMENDMENT | | CLAIMS REMAINING AFTER AMENDMENT | | HIGHEST NUMBER PREVIOUSLY PAID FOR | PRESENT EXTRA | RATE ($) | ADDITIONAL FEE ($) | | RATE ($) | ADDITIONAL FEE ($) |
| | Total (37 CFR 1.16(i)) | * | Minus | ** | = | X $ = | | OR | X $ = | |
| | Independent (37 CFR 1.16(h)) | * | Minus | *** | = | X $ = | | OR | X $ = | |
| | □ Application Size Fee (37 CFR 1.16(s)) | | | | | | | | | |
| | □ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j)) | | | | | | | OR | | |
| | | | | | | TOTAL ADD'L FEE | | OR | TOTAL ADD'L FEE | |

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/HENRIETTA DENDY/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

*If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.*

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 12/713,637 | 02/26/2010 | Sean BARGER | EQUI0016D2 | 5103 |

22862 7590 04/05/2011
GLENN PATENT GROUP
3475 EDISON WAY, SUITE L
MENLO PARK, CA 94025

| EXAMINER | |
|---|---|
| MURPHY, CHARLES C | |
| **ART UNIT** | **PAPER NUMBER** |
| 2455 | |

| NOTIFICATION DATE | DELIVERY MODE |
|---|---|
| 04/05/2011 | ELECTRONIC |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptomatters@glenn-law.com
GlennDocketing@cpaglobal.com

| | | | |
|---|---|---|---|
| ***Office Action Summary*** | **Application No.**<br>12/713,637 | **Applicant(s)**<br>BARGER ET AL. | |
| | **Examiner**<br>CHARLES MURPHY | **Art Unit**<br>2455 | |

***-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --***

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *13 January 2011*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

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

**Disposition of Claims**

4)☒ Claim(s) *1 and 3-11* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1 and 3-11* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

## *Detailed Action*

**1.** This Office Action is responsive the Amendment filed on 01/13/2011. Claims 1 and 3-11 have been amended. Claims 1 and 3-11 remain pending for examination. Claim 2 has been withdrawn from examination by applicant.

## *Claim Objections*

Claims 1 and 3-10 objected to because of the following informalities: The parent claim of claims 3-10 recites a “non-transitory computer program product” while claims 3-10 recite a “computer program product”. Claim 1 also recites a non-transitory computer program product comprising one ore more computer readable physical **and/or** memory storage media. The use of the term “and/or” is confusing and appropriate correction is required.

## Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claim 1 and 3-10 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

As per claim 1, the claim limitation recites “computer readable physical and/or memory storage media". However, the usage of the phrase ""computer readable physical and/or memory storage media" is broad enough to include both "non-transitory" and "transitory" (moving electrons, etc) media. Although the preamble of the claim recites a non-transitory computer program product, the first limitation in claim 1 recites a computer readable and/or memory storage media. The specification does not clearly limit the utilization of a non-transitory computer readable medium and, thus does not constitute functional descriptive material. Therefore, when the broadest reasonable interpretation of a claim covers a signal per se, the claim must be rejected under 35 U.S.C. § 101 as covering non-statutory subject matter. See *In re Nuijten,* 500 F.3d 1346, 1356-57 (Fed. Cir. 2007) (transitory embodiments are not directed to statutory subject matter).

A claim drawn to such a computer readable medium that covers both ***transitory*** and ***non-transitory*** embodiments may be amended to narrow the claim to cover only statutory embodiments to avoid a rejection under 35 U.S.C. § 101 by adding the limitation "***non-transitory***" to the claim. *Cf. Animals - Patentability*, 1077 *Off. Gaz. Pat. Office* 24 (April 21, 1987) Specifically, the Applicant may amend the first limitation of claim 1 to read a "***non-transitory computer readable physical and/or memory storage media***" or the like to further narrow the claim to cover only statutory embodiments.

The dependent claims included in the statement of rejection but not specifically addressed in the body of the rejection have inherited the deficiencies of their parent claim and have not resolved the deficiencies. Therefore, they are rejected based on the same rationale as applied to their parent claims above.

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

1. **Claims 1 and 3-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bhagwat et al. US Patent 6,563,517 (hereinafter referred to as Bhagwat) in view of Krishnaswamy et al. US Patent 6,909,708 (hereinafter referred to as Krishnaswamy)**

2. Regarding claim 1, **Bhagwat** teaches an computer program product when executed causes a computer system to perform a method comprising:

   one or more computer-readable physical and/or memory storage media, the one or more computer-readable memory media having stored thereon computer executable instructions that, when executed by one or more processors of the computing system, cause the computing system to perform the following ***(Bhagwat, col. 3, lines 9-15)*** *(**Bhagwat** teaches using a computer readable medium to implement the invention)**(Bhagwat, col. 14, lines 1-10)***:

an act of receiving a request for media content to be delivered to a client presentation system;*(**Bhagwat** teaches receiving a request for media to be sent to a client)*

after receiving the request, an act of identifying transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system; *(**Bhagwat** teaches using various parameters transcode media before it is delivered to a requesting client)**(Bhagwat, col. 3, lines 9-20; col. 5, lines 39-45)***

an act of causing the transcoded media content to be delivered to the client presentation system; and *(**Bhagwat** teaches delivering the requested transcoded content to the requesting client))**(Bhagwat, col. 3, lines 9-20; col. 5, lines 39-45; cool 6, lines 45-50)***

an act of caching the transcoded media content. *(**Bhagwat** teaches caching a transcoded media file)**(Bhagwat, col. 5, lines 15-25 and 50-57; cool 6, lines 45-50)*** the act of receiving a request for media content wherein the requested media content has a limited number of base transcoding profiles associated therewith, each base transcoding profile corresponding to a cached version of the requested media content; *(**Bhagwat** teaches a multi resolution cache in a transcoding proxy. This multi resolution cache stores the original (base) file that is to be transcoded as well as various reduced resolution versions of the original file that have been transcoded for specific devices. The parameters used to transcode this base file into a specified media can be considered transcoding profiles used to transcode the base file. )**(Bhagwat, col. 5, lines 10-25 and 50-57; cool 6, lines 45-50, col. 13, lines 20-40)***

**Bhagwat** does not teach:

an act of transcoding the requested media content in accordance with the identified transcoding parameters, such that the identified transcoding parameters are used to perform additional incremental transcoding on top of the base transcoding profile;

In an analogous art, **Krishnaswamy** teaches:

an act of transcoding the requested media content in accordance with the identified transcoding parameters, such that the identified transcoding parameters are used to perform additional incremental transcoding on top of the base transcoding profile; *(**Krishnaswamy** teaches to use a base file that has already been transcoded, and convert the base file into various formats.)**(Krishnaswamy, col. 135, lines 25-35)***

At the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teachings of **Bhagwat** and **Krishnaswamy** before him or her, to modify **Bhagwat** to include the teachings of **Krishnaswamy** because this would create a system where users can manage and control a network while a network operator can maintain network quality and routing selection (**Krishnaswamy, col. 2, lines 60-63)**

**3.** Regarding claim 3, **Bhagwat**-**Krishnaswamy** further teaches the computer program product of claim 1, wherein:

the act of receiving a request for media content is performed by receiving the request over the Internet. *(**Bhagwat** teaches a client computer and a* *web server communicating with each other over the internet)* ***(Bhagwat, col. 3, lines 9-15)***

**4.** Regarding claim 4, **Bhagwat**-**Krishnaswamy** further teaches the computer program product of claim 1, wherein:

an act of receiving a request for the media content to be delivered to a second client presentation system; *(**Bhagwat** teaches a client computer and a web server communicating with each other over the internet)* ***(Bhagwat, col. 3, lines 9-15)***

an act of determining that the transcoding parameters to be applied to the requested media content prior to delivery to the second client presentation system are the same as the transcoding parameters that were applied to the requested media content prior to delivery to the first client presentation system; and*(**Bhagwat** teaches first checking a cache after a request is received by a client for a media object. If the media object transcoded with the required parameters is cached then the cached media object is provided to the requesting client.)****(Bhagwat, col. 5, lines 50-67)***

**5.** Regarding claim 5, **Bhagwat**-**Krishnaswamy** further teaches the computer program product of claim 1, wherein:

the requested media content is video media. *(**Bhagwat** teaches a client computer with the ability to request video content)* ***(Bhagwat, col. 6, lines 19-25)***

**6.** Regarding claim 6, **Bhagwat**-**Krishnaswamy** further teaches the computer program product of claim 1, wherein:

the identified transcoding parameters includes at least video size parameters. *(**Bhagwat** teaches various transcoding parameters including the resolutions of a media object.)****(Bhagwat, col. 5, lines 15-25; col. 6, lines 5-15)***

**7.** Regarding claim 7, **Bhagwat**-**Krishnaswamy** further teaches the computer program product of claim 1, comprising:

prior to the act of transcoding, an act of determining that a transcoded representation of the requested media content transcoded using the identified transcoding parameters does not already exist. *(**Bhagwat** teaches checking if a request media file is cached at the requested resolution or transcoding level. These parameters are what is used to transcode the media file.)****(Bhagwat, col. 5, lines 50-60)***

**8.** Regarding claim 8, **Bhagwat**-**Krishnaswamy** further teaches the computer program product of claim 1, wherein:

the act of identifying transcoding parameters comprises an act of selecting a transcoding profile from a set of pre-existing transcoding profiles. *(**Bhagwat** teaches choosing a set of transcoding parameters based on user's preferences.)**(Bhagwat, col. 13, lines30-40)***

**9.** Regarding claim 9, **Bhagwat**-**Krishnaswamy** further teaches the computer program product of claim 1, comprising:

an act of reading one or more transcoding parameters from the request; and *(**Bhagwat** teaches a client computer sending a request to a server for a requested media file. The server then, using a user's preferences, transcodes the media file and transmits it back to the requesting client.)**(Bhagwat, col. 13, lines 30-40 and col. 11, lines 45-67 )***

an act of identifying the one or more transcoding parameters from the request as being the identified transcoding parameters to be applied. *(**Bhagwat** teaches choosing a set of transcoding parameters based on user's preferences.)**(Bhagwat, col. 13, lines 30-40 and col. 11, lines 45-67)***

**10.** Regarding claim 10, **Bhagwat**-**Krishnaswamy** further teaches the computer program product of claim 2, comprising:

an act of incrementally performing further transcoding of the pre-existing base transcoded version in accordance with a remaining portion of the identified transcoding parameters. *(**Bhagwat** teaches incrementally transcoding a file)**(Bhagwat, col. 12, lines 14-16)***

**Bhagwat** does not teach:

an act of selecting a pre-existing base transcoded version of the requested media content that has been transcoded in accordance with only a portion of the identified transcoding parameters; and

In an analogous art, **Krishnaswamy** teaches:

an act of selecting a pre-existing base transcoded version of the requested media content that has been transcoded in accordance with only a portion of the identified transcoding parameters; and *(**Krishnaswamy** teaches to use a base file that has already been transcoded, and convert the base file into various formats.)**(Krishnaswamy, col. 135, lines 25-35)***

At the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teachings of **Bhagwat** and

**Krishnaswamy** before him or her, to modify **Bhagwat** to include the teachings of **Krishnaswamy** because this would create a system where users can manage and control a network while a network operator can maintain network quality and routing selection (**Krishnaswamy, col. 2, lines 60-63)**

**11. Claim 11 rejected under 35 U.S.C. 103(a) as being unpatentable Bhagwat et al. US Patent 6,563,517 (hereinafter referred to as Bhagwat) in view of Krishnaswamy et al. US Patent 6,909,708 (hereinafter referred to as Krishnaswamy) in further view of Garofalakis et al. US Patent 5,845,279 (hereinafter referred to as Garofalakis)**

**12.** Regarding claim 11, **Bhagwat** teaches a method comprising:

an act of receiving a request for media content to be delivered to a client presentation system for media content;*(**Bhagwat** teaches receiving a request for media to be sent to a client)**(Bhagwat, col. 3, lines 9-15)*** *(**Bhagwat** also teaches using a computer readable medium to implement the invention)**(Bhagwat, col. 14, lines 1-10)***

after receiving the request, an act of identifying transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system; *(**Bhagwat** teaches using various parameters transcode media before it is delivered to a requesting client)**(Bhagwat, col. 3, lines 9-20; col. 5, lines 39-45)***

an act of determining that the transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system are the same as transcoding parameters that are being applied to the requested media content prior to delivery to another client presentation system; and*(**Bhagwat** teaches checking if a request media file is cached at the requested resolution or transcoding level. These parameters are what is used to transcode the media file.)**(Bhagwat, col. 5, lines 50-60)***

wherein the requested media content has a limited number of base transcoding profiles associated therewith, each base transcoding profile corresponding to a cached version of the requested media content; *(**Bhagwat** teaches a multi resolution cache in a transcoding proxy. This multi resolution cache stores the original (base) file that is to be transcoded as well as various reduced resolution versions of the original file that have been transcoded for specific devices. The parameters used to transcode this base file into a specified media can be considered transcoding profiles used to transcode the base file. )**(Bhagwat, col. 5, lines 10-25 and 50-57; cool 6, lines 45-50, col. 13, lines 20-40)***

**Bhagwat** does not teach:

> an act of transcoding the requested media content in accordance with the identified transcoding parameters, such that the identified transcoding parameters are used to perform additional incremental transcoding on top of the base transcoding profile;

In an analogous art, **Krishnaswamy** teaches:

> an act of transcoding the requested media content in accordance with the identified transcoding parameters, such that the identified transcoding parameters are used to perform additional incremental transcoding on top of the base transcoding profile; *(**Krishnaswamy** teaches to use a base file that has already been transcoded, and convert the base file into various formats.)**(Krishnaswamy, col. 135, lines 25-35)***

At the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teachings of **Bhagwat** and **Krishnaswamy** before him or her, to modify **Bhagwat** to include the teachings of **Krishnaswamy** because this would create a system where users can manage and control a network while a network operator can maintain network quality and routing selection (**Krishnaswamy, col. 2, lines 60-63)**

**Bhagwat- Krishnaswamy** does not teach:

an act of delivering the transcoded media content to both client presentation systems concurrently.

In an analogous art, **Garfalakis** teaches:

an act of delivering the transcoded media content to both client presentation systems concurrently.*(**Garofalakis** teaches the ability to deliver media concurrently to multiple systems)**(Garofalakis, col. 2, lines 50-55)***

At the time of the invention, it would have been obvious to one of ordinary skill in the art, having the teachings of **Bhagwat- Krishnaswamy** and **Garfalakis** before him or her, to modify **Bhagwat- Krishnaswamy** to include the teachings of **Garfalakis** because this would create a system that would schedule and transmit various media segments of different popularity and lengths at different times. (**Garfalakis, col. 2, lines 60-63)**

***Response to Arguments***

Applicant's arguments filed 01/13/2011 have been fully considered but they are not persuasive.

Applicant states:

> *Specifically neither Bhagwat nor Krishnaswamy teach or suggest transcoding of content based on transcoding parameters included in a request for content, wherein requested media that has associated with it a limited set of transcoding profiles, each corresponding to a limited number of base transcoding profiles, and performing transcoding by performing additional incremental transcoding on top of the base transcoding, nor does the Examiner suggest that he does.* ***(Remarks, page 9)***

The Examiner respectfully disagrees. **Bhagwat** teaches a client computer sending a request to a server for a requested media file. Included within the request are the user's preferences. These preferences are selected via a user interface that is used to compare image quality vs. download time. Once the user's preferences have been selected a request is sent to the server which, using a user's preferences, transcodes the media file and transmits it back to the requesting client.*)**(Bhagwat, col. 13, lines 30-40; and col. 11, lines 45-67)***

**Krishnaswamy** teaches to receiving an input stream and transcoding that input stream into a base format. This base format can then be transcoded again into various other formats for use on specified devices. **(Krishnaswamy, col. 135, lines 25-35)**

### ***Conclusion***

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHARLES MURPHY whose telephone number is (571)270-5444. The examiner can normally be reached on 8AM - 5PM Monday - Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Rupal Dharia can be reached on 571-272-3880. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Rupal D. Dharia/
Supervisory Patent Examiner, Art Unit 2456

/CHARLES MURPHY/
Examiner, Art Unit 2455
3/29/2011

| | | | |
|---|---|---|---|
| ***Notice of References Cited*** | Application/Control No.<br>12/713,637 | Applicant(s)/Patent Under Reexamination<br>BARGER ET AL. | |
| | Examiner<br>CHARLES MURPHY | Art Unit<br>2455 | Page 1 of 1 |

**U.S. PATENT DOCUMENTS**

| * | | Document Number<br>Country Code-Number-Kind Code | Date<br>MM-YYYY | Name | Classification |
|---|---|---|---|---|---|
| * | A | US-6,563,517 | 05-2003 | Bhagwat et al. | 715/735 |
| * | B | US-6,909,708 | 06-2005 | Krishnaswamy et al. | 370/352 |
| * | C | US-5,845,279 | 12-1998 | Garofalakis et al. | 1/1 |
| | D | US- | | | |
| | E | US- | | | |
| | F | US- | | | |
| | G | US- | | | |
| | H | US- | | | |
| | I | US- | | | |
| | J | US- | | | |
| | K | US- | | | |
| | L | US- | | | |
| | M | US- | | | |

**FOREIGN PATENT DOCUMENTS**

| * | | Document Number<br>Country Code-Number-Kind Code | Date<br>MM-YYYY | Country | Name | Classification |
|---|---|---|---|---|---|---|
| | N | | | | | |
| | O | | | | | |
| | P | | | | | |
| | Q | | | | | |
| | R | | | | | |
| | S | | | | | |
| | T | | | | | |

**NON-PATENT DOCUMENTS**

| * | | Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages) |
|---|---|---|
| | U | |
| | V | |
| | W | |
| | X | |

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

| *Search Notes* | Application/Control No. | Applicant(s)/Patent Under Reexamination |
|---|---|---|
| | 12713637 | BARGER ET AL. |
| | **Examiner** | **Art Unit** |
| | CHARLES MURPHY | 2455 |

## SEARCHED

| Class | Subclass | Date | Examiner |
|---|---|---|---|
| 709 | 263 234 231 226 219 217 204 203 200 | 10/7/2010 | Charles Murphy |
| 370 | 552 468 431 353 352 330 280 241 | 10/7/2010 | Charles Murphy |
| 709 | 263 231 226 219 217 204 203 | 3/22/2011 | Charles Murphy |
| 370 | 468 431 353 352 241 | 3/22/2011 | Charles Murphy |

## SEARCH NOTES

| Search Notes | Date | Examiner |
|---|---|---|
| Searched East | 10/7/2010 | Charles Murphy |
| Searched East | 3/22/2011 | Charles Murphy |
| Consulted Wing Chan for potential 101 issue for claims 1-10 | 3/28/2011 | Charles Murphy |

## INTERFERENCE SEARCH

| Class | Subclass | Date | Examiner |
|---|---|---|---|
| | | | |

/CHARLES MURPHY/
Examiner.Art Unit 2455

|  |  |  |
|---|---|---|
| ***Index of Claims*** | **Application/Control No.**<br>12713637 | **Applicant(s)/Patent Under Reexamination**<br>BARGER ET AL. |
|  | **Examiner**<br>CHARLES MURPHY | **Art Unit**<br>2455 |

|  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
| ✓ | **Rejected** | - | **Cancelled** | N | **Non-Elected** | A | **Appeal** |
| = | **Allowed** | ÷ | **Restricted** | I | **Interference** | O | **Objected** |

□ **Claims renumbered in the same order as presented by applicant** □ **CPA** □ **T.D.** □ **R.1.47**

| **CLAIM** |  | **DATE** |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
| Final | Original | 10/07/2010 | 03/22/2011 |  |  |  |  |  |  |  |
|  | 1 | ✓ | ✓ |  |  |  |  |  |  |  |
|  | 2 | ✓ | - |  |  |  |  |  |  |  |
|  | 3 | ✓ | ✓ |  |  |  |  |  |  |  |
|  | 4 | ✓ | ✓ |  |  |  |  |  |  |  |
|  | 5 | ✓ | ✓ |  |  |  |  |  |  |  |
|  | 6 | ✓ | ✓ |  |  |  |  |  |  |  |
|  | 7 | ✓ | ✓ |  |  |  |  |  |  |  |
|  | 8 | ✓ | ✓ |  |  |  |  |  |  |  |
|  | 9 | ✓ | ✓ |  |  |  |  |  |  |  |
|  | 10 | ✓ | ✓ |  |  |  |  |  |  |  |
|  | 11 | ✓ | ✓ |  |  |  |  |  |  |  |

## EAST Search History

## EAST Search History (Prior Art)

| Ref # | Hits | Search Query | DBs | Default Operator | Plurals | Time Stamp |
|---|---|---|---|---|---|---|
| L1 | 1 | ("6563517").PN. | US-PGPUB; USPAT; USOCR | OR | OFF | 2011/03/22 16:33 |
| L2 | 1 | ("6909708").PN. | US-PGPUB; USPAT; USOCR | OR | OFF | 2011/03/22 16:33 |
| L3 | 1 | 1 preferences same request | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2011/03/22 16:50 |
| S1 | 1 | ("20090254672").PN. | US-PGPUB; USPAT; USOCR | OR | OFF | 2010/10/07 14:44 |
| S2 | 74 | ("5088052" \| "5355472" \| "5442771" \| "5530852" \| "5701451" \| "5708845" \| "5710918" \| "5737619" \| "5745908" \| "5758110" \| "5761555" \| "5761655" \| "5793964" \| "5819261" \| "5822436" \| "5845084" \| "5845299" \| "5860068" \| "5860073" \| "5861881" \| "5862325" \| "5864337" \| "5870552" \| "5880740" \| "5884337" \| "5890170" \| "5890170" \| "5895476" \| "5895476" \| "5895477" \| "5903892" \| "5937160" \| "5943680" \| "5943680" \| "5956737" \| "5956737" \| "6009436" \| "6456305" \| "6484149" \| "6563517" \| "6591280" \| "6591280" \| "6623529" \| "6623529"). PN. | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 14:48 |
| S3 | 2 | S2 (transcod$4) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 14:49 |

| | | | | | | |
|---|---|---|---|---|---|---|
| S4 | 74 | ("5088052" \| "5355472" \| "5442771" \| "5530852" \| "5701451" \| "5708845" \| "5710918" \| "5737619" \| "5745908" \| "5758110" \| "5761555" \| "5761655" \| "5793964" \| "5819261" \| "5822436" \| "5845084" \| "5845299" \| "5860068" \| "5860073" \| "5861881" \| "5862325" \| "5864337" \| "5870552" \| "5880740" \| "5884337" \| "5890170" \| "5890170" \| "5895476" \| "5895476" \| "5895477" \| "5903892" \| "5937160" \| "5943680" \| "5943680" \| "5956737" \| "5956737" \| "6009436" \| "6456305" \| "6484149" \| "6563517" \| "6591280" \| "6591280" \| "6623529" \| "6623529").PN. | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 14:51 |
| S5 | 1 | S4 (transcod$4 same cache) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 14:51 |
| S6 | 34 | (transcod$4) same (content or media or file) near4 (concurrent$3) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 17:40 |
| S7 | 0 | (transcod$4) same (content or media or file) near4 (concurrent$3) @ad<"19991021" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 17:42 |
| S8 | 0 | (transcod$4) same (content or media or file) near4 (concurrent$3) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 17:42 |
| S9 | 15 | (transcod$4) (content or media or file) near4 (concurrent$3) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 17:42 |

| | | | | | | |
|---|---|---|---|---|---|---|
| S10 | 10 | (transcod$4) (content or media or file) same (deliver$3 or send$3 or transmit$3) same (concurrent$3 or simultan $4) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 17:43 |
| S11 | 0 | (transcod$4 same parameter) (content or media or file) same (deliver$3 or send$3 or transmit$3) same (concurrent$3 or simultan $4) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:18 |
| S12 | 2709 | (content or media or file) same (deliver$3 or send $3 or transmit$3) same (concurrent$3 or simultan $4) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:25 |
| S13 | 763 | (content or media or file) near5 (deliver$3 or send $3 or transmit$3) same (concurrent$3 or simultan $4) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:25 |
| S14 | 184 | (content or media or file) near5 (deliver$3 or send $3 or transmit$3) near5 (concurrent$3 or simultan $4) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:25 |
| S15 | 41 | S14 (transcod$3 or convert) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:26 |
| S16 | 1 | S14 (transcod$3) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:26 |
| S17 | 25 | S14 (media.ti.) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:27 |

| | | | | | | |
|---|---|---|---|---|---|---|
| S18 | 3 | (content or media or file) near5 (transcod$3 or convert$3) same (deliver $3 or send$3 or transmit $3) near5 (concurrent$3 or simultan$4) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:30 |
| S19 | 157 | (content or media or file) near5 (transcod$3 or convert$3) (deliver$3 or send$3 or transmit$3) near5 (concurrent$3 or simultan$4) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:30 |
| S20 | 6 | S19 (media.ti.) | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:30 |
| S21 | 22 | (content or media or file or audio or video or movie) near5 (transcod$3 or convert$3) same (deliver$3 or send$3 or transmit$3) near5 (concurrent$3 or simultan $4) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:31 |
| S22 | 1 | ("5845279").PN. | US-PGPUB; USPAT; USOCR | OR | OFF | 2010/10/07 18:35 |
| S23 | 0 | S22 concurent$3 | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:35 |
| S24 | 0 | S22 concurent$4 | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:35 |
| S25 | 1 | S22 concurrent$4 | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2010/10/07 18:35 |

| | | | | | | |
|---|---|---|---|---|---|---|
| S26 | 13 | (transcod$4) same (content or media or file) near4 (base or primary) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2011/03/16 16:10 |
| S27 | 13 | (transcod$4) same (content or media or file) near4 (base or primary or foundation) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2011/03/16 16:10 |
| S28 | 13 | (transcod$4) same (content or media or file) near4 (base or primary or foundation or prilimnary) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2011/03/16 16:11 |
| S29 | 13 | (transcod$4) same (content or media or file) near4 (base or primary or foundation or preliminary) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2011/03/16 16:12 |
| S30 | 0 | (transcod$4) near5 (profile) same (content or media or file) near4 (base or primary or foundation or preliminary) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2011/03/16 16:13 |
| S31 | 1 | (transcod$4) near5 (profile) same (content or media or file) @ay<"1999" | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2011/03/16 16:13 |
| S32 | 1 | ("6359902").PN. | US-PGPUB; USPAT; USOCR | OR | OFF | 2011/03/17 17:35 |
| S33 | 0 | S32 cache | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2011/03/17 17:35 |
| S34 | 1 | ("6563517").PN. | US-PGPUB; USPAT; USOCR | OR | OFF | 2011/03/22 15:22 |
| S35 | 1 | ("5845279").PN. | US-PGPUB; USPAT; USOCR | OR | OFF | 2011/03/22 15:23 |
| S36 | 1 | ("6909708").PN. | US-PGPUB; USPAT; USOCR | OR | OFF | 2011/03/22 15:23 |

| | | | | | | |
|---|---|---|---|---|---|---|
| S37 | 3 | S34 or S35 or S36 | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2011/03/22 15:23 |
| S38 | 0 | S37 cache same media | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2011/03/22 15:23 |
| S39 | 2 | S37 cache | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2011/03/22 15:23 |
| S40 | 1 | S37 cache same transcode $4 | US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB | AND | ON | 2011/03/22 15:24 |

**EAST Search History (Interference)**

<This search history is empty>

**3/22/2011 5:19:59 PM**
**C:\ Documents and Settings\ cmurphy1\ My Documents\ EAST\ Workspaces\ 12713637.wsp**

Doc code: RCEX
Doc description: Request for Continued Examination (RCE)

PTO/SB/30EFS (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

# REQUEST FOR CONTINUED EXAMINATION(RCE)TRANSMITTAL
## (Submitted Only via EFS-Web)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Application Number | 12/713,637 | Filing Date | 2010-02-26 | Docket Number (if applicable) | EQUI0016D2 | Art Unit | 2455 |
| First Named Inventor | Sean Barger | | | Examiner Name | Charles C. Murphy | | |

**This is a Request for Continued Examination (RCE) under 37 CFR 1.114 of the above-identified application.**
Request for Continued Examination (RCE) practice under 37 CFR 1.114 does not apply to any utility or plant application filed prior to June 8, 1995, or to any design application. The Instruction Sheet for this form is located at WWW.USPTO.GOV

SUBMISSION REQUIRED UNDER 37 CFR 1.114

Note: If the RCE is proper, any previously filed unentered amendments and amendments enclosed with the RCE will be entered in the order in which they were filed unless applicant instructs otherwise. If applicant does not wish to have any previously filed unentered amendment(s) entered, applicant must request non-entry of such amendment(s).

□ Previously submitted. If a final Office action is outstanding, any amendments filed after the final Office action may be considered as a submission even if this box is not checked.

- □ Consider the arguments in the Appeal Brief or Reply Brief previously filed on ______
- □ Other ______

☒ Enclosed

- ☒ Amendment/Reply
- □ Information Disclosure Statement (IDS)
- □ Affidavit(s)/ Declaration(s)
- □ Other ______

MISCELLANEOUS

□ Suspension of action on the above-identified application is requested under 37 CFR 1.103(c) for a period of months ______
(Period of suspension shall not exceed 3 months; Fee under 37 CFR 1.17(i) required)

□ Other ______

FEES

☒ **The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.**
The Director is hereby authorized to charge any underpayment of fees, or credit any overpayments, to Deposit Account No 071445

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

☒ Patent Practitioner Signature
□ Applicant Signature

| Signature of Registered U.S. Patent Practitioner | | | |
|---|---|---|---|
| Signature | /MAG/ | Date (YYYY-MM-DD) | 2011-06-30 |
| Name | Michael A. Glenn | Registration Number | 30176 |

This collection of information is required by 37 CFR 1.114. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450.

*If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.*

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

| | | |
|---|---|---|
| First Named Inventor | : | Sean Barger |
| Serial No. | : | 12/713,637 |
| Filed | : | February 26, 2010 |
| Art Unit | : | 2455 |
| Confirmation Number | : | 5103 |
| Examiner | : | Charles C. Murphy |
| Title | : | Automated Media Delivery System |
| Attorney Docket No. | : | EQUI0016D2 |

June 30, 2011

Commissioner for Patents
Mail Stop Amendment
P.O. Box 1450
Alexandria, VA 22313-1450

### RESPONSE TO OFFICE ACTION

Applicant submits this Response to the Office Action dated April 5, 2011 in connection with the above-identified patent application.

A **Listing of Claims** begins on Page 2 of this paper, and

**Remarks** begin on Page 11 of this paper.

The Commissioner is authorized to charge any fees that may be due and to credit any overpayments to Deposit Account 07-1445 (Order No. EQUI0016D2). Applicant considers this document to be filed in a timely manner.

## LISTING OF CLAIMS

1. (Currently Amended) A non-transitory computer program product for causing a computing system to dynamically transcode media content to be presented on a client presentation system, the computer program product comprising:

one or more computer-readable ~~physical and/or memory storage~~ media, the one or more computer-readable ~~memory~~ media having stored thereon computer-executable instructions that, when executed by one or more processors of the computing system, cause the computing system to perform the following:

an act of receiving a request for media content to be delivered to a client presentation system, wherein the requested media content has a limited number of base transcoding profiles associated therewith, each base transcoding profile corresponding to a cached version of the requested media content;

at the time of ~~after receiving~~ the request, and without input by a network administrator, an act of automatically identifying transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system, wherein identification of transcoding parameters is based on one or more formats of any client presentation system;

an act of transcoding the requested media content in accordance with the identified transcoding parameters, such that the identified transcoding parameters are used to perform additional incremental transcoding on top of the base transcoding profile;

an act of causing the transcoded media content to be delivered to the client presentation system; and an act of caching the transcoded media content.

2. (Cancelled).

3. (Currently Amended) A non-transitory computer program product in accordance with claim 1, wherein the act of receiving a request for media content is performed by receiving the request over the Internet.

4. (Currently Amended) A non-transitory computer program product in accordance with claim 1, wherein the client presentation system is a first client presentation system, the method further comprising:

an act of receiving a request for the media content to be delivered to a second client presentation system;

an act of determining that the transcoding parameters to be applied to the requested media content prior to delivery to the second client presentation system are the same as the transcoding parameters that were applied to the requested media content prior to delivery to the first client presentation system; and

an act of delivering the cached transcoded media content, rather than separately transcoding the requested media content again for the second client presentation system.

5. (Currently Amended) A non-transitory computer program product in accordance with claim 1, wherein the requested media content is video media.

6. (Currently Amended) A non-transitory computer program product in accordance with claim 1, wherein the identified transcoding parameters includes at least video size parameters.

7. (Currently Amended) A non-transitory computer program product in accordance with claim 1, wherein the method further comprises:

prior to the act of transcoding, an act of determining that a transcoded representation of the requested media content transcoded using the identified transcoding parameters does not already exist.

8. (Currently Amended) A non-transitory computer program product in accordance with claim 1, wherein the act of identifying transcoding parameters comprises an act of selecting a transcoding profile from a set of pre-existing transcoding profiles.

9. (Currently Amended) A non-transitory computer program product in accordance with claim 1, wherein the act of identifying transcoding parameters comprises:

an act of reading one or more transcoding parameters from the request; and

an act of identifying the one or more transcoding parameters from the request as being the identified transcoding parameters to be applied.

10. (Currently Amended) A non-transitory computer program product in accordance with claim 1, wherein the act of act of transcoding the requested media content in accordance with the identified transcoding parameters comprises:

an act of selecting a pre-existing base transcoded version of the requested media content that has been transcoded in accordance with only a portion of the identified transcoding parameters; and

an act of incrementally performing further transcoding of the pre-existing base transcoded version in accordance with a remaining portion of the identified transcoding parameters.

11. (Currently Amended) A method for accessing dynamically transcoding media content, the method comprising:

an act of receiving a request for media content to be delivered to a client presentation system for media content, wherein the requested media content has a limited number of base transcoding profiles associated therewith, each base transcoding profile corresponding to a cached version of the requested media content;

at the time of ~~after receiving~~ the request, and without input by a network administrator, an act of automatically identifying transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system, wherein identification of transcoding parameters is based on one or more formats of any client presentation system;

an act of determining that the transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system are the same as transcoding parameters that are being applied to the requested media content prior to delivery to another client presentation system;

an act of transcoding the requested media content in accordance with the identified transcoding parameters, such that the identified transcoding parameters are used to perform additional incremental transcoding on top of the base transcoding profile; and

an act of delivering the transcoded media content to both client presentation systems concurrently.

12. (New) A computer program product for causing a computing system to dynamically transcode media content to be presented on a client presentation system, the computer program product comprising:

one or more computer-readable physical and/or memory storage media, the one or more computer-readable memory media having stored thereon computer-executable instructions that, when executed by one or more processors of the computing system, cause the computing system to perform the following:

maintaining a transcoding profile library, the transcoding profile library comprising a plurality of transcoding profiles for media content;

an act of receiving a request for media content to be delivered to a client presentation system, wherein the requested media content has a limited number of base transcoding profiles associated therewith, each base transcoding profile corresponding to a cached version of the requested media content;

an act of identifying transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system;

determining whether an existing base transcoding profile exists within the transcoding profile library corresponds to the requested media content transcoded for delivery to the client presentation system;

an act of transcoding the requested media content in accordance with the identified transcoding parameters, such that the identified transcoding parameters are used to perform additional incremental transcoding based on multiple transcoding parameters on top of a base transcoding profile;

an act of delivering the transcoded media content to the client presentation system;

adding a new transcoding profile to the transcoding profile library corresponding to the identified transcoding parameters and the base transcoding profile; and

an act of caching the transcoded media content.

13. (New) A computer program product in accordance with Claim 2, wherein the act of receiving a request for media content is performed by receiving the request over the Internet.

14. (New) A computer program product in accordance with Claim 2, wherein the client presentation system is a first client presentation system, the method further comprising:

an act of receiving a request for the media content to be delivered to a second client presentation system;

an act of determining that the transcoding parameters to be applied to the requested media content prior to delivery to the second client presentation system are the same as the transcoding parameters that were applied to the requested media content prior to delivery to the first client presentation system; and

an act of delivering the cached transcoded media content, rather than separately transcoding the requested media content again for the second client presentation system.

15. (New) A computer program product in accordance with Claim 14, wherein act of caching the transcoded media content comprises an act of first caching a transcoded stream of the transcoded media content while the act of transcoding is still occurring, and further caching a transcoded file of the transcoded media content after the act of transcoding has completed.

16. (New) A computer program product in accordance with Claim 15, wherein the act of delivery the cached transcoded media content comprises:

an act of providing the transcoded stream to the second client presentation system.

17. (New) A computer program product in accordance with Claim 15, wherein the act of delivery the cached transcoded media content comprises:

an act of providing the transcoded file to the second client presentation system.

18. (New) A computer program product in accordance with Claim 13, wherein the requested media content is video media.

19. (New) A computer program product in accordance with Claim 13, wherein the identified transcoding parameters includes at least video size parameters.

20. (New) A computer program product in accordance with Claim 13, wherein the method further comprises:

prior to the act of transcoding, an act of determining that a transcoded representation of the requested media content transcoded using the identified transcoding parameters does not already exist.

21. (New) A computer program product in accordance with Claim 13, wherein the act of identifying transcoding parameters comprises an act of selecting a transcoding profile from a set of pre-existing transcoding profiles.

22. (New) A computer program product in accordance with Claim 13, wherein the act of identifying transcoding parameters comprises:

an act of reading one or more transcoding parameters from the request; and

an act of identifying the one or more transcoding parameters from the request as being the identified transcoding parameters to be applied.

23. (New) A computer program product in accordance with Claim 13, wherein the act of act of transcoding the requested media content in accordance with the identified transcoding parameters comprises:

an act of selecting a pre-existing base transcoded version of the requested media content that has been transcoded in accordance with only a portion of the identified transcoding parameters; and

an act of incrementally performing further transcoding of the pre-existing base transcoded version in accordance with a remaining portion of the identified transcoding parameters.

## REMARKS

Applicants respectfully request further examination and reconsideration in view of the above amendments and arguments set forth fully below.

### Claim Objections

Within the Office Action, Claims 1 and 3-10 were objected to for various formalities. The Applicants amend Claims 1 and 3-10 herein, thereby rendering the objections moot.

### Claim Rejections under 35 U.S.C. § 101

Within the Office Action, Claims 1-11 were rejected under 35 USC 101 for allegedly being directed to nonstatutory subject matter. Applicants amend the claims herein, thereby rendering the rejection moot.

### Claim Rejections under 35 U.S.C. § 103

Also within the Office Action, Claims 1 and 3-10 were rejected under 35 U.S.C. § 103(a) as being unpatentable over United States Patent No. 6,563,517 to Bhagwat (hereinafter referred to as "Bhagwat") in view of United States Patent No. 6,909,708 to Krishnaswamy et al. (hereinafter referred to as "Krishnaswamy").

To establish a *prima facie* case of obviousness of a claimed invention, all the claimed features must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). The Applicants respectfully traverse this rejection because neither Bhagwat nor Krishnaswamy disclose all of the limitations of Claims 1 and 3-10.

Specifically neither Bhagwat nor Krishnaswamy teach or suggest:

> "at the time of the request, and without input by a network administrator, an act of automatically identifying transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system, wherein identification of transcoding parameters is based on one or more formats of any client presentation system."

Within the Office Action, the Examiner states that "Bhagwat teaches delivering the requested transcoded content to the requesting client." The Applicants disagree; however, for the sole purpose of expediting prosecution, the Applicants amend the claims to recite that the transcoding parameters are identified "at the time of the request" and "without input by the network administrator." Furthermore, while Bhagwat is primarily involved with transcoding based on network bandwidth, the Applicants amend the claims to specify that "identification of transcoding parameters is based on one or more formats of any client presentation system." Accordingly, Bhagwat does not teach or suggest: "at the time of the request, and without input by a network administrator, an act of automatically identifying transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system, wherein identification of transcoding parameters is based on one or more formats of any client presentation system."

Likewise, Krishnaswamy does not teach or suggest the above limitation, nor does the Examiner suggest that it does.

For at least this reason, Claims 1 and 3-10 are not rendered obvious by a hypothetical combination of Bhagwat and Krishnaswamy.

Also within the Office Action, Claim 11 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Bhagwat in view of Krishnaswamy and further in view of United States Patent No. 5,845,279 to Garofalakis et al. (hereinafter referred to as "Garofalakis"). The Applicants respectfully traverse this rejection neither Bhagwat, Krishnaswamy, nor Garofalakis, neither alone nor in combination, disclose all of the limitations of Claim 11.

Within the Office Action, the Examiner admits that neither Bhagwat nor Krishnaswamy teach or suggest "an act of delivering the transcoded media content to both client presentation systems concurrently." The Applicants agree. However, the Examiner cites Garofalakis to allege that the hypothetical combination of Bhagwat, Krishnaswamy, and Garofalakis teach each of the Claim limitations. However, neither Bhagwat, Krishnaswamy, nor Garofalakis teach or suggest"

> "at the time of the request, and without input by a network administrator, an act of automatically identifying transcoding parameters to be applied to the requested media content prior to delivery to the client presentation system, wherein identification of transcoding parameters is based on one or more formats of any client presentation system."

As explained above, Bhagwat and Krishnaswamy fail to teach the above limitation. Additionally, Garofalakis fails to the this feature, nor does the Examiner suggest that he does.

For at least this reason, Claim 11 is not rendered obvious by a hypothetical combination of Bhagwat, Krishnaswamy and Garfalakis.

For the record, the Applicant respectfully traverses any and all factual assertions in the file that are not supported by documentary evidence. Such include assertions based on findings of inherency, assertions based on Official Notice, and any other assertions of what is well known or commonly known in the prior art.

Additionally, the foregoing amendments are made solely in the interest of expediency, in recognition of the Office policy of compact prosecution. They do not indicate agreement by Applicant with the Office's position, nor do they reflect intent to forsake Claim scope. In fact, Applicant expressly reserves the right to pursue patent protection of a scope it is reasonably entitled to in future submissions to the Office.

## CONCLUSION

Applicant respectfully posits that the pending claims have been distinguished from the art of record, and that all objections to and rejections of the claims have been overcome. Accordingly, Applicant respectfully requests allowance. Should the Examiner deem it helpful he is encouraged to contact Applicant's attorney at (650) 474-8400.

Respectfully submitted,

Michael A. Glenn
Reg. No. 30,176

Customer No. 22862

# Electronic Patent Application Fee Transmittal

| | |
|---|---|
| **Application Number:** | 12713637 |
| **Filing Date:** | 26-Feb-2010 |
| **Title of Invention:** | Automated Media Delivery System |
| **First Named Inventor/Applicant Name:** | Sean BARGER |
| **Filer:** | Michael Glenn/Jessica Pallach |
| **Attorney Docket Number:** | EQUI0016D2 |

Filed as Small Entity

## Utility under 35 USC 111(a) Filing Fees

| Description | Fee Code | Quantity | Amount | Sub-Total in USD($) |
|---|---|---|---|---|
| **Basic Filing:** | | | | |
| **Pages:** | | | | |
| **Claims:** | | | | |
| **Miscellaneous-Filing:** | | | | |
| **Petition:** | | | | |
| **Patent-Appeals-and-Interference:** | | | | |
| **Post-Allowance-and-Post-Issuance:** | | | | |
| **Extension-of-Time:** | | | | |

| Description | Fee Code | Quantity | Amount | Sub-Total in USD($) |
|---|---|---|---|---|
| **Miscellaneous:** | | | | |
| Request for continued examination | 2801 | 1 | 405 | 405 |
| | | **Total in USD ($)** | | **405** |

| Electronic Acknowledgement Receipt | |
|---|---|
| EFS ID: | 10433375 |
| Application Number: | 12713637 |
| International Application Number: | |
| Confirmation Number: | 5103 |
| Title of Invention: | Automated Media Delivery System |
| First Named Inventor/Applicant Name: | Sean BARGER |
| Customer Number: | 22862 |
| Filer: | Michael Glenn/Jessica Pallach |
| Filer Authorized By: | Michael Glenn |
| Attorney Docket Number: | EQUI0016D2 |
| Receipt Date: | 30-JUN-2011 |
| Filing Date: | 26-FEB-2010 |
| Time Stamp: | 18:24:32 |
| Application Type: | Utility under 35 USC 111(a) |

## Payment information:

| | |
|---|---|
| Submitted with Payment | yes |
| Payment Type | Deposit Account |
| Payment was successfully received in RAM | $405 |
| RAM confirmation Number | 8073 |
| Deposit Account | 071445 |
| Authorized User | |

## File Listing:

| Document Number | Document Description | File Name | File Size(Bytes)/ Message Digest | Multi Part /.zip | Pages (if appl.) |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| 1 | | RCEresponse-EQUI0016D2--efiled063011.pdf | 878646<br>0536a941c1346fa3f792078692b4f226716ff9a7 | yes | 17 |

**Multipart Description/PDF files in .zip description**

| Document Description | Start | End |
|---|---|---|
| Request for Continued Examination (RCE) | 1 | 2 |
| Amendment Submitted/Entered with Filing of CPA/RCE | 3 | 3 |
| Claims | 4 | 12 |
| Applicant Arguments/Remarks Made in an Amendment | 13 | 17 |

**Warnings:**

**Information:**

| | | | | | |
|---|---|---|---|---|---|
| 2 | Fee Worksheet (SB06) | fee-info.pdf | 30090<br>347184b11f2e26a231ef0762595bbdbbc703de43 | no | 2 |

**Warnings:**

**Information:**

| | |
|---|---|
| **Total Files Size (in bytes):** | 908736 |

**This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.**

**New Applications Under 35 U.S.C. 111**
**If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.**

**National Stage of an International Application under 35 U.S.C. 371**
**If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.**

**New International Application Filed with the USPTO as a Receiving Office**
**If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.**

Document code: WFEE

United States Patent and Trademark Office
Sales Receipt for Accounting Date: 07/08/2011

```
JMASON     SALE   #00000001    Mailroom Dt:  06/30/2011    071445   12713637
                       01     FC : 2202                  52.00  DA
```

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

| | | | |
|---|---|---|---|
| **PATENT APPLICATION FEE DETERMINATION RECORD**<br>Substitute for Form PTO-875 | Application or Docket Number<br>12/713,637 | Filing Date<br>02/26/2010 | □ To be Mailed |

APPLICATION AS FILED – PART I

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | (Column 1) | (Column 2) | SMALL ENTITY ☒ | | OR | OTHER THAN SMALL ENTITY | |
| FOR | NUMBER FILED | NUMBER EXTRA | RATE ($) | FEE ($) | | RATE ($) | FEE ($) |
| □ BASIC FEE (37 CFR 1.16(a), (b), or (c)) | N/A | N/A | N/A | | | N/A | |
| □ SEARCH FEE (37 CFR 1.16(k), (i), or (m)) | N/A | N/A | N/A | | | N/A | |
| □ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q)) | N/A | N/A | N/A | | | N/A | |
| TOTAL CLAIMS (37 CFR 1.16(i)) | minus 20 = | * | X $ = | | OR | X $ = | |
| INDEPENDENT CLAIMS (37 CFR 1.16(h)) | minus 3 = | * | X $ = | | | X $ = | |
| □APPLICATION SIZE FEE (37 CFR 1.16(s)) | If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s). | | | | | | |
| □ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j)) | | | | | | | |
| | | | TOTAL | | | TOTAL | |

* If the difference in column 1 is less than zero, enter "0" in column 2.

APPLICATION AS AMENDED – PART II

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | (Column 1) | | (Column 2) | (Column 3) | SMALL ENTITY | | OR | OTHER THAN SMALL ENTITY | |
| AMENDMENT | **06/30/2011** | CLAIMS REMAINING AFTER AMENDMENT | | HIGHEST NUMBER PREVIOUSLY PAID FOR | PRESENT EXTRA | RATE ($) | ADDITIONAL FEE ($) | | RATE ($) | ADDITIONAL FEE ($) |
| | Total (37 CFR 1.16(i)) | * 22 | Minus | ** 20 | = 2 | X $26 = | 52 | OR | X $ = | |
| | Independent (37 CFR 1.16(h)) | * 3 | Minus | ***3 | = 0 | X $110 = | 0 | OR | X $ = | |
| | □ Application Size Fee (37 CFR 1.16(s)) | | | | | | | | | |
| | □ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j)) | | | | | | | OR | | |
| | | | | | | TOTAL ADD'L FEE | **52** | OR | TOTAL ADD'L FEE | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | (Column 1) | | (Column 2) | (Column 3) | | | | | |
| AMENDMENT | | CLAIMS REMAINING AFTER AMENDMENT | | HIGHEST NUMBER PREVIOUSLY PAID FOR | PRESENT EXTRA | RATE ($) | ADDITIONAL FEE ($) | | RATE ($) | ADDITIONAL FEE ($) |
| | Total (37 CFR 1.16(i)) | * | Minus | ** | = | X $ = | | OR | X $ = | |
| | Independent (37 CFR 1.16(h)) | * | Minus | *** | = | X $ = | | OR | X $ = | |
| | □ Application Size Fee (37 CFR 1.16(s)) | | | | | | | | | |
| | □ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j)) | | | | | | | OR | | |
| | | | | | | TOTAL ADD'L FEE | | OR | TOTAL ADD'L FEE | |

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/JAMES MASON/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

*If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.*

| | |
|---|---|
| **Form 1449 (Modified)** | **Serial No.: 12/713,637** |
| | **Atty. Docket No.: EQUI0016D2** |
| **Information Disclosure** | **First Named Inventor: Sean BARGER** |
| **Statement By Applicant** | **Art Unit: 2455** |
| | **Confirmation No.: 5103** |
| (Use Several Sheets if Necessary) | **Filing Date: February 26, 2010** |

**U.S. Patent Documents**

| Examiner Initials | No. | Patent No. | Issue Date | Patentee | Filing Date |
|---|---|---|---|---|---|
| | 1 | 5,088,052 | 02/1992 | Spielman, et al. | |
| | 2 | 5,355,472 | 10/1994 | Lewis | |
| | 3 | 5,442,771 | 08/1995 | Filepp, et al. | |
| | 4 | 5,530,852 | 06/1996 | Meske, Jr., et al. | |
| | 5 | 5,701,451 | 12/1997 | Rogers, et al. | |
| | 6 | 5,708,845 | 01/1998 | Wistendahl, et al. | |
| | 7 | 5,710,918 | 01/1998 | Lagarde, et al. | |
| | 8 | 5,737,619 | 04/1998 | Judson | |
| | 9 | 5,745,908 | 04/1998 | Anderson, et al. | |
| | 10 | 5,758,110 | 05/1998 | Boss, et al. | |
| | 11 | 5,761,655 | 06/1998 | Hoffman, Michael T. | |
| | 12 | 5,793,964 | 08/1998 | Rogers, et al. | |
| | 13 | 5,819,261 | 10/1998 | Takahashi, et al. | |
| | 14 | 5,822,436 | 10/1998 | Rhoads | |
| | 15 | 5,845,084 | 12/1998 | Cordell, et al. | |
| | 16 | 5,845,279 | 12/1998 | Garofalakis, et al. | |
| | 17 | 5,845,299 | 12/1998 | Arora, et al. | |
| | 18 | 5,860,068 | 01/1999 | Cook | |
| | 19 | 5,860,073 | 01/1999 | Ferrel, et al. | |
| | 20 | 5,861,881 | 01/1999 | Freeman, et al. | |
| | 21 | 5,862,325 | 01/1999 | Reed, et al. | |
| | 22 | 5,864,337 | 01/1999 | Marvin, John | |
| | 23 | 5,870,552 | 02/1999 | Dozier, et al. | |
| | 24 | 5,880,740 | 03/1999 | Halliday, et al. | |
| | 25 | 5,890,170 | 03/1999 | Sidana | |
| | 26 | 5,895,476 | 04/1999 | Orr, et al. | |
| | 27 | 5,895,477 | 04/1999 | Orr, et al. | |
| | 28 | 5,903,892 | 05/1999 | Hoffert, et al. | |
| | 29 | 5,937,160 | 08/1999 | Davis, et al. | |
| | 30 | 5,943,680 | 08/1999 | Ohga, et al. | |
| | 31 | 5,956,737 | 09/1999 | King, et al. | |
| | 32 | 6,009,436 | 12/1999 | Motoyama, et al. | |
| | 33 | 6,456,305 | 09/2002 | Qureshi, et al. | |
| | 34 | 6,483,851 | 11/2002 | Neogi | |
| | 35 | 6,484,149 | 11/2002 | Jemmes, et al. | |

| | | | | | |
|---|---|---|---|---|---|
| | 36 | 6,563,517 | 05/2003 | Bhagwat, et al. | |
| | 37 | 6,591,280 | 07/2003 | Orr | |
| | 38 | 6,623,529 | 09/2003 | Lakritz | |
| | 39 | 6,909,708 | 06/2005 | Krishnaswamy, et al. | |
| | 40 | 6,938,073 | 08/2005 | Mendhekar, et al. | |
| | 41 | 7,284,201 | 10/2007 | Cohen-Solal, Eric | |
| | 42 | 7,477,688 | 01/2009 | Zhang, et al. | |
| | 43 | 7,673,063 | 03/2010 | Xie, et al. | |

### Published U.S, Patent Application

| Examiner Initials | No. | Publication No. | Publication Date | Applicant |
|---|---|---|---|---|
| | 41 | 2006/0015580 | 01/2006 | Gabriel, et al. |
| | 42 | 2008/0155230 | 06/2008 | Robbins et al |
| | 43 | 2009/0254672 | 10/2009 | Zhang |
| | 44 | 2009/0070485 | 03/2009 | Barger, et al. |
| | 45 | 2009/0089422 | 04/2009 | Barger, et al. |

### Foreign Patent or Published Foreign Patent Application

| Examiner Initials | No. | Document No. | Publication Date | Applicant |
|---|---|---|---|---|
| | 46 | AU-A-53031/98 | 08/1998 | Dudley, John Mills |
| | 47 | EP 0747842 | 12/1996 | International Business Machines Corp. |
| | 48 | EP 0782085 | 07/1997 | International Business Machines Corp. |
| | 49 | EP 0818907 | 01/1998 | AT&T Corp |
| | 50 | EP 0843276 | 05/1998 | Canon Information Systems Inc. |
| | 51 | EP 0876034 | 11/1998 | International Business Machines Corp. |
| | 52 | EP 0883068 | 12/1998 | Home Information Services, Inc. |
| | 53 | EP 0886409 | 12/1998 | Digital Vision Laboratories Corporation |
| | 54 | EP 0895171 | 02/1999 | Neoforma, Inc. |
| | 55 | EP 0926607 | 06/1999 | Ricoh KK |
| | 56 | EP 0949571 | 10/1999 | Xerox Corp |
| | 57 | WO 97/49252 | 12/1997 | Senthilkumar, et al. |
| | 58 | WO 98/40842 | 09/1998 | Computer Information and Sciences, Inc. |
| | 59 | WO 98/43177 | 10/1998 | Intel Corp |

### Non-Patent Literature Documents

| Examiner Initials | No. | Author, Title, Date, Place (e.g. Journal) of Publication |
|---|---|---|
| | A | BERINSTEIN, Paula; "The Big Picture; Text and Graphics on UMI's ProQuest Direct: The Best (Yet) of Both Words"; March 1997; retrieved on 3/23/04 from website: http://www.infotoday.com/online/MarOL97/picture3.html, 11 pages. |
| | B | BULTERMAN, DICK.C.A.; Models, Media and Motion: Using the Web to Support Multimedia Documents; Proceedings of 1997 Int'l Conf. on Multimedia Modeling; p. 17-20; November 1997; SINGAPORE |
| | C | DOBSON, R.; Animating Your Web Pages with Direct Animation; Web Techniques; vol.3, no. 6, p. 49-52; June 1998 |

| | | |
|---|---|---|
| | D | McNeil, Sara; Research Interests; retrieved on March 18, 2004 from website: http//www.coe.uh.edu/`smcneil/research.htm, 3 pages. |
| | E | MOHLER, J.L.; Migrating Course Materials to the World Wide Web: A Case Study of the Department of Technical Graphics at Purdue University; Computer Networks and ISDN Systems; Vol. 30, Issues 20-21, pp.1981-1990; November 12, 1988 |
| | F | Sakaguchi, et al.; " A browsing tool for multi-lingual documents for users without multi-lingual fonts"; 1996; ACM International Conference On Digital Libraries, pp. 63-71 |
| | G | Tables of Contents service for Computers & Geosciences; Copyright 1997; Computers and GeoSciences, Volume 23, Issue 5, retrieved on 3/18/04 from website: http://library.iem.ac.ru/comp&geo/00983004/sz977014.html, 2 pages. |
| | H | Zaiane, et al.; "Mining multimedia data"; Nov. 1998; ACM Conference of the Center for Advanced Studies on Collaborative research, pp. 1-18 |

Examiner's Signature:________________________________ Date:______________

Examiner: Initial citation considered. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

# Electronic Acknowledgement Receipt

| | |
|---|---|
| **EFS ID:** | 10439689 |
| **Application Number:** | 12713637 |
| **International Application Number:** | |
| **Confirmation Number:** | 5103 |
| **Title of Invention:** | Automated Media Delivery System |
| **First Named Inventor/Applicant Name:** | Sean BARGER |
| **Customer Number:** | 22862 |
| **Filer:** | Michael Glenn/Christine Ortt |
| **Filer Authorized By:** | Michael Glenn |
| **Attorney Docket Number:** | EQUI0016D2 |
| **Receipt Date:** | 01-JUL-2011 |
| **Filing Date:** | 26-FEB-2010 |
| **Time Stamp:** | 14:28:40 |
| **Application Type:** | Utility under 35 USC 111(a) |

## Payment information:

| | |
|---|---|
| Submitted with Payment | no |

## File Listing:

| Document Number | Document Description | File Name | File Size(Bytes)/ Message Digest | Multi Part /.zip | Pages (if appl.) |
|---|---|---|---|---|---|
| 1 | Foreign Reference | 46-AUA53031-98.pdf | 1705632<br>410c5e3f5665f1ac14f8282911acfdba41cf54a1 | no | 32 |

**Warnings:**

**Information:**

| | | | | | |
|---|---|---|---|---|---|
| 2 | Foreign Reference | 47-EP0747842.pdf | 1664575<br>2af9850cc271a2199bf6d59e70568e93d507f6c1 | no | 27 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 3 | Foreign Reference | 48-EP0782085.pdf | 791335<br>2b7c787340dafd993b77b5b55fa1e4343e211d36 | no | 18 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 4 | Foreign Reference | 49-EP0818907.pdf | 542895<br>b690be55020c3094e256e0988aa06af4a7fe681d | no | 8 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 5 | Foreign Reference | 50-EP0843276.pdf | 1854757<br>eb56a7b43f7c18e6a7d7c9a1e2b128bac6789fe3 | no | 42 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 6 | Foreign Reference | 51-EP0876034.pdf | 1026073<br>26f5619a632d73360d25feb2506f03da590b7130 | no | 16 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 7 | Foreign Reference | 52-EP0883068.pdf | 1172032<br>fa47e50bf1595b8f41ebc3335f55a7d26cb432ef | no | 22 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 8 | Foreign Reference | 53-EP0886409.pdf | 1365504<br>dbb211b2c33c03334e6171f7d45569c580bcc3b4 | no | 28 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 9 | Foreign Reference | 54-EP0895171.pdf | 2463238<br>d4e5c48419de6f5f3edab9a42a57b6686ef1603e | no | 39 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 10 | Foreign Reference | 55-EPO926607.pdf | 3783168<br>82f99a0149416291f9775b3e6a903a8202ac47b0 | no | 74 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 11 | Foreign Reference | 56-EPO949571.pdf | 2246210<br>e090ac4df027a18cb51a141b53a1f824a9b3788b | no | 40 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 12 | Foreign Reference | 57-WO97-49252.pdf | 1511607<br>7bfa5f299ba517e99c8f27e90c68e0efe07676b3 | no | 50 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 13 | Foreign Reference | 58-WO98-40842.pdf | 3996847<br>f631e807a428f3eb336305478b1ce88ffbff0020 | no | 118 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 14 | Foreign Reference | 59-WO98-43177.PDF | 1839850<br>054919e44072c80d458b1a87557159880acbaef1 | no | 46 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 15 | Non Patent Literature | A-Berinstein-TheBigPicture.PDf | 526864<br>055c120228c026c7ce046cec7039c80a2b8b6968 | no | 11 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 16 | Non Patent Literature | B-Bulterman-ModelsMediaAndMotion.PDF | 690629<br>0345df97b463fa006f0df427272df739d9fad73d | no | 12 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 17 | Non Patent Literature | C-Dobson-AnimatingYourWebPages.PDF | 293322<br>759a70d5547c34da189436bcd3acf15a66c914a7 | no | 5 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 18 | Non Patent Literature | D-McNeil-ResearchInterests.PDF | 100387<br>a9668ca00802930bbdd9f94836d40b2e61c13419 | no | 3 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 19 | Non Patent Literature | E-Mohler-MigratingCourseMaterials.PDF | 77628<br>7f940080689bd2c8d05a89945320fcabae8e32e6 | no | 2 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 20 | Non Patent Literature | F-Sakaguchi-ABrowsingTool.PDF | 494426<br>1d96d1dc1240a3508d5539ad0bc687b614dae8fd | no | 9 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 21 | Non Patent Literature | G-GeoSciences-TableOfContents.PDF | 45795<br>c41562a3bd6be90a4d9142499ca7dbeb799677e2 | no | 2 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 22 | Non Patent Literature | H-Zaiane-MiningMultimediaData.pdf | 1370264<br>564270bf9ee4122477794da53486be356ccdad6d | no | 19 |
| **Warnings:** | | | | | |
| **Information:** | | | | | |
| 23 | | EQUI0016D2-IDS-2011-07-01.pdf | 804582<br>4b337ac5cd7971495b03ea7c6fa96c6fc3cab039 | yes | 5 |

**Multipart Description/PDF files in .zip description**

| Document Description | Start | End |
|---|---|---|
| Transmittal Letter | 1 | 2 |
| Information Disclosure Statement (IDS) Form (SB08) | 3 | 5 |

| | |
|---|---|
| **Warnings:** | |
| **Information:** | |
| **Total Files Size (in bytes):** | 30367620 |

**This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.**

**New Applications Under 35 U.S.C. 111**
**If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.**

**National Stage of an International Application under 35 U.S.C. 371**
**If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.**

**New International Application Filed with the USPTO as a Receiving Office**
**If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.**

ORIGINAL

# AUSTRALIA

Patents Act 1990

# COMPLETE SPECIFICATION

**APPLICANT:** Dudley John MILLS

**ADDRESS:** 30 Hutchison Crescent,
Kambah, ACT 2902

**ACTUAL INVENTOR:** Dudley John MILLS

**ADDRESS FOR SERVICE:** Dudley J. Mills,
30 Hutchison Crescent,
Kambah, ACT2902

ASSOCIATED PROVISIONAL: PO5254 Filed 21 February 1997

**INVENTION TITLE:** NETWORK-BASED CLASSIFIED
INFORMATION SYSTEMS

The following is a full description of the invention including the best method of performing it known to me:

AU9853031

# (12) PATENT ABSTRACT (11) Document No. AU-A-53031/98
# (19) AUSTRALIAN PATENT OFFICE

(54) Title
NETWORK-BASED CLASSIFIED INFORMATION SYSTEMS

International Patent Classification(s)
(51)6 G06F 017/30

(21) Application No. : 53031/98 (22) Application Date : 10/02/98

(30) Priority Data

| (31) Number | (32) Date | (33) Country |
|---|---|---|
| PO5254 | 21/02/97 | AU AUSTRALIA |

(43) Publication Date : 27/08/98

(71) Applicant(s)
DUDLEY JOHN MILLS

(72) Inventor(s)
DUDLEY JOHN MILLS

(57)

A system for automatically creating databases containing industry, service, product and subject classification data, contact data, geographic location data (CCG-data) and links to web pages from HTML, XML or SGML encoded web pages posted on computer networks such as the Internet or Intranets. The web pages containing HTML, XML or SGML encoded CCG-data, database update controls and web browser display controls are created and modified by using simple text editors, HTML, XML or SGML editors or purpose built editors. The CCG databases may be searched for references (URLs) to web pages by use of enquiries which reference one or more of the items of the CCG-data. Alternatively, enquiries referencing the CCG-data in the databases may supply contact data without web page references. Data duplication and coordination is reduced by including in the web page CCG-data display controls which are used by web browsers to format for display the same data that is used to automatically update the databases.

TITLE: NETWORK BASED CLASSIFIED INFORMATION SYSTEMS

## FIELD OF INVENTION

This invention relates to network based classified information systems, to methods of automatically building searchable databases of classified information derived from web pages posted on a network, and, to web pages for use in such systems and methods.

The information systems and databases of most relevance to this invention are those which include classified product and service catalogues similar to the Yellow Pages telephone books, contact indexes similar to the White Pages telephone books, and/or subject indexes similar to Library catalogues. Such information systems and databases typically include sets of associated classification, contact and/or geographic items of information. For convenience, classification, contact and/or geographic information will be hereinafter called CCG-data.

The networks with which this invention is concemed are the worldwide public computer/communications network commonly known as the Internet and private networks – sometimes called intranets – which allow common access to markup documents on computers connected to the network. Markup documents are text files prepared using various markup languages such as HyperText Markup Language (HTML) and Extensible Markup Language (XML) which are implementations (or dialects) of the Standard Generalised Markup Language (SGML). The system of accessible files on the Internet is called the World Wide Web (WWW) and the markup documents themselves are commonly called 'web pages'. A web page is said to be 'posted' on a network when it is stored on computer-readable media of a host network computer as a file which is generally accessible to network users. A web page is transported from the host computer to a requesting computer through intermediate network computers as a computer-readable signal embodied in a carrier wave. Though this invention is not limited to Internet based information systems, these terms are used for convenience.

## BACKGROUND TO THE INVENTION

It has been estimated that there are about 100 million web pages on the Internet and that the number is doubling every two years. Many of these pages include information concerning commercially offered goods and services and often include contact details. But the difficulty of locating such information is increasing faster than the growth in the number of web pages.

To assist network users locate web pages of interest, certain network service providers create indexes (or databases) of the contents of web pages posted (stored on computer readable media so as to be generally accessible) on the network and provide 'search engines' to use the indexes. These indexes are often created automatically by the use of 'web crawlers' which (i) interrogate computer after computer on the network to locate successive web pages and (ii) index the words in each web page encountered against the network address (eg Internet Protocol Address or IPA) and filing system path or universal resource locator (URL) at which the web page is accessible. Hereinafter the terms URL and URI (Uniform Resource Identifier) are taken to be identical in meaning and to signify network addresses and filing system paths. Usually, the indexes consist of a list of unique words with each word having an associated list of URLs of the web pages wherein the word was found to occur during interrogation. The URL serves as a 'hyperlink' which, if selected by a user/searcher, results in the associated web page being automatically transmitted from the computer where it is posted on the network to the user/searcher's computer where it may be displayed or otherwise processed. The sending and receiving of files in this way is greatly assisted by user interface programs called 'web browsers' (or more simply, 'browsers') such as Netscape and Microsoft Internet Explorer.

The search for web pages of interest using search engines leaves much to be desired:

- simple searches (those using a few keywords in simple combinations) often yield far too many web page references (URLs) to permit them to be interrogated one-by-one,
- complex searches (those using many keywords and/or complex Boolean expressions) require considerable expertise to undertake,
- even using optimum search criteria, many irrelevant web pages are referenced because of inconsistent use of terminology by those who author the original web pages,
- even using optimum search criteria, many relevant pages are missed, again because of inconsistent use of terminology by web page authors, and
- because items of information included in the body of web pages cannot be 'understood' or associated in useful ways by web crawlers; that is recognised as, say, a surname, a street name, a geographic locality, or type of goods or services and, say, a surname strongly associated with a street name, a geographic locality, or a type of goods or service.

The result is that information provided by search engines from databases which are automatically compiled using web crawlers is a very poor equivalent of the common Yellow Pages and White Pages directories which serve the telephone industry (though these directories are not, of course, automatically compiled from web pages).

In an attempt to improve the usefulness of automatically compiled network databases, some search engine providers make use of information contained in URLs, such as the country code and top level domain name codes such as 'com', 'edu', 'net' and 'org' which is sometimes used to signify the subject matter of web pages. It has been proposed to add more content classifying codes to URLs (eg, "chem" to signify chemical subject matter) to allow specialised databases – national, commercial, chemical, etc – to be generated. However, this proposal has serious drawbacks:

- URLs are Internet addresses and it is in principle undesirable to confuse the address function of a URL with that of representing a list of web page classifications or contact details.
- A URL is an inappropriate container of multiple web page classification codes and contact details because the length of the URL would cause it to become unwieldy as an Internet address.
- Including in a URL classification codes drawn from a list of thousands of codes would compromise the mnemonic quality of Internet addresses such as "www.yellowpages.com".
- There is substantial overlap in the subject matter contained in web pages having the various top level domain name codes.
- There is no consensus on, or standard for, content classification codes in URLs.

Another proposal to add content classification data to web pages has arisen from the wish to identify pages containing material that may be offensive to some viewers, or should not be accessed by minors. The Platform for Internet Content Selection (PICS) (see http://www.w3.org/pub/WWW/PICS and other documents at www.w3.org) is a web page ratings standard similar in principle to the ratings systems for motion pictures. This system allows page authors to "internally" self classify their pages through use of the "<meta...>" HTML element. Alternatively, "external" PICS ratings of web pages may be obtained from ratings service providers accessed each time a URL is selected. In practice, the ratings service providers have adopted very limited range of web page classifications. For example, Ararat Software's Commercial Rating System (see http://www.ararat.com.ratings/ararat10.html) provides just 5 categories of web page content; commercial content, technical/customer support, ordering information, downloading information and contact information. In other examples, CyberPatrol (http://www.microsys.com/pics/pics_msi.htm) provides 16 categories, the Recreational Software Advisory Council (http://www.rsac.org/faq.html) provides 4 categories, SafeSurf (http://www.safesurf.com/ssplan.htm) provides 11 categories and Vancouver Webpages Rating Service (http://vancouver-webpages.com/VWP1.0/ provides 11 categories. None of the categories provide classification of web pages by industry, service, product or subject with sufficient specificity to be useful when searching for web pages. Rather, the categories are intended to prevent web browsers from displaying web pages unsuitable for particular types of web browser users. Such rating systems are not intended to be used for the automated creation of Yellow or White pages like databases from web pages and are unsuitable for that purpose because they can not represent contact details. Further, the ratings data may only be encoded in the <meta...> element in the <head> of an HTML document drastically limiting the type and usefulness of the data that can be encoded.

Another proposal for classifying the content of web pages, the "Meta Content Framework" (MCF - see http://mcf.research.apple.com/mcf.html"), requires the content of web pages to be classified and the classification data to be held in a separate non-HTML data file with a MIME type of text/mcf. Storing data in non-HTML encoded documents which describes the content of HTML encoded documents is a technical and economic barrier to the adoption by search engine providers of the proposal. The MCF proposal is thus entirely unsuited to the automated creation of Yellow or White pages like databases from HTML encoded web pages (MIME type text/html) because data stored according to the MCF proposal is not stored in HTML encoded web pages.

The "Electronic Business Card", vCard, (see "vCard The Electronic Business Card" Version 2.1, versit Consortium Specification, Sept 18, 1996 or ftp://ds.internic.net/internet-drafts/draft-ietf-asid-mime-vcard-01.txt) uses non-HTML data file (MIME Content Types of "text/plain" or the non-standard "text/X-vCard") containing contact information equivalent to an extended White Pages entry which can be exchanged on a network using Simple Mail Transfer Protocol (SMTP) or using HTTP. It can be associated with a web page by use of a URL in the web page which refers to the vCard information (eg <a href="http://www.thing.com/vCard.vcf">My vCard</a>). Version 2.1 vCard standard data file format (published 18 September 1996) provides for the inclusion of many items of contact information. The vCard specification recommends that, where possible, there should be consistent mapping of vCard property names to HTML "<input>" element attribute names (eg vCard property name "TITLE" maps to HTML "<input name= 'title'>"). The intention is to facilitate the transfer of vCard data into web page input forms by pasting from a clipboard or by dragging from other computer applications. The VCard proposal is unsuited to the automated creation of Yellow or White pages like databases from HTML encoded web pages because data stored according to the VCard proposal is not stored in HTML encoded web pages.

The inclusion of classified information in separate documents (such as Meta Content files or vCards) has the disadvantage that there is necessarily much duplication of data and coordination of modifications between the separate documents and the web pages. This must be done to allow a person who has accessed a web page using an HTML compliant browser to determine whether it is worth calling up the associated file or vice versa. Also, to allow portions of web pages to be classified, web page contextual information would have to be duplicated in the separate document. vCards in particular do not provide this functionality. Another disadvantage is that non-HTML documents such as vCards contain no details as to how the data they contain is to be displayed. In the display of HTML documents the position, font, size, colour of the text and other elements of the document are of great importance. The restriction of address data in a vCard to untagged ordinally organised fields is inflexible. For example, multiple instances of extended parts of the address are not possible. Also components of names, addresses and telephone numbers and so forth are insufficiently identified.

The Online Computer Library Center Inc (OCLC, Dublin, Ohio, USA) proposal, known as the "Dublin Core", proposes to classifying scholarly web pages by subject (topic of the work, or keywords that describe the content of the work), title, author, publisher, other agent, date, object type (genre of the object such as home page, novel, poem etc), form, identifier, source, language, relationship and coverage (spatial and temporal) (see http://www.oclc.org:5046/~weibel/html-meta.html and other documents at www.oclc.org). This proposal does not include industry, service, product or subject classifications. It also does not include contact details. Names such as that of the author are not specified in sufficient detail to avoid ambiguities such as which is the author's first and last names. The proposal specifies that the details are encoded using the <meta...> element in the <head> of web pages. The proposal is unsuited to the automated creation of Yellow or White pages like databases from web pages because the proposal does not provide for classification of web pages and does not provide adequate contact details. Further, the use of keywords for describing the content of the work adds very little to the effectiveness of indexing of web pages since the web pages are usually indexed on every word of their content and most often the key words would simply be a duplication of words already contained in the document.

It has also been proposed to use the Dewey Decimal System (see http://orc.rsch.oclc.org:6109/eval_dc.html and http://orc.rsch.oclc.org:6109/bintro.html) to rank electronic documents against a Dewey Decimal subject classification. The proposal suggests automatically assigning Dewey Decimal subject classification codes to documents during automated indexing and cataloguing but does not specify the exact nature of the assignment although it is implied that the codes are stored separately from the documents. The proposal admits that such automated classification is less satisfactory than human classification. The proposal is unsuited to the automated creation of Yellow or White pages like databases from web pages because the accuracy of classification is inadequate, does not provide for inclusion of industry, service or product classifications and does not provide for inclusion of contact details. Deriving a subject classification code from an analysis of every word and phrase in a web page is computationally expensive.

The HTML 3.0 standard (see page 23 of the www.w3.org document "draft-ietf-html-specv3-00.txt") provides "class" as an attribute of almost all HTML "<body>" elements. The "class" attribute is intended to be used with style sheets. Style sheets provide a means by which the display of HTML documents may be altered to suit the needs of different classes of browser users. For example, <div class="appendix"> could be used to define a division that acts as an appendix, <h2 class="section"> could be used to define a level 2 header that acts as a section header, although, of course, any string of characters could be defined for those purposes. The "class" attribute, although never having been suggested for holding goods and services classifications, is not suited for such a use as it is, in any case, undesirable to confuse the style sheet function of the "class" attribute.

The HTML 3.0 and earlier standards provided the HTML elements "<person>" and "<address>" but do not specify the form of the content or method of validating the content of those elements. A person's name may be written as first name followed by last name or last name followed by first name. Similarly, different conventions exist for writing addresses. Similar ambiguities arise in the ill defined format of the HTML elements "<person>" and "<address>". As such they are of little use in the automatic compilation of searchable databases.

The XML language (see: http://textuality.com/sgml-erb/WD-xml.html) was developed to extend HTML so that software vendors can add new elements and new element attributes to HTML which are not specifically defined in any HTML standard. The intention is to ensure that all new elements and attributes could be parsed by all XML parsers even if the new elements held no significance for any particular XML parser. However, like HTML, XML does not provide a standard for the representation of industry, service, product or subject classification, contact or geographic location details within an web page.

Of course, many useful databases of the Yellow Pages or White Pages type are made available by service providers on networks, but they are not compiled automatically by using web crawlers to scan HTML web pages posted on a network. For example, http://www.yellowpages.com.au and http://www.mcp.com provide classified advertisements of the Yellow Pages type with links to the web pages of paying advertisers or subscribers. There are also directories of email addresses which approximate the White Pages directories, listing the names of individuals and organisations and contact details, (eg http://www.bigbook.com and http://query1.whowhere.com). However, these email directories require listers to manually add their directory entries and enquirers to be aware of and to find the directory enquiry web page. They cannot be automatically generated by scanning web pages using web crawlers since there is no adequate mechanism to relate email addresses to the names of people and organisations and their other contact details which may also exist in the same web page.

## OBJECTIVES OF THE INVENTION

The general object of the invention is to provide improved methods for automatically building searchable databases of classification, contact, and/or geographical information by using web crawlers to interrogate web pages posted on a network. [For convenience, this information is collectively referred to as CCG-data].

Other non-essential objectives are to provide methods for including and/or displaying CCG-data within web pages accessed by browsers, for automatically extracting CCG-data from web pages posted on a network and for using the same, and/or to provide methods for searching automatically compiled databases using such data.

Another subsidiary objective of the invention is to provide a new form of web page which is better suited to the automatic compilation (using web crawlers) of databases constructed by the automatic scanning of many such pages posted on a network.

## OUTLINE OF THE INVENTION

The invention is based upon the realisation that highly useful databases can be automatically built by successively interrogating web pages posted on a network if one or more HTML encoded CCG phrases are included in the web pages. A CCG phrase is one containing CCG-data in a form which is directly accessible and identifiable. CCG phrases may also include one or more items which provide the web page author with control over how the CCG-data is applied to the database.

Data duplication can be reduced if some of the CCG-data in the coded CCG phrases can be displayed by browsers as well as being used to update databases. Errors due to inexactly duplicated data are also eliminated. Accordingly, it is envisaged that CCG phrases may include one or more items which provide the web page author with control over how the CCG-data is displayed by a browser.

HTML (including version 2 and version 3) and XML are evolving applications (sub-sets or dialects) of ISO Standard 8879 1986 known as Standard Generalised Markup Language (SGML). HTML, in large part, is a language used to describe how text (unstructured data) and graphics is to be formatted for display. The HTML language consists of a finite number of "elements" (for example: "<BR>" where "BR" is the element name, also called the tag name) which may contain "attributes" (for example; "<DL COMPACT>" where "COMPACT" is an attribute named "COMPACT") and may contain values associated with attributes (for example; "<FONT SIZE=+1>" where +1 is the attribute value of the attribute named "SIZE"). XML is a language used to describe structured data. The XML language is similarly composed of elements, attributes and values with a similar syntax to HTML but unlike HTML the element names which may be used are not restricted and the meaning of the XML data may be interpreted in any convenient manner. While the XML language is mute about how data described by XML is to be formatted for display, the data may be used by computer programs for any purpose including description of how XML coded data is displayed. However, due to its historic importance in connection with web pages, the term "HTML" is herein used to refer to all markup languages which are subsets or complete sets of the SGML language. In particular, the term "HTML encoded CCG phrase" and the synonymous term "CCG phrase" are herein used to refer to CCG-data encoded in a subset or complete set of the SGML language. Herein, a "web page" is a document adapted to be or actually accessible through a network and encoded in a subset or complete set of the SGML language.

For convenience, CCG items in HTML encoded CCG phrases, whether they are syntactically represented as elements or as attributes, will be referred to hereinafter as CCG attributes.

A CCG phrase includes at least one of the following identifiable types of CCG-data attributes:

- industry, product, service, and/or subject classifications,
- contact categories, contact person(s) and/or organisation(s) names, titles or associations, contact details including physical and postal addresses, telephone and fax numbers, email and Internet or network addresses or locations, public keys, and
- geographic location details.

A CCG phrase may also include any of the following identifiable types of CCG control attributes:

- database control attributes to indicate which parts of the data are to be used to update databases, and
- display control attributes to indicate how browsers are to display the data.

By virtue of occurring in the same CCG phrase, a plurality of CCG-data attributes are associated with each other.

By virtue of their occurrence in the same CCG phrase, CCG-data attributes are idententified as a set of associated attributes. However the degree of association between attributes can be controlled by the inclusion in the phrase of database control attributes.

The start and end of CCG phrases should be identifiable to clearly distinguish these phrases from other data. To identify the beginning and end of a CCG phrase, at least one HTML element should have a CCG specific HTML element name or CCG specific attribute name or

CCG specific value. Each CCG attribute may consist, with or without other incidental characters, of a CCG attribute name and/or a CCG value or values. Preferably, each CCG phrase is contained in the "<body>" of the web page.

Two examples of a CCG specific HTML element are: "<CCG ...>" or "<CCG ... />" or "<CCG>...</CCG>". (Where a CCG phrase is coded in XML, the elements "<XML>" and "</XML>" may also be needed at the start and end of the CCG phrase.) A less satisfactory example is: "<!–CCG ...–> where the characters "CCG" after HTML comment element name "!–" are used to signify that the comment contains CCG-data. An example of the use of a CCG specific attribute name is: "<START CCG>"..."<END CCG>". An example of the use of a CCG specific value is: "<START TYPE='CCG'>"..."<END TYPE='CCG'>". Obviously, other character strings could be substituted for the element name, element attribute name or element attribute value "CCG" string of the examples.

The codes "<CCG ...>" and "<CCG ... />" are compatible with most HTML specifications, but being non-standard HTML, most web browsers do not display any text or attributes (eg PQ="AQD") within the angle brackets "<" and ">". These codes are preferred where display of the CCG data is not required and compatibility with older browsers is required (eg CCG phrases containing only classification values).

From one aspect, therefore, the invention comprises a web page for posting on a network, the web page being characterised by the inclusion of at least one CCG phrase in the "<body>" of the page, the CCG phrase being such that the CCG attributes contained therein are accessible and identifiable by (i) HTML compliant editors and/or (ii) HTML compliant web crawlers for the automatic construction of databases of classified information, and/or (iii) HTML compliant browsers for display on the computer screens of network users.

From another aspect, the invention comprises a method of constructing web pages of the above described type. The web pages may be constructed on digital computers using simple text editors such as Microsoft Windows Notepad, or preferably, purpose built human controlled editors or automated composing programs which embody knowledge of HTML and CCG syntax and grammar. Which ever process is used, CCG attributes are selected and inserted, modified, deleted and/or organised to form a valid CCG phrases in HTML encoded documents and the documents are posted on computer readable storage devices of computers connected to a computer network so that the documents are generally available to computers on the network.

From another aspect, the invention comprises a method of populating a database with CCG-data extracted from web pages. Web pages posted on a network are successively retrieved by a digital computer program (eg: a web crawler) and CCG phrases contained therein are identified and at least some of the CCG attributes found within the CCG phrases are extracted. The CCG attribute names are used to determine the type of data in the associated values. Generally the CCG attributes of interest are those relating to classification, contact and geographic data and database update controls while the attributes of little or no of interest in relation to database updating are those relating to display controls. Of course, the CCG-data extracted need only be that relevant to the particular database being updated. For example, one database may have been designed to index only web page classifications and URLs while another database may have been designed to index only contact details. Databases also differ in their internal representation of data and means of associating data. For example, some use "flat file" tables, others use pointers to data to create network associations while others use hashing and buckets.

The conventional nomenclature differs considerably between different types of database. Depending on the particular database nomenclature, data of the same type is said to be stored in table columns, fields, attributes and properties. The terms column and field are somewhat related to the physical representation of the data in files while attribute and property is more related to the logical representation of data. To avoid confusion, with the terms "HTML attribute", "CCG attribute" or just "attribute", hereinafter a database property means both a type of data stored in the database and a place in the database where data of the same type is stored. Database properties are referred to by a name ("property name") or similar reference and contain values. For example, a database property with the name "City name" and which contains values which are all the names of cities may be defined as a "City name" type database property.

Whichever style of database is used, it is preferred that the database update program relate the CCG attributes to corresponding database properties used by the database update process so that the database property values are updated with CCG values in a manner which preserves the distinctness, content and meaning of the CCG values and, preferably, preserves the CCG value associations expressed in the CCG phrase as sets of associated database property values of different types.

In some cases, it is desired to know the address of the web page from which the CCG values were extracted. For example, the purpose of building a database might be to allow searching of the database by web page classification to provide a list URLs of web pages or URLs of portions of web pages which contain matching CCG classifications. The URLs could then be inserted in an HTML document and transmitted to a web browser as a list of references to web pages matching a search expression. In that example, associating the URL of a web page or the URL of a portion of a web page with the CCG values extracted from the same web page or web page portion is important and the URL or means of reconstructing it must be available and supplied to the database update process. In one style of database, the values of the same type are held separate rows in a column (property) of a database table, and pointers held in another column (property) are associated with the values by sharing the same table row. The table row constitutes a set of associated property values. Each pointer points to a bucket (block of data) containing a list of URLs or pointers to URLs held in a separate bucket or table. In another style of database, values of different types are held in different tables together with a set number, pointer or similar code which is used to indicate which values are associated as members of the same set. In one variation, the values of set members are prefixed with a code indicating the type of value and all values are held in the same column of a table. If the purpose of the database is to hold contact data, recording the web page URL in the database might not be required although if the URL is not present in the database, updating changes in the CCG contact details contained within a web page is more difficult. Of course, one database may be used to record all types of CCG values contained in web pages and associate with each other any and all values extracted from the same web page or even from other web pages.

From another aspect, the invention comprises a method of searching the databases constructed as outlined above. These databases may be used for a variety of searching purposes. For example, to find web page URLs by using the association of web page URLs with industry, service, product or subject classification or a person's or organisation's name or address or geographic location values or any combination thereof. In another example, the databases may be used to find the contact details for people or organisations by name or location of industry, service, product or web page subject type and so forth by using the association between items of the contact details in the database without having to retrieve web pages associated with the contact details.

More particularly, the searching method involves finding URL references, or finding sets of associated database property values, from databases containing CCG-data. The method including steps of parsing a query phrase received from a computer network to extract query relational expressions and, from each expression, deriving a query field name, query relational operator and query value, determining the type of the query field by reference to its name, relating the query field to a corresponding database property according to type and locating CCG-data database property values in the database property which return a true value when tested against the query value using the query relational operator. Finally, the URL references or the sets of property values associated with the so located CCG-data database property values are extracted.

Database queries are usually expressed in a query language in the form of a phrase or sentence. In query by example style enquiry systems, the user types values into input fields on a form and a program extracts the input values and uses the values to automatically compose a query phrase or sentence. There are many existing examples of query languages used in connection with databases. Generally, they consist of relational expressions (eg Field=Value), logical expressions and grouping of relational and logical expressions by means such as parentheses. They may also contain sorting and output formatting expressions. Often abbreviated notation is used in the expressions such as leaving out field names or relational operators which are then inferred from the value in the expression or implied by default. In an enquiry the nature and format of the output may also be implied, such as a list of URLs of web pages or a list of contact details. Whatever is the mechanism of any particular database, the query expression needs to be parsed and fields in the query expression, explicit, default, implied or inferred, need be related to database properties of similar type. In some styles of database enquiry the query expression is evaluated against each row of a table or record of a file to find rows or records (ie a set of associated property values) which match the query expression. In other styles, sub-sets of the values of the properties are selected according to the interpretation of relational expressions in the query expression and the sub-sets are combined according to logical and grouping expressions in the query to find the sets of associated property values which match the query expression. Often, to make logical operations which combine the selected sub-sets more efficient, it is not the values which are selected but pointers to the values (eg Table name and table row) or unique keys (eg URLs or pointers to URLs) associated with the values. For example, the AND logical operator is often used to combine two lists so that only values or pointers or keys common to both lists are found in the combined list. Usually, the query produces a result list which is then provided to other processes. For example, a list of URLs of web pages is processed to produce an attractively formatted HTML encoded document containing the URLs and is sent to a web browser to allow an enquirer to retrieve interesting web pages. In another example, the contact details associated in the database with each value or pointer in the result list are retrieved from the database and presented as a report in the form of an HTML encoded document and is sent to a web browser for viewing.

From another aspect, the invention comprises a method of displaying CCG-data contained in CCG phrases within web pages which are displayed by a web browser executing on a digital computer. While a web page is loading or has loaded in a web browser, the web browser parses the web page and displays the text (or data) of the web page on a display device connected to the computer. When the web browser parser encounters CCG phrases, the web browser may display the CCG-data (element and/or attribute names (or translations of element and/or attribute names) and/or values) in a number of browser specific ways. For example, the web browser may by default not display any CCG-data, display all CCG-data, not display any CCG-data until a CCG display control attribute explicitly states that subsequent data should be displayed or display all CCG-data until a CCG display control attribute explicitly states that subsequent data should not be displayed. The web browser may also use CGA display controls specifying the size, font, position and so forth to alter the display of the CCG-data.

**DESCRIPTION OF EXAMPLES**

Having indicated the nature of the present invention, examples or embodiments thereof will now be described by way of illustration only.

Example 1: HTML Syntax Suitable for Representing a CCG Phrase

The following is an example of HTML element syntax suitable for representing CCG phrases in which a control (e.g. "SHOW") may be "good until countermanded" and thus apply to more than one field:

```
<CCG HREF="url"
{{NAME="label"    |    ID="identifier_code"}    &|    {LANG="language_code"    &
CLASS="Class_name"}
{
        {SET_SEPARATOR} &|
        {INDEX | NOINDEX} &|
        {SHOW | HIDE} &|
        {XPOS="horizontal_position_number"} &|
        {YPOS="vertical_position_number"} &|
        {NEWLINE} &|
        {ALIGN=centre | left | right | justify} &|
        {SIZE=[+/-]1 | 2 | 3 | 4 | 5 | 6 | 7} &|
        {COLOR="#rrggbb" | "colour_name"} &|
        {FACE="type_face_name"} &|
        {BLINK &| BOLD &| UNDERLINE &| ITALIC &| STRIKE} &|
        {SUBSCRIPT | SUPERSCRIPT} &|
        {CLEAR{=left | right | all}}
        {NORMAL} &|
        {{{CONTACT &| COPYRIGHT &| DEVELOPER} &|
        {PERSONAL &| BUSINESS &| ASSOCIATION} &|
        {attribute_name="attribute_value(s)"}
}
...
>
```

where: the ellipsis "..." implies optional repetition of the braced ("{" "}") items; the braces are used to group items and are not CCG syntactic elements; "&" (and) implies items must occur together; "|" (or) implies only one item must occur; and "&|" .(and/or) implies any including none of the items may appear together.

*Using the syntax of this example*, each CCG phrase is represented as an HTML element, the element name being "CCG" and the CCG-data (eg attribute_name="attribute_value") and CCG controls (eg SIZE=+1) are represented as attributes of the HTML element. Some of the attributes (eg SIZE) having explicit values (eg +1) and some attributes have implied values depending on the presence or absence in a CCG phrase (eg when the attribute BUSINESS is present it has the implied value of True and the implied value of False when absent).

Representation in XML syntax requires, at most, only a simple translation. All the items, such as "NORMAL" and "attribute_name" may remain unchanged as attributes of the element named "CCG" (eg <CCG size=+1/>). However, when a CCG phrase is encoded in XML, it is preferred that the items are represented as XML elements. For example attribute "SIZE=+1" can be represented as element "<size>+1</size>" or "<size value=+1/>" and "NORMAL" can be represented as "<normal/>.

In this example, the attributes, ID, LANG and CLASS take their meanings from HTML 3.0. The "url" in HREF="url" or may be a link with or without destination anchor labels. For example the URL http://www.w3.org/docs.html does not contain a destination anchor label (or identifier) while http://www.w3.org/docs.html#searching does contain the destination anchor label "#searching" which is intended refer to an anchor in docs.html such as <A NAME="searching">...</A>. There is some confusion in various HTML standards documentation about the distinction between the expression NAME="label" and the expression ID="identifier_code". For most practical purposes the two expressions have the same function or meaning: to uniquely identify within a document a position in or portion of that document.

Database control attributes:
"Set_separator" indicates the end of association between preceding and following data other than through the weaker mutual association with the same CCG phrase or web page; the data are divided into sets. "Index | Noindex" indicates that the following data are / are not to be indexed by a web crawler. These attributes have an implied attribute value of 'True' if present in and 'False' when absent from a CCG phrase.

Display control attributes:
"Show | Hide" indicates that a browser should show / not show the following data. Xpos and Ypos indicate the position (for example in pixel or physical units) on the browser screen where the data is to be displayed. "Newline" may be used in addition or as an alternative method of placing text on a browser screen. "Align" indicates the positioning of data on a browser screen relative to the cursor position set by "Xpos", "Ypos" or "Newline". "Size", "Colour" and "Face" indicates the size, colour and type face or font of the following data when displayed on an browser screen. "Blink", "Bold", "Underline", "Italic", "Strike", "Superscript" and "Subscript" indicates that the following data should be displayed blinking, bold, underlined, italicised, struck through, superscripted or subscripted. "Clear" indicates that the browser screen in the region where data will be displayed should be cleared to background before displaying the following data. "Normal" indicates the data is to be displayed without the "Blink", ..., "Clear" characteristics. The display controls which consist of an attribute name without an explicit value have an implied value of 'True' when present and 'False' when absent.

CCG-data attributes:
"Contact &| Copyright &| Developer" indicates that the following CCG-data refers to details for a person or organisation and/or to the copyright owner and/or to the HTML or web page developer. "Personal &| Business &| Association" indicates that the following data refers to details for a person and/or business and/or association. The previous CCG-data attributes have an implied attribute value of 'True' if present in a CCG phrase or set and 'False' when

absent from a CCG phrase or set. The attribute_name could be standard CCG attribute names or synonyms of standard CCG attribute names or abbreviations of CCG attribute names which refer to the following types of CCG attribute values where square brackets "[" and "]" surround suggested attribute names:

- industry or service or product or subject classifications and sub-classifications:
    - classification name [CN],
    - classification codes [CC].
- display only text [TEXT].
- contact:
    - person:
        - courtesy title [PNC],
        - first given name [PNG],
        - other given names [PNO],
        - family name [PNF],
        - name suffix [PNS],
        - qualifications [PQ],
        - associations [PA],
        - contact person title [P-T],
        - contact person role [PR].
    - organisation:
        - name [ON],
        - unit [OU],
        - identifier [OID].
    - physical or post or delivery address:
        - type [AT] (= "PHYSICAL" &| "POST-OFFICE" &| "POSTAL" &| "DELIVERY")
        - post office box number [AP#]
        - post office name [APN]
        - room or suite or office or unit or flat or apartment name &| number [AB#],
        - floor name &| number [ABF],
        - building name [ABN],
        - lane or street or road or highway number [AS#],
        - lane or street or road or highway name [ASN],
        - suburb or town or city name [ACN],
        - region or state or territory or province name [ARN],
        - post code [APC],
        - country or nation name [ANN],
    - telephone:
        - type [TT] (= "PREFERRED" &| "VOICE" &| "MOBILE" &| "CAR" &| "MESSAGE" &|"PAGER" &| "FACSIMILE" &| "MODEM" &| "ISDN" &| "VIDEO")
        - nation or country code number [TC#],
        - trunk access number [TT#],
        - area code number [TA#],
        - local number [TL#],
    - email:
        - type [ET] (= "INTERNET" | {other}),
        - mailer [EM],
        - address [EA],
    - Internet address:
        - url [IURL].
    - date & time:

        - date & time from [DTF],
        - date & time to [DTT],
        - weekday from [DTWF],
        - weekday to [DTWT],
        - weekday time from [DTWFT],
        - weekday time to [DTWTT],
        - time zone [DTZ].
    - brand name [BN].
    - public key:
        - key type[KT].
        - key [K],
- geographical:
    - location units [GLU],
    - location [GL],
    - serviced region units [GLRU].
    - serviced region [GLR],

Suggested attribute name [CN] is the name of an attribute associated with the attribute value containing "classification name" type data. For example, the [CN] attribute value could be the name of a proprietary or national or international or other industry classification standard such as the Australian and New Zealand Standard Industry Classification or "ANZSIC" for short or the U.S. Bureau of the Census Industrial Classifications (USBCIC). The associated classification codes [CC] attribute value could contain the codes and/or descriptions of the codes of the named standard with or without modifications, deletions or extensions. For example: CN="ANZSIC" CC="61;Road transport" or CN="USBCIC" CC="581;Hardware store". Service classifications such as the International Standard Classification of Occupations could be used. For example: CN="ISCOO" CC="4430;Auctioneer" Product classifications such as the Harmonised Commodity Description And Coding System could be used. For example: CN="HSC" CC="8411;Turbojets, turbo-propellers & other gas turbines; parts thereof" For subject classifications, Dewey Decimal, and/or Universal Decimal and/or Library of Congress and/or Bliss and/or Colon Classification could be used. For example: CN="DDC" CC="577.699;Sea shore ecology" The inclusion of subject classifications provides a very simple, straightforward method of classifying the subject matter of an HTML document which could be attractive to commercially oriented copyright owners.

The text ([TEXT]), person ([PNC] - [PR]), organisation ([ON] - [OID]), physical or post or delivery address ([AT] - [ANN]), telephone ([TT] - [TL#]), email address ([ET] - [EA]) and Internet address [IURL] are intended to be associated with each other in the obvious manner. Date & time(s) ([DTF] - [DTZ]) are intended to indicate the times at which the address and/or telephone and/or email will be serviced by the associated person(s) and/or organisation(s). The brand name ([BN]) attribute is intended to hold commercial brand names. Public key ([KT] - [K]) is intended to hold public encryption keys for secure communication with the contact person or organisation.

The geographical location [GL] could be a latitude and longitude (eg E148D31'12.5",S36D40',09.6" or E148.5201,S36.6693 or -148.5201,-36.6693), or a Universal Grid Reference (eg 55FV364402) or other global, national, regional or local location reference with units as specified [GLU], which is typed in or obtained by pointing to a digitally encoded map or other methods. In more populated regions of some countries such as the U.S., street addresses and post codes are associated with a moderately accurate geographic location and can be used to interpolate geographic location data where geographic location data is not explicitly stated in the CCG-data. Using a universally recognised code such as latitude and longitude has advantages when used with international mediums like the Internet. Geographical location is intended to be associated with a post, delivery address or physical address such as place of business or residence. A CCG compliant browser could use this reference to display a map centred on that geographic location. The purpose of the geographical location data is to allow browser users to specify search engine search criteria which will result in the search engine selecting only those Internet accessible documents which provide details about providers which are within a specified region. The serviced region [GLR] is intended to indicate the preferred area of operation of providers expressed in terms of serviced region units [GLRU]. A radial distance (eg in kilometres) or alternate means of expressing an area of interest around a geographic point, such as polygons, are envisaged.

It is envisaged that the CCG attribute_value could be composed of more than one value (actually sub-value) wherein specific characters or character strings separate individual values.

While specific instances of element names and types have been given in this example, of more importance is the type of data and type controls over the display and indexing of the data. As an alternative to the preferred immediately following example where the CCG-data is lumped together under the HTML element named "CCG", certain elements of the data, for example the classification data, could be lumped under separate HTML elements with distinctly different names thereby separating CCG classification data from CCG contact data. However, this is not preferred because the strength of association between the two types of data is weakened.

Example 2: Classification of Portion of a Web Page.

Where it is desired to classify a portion of a web page, such as a paragraph about a product, simple CCG-data may be used in conjunction with the syntax of Example1. For example:

```
<A NAME="Radios">AM-FM radio receivers: </A>
     <CCG HREF="#Radios">
          CN="ANZSIC"
          CC="E23.34.78;Electrical equipment - radio receivers AM"
          CC="E23.34.79;Electrical equipment - radio receivers FM"
     </CCG>
We won't be beaten on the price of these high quality receivers ....
```

In this example, the CCG prase appears after the related anchor (<A NAME=...</A>). However, while such proximity visually provides an obvious association between the anchor and related CCG phrase, it is intended that CCG phrase containing the attribute HREF related to a specific anchor could appear anywhere within the body of a web page and remain related to the named anchor. The CCG phrase containing the attribute HREF could appear in a separate document and thereby relate the CCG-data to the entire document or to a named anchor although, as previously noted, coordinating separate documents can be problematic. In the absence of the HREF and NAME attributes, it is also intended that the CCG-data apply to the whole web page.

Example 3 Classification of Portion of a Web Page using XML Syntax

Using XML syntax and similar attribute names to those of Example 2 the HTML fragment of Example 2 may be rewritten as:

```
<A NAME="Radios">AM-FM radio receivers: </A>
     <XML>
```

```
<CCG>
      <HREF>"#Radios"</HREF>
      <CN>"ANZSIC"</CN>
      <CC>"E23.34.78;Electrical equipment - radio receivers AM"</CC>
      <CC>"E23.34.79;Electrical equipment - radio receivers FM"</CC>
</CCG>
</XML>
```

We won't be beaten on the price of these high quality receivers ....

This example demonstrates that the translation of CCG-data from HTML to XML (and the reverse) involves simple syntactical and grammatical translations. Of course, the resulting HTML and XML, while "well formed" might not be recognised or, if recognised, might not be understood by some parsers.

Example 4: Constructing a Web Page Containing CCG-data

As an example, a web page developer, Alice Jamieson, is preparing an advertisement for a local electrician John Williams, trading as Kelso Electrical, who wants to advertise on the web for business within 30 kilometres from his office located at 18 Raglan Street, Kelso, New South Wales. Alice uses a graphical user interface web page authoring tool capable of creating and modifying web pages containing HTML (and XML) CCG phrases by accepting inputs from a user. The tool executes on a digital computer having input devices such as a keyboard, mouse, light pen and touch pad, display devices such as a CRT, LED arrays, liquid crystal arrays and computer-readable media such as magnetic and optical disks, memory arrays, magnetic tape and the like.

The authoring tool also embodies knowledge of the content and structure of CCG phrases such as the attribute names, valid ranges and sets of associated attribute values, the normal order of the attributes in the CCG phrase and interdependencies between attribute values. The tool provides a window where web pages may be viewed in layout (browser) mode and another window where the HTML code may be viewed in editing mode. The tool also provides means of inserting, deleting, modifying and organising HTML elements, changing font size, face and colour and so forth. The tool provides means for the user to build CCG phrases by using input devices to select an edit control representing various types of CCG attributes from a list which the tool then inserts in the body of a web page together with, when not already present, HTML code indicative of the start and end of a CCG phrase. The user then types in the value in the attribute. Similarly, the tool provides means of converting web page text to CCG attributes. Using input devices, the user selects the text to be converted to a CCG attribute then selects an edit control from a list; the tool then inserts the HTML code necessary to encode the text as a CCG attribute. However, these semi-manual methods of creating and modifying CCG phrases are inefficient and error prone. The tool also provides a button, which can be activated by using input devices, for access to CCG phrase editing functions. The CCG editing functions consist of a means of extracting the CCG values from existing CCG phrases in the web page being edited, forms for entering and modifying the extracted CCG values, a layout view browser window for altering how the CCG-data displays (position, font size, face, colour, bold, normal, hiding or showing and so forth), a data view browser window to alter which CCG-data values are to be indexed or not indexed in search engine databases, and a means of deleting existing CCG phrases from web pages and inserting new or changed CCG phrases in web pages. Editing cursors marking the current location at which text and/or data may be inserted, deleted or modified are provided in each window and form.

In the current example, the web page initially contains no CCG phrase. Clicking the CCG editing function button of the authoring tool causes a form to appear. The form contains prompts related to CCG attribute names and associated data input fields related to the CCG attribute values associated with the CCG attribute names, that is CCG-data. The fields are blank because, in the web page layout view, the edit cursor is not over a CCG phrase (and can not be since the web page initially contains no CCG phrase). The service classifications relevant to the web age, John Williams physical business contact address, phone and fax numbers, email address and geographic location and his post office business contact addresses are entered into the forms using a keyboard and mouse. The developer, Alice Jamieson, also includes her basic contact details where provided for on the form. The forms use drop down lists to select address blocks (eg physical and post office) for editing. Logic associated with the forms validates the CCG attribute values and interdependencies. Input devices are then used to control the CCG-data layout view browser to modify the appearance of the CCG-data such as font size and colour and positioning. In the layout browser, input devices communicating with the edit cursor are used to highlight individual items and blocks of items to be changed. The post office address is highlighted as a block and moved into position in line with the physical address. The CCG-data view window is then used to check which data items are to be indexed by search engines. In this example all CCG-data (ie all CCG attribute values except display control values and database control values) are to be indexed. Input devices are used to control the edit cursor to highlight the entire data and a mouse is used to click (activate) a button to mark all the data for indexing. Then another button is clicked which builds an HML encoded CCG phrase of CCG attributes derived from the CCG-data values, display control values and database control values and inserts the CCG phrase in the web page at the location pointed to in the web page layout browser window.

The HTML code editing mode window was called up which revealed the following HTML encoded CCG phrase in the web page:

```
<XML>
<CCG>
      <INDEX/>
      <HIDE/>
      <CN>ANZSIC</CN>
      <CC>D36.11.45;Electrical contractors - residential</CC>
      <CC>D36.11.46;Electrical contractors - industrial</CC>
      <SHOW/>
      <CONTACT/> <COPYRIGHT/>
      <BUSINESS/>
      <XPOS>50</XPOS>
      <YPOS>320</YPOS>
      <ALIGN>centre</ALIGN>
      <SIZE>3</SIZE>
      <COLOR>black</COLOR>
      <FACE>Times New Roman</FACE>
      <BOLD/>
      <CLEAR>all</CLEAR>
      <TEXT>Contact :</TEXT>
      <PNC>Mr</PNC>
      <PNG>John</PNG>
      <PNF>Williams</PNF>
      <PQ>AIE</PQ>
```

```
<PA>ARUC</PA>
<NEWLINE/>
<PT>Managing Director</PT>
<NEWLINE/>
<ON>Kelso Electrical Pty. Ltd.</ON>
<NEWLINE/>
<NORMAL/> <ITALIC/>
<SIZE>-2</SIZE>
<TEXT>NSW License 45678C</TEXT>
<NEWLINE/>
<NORMAL/> <BOLD/>
<SIZE>+2</SIZE>
<AT>PHYSICAL</AT>
<AS#>18<AS#>
<ASN>Raglan Street<ASN>
<NEWLINE/>
<ACN>Kelso</CAN>
<NEWLINE/>
<ARN>NSW<ARN>
<NEWLINE/>
<HIDE/>
<ANN>Australia</ANN>
<NEWLINE/>
<SHOW/>
<TEXT>Phone:</TEXT>
<TT>PREFERRED ; VOICE ; MESSAGE</TT>
<HIDE/>
<TC#>61</TC>
<SHOW/>
<TT#>0</TT#>
<TA#>63</TA#>
<TL#>456-7828</TL#>
<TEXT>    Fax:</TEXT>
<TT>FACSIMILE</TT>
<HIDE/>
<TC#>61</TC#>
<SHOW/>
<TT#>0</TT#>
<TA#>63</TA#>
<TL#>456-7829</TL#>
<NEWLINE/>
<ET>INTERNET</ET>
<EA>johnw@firefly.com.au<EA>
<TEXT>   </TEXT>
<GLU>LatLong</GLU>
<GL>="33.3978S;148.5679E</GL>
<GLRU>Km</GLRU>
<GLR>30 </GLR>
<SET_SEPARATOR/>
<XPOS>250</XPOS>
```

```
          <YPOS>320</YPOS>
          <NEWLINE/>
          <NEWLINE/>
          <TEXT>Or write to us at :</TEXT>
          <NEWLINE/>
          <ON>Kelso Electrical Pty. Ltd.</ON>
          <NEWLINE/>
          <AT>POST-OFFICE</AT>
          <AP#>P.O. Box 187</AP#>
          <NEWLINE/>
          <APN>Sunny Comer</APN>
          <TEXT> </TEXT>
          <APC>2795</APC>
          <NEWLINE/>
          <HIDE/>
          <ANN>Australia</ANN>
          <SET_SEPARATOR/>
          <HIDE/>
          <DEVELOPER/>
          <BUSINESS/>
          <PNG>Alice</PNG>
          <PNF>Jamieson</PNF>
          <ET>INTERNET</ET>
          <EA>alijam@firefly.com.au</EA>
          <IURL>http://www.firefly.com.au/~aljam/<IURL>
     </CCG>
     </XML>
```

In the web page layout browser window the CCG-data displayed as follows:

```
Contact :                              Or write to us at:
Mr John Williams, AIE, ARUC,
Managing Director
Kelso Electrical Pty. Ltd.             Kelso Electrical Pty Ltd
NSW License 45678C                          P.O. Box 187
18 Raglan Street                    Sunny Comer 2795
Kelso
NSW
Phone: 063-456-7828  Fax: 063-456-7829
Email: johnw@firefly.com.au  Map
```

Having encoded the web page in this way, Alice then posts it on the storage device of a digital computer connected to the Internet from where it can be retrieved through the Internet using the URL "http://www.firefly.com.au/~johnw/index.html"

Example 4: Constructing a Database from Web Pages Containing CCG-data

During a routine sweep of Internet connected web page servers, a web crawler (or robot) operating on a server named "ccg.search.com" executing on an Internet connected digital computer discovers the URL "http://www.firefly.com.au/~johnw/index.html" in a document it had previously retrieved through the Internet. The web crawler decides that the URL matches it's selection criteria because the URL contains the suffix ".html". The web crawler then successfully retrieves the document by extracting from the URL the address of the computer hosting the document, addressing and sending a message (including the address of the web crawler) requesting the web page through the network to the web page host computer using TCP/IP protocol, the host computer then reads the document, addresses and sends the document to the web crawler using TCP/IP protocol, the web crawler then waiting until it has received all parts of the web page from the host computer before proceeding. It inspects the contents of the document and finds that it matches the additional selection criteria that it is an HTML encoded document. The web crawler program, depending on its state and logic, then parses the document, strips out and saves some or all of the URLs in the document for future examination. The web crawler program then passes the document, together with the URL of the document through a network communications channel to an indexing program executing on a different computer. The indexing computer has database updating software which manipulates a database stored on computer-readable media.

The indexing program parses the document, from first to last character, indexing some of the meta data in the <head> of the document and the words in the text of the document with respect to the document URL. In the database of this example, unique words extracted from the documents already indexed are held in separate rows of a column of a database table and in another column of the same table on each row is an associated pointer to the first bucket or block of URLs of documents containing the word associated with the pointer. As new words are found, the new word is added as a new row in the word column of the table, a new bucket is created, the URL of the document containing the new word is inserted into the bucket and a pointer to the new bucket is written in the new row pointer column. When the same word is found in another document, the row in the table of the word is found, the pointer is retrieved from the table, the bucket pointed to by the pointer is retrieved and the URL of the other document is inserted in the bucket. Where a bucket becomes full of URLs, a new bucket is created and a pointer to the new bucket for holding additional URLs is placed in the full bucket. Deletion of words and URLs of changed or no longer existing documents is also provided for.

In addition to indexing words extracted from the text of the document, the indexing program also indexes the CCG-data in the document as well as indexing words found in the CCG-data. When the parser finds HTML element "<XML>" in the document it switches into XML parsing mode and switches out of that mode when "</XML> is found. When the element "<CCG>" is found, the parser switches into the CCG parsing mode and switches out of that mode when "</CCG>" is found.

The example database has a CCG-data attribute name to database property name correspondence table to show the relationship between the CCG-data attribute names and the database tables and columns (properties) where the CCG-data attribute values are to be stored in the database as database property values. The database property values and associated URLs are stored in much the same way as for words extracted from text as outlined above. However, CCG contact data, for example, which consists of several distinct CCG-data attributes which are related (eg street name, city), is stored in a database table having a column (property) related to each distinct CCG contact attribute name and each separate CCG contact data set (eg person's name, address, telephone number) as separated by "<CCG>", "<SET_SEPARATOR>" and "</CCG>" is held in a separate row in the table. The values stored in each row are considered to be a set of associated property values of different types.

The indexing program, during parsing the document of Example 2 above, encounters the "<CCG>" element and enters the CCG parsing mode. The parser knows to ignore display control attributes and to consider database control elements in the CCG phrase. The example indexing program opts to index all other CCG-data contained in the attribute values until explicitly instructed not to index the attribute values by encountering the "<NOINDEX/>" database control element and then to recommence indexing when the "<INDEX/>" database control element is encountered.

Taking each CCG-data attribute name and associated attribute value(s) in succession, the example indexing program uses the correspondence table to translate the CCG-data attribute name to the database table and column (property) names where the CCG-data attribute value(s) are to be stored as database property value(s). The indexing program may opt to translate the CCG-data attribute values to database property values by, for example, converting character strings of digits to binary encoded decimal representation, the string "True" to a single bit representation and the like. The indexing program then adds or updates the database property value(s), using the database table and column (property) names (or similar references) obtained by translation, in much the same manner as outlined above for the update of the database using words extracted from the document text, including associating the data to the document URL where desired. Where the CCG-data contains a "HREF" attribute (or similar), the URL associated with the other CCG-data is a URL taken from the "HREF" attribute value or composed of the document URL and the "HREF" attribute value if the attribute value is a partial or relative URL. Some CCG attributes, such as "<BUSINESS/> have only an implied value of true if the attribute is present and false if the attribute is absent, the "<SET_SEPARATOR/>", "<CCG>" and "</CCG>" resetting such values to false. However, where attribute value(s) associated with different attribute names are still related, such as a person's name and a street name, the related values of different types are stored on the same row of the same database table but in a different column (database property) to preserve the relationship. "<SET_SEPARATOR/>" limits the degree of relatedness between, for example, a person's name occuring before the separator and a street name occurring after the separator. Using the example document and using the same database column (property) names as used for the CCG-data attribute names a portion of the table constructed database table would look like:

| | PNC | PNG | PNF | PQ | PA | PT | | URL |
|---|---|---|---|---|---|---|---|---|
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| ... | Mr | John | Williams | AIE | ARUC | Managing Director | ... | (pointer) |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |

Difficulties not highlighted by this example are the need to handle properties having multiple values of the same type, "sparse rows" where only a few values are not null (blank) and tables with extremely large numbers of rows. For example, the CCG-data of this example could have contained multiple values of personal qualifications ("PQ"). To represent this type of data using a 2 dimensional table database system, the database would be "normalised" so that the multiple values were stored in a separate table and keys or pointers were used to relate the relate the items in the two tables. Numerous alternate database systems, for example those based on key hashing and data buckets, or tagging data values with prefixes or suffixes related to the type of data value may be used. Preferably, however, whatever database system is used, it should preserve the associations of CCG-data items present in the CCG phrases.

Because the geographic location data was missing from the postal address of the CCG-data in the example document, but a post code was present, the indexing program inferred the geographic location from the post code.

Example 6: Finding Web Page References Using a CCG Database

As an example, Kevin Robson lives in Sydney but owns and has rented out a house in Bathurst. He wants to use the web to find some electricians based in the general Bathurst region (not only in Bathurst City) to contact for estimating the cost of modifying the wiring in the house. He uses his web browser to open the web page "http://www.ausline.com.au/web_search.html" containing AusLine's search engine web page search criteria input form encoded using the HTML "<form>" element.

The search criteria input form contains several input fields including those labelled "Service classification", "Key words", "City./Suburb/Town", "Country", "Lat/Long" and "Radius". The form also displays a button labelled "Map" to allow latitude and longitude to be selected by pointing to map images. The word "electrician" is typed into the "Service classification" field, "house wiring" into the "Keywords" field, "Bathurst" into the "City/Suburb/Town" field and "10" into the field "Radius". The country "Australia" was already showing in the country field because the web page server had received cookie data from the browser indicating that that was the country used when the browser last used the web page. The "submit search" button on the web page was clicked. The browser transmitted a message using TCP/IP protocol to the AusLine server containing the input field values encoded in the header of the message.

After a short delay, the search result HTML encoded web page was returned. Clicking on the "Service classification" input field drop down list box to check the classifications used in the search revealed three items:

- Electrical contractors - residential
- Electrical contractors - industrial
- Electrical engineers

The search engine attached to the server obtained those classifications by using word stemming and searching the text of the service classifications held in it's database. The Lat/Long field contained the value "33.3856S;148.5743E" which the search engine obtained by looking up the latitude and longitude of the town "Bathurst" in the country "Australia" in it's database. Clicking on the "Map" button retrieved a web page having the image of a map centred on the town of Bathurst and showing the area 20 Km around it. The search engine obtained the map by making a request to another Internet connected server and supplying the latitude, longitude and radius. Clicking on the browser "Back" button returned to the search results page.

The search results contained 8 titles, brief descriptions and URLs including a reference containing the URL "http://www.firefly.com.au/~johnw/index.html". Retrieving each in turn revealed that all were well focused according to the search criteria being related to electricians, electrical contractors and engineers in the Bathurst area. The search engine obtained these references to web pages by:

- searching it's database of service classification titles with words stemming from "electrician" which resulted in three service classification codes,
- searching it's database using the three service classification codes to obtain an intermediate list of URLs of web pages containing those CCG codes

- searching it's database for the two keywords to obtain an intermediate list of URLs of web pages containing those words in the web page text,
- Searching it's database to find the latitude and longitude of Bathurst, Australia,
- searching it's database to obtain an intermediate list of web pages which contain latitude and longitude data lying within 10 Km of the latitude and longitude of Bathurst, Australia,
- producing as a result list, a list of URLs which are common to all the intermediate lists,
- obtaining from it's database the title and brief description of the web pages,
- formatting the titles, descriptions and URLs into an HTML encoded report,
- transmitting the report to the enquiring web browser.

Example 7: Finding Contact Details Using a CCG Database

As an example, Jim Jones of Jones and Sons wants to send a recall notice about a faulty batch of UV stabilised electrical power cable to all Electrical contractors and Electrical wholesalers in Australia who have email addresses. He uses his web browser to open the web page "http://www.ausline.com.au/contact_search.html" containing AusLine's search engine contact search criteria input form encoded using the HTML "<form>" element.

The search criteria input form contains several input fields including those labelled "Service classification", "Country" and "Output format". The word "electric*" is typed into the "Service classification" field, the word "Australia" is typed into the "Country" field and the "Tabular - Name & Email" option in the "Output format" drop down list box is selected. The "Submit search" button on the web page is clicked. The browser transmits a message using TCP/IP protocol to the AusLine server containing the input field values encoded in the header of the message.

After a short delay, the search result HTML encoded web page is returned. Clicking on the "Service classification" input field drop down list box to check the classifications used in the search revealed too many classifications for the result to be sufficiently focused. The following four classifications were selected from the list:

- Electric cable - ducting systems
- Electrical contractors - residential
- Electrical contractors - industrial
- Electrical wholesalers

and the "Submit search" button is pressed again to refine the search.

The search results contained 3,473 names and associated email addresses and URLs to full contact details. Jim saved the search result page on his computer so that he could use his email program to send the recall notice to each email address in the list. The email address "johnw@firefly.com.au" was included in the list.

The search engine obtained these references to web pages by:

- searching it's database using the four service classification titles which resulted in four service classification codes,
- searching it's database using the four service classification codes to obtain an intermediate list of database primary keys of database table rows containing those service classification codes in the database Service classification attribute,
- searching it's database using the country name "Australia" to obtain an intermediate list of database primary keys of database table rows containing that word in the database Country attribute,

- producing as a result list, a list of database primary keys which are common to both the intermediate lists,
- obtaining from it's database using the result list the values of the name and email attributes,
- using the HTML <table> element to format the name values, email values and full detail URLs into an HTML encoded report,
- transmitting the report to the enquiring web browser.

This example relates to finding sets of associated database contact values without requiring references to web pages. However, finding other sets of associated database values such as sets of associated industry classification values and geographic location values might also be useful for some purposes.

Thus it is appreciated that the afore stated goals, advantages and objectives are achieved by the teachings herein. In particular it is seen that, unlike the prior art, efficiently searchable Yellow pages and White pages databases and the like may be automatically constructed from HTML encoded web pages. Additionally the database entries may be automatically linked to specific web pages and portions of web pages allowing convenient methods of indexing of product and service catalogues and the like. It is also appreciated that simpler methods of constructing databases suited to a variety of other uses such as industry and subject directories are also provided.

From the foregoing teachings and with the knowledge of those skilled in the art, it is apparent that other modifications and adaptations of the invention will become apparent. For example, the method steps disclosed and claimed herein may be practiced in a variety of different orders. CCG-data may take on a variety of different forms within the meaning of the claims. Thus, it is our intention to include within the scope of the claims not only the invention literally embraced by the language of the claims but to include all such modifications and adaptations which may come to those skilled in the art.

What I claim is:

1. An HTML encoded web page embodied on a computer-readable medium, said web page comprising at least one HTML encoded CCG phrase, each CCG phrase comprising:
   a) HTML code indicative of the start of a CCG phrase,
   b) at least one CCG-data attribute, and
   c) HTML code indicative of the end of a CCG phrase.

2. An HTML encoded web page embodied on a computer-readable medium, said web page comprising at least one HTML encoded CCG phrase, each CCG phrase comprising:
   a) HTML code indicative of the start of a CCG phrase,
   b) at least two CCG-data attributes,
   c) at least one database control attribute separating said CCG-data attributes into at least two sets of CCG attributes, and
   d) HTML code indicative of the end of a CCG phrase.

3. An HTML encoded web page embodied on a computer-readable medium, said web page comprising at least one HTML encoded CCG phrase, each CCG phrase comprising:
   a) HTML code indicative of the start of a CCG phrase,
   b) at least one CCG-data attributes,
   c) at least one attribute of: database control attributes, display control attributes; and
   d) HTML code indicative of the end of a CCG phrase.

4. A computer implemented method of building a web page comprising at least one HTML encoded CCG phrase, the method comprising the steps of:
   a) displaying a web page on a computer display device,
   b) displaying an edit cursor indicating a character position on said display device and a corresponding character position in said web page, said edit cursor being positionable within the display of said web page by use of computer input devices,
   c) separately displaying on said computer display device a set of edit controls representing CCG-data attribute types,
   d) positioning said edit cursor within said display of said web page using said input devices,
   e) selecting an edit control from said set of edit controls using said input devices,
   f) relating said selected edit control to a corresponding CCG-data attribute name,
   g) constructing a CCG-data attribute character string comprising a character string representing said attribute name and another character string representing an empty CCG-data value,
   h) if the said edit cursor is positioned outside a CCG phrase,
      i) inserting into said web page, at the character position indicated by said edit cursor, a start character string comprising HTML code indicative of the start of a CCG phrase,
      ii) inserting into said web page, immediately after the end of said start character string, an end character string comprising HTML code indicative of the end of a CCG phrase, and
      iii) positioning said edit cursor between said start and end character strings,

i) inserting said CCG-data attribute character string into said web page at the character position indicated by said edit cursor,

j) positioning said edit cursor at the character position in said web page of the CCG-data value of said inserted CCG-data attribute character string,

k) inputting characters using a keyboard,

l) inserting said input characters into said web page at the character position indicated by said edit cursor, thereby converting said empty CCG-data value to a non-empty CCG-data value, and

m) writing said web page on computer-readable media.

5. A computer implemented method of building a web page comprising at least one HTML encoded CCG phrase, the method comprising the steps of:

a) displaying a web page on a computer display device,

b) displaying a start edit cursor and an end edit cursor on said display device, each said edit cursors indicating a character position on said display device and a corresponding character position in said web page, said edit cursors being positionable within the display of said web page by use of computer input devices,

c) separately displaying on said computer display device a set of edit controls representing CCG-data attribute types,

d) selecting a string of web page characters on said display device using said input devices to position said start edit cursor to indicate the start said string of web page characters and said end edit cursor to indicate the end of said string of web page characters,

e) selecting an edit control from said set of edit controls using said input devices,

f) relating said selected CCG-data control to a corresponding CCG-data attribute name,

g) constructing a CCG-data attribute character string comprising a character string representing said attribute name and another character string representing a CCG-data value containing said string of web page characters,

h) deleting said string of web page characters from said wen page,

i) if the said start edit cursor is positioned outside a CCG phrase,

   i) inserting into said web page, at the character position indicated by said start edit cursor, a start character string comprising HTML code indicative of the start of a CCG phrase,

   ii) inserting into said web page, immediately after the end of said start character string, an end character string comprising HTML code indicative of the end of a CCG phrase, and

   iii) positioning said start edit cursor between said start and end character strings,

j) inserting said CCG-data attribute character string into said web page at the character position indicated by said start edit cursor, thereby converting said string of web page characters to a CCG-data attribute value contained within a CCG-data attribute contained within CCG-phrase, and

k) writing said web page on computer-readable media.

6. A computer implemented method of building a web page comprising at least one HTML encoded CCG phrase, the method comprising the steps of:

a) displaying a CCG-data input form on a computer display device,

b) inputting CCG-data values into fields of said data input form using computer input devices,

c) inserting into the body of a web page a start character string comprising HTML code indicative of the start of a CCG phrase,

d) inserting into said web page body immediately after the end of said start character string an end character string comprising HTML code indicative of the end of a CCG phrase,

e) extracting successive field values from said data entry form together with related field value type information,

f) relating the type of each extracted field value to a corresponding CCG-data attribute name,

g) constructing a CCG-data attribute character string comprising a character string representing said attribute name and another character string representing said field value,

h) inserting said CCG-data attribute character string into said web page between said start and end character strings.

i) writing said web page on computer-readable media.

7. A computer implemented method of building a database which comprises sets of associated property values wherein each set includes at least two property values of different types, the property values being any of classification values, contact values, geographic location values, hereinafter collectively referred to as CCG-data, the method comprising the steps of:

a) retrieving successive web pages from a computer network, each web page being identified by a URL,

b) searching each web page for a CCG phrase that includes a plurality of different types of CCG-data attributes,

c) extracting a plurality of said attributes from said phrase,

d) from each extracted attribute, deriving an attribute name and a related attribute value,

e) determining the type of said extracted attribute and said attribute value by reference to said attribute name,

f) relating said type of attribute value so determined to a corresponding type of database property value,

g) relating the URL of said web page to an other type of database property value,

h) writing said derived attribute value to the database property value of said determined corresponding type in a set of associated property values, and

i) writing the URL of said web page to a database property value of said other type in said set of associated property values.

8. A computer implemented method of building a database which comprises sets of associated property values wherein each set includes at least two property values of different types, the property values being any of classification values, contact values, geographic location values, hereinafter collectively referred to as CCG-data, the method comprising the steps of:

a) retrieving successive web pages from a computer network, each web page being identified by a URL,

b) searching each web page for a CCG phrase that includes at least one type of CCG-data attribute,

c) extracting at least one said attribute from said phrase,

d) from each extracted attribute, deriving an attribute name and a related attribute value,

e) determining the type of said extracted attribute and said attribute value by reference to said attribute name,

f) relating said type of attribute value so determined to a corresponding type of database property value,

g) relating the URL of said web page to an other type of database property value,

h) writing said derived attribute value to the database property value of said determined corresponding type in a set of associated property values, and

i) writing the URL of said web page to a database property value of said other type in said set of associated property values.

9. A computer implemented method of building a database which comprises sets of associated property values wherein each set includes at least two property values of different types, the property values being any of classification values, contact values, geographic location values, hereinafter collectively referred to as CCG-data, the method comprising the steps of:

a) retrieving successive web pages from a computer network,

b) searching each web page for a CCG phrase that includes a plurality of different types of CCG-data attributes.

c) extracting a plurality of said attributes from said phrase,

d) from each extracted attribute, deriving an attribute name and a related attribute value,

e) determining the type of said extracted attribute and said attribute value by reference to said attribute name,

f) relating said type of attribute value so determined to a corresponding type of database property value, and

g) writing said derived attribute value to the database property value of said determined corresponding type in a set of associated property values.

10. A computer implemented method of finding references to web pages posted on computer network the method using a database comprising sets of associated property values, the property values being any of classification values, contact values, geographic location values, hereinafter collectively referred to as CCG-data, and URL references, the method comprising the steps of:

a) receiving a query phrase including query relational expressions from a computer network,

b) parsing said query phrase and extracting each of said query relational expressions included therein,

c) from each extracted query relational expression, deriving a query field name,

d) determining the type of said query relational expression by reference to its derived query field name,

e) relating said type of query relational expression so determined to one of the following query relational expression types: CCG-data type, other type,

f) provided said query relational expression is a CCG-data type, deriving a query relational operator and query value related to its query field name from said query relational expression,

g) determining the type of said query value by reference to said query field name,

h) relating said type of query value so determined to a corresponding type of database property value,

i) locating database property values of said determined corresponding type which return a true value when tested against said query value using said query relational operator,

j) extracting from said database a list of the URL references associated with the so located database property values,

11. A computer implemented method of finding sets of associated database property values the method using a database comprising sets of associated property values wherein each set includes at least two property values of different types, the property values being any of classification values, contact values, geographic values, hereinafter collectively referred to as CCG-data, the method comprising the steps of:

a) receiving a query phrase including query relational expressions from a computer network,

b) parsing said query phrase and extracting each of said query relational expressions included therein,

c) from each extracted query relational expression, deriving a query field name,

d) determining the type of said query relational expression by reference to its derived query field name,

e) relating said type of query relational expression so determined to one of the following query relational expression types: CCG-data type, other type,

f) provided said query relational expression is a CCG-data type, deriving a query relational operator and query value related to its query field name from said query relational expression,

g) determining the type of said query value by reference to said query field name,

h) relating said type of query value so determined to a corresponding type of database property value,

i) locating database property values of said determined corresponding type which return a true value when tested against said query value using said query relational operator,

j) extracting from said database sets of associated database property values associated with the so located database property values.

12. A method of displaying a web page comprising at least one HTML encoded CCG phrase, the method comprising the steps of:

a) retrieving a web page from a computer network,

b) parsing said retrieved web page to locate an HTML code indicative of the start of a CCG phrase,

c) parsing said located CCG phrase and extracting successive CCG attributes contained therein until an HTML code indicative of the end of said CCG phrase is found,

d) from each extracted attribute, deriving an attribute name,

e) determining the type of said extracted attribute by reference to its derived attribute name,

f) relating said type of attribute so determined to one of the following attribute types: database control, display control, CCG-data,

g) provided said extracted attribute is not a database control type, deriving an attribute value related to its attribute name from said extracted attribute,

h) determining the type of said attribute value by reference to said attribute name.

*i)* *relating* said type of attribute value so determined to a corresponding type of parameter of a display-device-control-program,

j) writing said attribute value to said parameter, and

k) where said type of attribute is a CCG-data type, causing said display-device-control-program to effect display of said attribute value on a display device, formatted and positioned according said display-device-control-program parameters whereby successive values of CCG-data of the CCG phrase are displayed.

## ABSTRACT

A system for automatically creating databases containing industry, service, product and subject classification data, contact data, geographic location data (CCG-data) and links to web pages from HTML, XML or SGML encoded web pages posted on computer networks such as the Internet or Intranets. The web pages containing HTML, XML or SGML encoded CCG-data, database update controls and web browser display controls are created and modified by using simple text editors, HTML, XML or SGML editors or purpose built editors. The CCG databases may be searched for references (URLs) to web pages by use of enquiries which reference one or more of the items of the CCG-data. Alternatively, enquiries referencing the CCG-data in the databases may supply contact data without web page references. Data duplication and coordination is reduced by including in the web page CCG-data display controls which are used by web browsers to format for display the same data that is used to automatically update the databases.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 747 842 A1

# (12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
11.12.1996 Bulletin 1996/50

(51) Int. Cl.$^6$: G06F 17/30

(21) Application number: 96108976.0

(22) Date of filing: 05.06.1996

(84) Designated Contracting States:
AT BE CH DE ES FR GB IT LI NL SE

(30) Priority: 07.06.1995 US 479481

(71) Applicant: International Business Machines Corporation
Armonk, N.Y. 10504 (US)

(72) Inventors:
- Rogers, Richard Michael
  Beacon, New York 12508 (US)
- Lagarde, Konrad Charles
  Milford, Connecticut 06460 (US)

(74) Representative: Schäfer, Wolfgang, Dipl.-Ing.
IBM Deutschland
Informationssysteme GmbH
Patentwesen und Urheberrecht
70548 Stuttgart (DE)

## (54) A web browser system

(57) A World Wide Web browser makes requests to web servers on a network which receive and fulfill requests as an agent of the browser client, organizing distributed sub-agents as distributed integration solution (DIS) servers on an intranet network supporting the web server which also has an access agent servers accessible over the Internet. DIS servers execute selected capsule objects which perform programmable functions upon a received command from a web server control program agent for retrieving, from a database gateway coupled to a plurality of database resources upon a single request made from a Hypertext document, requested information from multiple data bases located at different types of databases geograhically dispersed, performing calculations, formatting, and other services prior to reporting to the web browser or to other locations, in a selected format, as in a display, fax, printer, and to customer installations or to TV video subscribers, with account tracking.

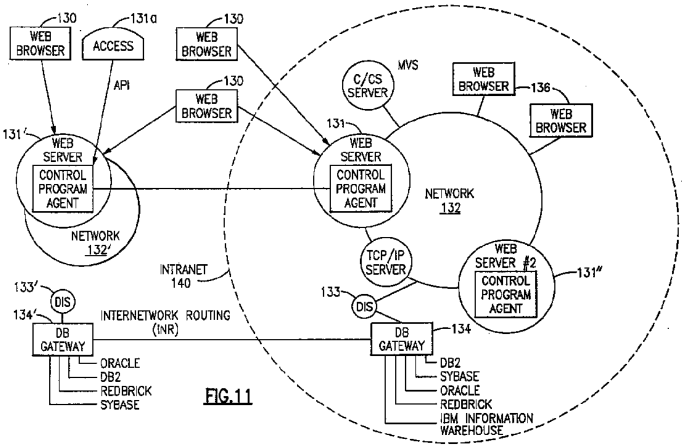


FIG.11

Printed by Rank Xerox (UK) Business Services
2.13.9/3.4

## Description

Copyright Authorization

FIELD OF THE INVENTION

This invention is related to computers and computer systems and particularly to a method and system for use of the World Wide Web and other sources of information and for utilization of existing equipment advantageously for web server data access over networks and the Internet.

RELATED APPLICATIONS

This application entitled "A Web Browser System", is related to other United States of America Patent applications filed concurrently herewith, and specifically to the applications entitled "Computer Network for WWW Server Data Access over Internet", USSN 08/474,571, filed June 7, 1995; and "A Service Agent for Fulfilling requests of a Web Browser", USSN 08/474,576, filed June 7, 1995; and "A Sub-Agent Service Agent for Fulfilling Requests of a Web Browser", USSN 08/474,575, filed June 7, 1995; and "A Method for Fulfilling Requests of a Web Browser" USSN 08/474,577, filed June 7, 1995; and "A Method for Distributed Task Fulfillment of Web Browser Requests",USSN 08/474,572, filed June 7, 1995.

These applications have a common assignee, International Business Machines Corporation, Armonk, New York.

GLOSSARY OF TERMS

While dictionary meanings are also implied by certain terms used here, the following glossary of some terms may be useful.

World Wide Web (WWW)

The Internet's application that lets people seeking information on the Internet switch from server to server and database to database by clinking on highlighted words or phrases of interest. An Internet WWW server supports clients and provides information.

Home page

A multi-media table of contents that guides a web user to stored information about an organization on the Internet.

Gopher

A menu-based search scheme, which as developed at the University of Minnesota, lets a user reach a destination on the Internet by selecting items from a series of text menus.

Access Agent

A logical component that provides support for different access protocols and data streams -- Frame Relay, HDLC (High Data Link Control) CBO (Continuous bit Operations, ATM (Asynchronous Transfer Mode), or TCP/IP.

Application Processing Agent

A data processing agent running in a server data processing system which performs tasks based on received requests from a client in a distributed environment. In our preferred embodiment, our application processing agent for database retrieval is our DIS server, a data interpretation system server and database gateway which is coupled to our web server HTTPD via a network. In our preferred embodiment an application processing agent employs executable object programs as command file objects, which in the preferred embodiment are capsule objects.

Client

A client is a computer serviced by the server which provides commands to the server.

Data Interpretation System (DIS).

IBM's object oriented decision support tool.

Capsule

A DIS capsule is a program created by a DIS programmer and executed in the DIS environment. A DIS capsule is a preferred example of a capsule object. A capsule object is a specialized form of a command file (which is a list of commands to be executed, as in an EXEC or *.BAT batch file. The capsule object is created with an object environment, as is supplied by IBM's DIS. Other object environments are IBM's SOM and DSOM, and Microsoft's COM environment.

Internet

The connection system that links computers worldwide in a web.

Server

A machine which supports one or more clients and is part of the web. Any computer that performs a task at the command of another computer is a server.

Slip or PPP connection.

Serial-line Internet protocol and point-to-point protocol, respectively, for providing a full access connection for a computer to the Internet. Transmission control protocol/Internet protocol. A packet switching scheme the Internet uses to chop, route, and reconstruct the data it handles, from e-mail to video.

InterNetwork Routing (INR)

The link between systems which routes data from one
physical unit to another according to the applicable protocol. The protocol will employ a URL address for Internet locations.

URL

Universal resource locater, a Web document version of an e-mail address. URLs are very cumbersome if they belong to documents buried deep within others. They can be accessed with a Hyperlink.

Web browser

An program running on a computer that acts as an Internet tour guide, complete with pictorial desktops, directories and search tools used when a user "surfs" the Internet. In this application the Web browser is a client service which communicates with the World Wide Web.

HTTPD

An IBM OS/2 Web Server or other server having Hypertext Markup Language and Common Gateway Interface. In our preferred embodiment, the HTTPD incorporates our control program agent and is supported by an access agent which provides the hardware connections to machines on the intranet and access to the Internet, such as TCP/IP couplings.

HTTP Hypertext transfer protocol

Hypertext transfer protocol. At the beginning of a URL "http:" indicates the file contains hyperlinks.

Hyperlink

A network address embedded in a word, phrase, icon or picture that is activated when you select the highlighted tidbit. Information about that item is currently retrieved to the client supporting a Web browser.

HyperText Markup Language (HTML)

HTML is the language used by Web servers to create and connect documents that are viewed by Web clients. HTML uses Hypertext documents. Other uses of Hypertext documents are described in U.S. Patents 5,204,947, granted April 20, 1993 to Bernstein et al.; 5,297,249, granted March 22, 1994 to Bernstein et al.; 5,355,472, granted October 11, 1994 to Lewis; all of which are assigned to International Business Machines Corporation, and which are referenced herein.

BACKGROUND OF THE INVENTIONS

The Internet is not a single network, it has no owner or controller, but is an unruly network of networks, a confederation of many different nets, public and private, big and small, that have agreed to connect to one another. An intranet is a network which is restricted and while it may follow the Internet protocol, none or only part of the network available from outside a "firewall" surrounding the intranet is part of the agreed connectionto the Internet. The composite network represented by these networks relies on no single transmission medium, bi-directional communication can occur via satellite links, fiber-optic trunk lines, phone lines, cable TV wires and local radio links. When your client computer logs onto the Internet at a university, a corporate office or from home, everything looks local, but the access to the network does cost time and line charges.

Until recently, "cruising or surfing" the Internet was a disorienting, even infuriating experience, something like trying to navigate without charts. The World Wide Web, a sub-network of the Internet, introduced about two years ago, made it easier by letting people jump from one server to another simply by selecting a highlighted word, picture or icon (a program object representation) about which they want more information -- a maneuver known as a "hyperlink". In order to explore the WWW today, the user loads a special navigation program, called a "Web browser" onto his computer. While there are several versions of Web browsers, IBM's example is the new WebExplorer which offers users of IBM's OS/2 Warp system software a consistent, easy to use desktop of pictorial icons and pull down menus. As part of a group of integrated applications available from IBM for OS/2 Warp called the IBM Internet Connection, lets users log onto the Internet.

To this point the World Wide Web (Web) provided by Internet has been used in industry predominately as a means of communication, advertisement, and placement of orders. As background for our invention there now exists a number of Internet browsers. Common examples are NetScape, Mosaic and IBM's Web Explorer. Browsers allow a user of a client to access servers located throughout the world for information which is stored therein and provided to the client by the server by sending files or data packs to the requesting client from the server's resources. An example of such a request might be something called GSQL (get SQL) which was a NCSA language and CGI server program developed to getting textual results for a client caller. Developed by Jason Ng at the University of Illinois, this document provided a way to map SQL forms against a database, and return the textual results to the client caller. This system is unlike the present invention, and presents difficulties which are overcome by our described system.

These servers act as a kind of Application Processing Agent, or (as they may be referred to) an "intelligent agent", by receiving a function request from a client in response to which the server which performs tasks, the function, based on received requests from a client in a distributed environment. This function shipping concept in a distributed environment was first illustrated by CICS as a result of the invention described in U.S. Patent 4,274,139 to Hodgkinson et al. This kind of function, illustrated by CICS and its improvements, has been widely used in what is now known as transaction processing. However, servers today, while performing many functions, do not permit the functions which we have developed to be performed as we will describe.

Now, "surfing" the Internet with the WWW is still a time consuming affair, and the information received is not generally useful in the form presented. Even with 14,400 baud connection to the Internet much line time is tied up in just keeping going an access to the Internet, and the users don't generally know where to go. Furthermore the coupling of resources available on a company's intranet and those available on the Internet has not been resolved. There is a need to reduce gateways, make better use of existing equipment, and allow greater and more effective usage of information which is resident in many different databases on many different servers, not only within a homogeneous network but also via the Internet and heterogeneous network systems.

The problems with creating access to the world via the Internet and still to allow internal access to databases has been enormous. However, the need for a system which can be used across machines and operating systems and differing gateways is strongly felt by users of the Internet today. Anyone who has spent hours at a WWW browser doing simple task knows how difficult it still is to navigate thorough arcane rules without knowing where to go and even if you know what you are doing spending hours doing routine tasks. Many needs exist. As one important instance, until now we know of no way to access data on multiple databases of different types using a single user request from a client.

This and other difficulties are solved by our invention.

## SUMMARY OF THE INVENTIONS

In accordance with our invention needless user intervention is eliminate or greatly reduced with a Web server supports an HTTPD which is provided with the capabilities of our control program agent which organizes sub-agents supporting command file objects or capsules to perform tasks in support of a Web browser's request for service as programmable functions receiving parameters as input and providing as their output handled by the control program agent task completed results for reporting in accordance with the Web browser request in the form and to the location determined by a request and handling these request without needless user intervention.

In accordance with our invention, we have created a way to allow Web users to request information that is created by a data interpretation system (DIS) and then presented by a web server to the user of the web. Our solution provides a way of requesting and processing and presenting information on the Web. In the process, data is retrieved from multiple sources which may be located remotely and accessed via an intranet routing and via the Web Internet and processed by our decision support capsules. Now companies and universities, and other users that want to access data located on different databases, want that data processed and formatted, and presented in a form the user desires, such as a graphical format. Our solution permits users to access information from various sources and obtain information at a desired location as a result of a single request which is responded to by an organization of facilities and command file sub-agent decision support capsule objects by our command program agent. Users of the information can be internal to a company, or external. The result can be furnished to a user at a location which is internal or external to the company, and as specified at a specified location with a form and format desired. This allows a report to be managed by the web support services we provide, and in a form consistent with the request, but without requiring a consistent interface solution.

In order to create a way for Web users to request information generation we provide a web server with a control program agent which is linked to a decision support tool of a data interpretation system server, the application processing agent, and then have that server retrieve, process, and format information which is presented to the user on the Web by the Web server. In our preferred embodiment, we have provided a link between a Hypertext Markup Language (HTML) document using a common gateway interface, and open data interpretation system server (ODAS). As a result, Web clients can request DIS reports to be generated, specify the parameters to be used in generating the reports, and then view the report results on a Web home page. The DIS capsule can generate graphical information, such as colored pie charts, line graphs, bar graphs, and other forms of generated information. Since the Web server is capably of presenting the results in desired formats, the full capabilities of a DIS report are utilized.

Our invention provides a method and system for allowing a user of a client to access and assemble information structured and reported to the user in accordance with his desires, selecting information for disparate servers which are located within a network can be an intranet or internal network, such as a LAN or WAN not normally accessible to the Internet, or coupled to the Internet. In accordance with our invention one can access data on multiple databases of different types using a single user request from a client. We also allow the facility for providing specialized specific requests to be created for routine use, as well as the facility to formulate generalized or specialized ad hoc requests. In addition, we provide besides query and update capability, the ability to perform calculations with respect to any retrieved data, to format the information in text or in graphics, and the facility of presenting the results to the client for display or other use.

The improvements which we have made achieve a means for accepting Web client requests for information, obtaining data from one or more databases which may be located on multiple platforms at different physical locations on an Internet or on the Internet, processing that data into meaningful information, and presenting that information to the Web client in a text or graphics display as a location specified by the request.

Our invention of providing a web server with a control program agent allows organization of decision support functions to be executed by application processing agent servers located throughout the Internet to gather and supply information not presently available with any existing resources without the need of endless intervention on the part of a requesting user of the WWW; further enabling an ordinary user to take advantage of expertise which is provided by programmable sub-agents developed by those with particular expertise in a given area as well as enabling use of standard routines commonly needed.

These improvements are accomplished by providing for Web clients to request information from an application processing agent in which the application processing agent server performs tasks based on received requests from a client in a distributed environment by a web server supported by an access agent link and control program agent which in turn causes a decision support function to be executed by the application processing agent server. This is performed within the distributed environment by the application processing agent server which forms part of a network coupled to and under control of the control program agent. According to our invention the decision support function is provided by a data interpretation system which functions as part of the application processing agent and the decision support function is programmable and generated by a data interpretation system, DIS or other decision support element performing similar functions, and provided in a form accessible to our control program agent which presents the output generated to be presented to the user on the Web who made the initial request. We have provided, in a preferred embodiment, a link between IBM's Hypertext Markup Language (HTML), the Common Gateway Interface (CGI), and the Open DIS Access Server (ODAS), all of which may be used on machines which are are commercially available from IBM. In order to write additional functions which develop our invention, the reader is referred to the Medaphor Data Interpretation System publication "Developing Applications with OpenDIS Access Service, Version 2.0, available from IBM, First Edition (September 1994) Part Number 315-0002-01 which is incorporated herein by reference.

Our improvements relating to our control program agent is in accordance with our preferred embodiment is normally installed on an IBM HTTPD which is an IBM OS/2 Web Server or other server having Hypertext Markup Language and Common Gateway Interface. In our preferred embodiment, the HTTPD incorporates our control program agent and is supported by an access agent which provides the hardware connections to machines on the intranet and access to the Internet, such as TCP/IP couplings. The hardware for the Web server is thus a workstation, such as IBM's PS/2 model 80 with OS/2. However, the HTTPD can be installed in PCs and upwardly also in machines which range across IBM's line of computers from powerful personal computers to mainframe systems which support MVS, IBM's operating system which enables multiple kinds of operating systems, including "UNIX" to coexist on a single platform. As a result of our invention Web clients can request DIS reports to be generated by the application processing agent specifying the parameters to be used in generating the reports, and then as a result of the request receive a result which is presented, as a visual display or otherwise, on a Web page for use by the requesting user. Our machine implementation allows a user having DIS access to generate graphical information such as colored pie charts, line graphs, bar graphs, etc. Since Web browsers such as IBM's Web Explorer are capable of displaying these formats, all the functions which can be created by a DIS capsule can be utilized by a user of our invention.

According to our improved method, an Internet World Wide Web user connects to a Web server through the use of a Web browser. In accordance with our preferred embodiment, we use HTML as the language used by Web servers to create and connect documents that are viewed by Web clients. HTML is an example of a hypertext language having the facility of clicking on a highlighted word, string of words, or image in order to move to another HTML document or invoke a program on the server. An example of a Web client would be a machine used by a person using IBM's Web Explorer product. In using our invention a user may click on the hypertext in a document to reference a function which will be provided by an application processing agent server. The user is able to connect to another document that may be on another Web server. HTML commands are used to reference other documents. HTML is used to reference programs available on a server, and pass parameters to those programs. The application processing agent server executes a program when it is referred to by a Web client via a control program agent resident, preferably, in a Web server.

The Web client selects the information that they wish to view by using the HTML created page, the Web server takes the client request and passes it to a C program implementation of our control program agent. Web servers, such as HTTPD for OS/2, with our control program agent are able to provide access to executable programs through the use of the Common Gateway Interface (CGI). When a program is referenced by the HTML, any parameters are passed to the program and it is executed. In our preferred embodiment we have used CGI to invoke programs that we have developed that will interface with the DIS product. CGI is an example of a software gateway from a Web server to programs outside the Web server application.

The control program agent that is called in this instance by the Web server through the CGI interface, passes the Web client request along to a data interpretation system DIS via a Open Dis Access Server ( ODAS ). ODAS is a feature of a data interpretation system DIS that allows programs to initiate DIS functions, such as invoking DIS capsules. Our control program agents interface with DIS through ODAS to submit DIS capsules for execution. DIS capsules are basically programs that DIS application programmers create with the DIS programming language. In accordance with our invention, we have written capsules which are executed as a DIS capsule on a server to gather data from one or more databases, process that data, and create a report in one of many formats, which we will describe by way of example. After the DIS capsule completes executing, in accordance with our preferred embodiment, the results that are generated during execution of a capsule are stored in a file on the application processing server.

After DIS creates a file that contains the formatted report results, our control program agents program dynamically creates HTML tags to present the formatted report back to the Web client on the Internet. Our control program agents using the CGI interface can create HTML commands dynamically. In this way a program can present information on a Web browser for the Web client.

After the DIS capsule has created the file containing the report request results, the control program creates HTML statements dynamically that display the report results to the Web browser.

Alternative means of presenting the data are shown by alternative routing. The user requesting the report may wish to have the report results sent to another location in addition to or instead of displaying the report results to the Web browser. This information is provided during the request phase. As a result of the alternative report request, and according to the parameters indicated therein, the report results can be sent by the control program via electronic mail, i.e. TCP/IP Sendmail facility and Lotus Notes, to one or more locations on the Internet. The report results can be sent as a file and as a note. The request can request a voice response, which can be routed to a voice response unit. Thus, with a call to a translator, the text can be converted to voice, and even translated along the way. The report results can also be sent to a fax machine, or to a computer that has the capability of receiving fax data.

We use these report concepts to present report files created by DIS capsules on the Web client display.

These and other improvements are set forth in the following detailed description. For a better understanding of the invention with advantages and features, refer to the description and to the drawings.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 shows schematically an overview of the preferred embodiment and particularly shows a typical distributed computer system which has internal and external networks including the Internet to connect clients to World Wide Web servers and other servers within the system in which our invention is situate.

FIGURE 2 shows a inquiry screen (home page) which is displayed on a client after the client is coupled to its server (which may be an Internet server) by a Web browser.

FIGURE 3 is a next screen which illustrates how a request is made according to a users desires, making a request in accordance with our invention with an input screen shown.

FIGURE 4 is a sample result screen which is returned to the client after the requested service is provided by the computer system network in accordance with our invention formatted according to the specifications of a DIS capsule.

FIGURE 5 is a next screen which illustrates how a request is made according to a users desires, making a request in accordance with our invention by selection from a menu and through the use of image mapping.

FIGURE 6 is an example of a graphical result screen which is returned to the client after the requested service is provided by the computer system network in accordance with our invention.

FIGURE 7 illustrates a flowchart showing data flow between a web server and decision support system tool such as IBM's Data Interpretation System (DIS), and shows the coupling of a Web client to a Web server and the coupling of a request to execute a DIS capsule and the coupling within the Web server from ODAS to a distributed DIS LAN with heterogeneous connections to multiple databases.

FIGURE 8 illustrates as a flow chart the functions of the control program for the web server.

FIGURE 9 illustrates by way of example a DIS capsule that creates a text report file.

FIGURE 10 illustrates by way of example a DIS capsule that creates a graphical report file.

FIGURE 11 illustrates an alternative configuration of the network system as it may be employed for permitting access to information available through homepages and in data warehouses where access to the homepage or database may or may not be restricted by a firewall.

(Note: For convenience of illustration, in the formal drawings FIGURES may be separated in parts and as a convention we place the top of the FIGURE as the first sheet, with subsequent sheets proceeding down and across when viewing the FIGURE, in the event that multiple sheets are used.)

Our detailed description explains the preferred embodiments of our invention, together with advantages and features, by way of example with reference to the following drawings.

## DETAILED DESCRIPTION OF THE INVENTION

Figure 1 illustrates a information delivery solution of a typical combination of resources including clients and servers which may be personal computers or workstations as clients, and workstations to mainframe servers as servers. The various elements are coupled to one another by various networks, including LANs, WANs, and other networks, which may be internal SNA networks or other like internal networks, and also providing access to the Internet, which couples the system to the world via Internet.

### The Preferred Embodiment

Turning now to our invention in greater detail, it will be seen from FIGURE 1 that our preferred embodiment provides a Web browser 10, which is coupled to a Web server 11. Our Internet WWW browser is an intelligent computer system, such as an IBM PS/2, or other computer, an IBM ThinkPad, an RS/6000 works as well and connections are made to the network via OS/2 WARP Connect, an IBM product. The Internet Web browser in the intelligent computer system which performs the Web browser function has IBM Web Explorer, or NetScape or Mosaic installed thereon. This computer system 10 is bi-directionally coupled with the OS/2 WARP Connect facility over a line or via a wireless system to our preferred computer system which we call our Web server. This system is a PS/2 or RS/6000 or other similar system which includes our control program agent 73, which will be discussed below. Web server 11, in our preferred embodiment is coupled again bi-directionally via a line or wireless coupling to a computer system, such as a PS/2 or RS/6000 or other server which supports and performs the server function of ODAS server 12, which is coupled to the distributed DIS network, here shown as LAN 13. ODAS 12 may be located on the same server as the Web server 11 or be located at a separate service machine, such as an IBM Digital Server. The Web server is logically coupled to our application processing agent server via a network. We call our application processing agent server a DIS File server 14 because is comprises a data interpretation system which supports the decision support functions we provide which is today most inexpensively provided by an IBM computer system which supports OS/2. In our preferred embodiment, the intranet network is a LAN. Thus the components of the DIS LAN 13 comprise a DIS File Server 14, a general purpose workstation 15 which can be used for capsule development, a local database server 16, a Capsule Server 17 for storing a plurality of DIS capsules ready for user, a Database Gateway Server 18 which performs the gateway functions to access databases which are linked to it, these databases include geographically distributed databases which can be located, for instance, in Chicago, New York, Dallas, Los Angeles, and each of which can be a different supported database, such as DB2 database 19, ORACLE database 20, Sybase database 21, Redbrick database 22. In our preferred embodiment all servers are coupled with a conventional LAN or WAN connection, with a preferred IBM token ring shown. Reference should also be had to our alternative preferred embodiment discussed below with respect to FIGURE 11.

Thus, in connection with the preferred embodiment of FIGURE 1 as well as with respect to FIGURE 11 it would be appreciated from the schematic overview illustrated by FIGURE 1 and FIGURE 11 that our invention may be employed in a distributed computer system environment which has internal or intranet networks represented in our preferred embodiment by the DIS Network 13 and external networks including the Internet to connect clients to World Wide Web servers and other servers within the system in which our invention is situate. Our invention makes use of the entire network. The Web browser 10 can make a request to the Web Server 11 for a report. The Web server 11 with the facilities we provide causes the application processing agent which includes our DIS server 14 and its supporting communication server, the database gateway server 18, to act as an agent to gather data from one or more of the multiple databases, including the local database 16, DB2 database 19, ORACLE database 20, Sybase database 21, Redbrick database 22. Further details with respect to the use of our invention for database retrieval of information from multiple databases are provided as to the actions of the application processing agent functions of the database server(s) 18 with reference to FIGURE 7.

Thus, returning to our simplified and preferred embodiment, FIGURE 2 shows a inquiry screen (home page) 29 in the form which is displayed on a client after the client is coupled to its server (which may be an Internet Web server 11) by a Web browser 10. The entire screen contains information and a plurality of objects. Once the home page is displayed, with appropriate descriptive guidance as illustrated by the FIGURE 2, the user can interact, for example, by clicking on image objects 30, 31, 32, 33, 34. As a example should the user want to make a special request in accordance with our invention, he could click on image 30. This would take the user to the next screen, illustrated by FIGURE 3. Alternatively the user could select by clicking on image 31 another menu screen, illustrated by FIGURE 5. At this point also, a specialized format could be selected by double clicking first on a format select image illustrated by image objects representing access to menu screens 32, 33, 34, one or more of which a gopher.

The use of selection of icon image object is a function provided by HTML and programmers knowing this language can readily create variants to the images and functions we have illustrated. Thus incorporated within the drawings are to be understood to be the variants that can thus be created using our examples, as well as extensions and combinations thereof.

When the user selected image 30 by clicking on the image 30, FIGURE 3 appears. FIGURE 3 is the next screen which illustrates how a request is made according to a users desires, making a request in accordance with our invention with an input screen shown. The content of FIGURE 3 is preformatted 40 except for the user entries which are to be entered in the data input fields 41. In this example the input field 41 is a userid. After a user has entered in field 41 an acceptable input, he would then click on instruction key 42. The instruction key illustrated is submit a request. At this point the Web server captures the information entered by the user, as described in FIGURE 7. It will be appreciated that the Web server captures the information entered by the user, including specialized input, as well as any "hidden" default information, which can include password authorizations, charge account identification, and other information that can be used by the system in responding to the request. Thus the system can assume that the "hidden" password is an authorization to perform some function, such as include information from confidential source, or exit to the Internet. The charge authorization can also be tracked and accumulated by the system as it parses through its functions to charge back chargeable usages. If a request is for an order of an item, the actual item requested can be shipped and billed with this information. Since these functions are "hidden" they do not appear in the FIGURE but included with a request. The return of the request is illustrated in FIGURE 4.

FIGURE 4 is a sample result screen which illustrates how a sample report conforming to the request results are presented to the client after the requested service is provided by the computer system network in accordance with our invention formatted according to the specifications of a DIS capsule which is illustrated by example in FIGURE 9. In this example, the return was a file, whose file name is displayed as P81484 at 43. Informative text accompanying the file is included as illustrated by the example information 44 The screen provides the content of file 43 in the requested form of preformatted text 50 in the form of a display of a text report generated by a DIS capsule stored in the DIS server 17. While we show text as the form the report results, the form of the request can be another form of presentation, as and image, a voice response, or other multimedia presentation.Reports can be returned translated into any desired language based upon the request, as may be provided by DIS capsule calls to a translator. These features are included in the result 50 report.

When the user selected image made by clicking on the image 32 in FIGURE 2, FIGURE 5 appears. FIGURE 5 is a next screen which illustrates how a request is made according to a users desires. A user makes a request, in this instance for sales results within the organization for YTD Catalog Revenue in accordance with our invention by entering text data into the data entry areas 41 and 42 of the formatted screen with information as to type of data selected 40A which will be translated into specific report information created by a DIS capsule.

FIGURE 6 is a sample result screen which illustrates how the request results are presented to the client after the requested service is provided by the computer system network in accordance with our invention formatted according to the specifications of a DIS capsule. In this instance selection of the object 32 links to the the screen of FIGURE 5, which in turn with the DIS capsule created the output shown in FIGURE 6. DIS Capsules will be illustrated by examples in FIGURE 9 and 10. In this example the output of the DIS capsule illustrated in FIGURE 10 is presented on the screen shown by FIGURE 6. The screen comprises a file name identifier, descriptive information 61, and preformatted text 60 which is the display of the named file P555119. This is the display of a graphic report showing what might be deemed (but is not) Confidential information relating to Catalog Revenue for 1995 YTD, with revenue given in $M, and breakout as to HDW, SFW, PMV, MN and MNT from selected locations in Chicago, New York, Dallas, and Los Angeles, all of which are located on different systems, and which, as illustrated in FIGURE 1, may be on different databases such as DB2, Oracle, and Sybase relational databases. This report was generated by a DIS capsule which is illustrated in FIGURE 9. This example illustrates how multiple actions can be taken on information retrieved. In this example data was translated into image material by calculation and formatting in the form of a graphic pie shaped report. Other image data could also be displayed, as frames of selected images, or a sequence of images in the form of a moving picture display, which can be outputted from a server as will be described in FIGURE 11.

FIGURE 7 illustrates a flowchart showing data flow between a web server and decision support system tool such as IBM's Data Interpretation System (DIS). FIGURE 7 shows the coupling of a Web client 71 (corresponding to Web browser 10 in FIGURE 1) to a Web server 72 (corresponding to Internet WWW server 11) and the coupling of a request to execute a DIS capsule.

The Web browser 71 can make a request to the Web Server 72 for a report through the use of HTML. The HTML document refers to our control program agent 73, which may be implemented with the C language or other language which can provide run code for the particular Web server which is employed. We illustrate our preferred program according to the description provided in FIGURE 8. The Web Server 72 passes request data to and invokes our control program 73 through the use of the CGI in accordance with our invention. The control program uses ODAS 74 in ODAS server 12 to set DIS capsule parameters and initiates the execution of a DIS capsule located in this embodiment in DIS capsule server 17 according to our preferred examples illustrated in FIGURES 9 and 10.

After a DIS capsule completes execution, the file created by the DIS capsule contains the formatted report results requested by the user. Our control program 73 dynamically creates the HTML statements that present the file to the Web browser 10 screen. Figure 7 shows the coupling within the Web server from ODAS 74 to a distributed DIS LAN 75 with heterogeneous connections to multiple databases DB2, Redbrick, Sybase and Oracle. Other sources of data can be linked to the LAN.

## Preferred Embodiment Interface between Server and DIS

Our preferred control program agent 73 in FIGURES 1 and 11 is illustrated in detail by way of the flowchart of FIGURE 8. In our preferred embodiment, this program can be written in C or other suitable language but for general appreciation of the details, we will describe the steps in detail. These steps can be implemented by programmers of ordinary skill in the art without undue experimentation after understanding the steps described below. The control program agent 73 is located in a Web server and provides an interface and execution functions. Thus in FIGURE 11 the function is provided between the Web Server 131 (corresponding to Internet WWW server 11 in FIGURE 1) and DIS which is located in a DIS server 133 (corresponding to server 14 in FIGURE 1) and for presentation of results according to the instructions of the Web browser 130 (corresponding to browser 10 in FIGURE 1) according to the request command, which in default is a return to the Web browser home page. This interface utilizes in our preferred embodiment the Web Server CGI and the DIS ODAS.

Before we proceed to the control program 73, it will be noted that in FIGURE 11 the Web Browser 130 will link to a Web Server 131 accessing it on the Internet though a unique ID called the uniform resource locater to access the node which we call the Web server 131. When that access takes place an HTML document is displayed by the Web server 131 to the Web browser 130, as shown in FIGURE 2. Now, the user makes his entries as described with respect to FIGURE 2. Next the HTML document refers to the control program agent 73 and the Web server 131 through the use of the CGI invokes our control program agent 73. The Web server 131 retrieves data entered by the user from the HTML document and passes that data to our control program agent 73 upon invocation.

The Web Server 131 has a gateway interface that allows the server to invoke a control program agent 73 running on it and to pass input parameters to the control program agent 73 (FIGURE 8) that were returned from the Hypertext document of the Web Browser. It will be appreciated that while weillustrate for our preferred example a single Web Server 131, the Hypertext document locates the particular Web Server that can support the request made by checking the details of the "hidden" defaults and those functions requested. Thus a menu request for a generalized search throughout the Internet may locate the particular service machine having an application processing agent which has the information desired. Once the control program 73 (FIGURE 8) is invoked, the steps programmed for the machine to follow begins with a step 110 illustrated in FIGURE 8. In reviewing this preferred control program agent it should be appreciate that steps 110 and step 111 are steps that are interchangeable in order and which obtain environment variable data from the HTML document return.

Thus step 110 obtains a PATH_INFO environment variable data. PATH_INFO contains data from the HTML document that referred the Web Server to our program. Specifically the data contains the name of the DIS capsule to call, the name of the file containing the HTML statements to use when building the HTML document that displays the DIS capsule results to the Web browser, and the type of file that the DIS capsule will create. All off this information is the variable data which is stored in a buffer environment in step 112, and which is used in subsequent steps.

Thus also, the control program proceeds with step 111 which may follow or precede or proceed in parallel with step 110 to obtain the QUERY_STRING environment variable data. QUERY_STRING contains data from the HTML document that referred the Web Server to our program. Specifically the data contains values selected by the user and / or default values selected by the HTML document designer. These values are set in the DIS capsule by our control program prior to execution of the DIS capsule.This information is used to set variables in the DIS capsule. All off this information is the variable data which is stored in a buffer environment in step 112, and which is used in subsequent steps.

Within the scope of the discussion of the control program agent illustrated by FIGURE 8 it should be appreciate that the steps 112 through 125 include the utilization of an API set that provides a method of invoking executable programs located in a service machine which we denote as a sub-agent which executes in step 122 object capsules from our subagent DIS file server 14. This provides functions such as queue and update functions for databases on multiple platforms and allows the processing of data retried from a database to be performed, including executing calculations, doing formatting, charging of accounts and the storing of results as a file accessible to the control program agent.During processing our control program agent 73 provides setups for API calls which occurs in steps WHAT ARE THESE STEPS. Thus the control program agent will proceed as with an API set with step 113.

With the variable information now stored in a buffer, in step 113 the control program retrieves from a store, all of the DIS capsules that are used and the variable names associated with each DIS capsule and loads into memory associated with the control program the DIS capsule names available and the variable names associated with each DIS capsule.

At that point in step 114 the control program is ready to and does initialize a connection between our control program and the ODAS through the use of an ODAS API. In other environments another API performing similar functions could be used.

At that point, if required for control by the decision support system, and as required by DIS, the control program would log onto the port or desktop for the assigned user. Thus, our control program agent 73 in step 115 logs onto a DIS "desktop", our DIS file server 14.

Once the DIS capsule information is loaded into control program memory, the control program can and does in step 116 retrieve from its memory the DIS capsule variable names associated with the DIS capsule name passed to our control program in the step 110 where PATH_INFO is provided.

Next, in step 117 the control program creates a data array stored in the control program memory containing the DIS capsule variable names and the values for them that were passed to our control program in the QUERY_STRING step. These two steps 116 and 117 should be done in order, even though steps 110 and 111 can have an arbitrary order. At this point in step 117 you are matching the DIS capsule variable names with the data that was passed to the control program in the QUERY_STING environment variables.

Next, in preparation for a report, in step 118 the program creates a unique filename which may include data originated by the HTML document's variables stored in step 112 (dotted line) to pass to the DIS capsule as a DIS variable for use in naming the report which will be created by the DIS capsule. As a result, the DIS capsule will create that file with the unique file name during its process.

In anticipation of DIS capsule execution, the values of variables used by the DIS capsule are obtained from the data array in the control program memory containing the DIS capsule variable names and the values for them that were passed to our control program in the QUERY_STRING step. This is done in step 119 using the ODAS API to set the DIS capsule variable values. At this point the capsule server 17 for the DIS server 133 attached to the Web Server 131 via network 132 will have a DIS capsule services queue. This queue is the queue of jobs being requested of the Dis Capsule Server 17. For the current job request (other like requests being perhaps still in the queue) we use the ODAS API to query the contents of the DIS Capsule Services queue. If the queue size is larger (>t) than a threshold level, then the process enters a wait state until the queue size is reduced to a tolerable level. The queue test of step 120 is a loop test which returns to test the queue size until a test answering "is the queue of a size that execution can proceed?" (<t)

is answered "YES".

Whenever the queue test is answered YES, at that point the ODAS API is used to submit a DIS capsule for execution in step 121.

After the ODAS API submits a DIS capsule for execution the particular request process being executed by the control program enters a wait state until completion of the DIS capsule execution. For this step of the process the control program uses the ODAS API to wait for completion of the DIS capsule execution performed by the DIS capsule execution 122. During a wait state other requests can be processed by the control program, as requests are fed through the control program as a pipeline, in this WAIT PIPE API step 123, so that the control program continually advances requests through the system.

During the wait state 123 the ODAS API looks for a completion signal. When that is received, the control program then in step 124 reads the file identified by the name passed to the control program in the first PATH_INFO step that contains the HTML statements which are to be presented with the DIS report results.

While in step 124 the control program reads the file identified, it dynamically creates new HTML statements to display the preformatted text to the Web browser.The new HTML statement include the information retrieved from the file in step 113 so that it can be displayed as a header 44 accompanying the report to be displayed, along with the filename 43.

At this point, in step 125 the control program tests for the kind of report to be created by obtaining information from stored variables and identifies output parameters, such as whether the report is to be a text report, or a graphical report. At this point the control program branches to the sequence applicable to the kind of report to be created. If the output is to be routed the the Web server 10, then the output is routed to the Web server in step 126.

If a text file report is created by the DIS capsule, that determines that a text display is to be reported and the the control program reads the file created by the DIS capsule and dynamically creates HTML statements to display the data lines to the Web browser.

If a graphics file is created by the DIS capsule, that determines that a graphics display is to be reported and the the control program dynamically creates the HTML statement to display the graphics file to the Web browser.

On the other hand, the control program agent allows alternative output direction, and if the output is another type, or an additional output, as for broadcast, it can be routed to another destination. In step 127, we illustrate how using the IBM Digital Server, output can be routed to a requestor selected resulting output selected from a group of possible output units, comprising fax, printer, retail or banking installations, or provided as a series of full motion videos or still frames which are can be transmitted to display devices, such as a TV set under control of end users with a set-top box cable control. These facilities are provided by providing the output of our control program agent from the web server to the alternative output device 127, In this case, the IBM Digital Server, which with an RS/6000 CPU, Network I/F Bus, DISKs, modems, and X.25 Data Switch provides the hardware to route the output to a variety of output devices, to fax, printer, retail, banking, TV or cable customers via the digital server service machine for full motion and still video, supplied with MPEG 2 and MPEG 1 protocol images respectively, to subscribers, Along the way, the output can be coupled to an auxiliary function, such as back-up or accounting processes 128 which allow for charging for system utilization and service charges for services and items requested. These processes will make use of hidden variables associated with the request, such as charge authorization. One of the hidden variables which may be associated with a request is a credit card number. The credit card number, is preferably encrypted, with a DES or RSA encryption utility, and this along with access authorization variables, will allow access to sensitive databases which reside behind firewalls. If selected data according to the request is permitted to the access authorized user at the location inside or outside the Internet, the data can be included in the results reported by our system to the Web browser.

Preferred Embodiment of text DIS capsule

In accordance with our invention, an HTML document, which is running on a web server, refers to the control program agent. The web server then invokes the control program agent. The control program agent has a to command files, which provide the preferred file command objects in the form of DIS capsule objects, or DIS capsules as they are known. The command file contains a list of available DIS capsules. Accordingly, there is not need for the HTML document to know how to get to the command file, as the control program supplies this access. A capsule object, as a DIS capsule, can call other routines which may be written in well known programming languages such as Visual Basic or C. These routines become part of the capsule object by the reference, and these routines perform such functions as account tracking, compression, calculation, handling specific custom outputs such as video, voice, translation, and enable programmability of the capsule objects. The capsule objects also have standard object capability, and we will illustrate these by way of the specific examples described.

It will be seen that the control program 73, described in detail in FIGURE 8 acts in concert with DIS capsule execution. The DIS capsule is an object program with executable additions which we have created to interact with the control program. It should be also understood that the DIS capsule object can perform programmable functions on data which is retrieved from databases. Not only can a DIS capsule get data, it can combine, reformat, and update, the data retrieved. It can act on the data to create new data and basically act as a dedicated processor processing data gathered or created during a Web browser request to output the end result to the user under programmable parameters determined by the creator of the DIS capsule, as they may be selected, if desired, by the user as part of the request. Thus the user entered inputs as part of his request, either free form or by selection of variables in the menus afforded to the user as illustrated by way of example in FIGURE 5.

DIS capsule objects are like some other objects. For instance in Microsoft's products, an example being the Excel (trademark of Microsoft) spreadsheet, one can click on an object portrayed on the screen and link a succession of objects to perform a specific function, such as take data from a spreadsheet and reformat it into a variety of selectable formats, such as text or graphic illustration. The kind of action to be taken is illustrated by an object on the screen, and linking of routines is done by a succession of clicks on icons representing the object.

In accordance with our preferred embodiment, a DIS capsule is used to invoke system resources. This is done by providing a list of commands, which may be those provided by a DIS processor itself, or written in Visual Basic or C by the programmer. The result is a command file, like an exec or command file in OS/2 or like a *.BAT file in DOS. These capsules perform the specific functions that are requested by the user from his initiation session. The user further qualifies the execution of the DIS capsule by providing parameters which are used in the invocation.

Now the DIS server 133 supports DIS, the program processor which supports DIS capsules by processing commands contained in the DIS capsule, either directly, in the case of DIS functions, or by to other system or user supplied functions. The user supplied functions comprise mainly those DIS functions which are supplied by DIS and illustrated in the manual "Developing Applications with OpenDIS Access Service, Version 2.0 of the OPEN Access Service." For those not familiar with command files, this manual is fully incorporated herein by this reference as available at the USPTO. An example of a system supplied function would be the base support for SQL queries of a specific database, which are invoked by the DIS capsule program.

In illustrating the specific examples of our invention illustrated in FIGURES 9 and 10, both illustrate linked objects according to a specified flow sequence within a DIS environment. The DIS environment contains numerous functions, including the Internetwork routing functions which the DIS capsules can invoke. Thus, a DIS object which queries a database, as illustrated, invokes the Internetwork routing functions to query databases where they are located on the network. If the preferred example of DIS environment is not supplied, a similar environment with program environment means which supports reaching a destination on the Internet by a link between systems which routes data from one physical unit to another according to the applicable protocol should be supplied. The protocol will employ a URL address for Internet locations.

FIGURE 9 illustrates by way of example a DIS capsule that creates a text report file. Referring to FIGURE 9, it will be seen that the capsule, represented by a series of linked objects, is supported by Internetwork processor support environment means 90. Within this environment an integrated capsule creates a text report file as a result of the object 95, make text. This object result file is the file 43 according to FIGURE 3 which is displayed at the browser. In the illustrated example, the multiple DIS capsule data retrieval command file 91(a)..91(n) initiates as a first step multiple queries to different databases which are specified by the parameters of the request. In this example, multiple queries are initiates as SQL type search requests as multiple steps 91(a)..91(n) executed by the DIS capsule server 133 with the Database Gateway 134 to select data from DB26000 databases located inside the intranet 140 and on the Internet by Internetwork routing to database gateway 134' and its DB26000 databases by step 91(a). The data is stored in a DIS declared buffer. Similarly, in parallel or successively, additional steps 91(b), 91(c), 91(d), and 91(n) retrieve data and store in their object buffer data retrieved from Sybase, Oracle, Redbrick, and IBM's Data Warehouse databases. Thus object 91(a) will query DB26000 and bring data back to DIS. Object 91(b) will query Oracle and bring data back to DIS. Object 91(c) will query Sybase and bring data back to DIS. Object 91(d) (shown as a dot in FIGURE 9) will query Redbrick and bring data back to DIS, and so on. The nth object 91(n) will query IBM's data warehouse and bring data back to DIS. In a subsequent linked processing step 92 data from the database queries in the first step is joined by joining object command file 92 and stored in a buffer related to this object. Object 92 will joint the data from the n locations searched in step 91. Thereafter, in a subsequent processing step performed by calculation object command file 93 on the joined data in the joined database result buffer of step 92, desired calculations performed in accordance with the parameters indicated by the request are done on the joined data. Thereafter, in accordance with the request parameters text is formatted to space delimited text by the format object command file 94. The results are stored in a buffer associated with format object command file 94. Thereafter, a make text command file 95 causes the formatted text to be created as a text file for the WWW server 131 to be stored in a file which is accessible to and can be retrieved and displayed by the control program agent 73, or directly displayed by the control program agent 73 in the form illustrated in FIGURE 4 at the Web browser 130. It will be noted we have illustrated this process as object capsules in a DIS internetworking environment. These object capsules are a specialized form of a command file, which can encompass additional commands called by an object.

Preferred Embodiment of graphics DIS capsule

FIGURE 10 illustrates by way of example a DIS capsule that creates a graphical report file. For simplicity, data in this FIGURE is also shown in a DIS environment 90. Retrieval object command file 101 illustrates a step of retrieval of data from one or more databases as specified in the parameters of the request, performing these retrieval steps as did retrieval object command files 91(a)..91(n). Thereafter, this data is plotted with the make plot object command file 102, with the results being stored in a buffer. The final step of creating a result- to-be-presented file, in this instance in the form of a bitmap ready for display to a Web browser 130 is created by the make bitmap (BMP) object command file 103. The example of a preferred bitmap object command which would be employed with todays Internet environment is a GIF image. Others can be used as well. Again the results are provided to the Web browser 130, by the action of the program command agent 73 on the Web Server 131, the results being illustrated by the pie-chart of FIGURE 6 in accordance with the parameters of the request for generating the graphical report illustrated by FIGURE 6.

Alternative Preferred Embodiments

Figure 11 illustrates an alternative configuration of the network system as it may be employed for permitting access to information available through homepages and in data warehouses where access to the homepage or database may or may not be restricted by a firewall. In Figure 11, the web browser(s) 130 accesses an associated Web Server 131, 131', 131"either by a coupling or addressing with a uniform resource locator (URL) the Web Server 131 which may be selected with a Hyperlink. This can be a direct coupling or an indirect coupling, as via a node locatable in a common access medium, such as provided by Internet resources accessible via a web browser, e.g. supporting Web Explorer, or Mosaic, NetScape, node 131 located somewhere on the Internet which utilizes our control program agent 73. Now node 131 which functions as a Web server is coupled via a token-ring network, SNA network, or other suitable network 132 (one of the any which may be used on the Internet as a transmission medium) with the facilities provided within what we will call our intranet, those facilities which are "proprietary" to the owner and which may be protected by firewalls at the intranet boundary 140. Now note that our control program 73 is resident within the Web Server 131 and functions as described in FIGURE 8 to couple to a DIS server 133 located within the intranet 140, which is preferably located behind a firewall as indicated in FIGURE 11. This DIS Server 133 is in turn coupled to our Database gateway 134. This database gateway is configured as illustrated also in FIGURE 1 for gathering information from databases coupled to it and located on servers for DB2, Oracle, Sybase, and Redbrick, as well as one for information warehouse functions. In our preferred embodiments these database units are IBM mainframe systems, as available commercially today, but they could be AS400s, RISC/6000, RISC/6000 SP or other systems supporting the databases.

The DIS Server is a server which supports DIS or similar decision support functions and the functions provided by our DIS capsules illustrated by FIGURE 9 and 10.

Now our Web browsers 130 can not only access information within the intranet, but can reach outside the intranet to gather information located elsewhere via the Internet. We will describe two examples of our preferred couplings to elements on the Internet. One example couples the database gateway 134 to another (a second) database gateway 134' via the Internet and it Internetwork routing (INR) protocol available from IBM as part of its current DIS product which can make use of UALs. The second database gateway 134' is coupled to its own (second) DIS server 133'. At this point the Web browser 130 can access data not only intranet, but also via the Internet to gather data from a database supported by DIS server 133' located outside the intranet. The Database server 134' would be able to gather information from any database coupled to it, as illustrated, assuming access is public or accessible after processing of a hidden variable access authorization.

However,the web browser(s) 130 can also access via Web Server 131 (with our control program 73 illustrated in detail in FIGURE 8) another Web server 131' which implements our control program 73. This Web server, for example, Web server 131' can also be coupled via its own (second) network 132' (which supports functions equivalent to network 132 and as illustrated in FIGURES 1 and 11) to an associated DIS Server 133' as illustrated to perform tasks like those we are describing from a request sent via the second network from its Web server 131'.

However, as another alternative example, Web server 131' with an appropriate API can access a directly coupled database available to the server, such as Microsoft's Access 131a. Thus small databases which have not yet invested in being able to gather information from an intranet resource, can use their own direct resources, and also be interrogated by the Web browser(s) 130, or another web browser 136. Remember that browser's 130 can also communicate with the Web server 131' across the Internet, just as can a Web browser 136 located on the intranet 140 inside of the firewall illustrated by the intranet 140 dashed line shown in FIGURE 11. With a browser 136 in place at the Web Server 131' location, that browser 136 can make requests, if authorized across the intranet to the Web Server 131 which can then utilize the DIS capsules provided by the DIS Server 133.

Physically, the network 132 will have its own access server 135 preferably in the form of a TCP/IP server 135 to make the physical connection across the Internet. We illustrate in FIGURE 11 this other logical layer as located on the network. This TCP/IP server supports the physical connections which are needed by the other logical higher levels of service supported on the network. The use of an InterNetwork Routing Protocol (INR) allows the logical coupling illustrated between a application processing server 134 to an external intranet application processing server 134'. On each network there can be one or more web servers. A Hypertext document request asking for a field to be seached, as by a Hyperlink, could index to a server directly, e.g. a second web server 134" on the same network which would have its own control program agent function duplicating the control program agent resident in web server 134. Thus at the request homepage a menu which say if "Art&Literature search", when selected in a Hyperlink setting, would index to a particular web server and a particular document within that web server's environment. This web server 134" besides being linked to its own application processing server 133" has a direct link, in the environment illustrated, to an MVS CICS, a transaction processing server for handling transaction processing. Such a solution allows CICS transaction processing to utilize the Internet to save transmission costs and still be located beneath a firewall for retention of data integrity. The outputs provided by the web server to the requested destination can be outside of the firewall, and in the form of results illustrated by the possible examples shown in FIGURES 3, 5 and 8.

While we have described our preferred embodiments of our invention, it will be understood that those skilled in the art, both now and in the future, may make various improvements and enhancements which fall within the scope of the claims which follow. These claims should be construed to maintain the proper protection for the invention first disclosed.

The following features - for themselves or in combination with other features - are also characteristics of the invention:

- The web browser system further comprises at least one intranet network supporting a plurality of web servers having a control program agent and having an access agent coupled to the Internet;
  at least one web server coupled to said web browser client for receiving and fulfilling a request as an agent of said client; and
  at least one intranet network supporting said web server and having an access agent coupled to the Internet;

- The web browser system further comprises a plurality of information access servers for acting as sub-agents for said web server during a process of fulfilling requests, said access servers having capsule objects which perform programmable functions which are executable upon a received command from said web server, at least one of said access servers being located on said intranet network; said web server including a control program agent for receiving a user request initiated at the web browser client for information and for transmitting said request to a sub-agent access server having capsule objects which execute upon command programmable functions requested by said web server.

## Claims

1. A web browser system, comprising:
   a web browser, means for associating said web browser with a homepage by a coupling or addressing with a uniform resource locator,
   a control program agent node located somewhere on the Internet supporting a control program agent coupled to and supporting said homepage by a coupling or addressing with a uniform resource locator,
   said control program agent node being coupled via a network with facilities provided within an intranet for private owner facilities and which may be protected by firewalls at the intranet boundary, said control program agent being coupled to a command file server and said command file server being coupled to a database gateway for gathering information from databases coupled to said database gateway and located on different database servers, said command file server supporting a plurality of command file objects which are programmed to perform web browser service support functions at the request of a user of said web browser to access information within the intranet and to gather information located elsewhere via the Internet as a sub-agent of said control program agent.

2. A web browser system according to claim 1,
   wherein by submission of a request at a web browser a user can not only access information within an intranet, but can reach outside the intranet to gather information located elsewhere via the Internet.

3. A web browser system according to claim 1,
   wherein there are on the networking including an intranet and the Internet a plurality of database gateways, and at the command of a command file running within a command file server one database gateway is coupled to another database gateway via the network by an inter-network routing protocol.

4. A web browser system according to claim 1,
   wherein there are on the networking including an intranet and the Internet a plurality of database gateways, and at the command of a command file running within a command file server one database gateway is coupled to another

database gateway via the network by an inter-network routing protocol invoking coupling of database gateways by UALs.

5. A web browser system according to claim 1, wherein
said second database gateway is coupled to its own command file server.

6. A web browser system according to claim 1,
wherein a web browser initiated request is distributed via an intranet to the Intranet whereby access of data is obtained not only intranet, but also via the Internet to gather data from a database supported by a command file server located outside the intranet.

7. A web browser system according to claim 1,
wherein a web browser initiated request is distributed via an intranet to the Internet whereby access of data is obtained not only internet, but also via the Internet to gather data from a database supported by a command file server located outside the internet and coupled to said command file server with public access or access obtained after processing of variable access authorization data provided thorugh said command file server.

8. A web browser system, comprising:
a web browser, means for associating said web browser with a homepage by a coupling or addressing with a uniform resource locator including a first control program agent node located somewhere on the Internet supporting a control program agent coupled to and supporting said homepage by a coupling or addressing with a uniform resource locator,
said second control program agent node being coupled via a network with facilities provided within an intranet for private owner facilities and which may be protected by firewalls at the intranet boundary,
a second control program agent node located somewhere on the Internet supporting a second control program agent by a coupling or addressing with a uniform resource locator,
said second control program agent node being coupled via network with facilities provided within an intranet for private owner facilities and which may be protected by firewalls at the intranet boundary,
said first control program agent being coupled to said second control program agent node located somewhere on the Internet supporting said second control program agent and coupled to and supporting a a command file server, said command file server being coupled to a database gateway for gathering information from databases coupled to said database gateway and located on different database servers, said command file server supporting a plurality of command file objects which are programmed to perform web browser service support functions at the request of a user of said web browser to access information within the intranet and to gather information located elsewhere via the Internet as a sub-agent of said control program agent.

9. A web browser system according to claim 8,
wherein said first control program agent resides on a first web server supporting said web browser and said second control program agent resides on a second web server which is coupled via its own network to an associated command file server to perform tasks requested by said web browser and communicated to said web browser after passing through multiple networks.

10. A web browser system, comprising:
a web browser, means for associating said web browser with a homepage including a first control program agent node supporting a control program agent coupled to and supporting said homepage and supporting an API to access a database available to said first control program agent node,
said control program agent and API enabling a user of said web browser to gather information from said database available to said first control program agent node and to gather information from an intranet resource and to provide access thereto in response to an interrogation initiated at a remote web browser.

11. A web browser system according to claim 10,
wherein said remote web browser is also coupled to a second control program agent node located on the Internet, said second control program agent node supporting a second control program agent supporting an API to access a database available to said first control program agent node via said second control program agent said second control program agent and API enabling a user of said web browser to gather information from a database available to said first control program agent node via said second control program agent node and to gather information from an intranet resource and to provide access thereto in response to an interrogation initiated at said web browser. across the Internet by a coupling or addressing with a uniform resource locator to said second control agent node and from resources available on an intranet coupled to said second control program agent node.

**12.** A web browser system according to claim 10,
wherein said second control program agent node is coupled via a network with facilities provided within an intranet for private owner facilities and which may be protected by firewalls at the intranet boundary,
said second control program agent node located somewhere on the Internet supporting said second control program agent by a coupling or addressing with a uniform resource locator,
said first control program agent being coupled to said second control program agent node located somewhere on the Internet supporting said second control program agent and coupled to and supporting a a command file server, said command file server being coupled to a database gateway for gathering information from databases coupled to said database gateway and located on different database servers, said command file server supporting a plurality of command file objects which are programmed to perform web browser service support functions at the request of a user of said web browswer to access information within the intranet and to gather information located elsewhere via the Internet as a sub-agent of said control program agent.

**13.** A web browser system according to claim 10,
wherein said web browser is at a web server location with said web server providing said control program agent node,
and browser requests, if authorized for access across said intranet, accesses a command file agent in a web server on said intranet providing said second command file agent node, which then utilize DIS capsules provided by a DIS Server functioning as a command file server.

**14.** A web browser system according to claim 10,
wherein the intranet network is provided with an access server to make physical connections across the Internet.

**15.** A web browser system according to claim 10,
wherein said DIS server invokes an InterNetwork Routing Protocol (INR) to create a logical coupling between an a application processing server to an external intranet application processing server.

**16.** A web browser system according to claim 10,
wherein a web browser request is a Hypertext document request asking for a field to be searched and indexing with a Hyperlink a web server providing a command file agent node on the same network which would have its own program control agent function duplicating the control program agent resident in said control program agent node.

**17.** A web browser system according to claim 10,
a web browser request homepage a menu topic, when selected in a Hyperlink setting, indexes index to a particular second control program agent node and a particular document within that second control program agent's environment.

**18.** A web browser system according to claim 10,
wherein said second control program agent node, besides being linked to its own application processing server has a direct link, to a a transaction processing server for handling transaction processing.

**19.** A web browser system according to claim 10,
wherein said transaction processing enables use of the Internet for authorized access and transmission of data, but said transaction processing server is located beneath a firewall for retention of data integrity.

**20.** A web browser system according to claim 10,
wherein said transaction processing server's output is provided to said second control program agent node server and from said second control program agent node server to a requested destination on the internet specified by the request initiated by said web browser.

**21.** A web server system for supporting a web browser, comprising
means for receiving from a world wide web browser a request to be fulfilled as an agent of the browser client,
a control program agent for organizing organizing distributed sub-agents as distributed integration solution servers on an intranet network supporting the web server which also has an access agent servers accessible over the Internet.

**22.** A web server system according to claim 21, further comprising, a plurality of distributed integration solution servers for executing selected capsule objects which perform programmable functions upon a received command from said web server control program agent.

23. A web server system according to claim 22, further comprising, a database gateway coupled to a plurality of database resources for supplying upon a singlr request made from a Hypertext document, requested information from multiple data bases located at different types of databases geographically dispersed.

24. A web server system according to claim 23, further comprising, command objects for performing calculations, formatting, and other services prior to reporting to the web browser or to other locations, in a selected format a requested result report selected from a set of result reports, including a display report, facsimile report, a printer report, a report to customer installations, and a report to TV video subscribers, with account tracking.

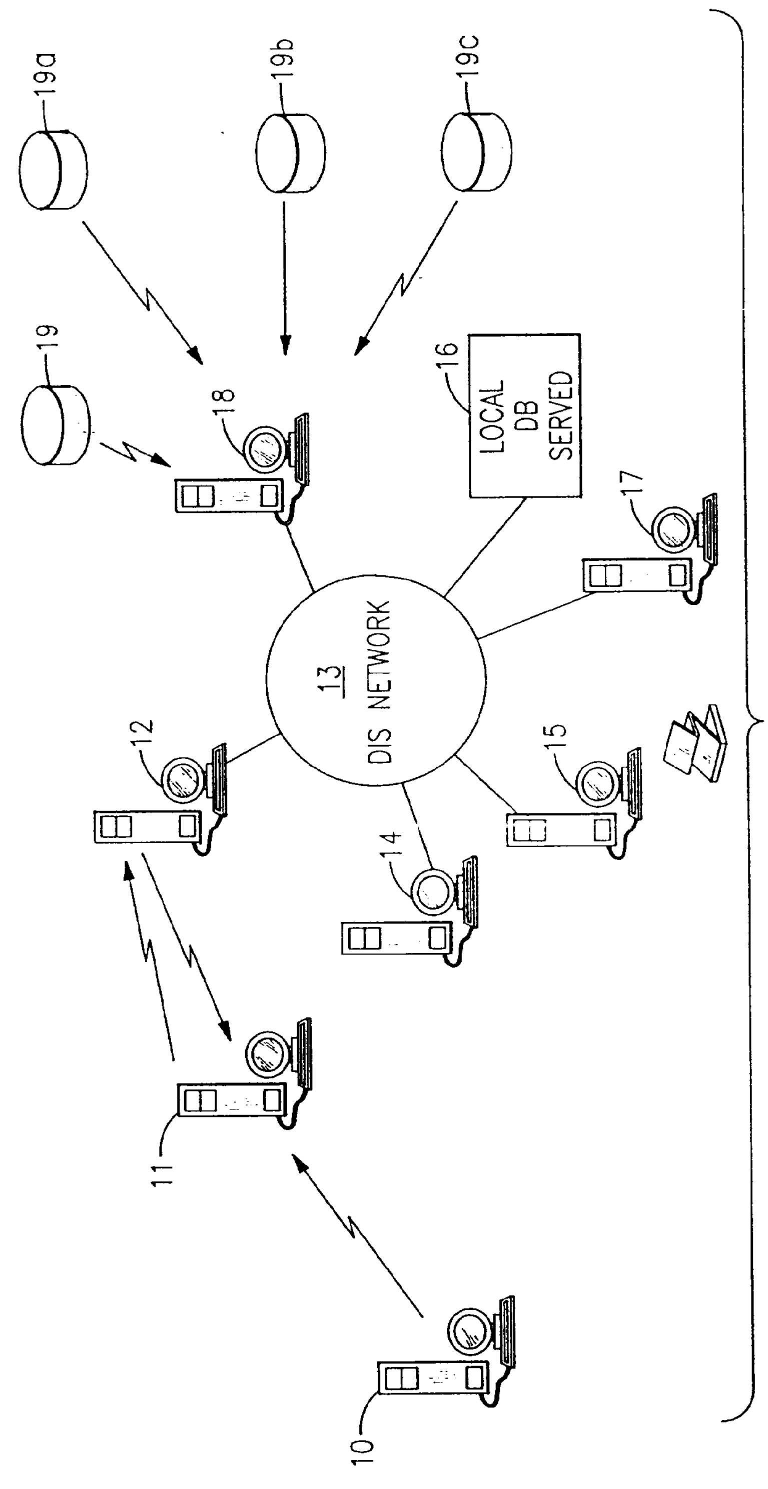
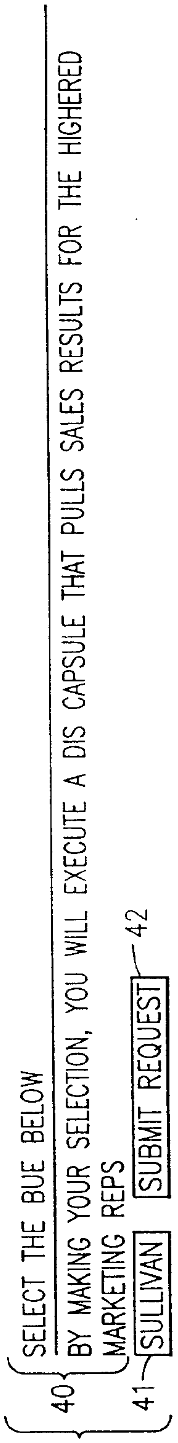
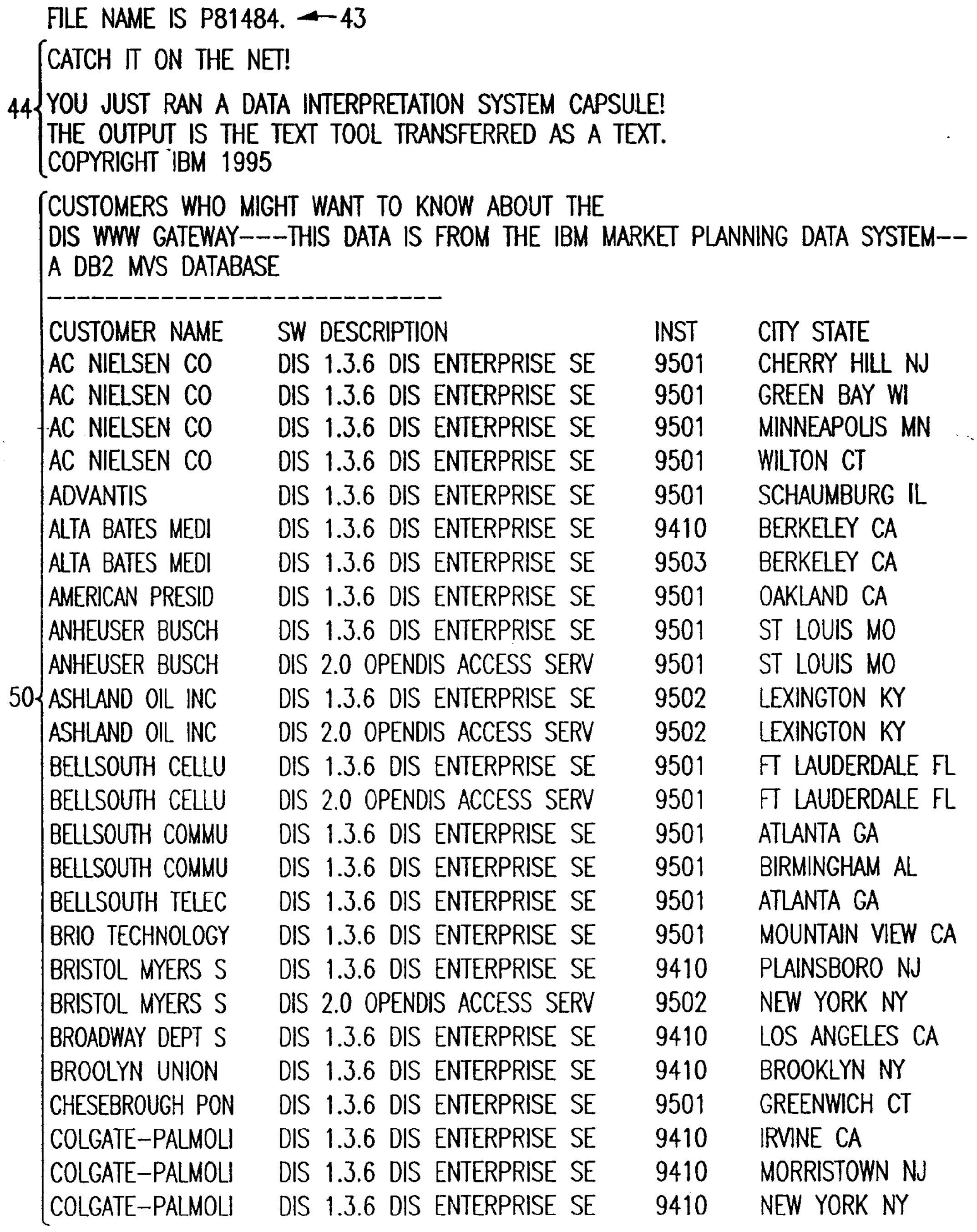


FIG.1


SELECT THE BUE BELOW

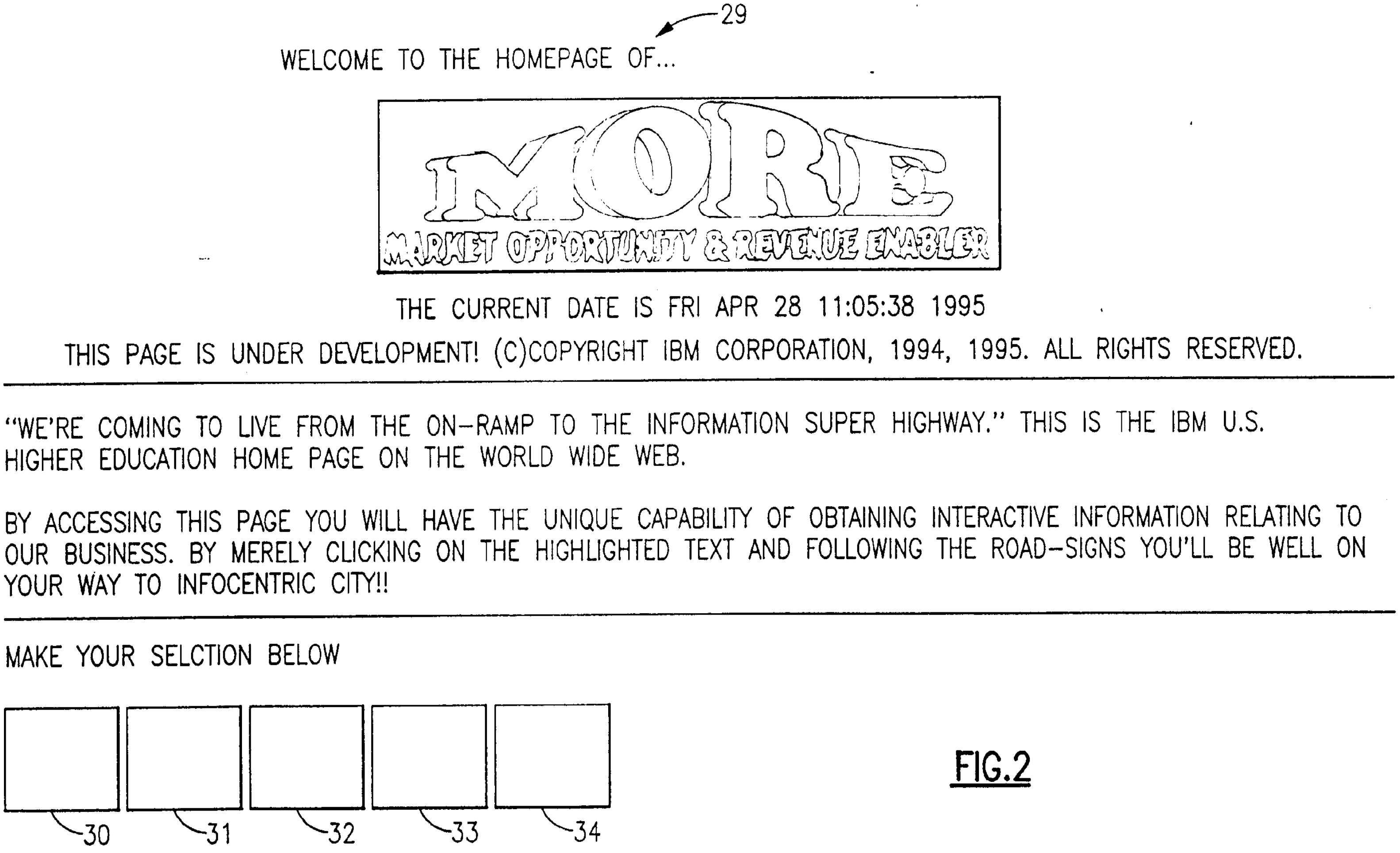
BY MAKING YOUR SELECTION, YOU WILL EXECUTE A DIS CAPSULE THAT PULLS SALES RESULTS FOR THE HIGHERED MARKETING REPS

FIG.3



```
FILE NAME IS P81484.
CATCH IT ON THE NET!
YOU JUST RAN A DATA INTERPRETATION SYSTEM CAPSULE!
THE OUTPUT IS THE TEXT TOOL TRANSFERRED AS A TEXT.
COPYRIGHT IBM 1995
CUSTOMERS WHO MIGHT WANT TO KNOW ABOUT THE
DIS WWW GATEWAY---THIS DATA IS FROM THE IBM MARKET PLANNING DATA SYSTEM--
A DB2 MVS DATABASE
--------------------------
CUSTOMER NAME    SW DESCRIPTION               INST    CITY STATE
AC NIELSEN CO    DIS 1.3.6 DIS ENTERPRISE SE  9501    CHERRY HILL NJ
AC NIELSEN CO    DIS 1.3.6 DIS ENTERPRISE SE  9501    GREEN BAY WI
AC NIELSEN CO    DIS 1.3.6 DIS ENTERPRISE SE  9501    MINNEAPOLIS MN
AC NIELSEN CO    DIS 1.3.6 DIS ENTERPRISE SE  9501    WILTON CT
ADVANTIS         DIS 1.3.6 DIS ENTERPRISE SE  9501    SCHAUMBURG IL
ALTA BATES MEDI  DIS 1.3.6 DIS ENTERPRISE SE  9410    BERKELEY CA
ALTA BATES MEDI  DIS 1.3.6 DIS ENTERPRISE SE  9503    BERKELEY CA
AMERICAN PRESID  DIS 1.3.6 DIS ENTERPRISE SE  9501    OAKLAND CA
ANHEUSER BUSCH   DIS 1.3.6 DIS ENTERPRISE SE  9501    ST LOUIS MO
ANHEUSER BUSCH   DIS 2.0 OPENDIS ACCESS SERV  9501    ST LOUIS MO
ASHLAND OIL INC  DIS 1.3.6 DIS ENTERPRISE SE  9502    LEXINGTON KY
ASHLAND OIL INC  DIS 2.0 OPENDIS ACCESS SERV  9502    LEXINGTON KY
BELLSOUTH CELLU  DIS 1.3.6 DIS ENTERPRISE SE  9501    FT LAUDERDALE FL
BELLSOUTH CELLU  DIS 2.0 OPENDIS ACCESS SERV  9501    FT LAUDERDALE FL
BELLSOUTH COMMU  DIS 1.3.6 DIS ENTERPRISE SE  9501    ATLANTA GA
BELLSOUTH COMMU  DIS 1.3.6 DIS ENTERPRISE SE  9501    BIRMINGHAM AL
BELLSOUTH TELEC  DIS 1.3.6 DIS ENTERPRISE SE  9501    ATLANTA GA
BRIO TECHNOLOGY  DIS 1.3.6 DIS ENTERPRISE SE  9501    MOUNTAIN VIEW CA
BRISTOL MYERS S  DIS 1.3.6 DIS ENTERPRISE SE  9410    PLAINSBORO NJ
BRISTOL MYERS S  DIS 2.0 OPENDIS ACCESS SERV  9502    NEW YORK NY
BROADWAY DEPT S  DIS 1.3.6 DIS ENTERPRISE SE  9410    LOS ANGELES CA
BROOLYN UNION    DIS 1.3.6 DIS ENTERPRISE SE  9410    BROOKLYN NY
CHESEBROUGH PON  DIS 1.3.6 DIS ENTERPRISE SE  9501    GREENWICH CT
COLGATE-PALMOLI  DIS 1.3.6 DIS ENTERPRISE SE  9410    IRVINE CA
COLGATE-PALMOLI  DIS 1.3.6 DIS ENTERPRISE SE  9410    MORRISTOWN NJ
COLGATE-PALMOLI  DIS 1.3.6 DIS ENTERPRISE SE  9410    NEW YORK NY
```

FIG.4



WELCOME TO THE HOMEPAGE OF...

THE CURRENT DATE IS FRI APR 28 11:05:38 1995

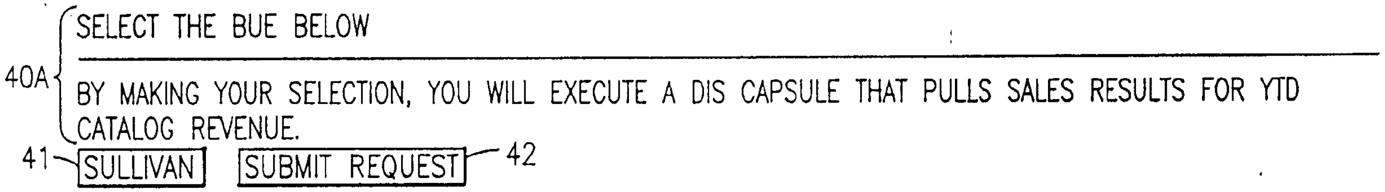
THIS PAGE IS UNDER DEVELOPMENT! (C)COPYRIGHT IBM CORPORATION, 1994, 1995. ALL RIGHTS RESERVED.

"WE'RE COMING TO LIVE FROM THE ON-RAMP TO THE INFORMATION SUPER HIGHWAY." THIS IS THE IBM U.S. HIGHER EDUCATION HOME PAGE ON THE WORLD WIDE WEB.

BY ACCESSING THIS PAGE YOU WILL HAVE THE UNIQUE CAPABILITY OF OBTAINING INTERACTIVE INFORMATION RELATING TO OUR BUSINESS. BY MERELY CLICKING ON THE HIGHLIGHTED TEXT AND FOLLOWING THE ROAD-SIGNS YOU'LL BE WELL ON YOUR WAY TO INFOCENTRIC CITY!!

MAKE YOUR SELCTION BELOW

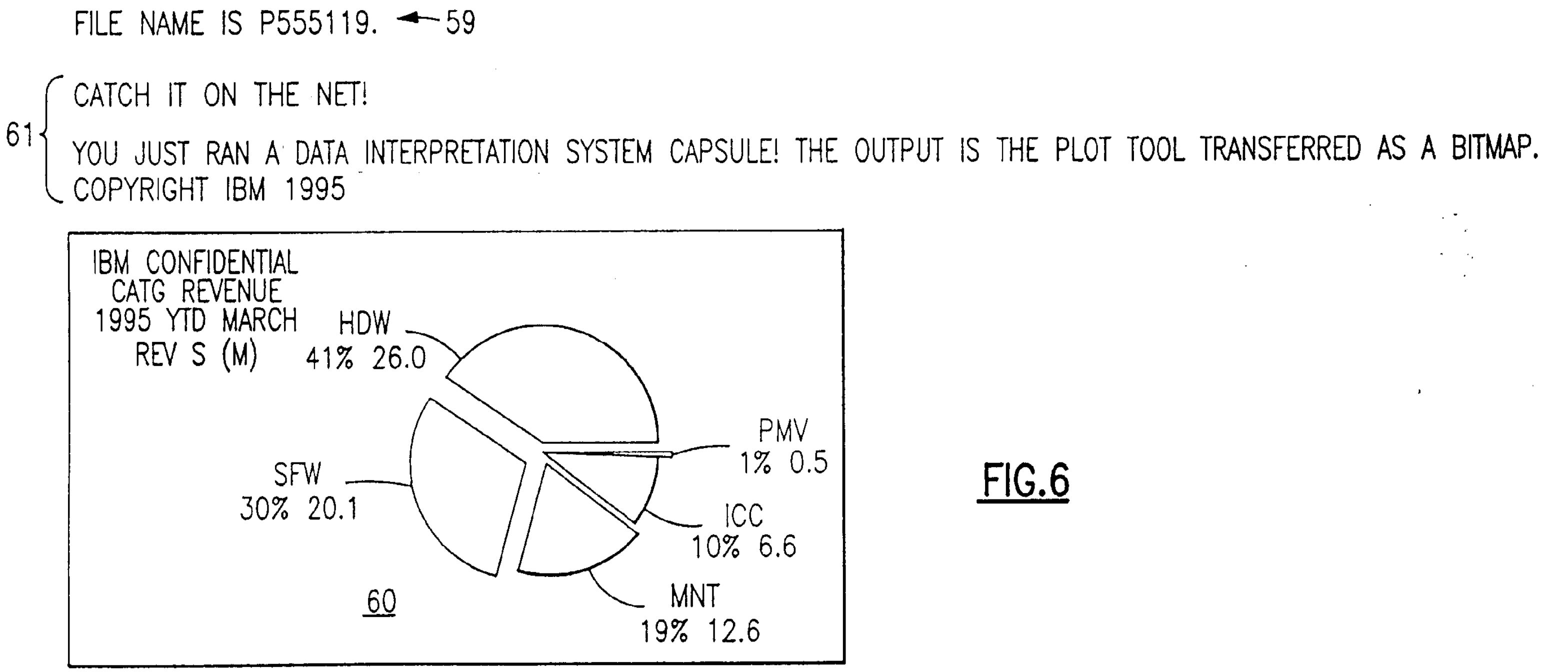
FIG.2



SELECT THE BUE BELOW

BY MAKING YOUR SELECTION, YOU WILL EXECUTE A DIS CAPSULE THAT PULLS SALES RESULTS FOR YTD CATALOG REVENUE.

FIG.5



FILE NAME IS P555119.

CATCH IT ON THE NET!

YOU JUST RAN A DATA INTERPRETATION SYSTEM CAPSULE! THE OUTPUT IS THE PLOT TOOL TRANSFERRED AS A BITMAP.
COPYRIGHT IBM 1995

FIG.6

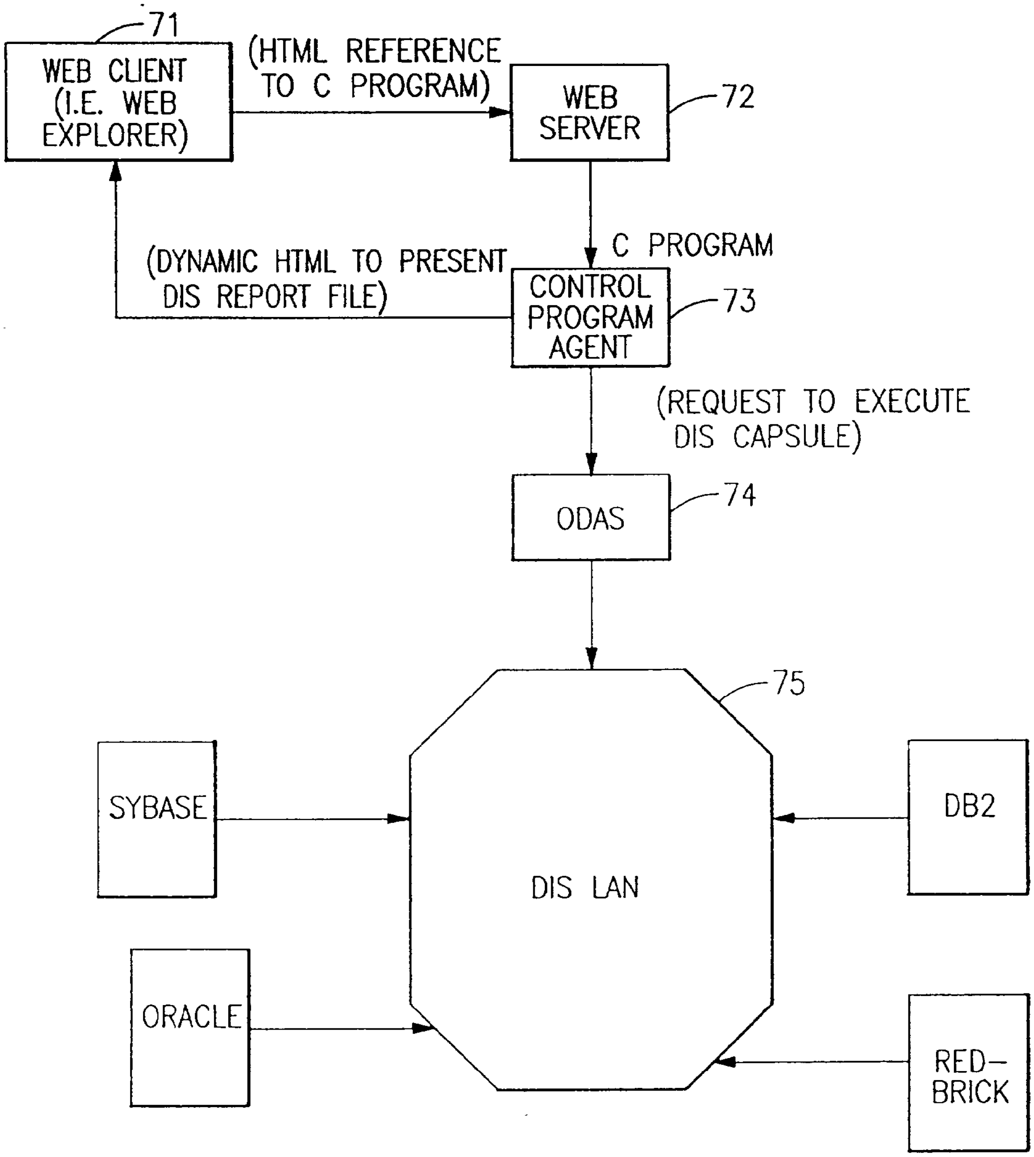

71
WEB CLIENT (I.E. WEB EXPLORER)
(HTML REFERENCE TO C PROGRAM)
WEB SERVER
72
C PROGRAM
(DYNAMIC HTML TO PRESENT DIS REPORT FILE)
CONTROL PROGRAM AGENT
73
(REQUEST TO EXECUTE DIS CAPSULE)
ODAS
74
75
SYBASE
DIS LAN
DB2
ORACLE
RED-BRICK

FIG.7

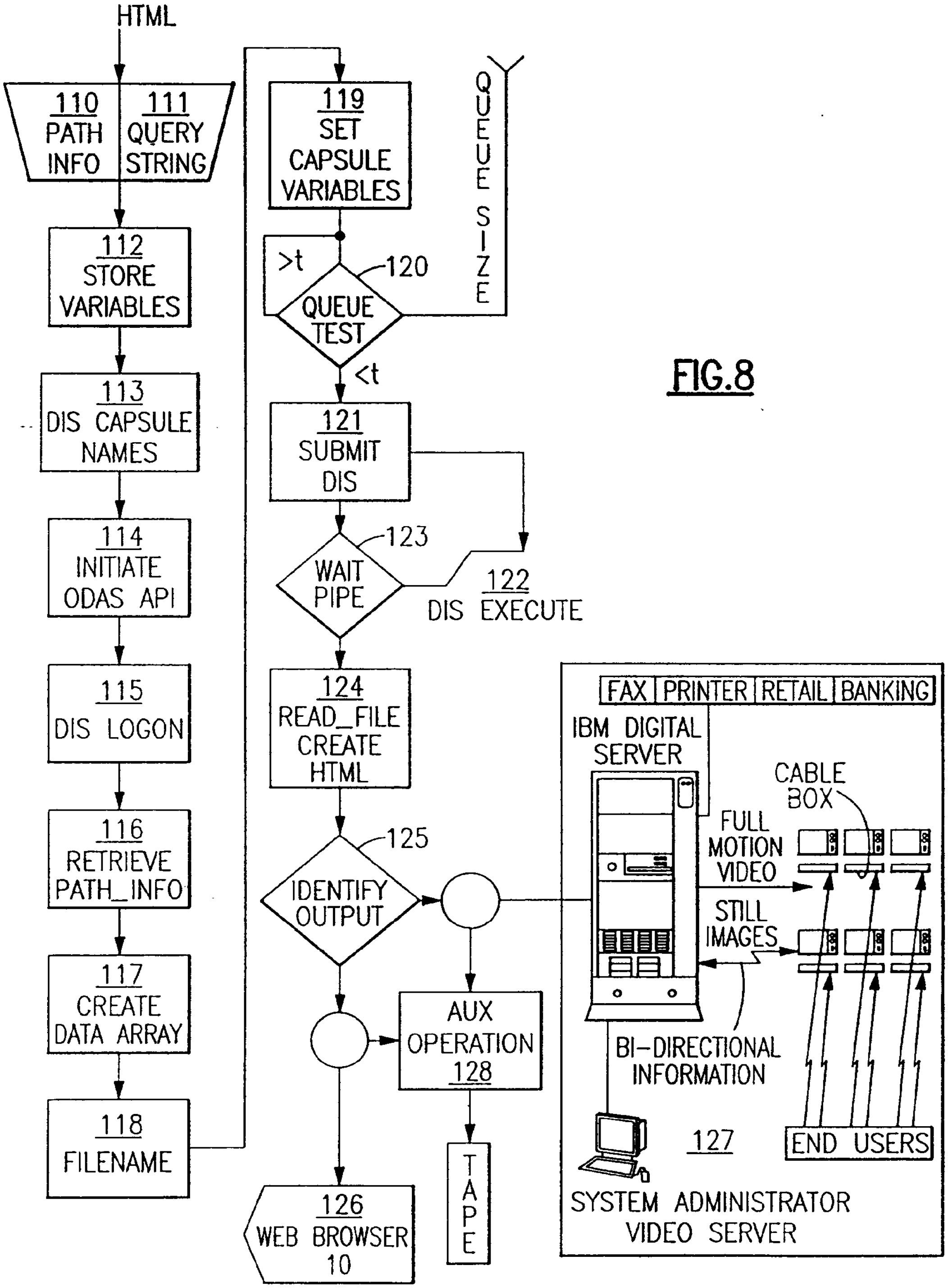

HTML
110 PATH INFO
111 QUERY STRING
112 STORE VARIABLES
113 DIS CAPSULE NAMES
114 INITIATE ODAS API
115 DIS LOGON
116 RETRIEVE PATH_INFO
117 CREATE DATA ARRAY
118 FILENAME
119 SET CAPSULE VARIABLES
QUEUE SIZE
>t
120
QUEUE TEST
<t
121 SUBMIT DIS
123
WAIT PIPE
122 DIS EXECUTE
124 READ_FILE CREATE HTML
125
IDENTIFY OUTPUT
AUX OPERATION 128
TAPE
126 WEB BROWSER 10
FAX
PRINTER
RETAIL
BANKING
IBM DIGITAL SERVER
CABLE BOX
FULL MOTION VIDEO
STILL IMAGES
BI-DIRECTIONAL INFORMATION
127
END USERS
SYSTEM ADMINISTRATOR VIDEO SERVER

FIG.8

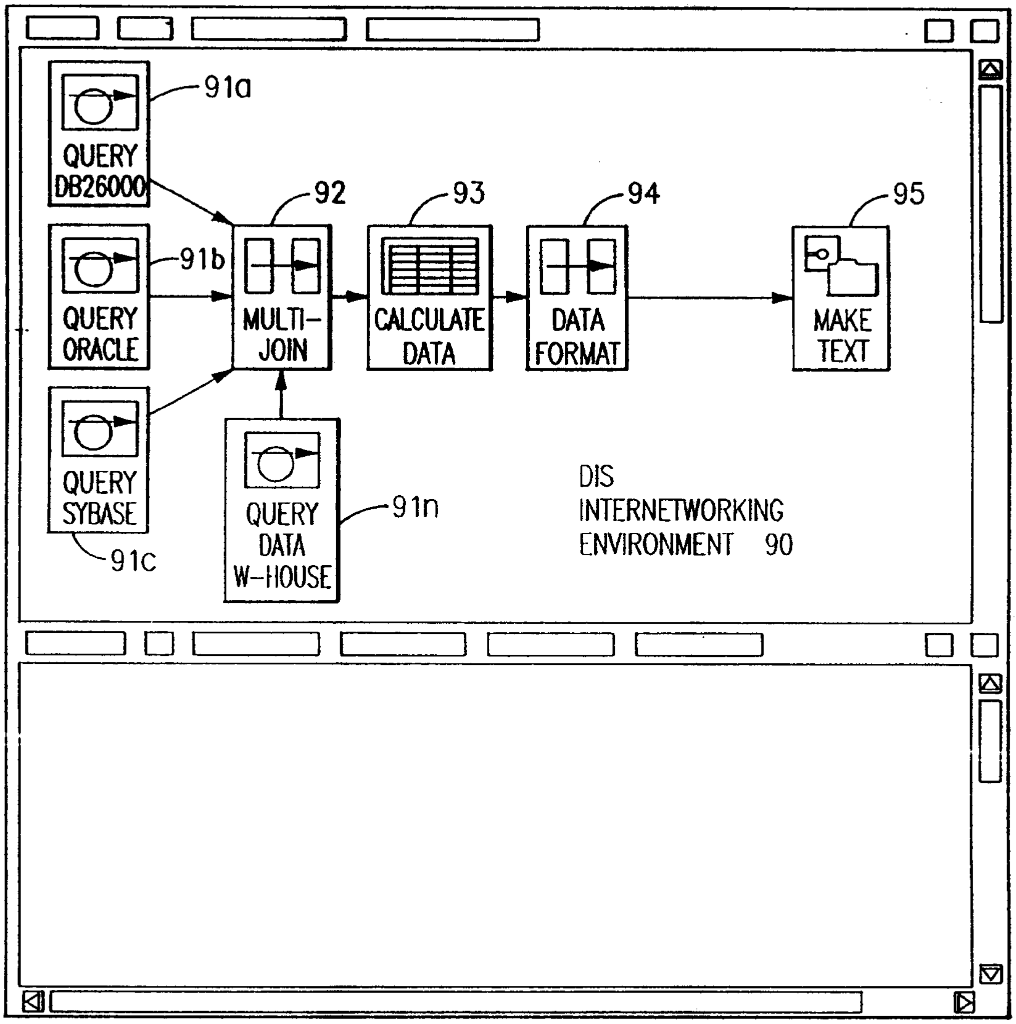

91a
QUERY DB26000
92
93
94
95
91b
QUERY ORACLE
MULTI-JOIN
CALCULATE DATA
DATA FORMAT
MAKE TEXT
QUERY SYBASE
91c
QUERY DATA W-HOUSE
91n
DIS INTERNETWORKING ENVIRONMENT 90

FIG.9

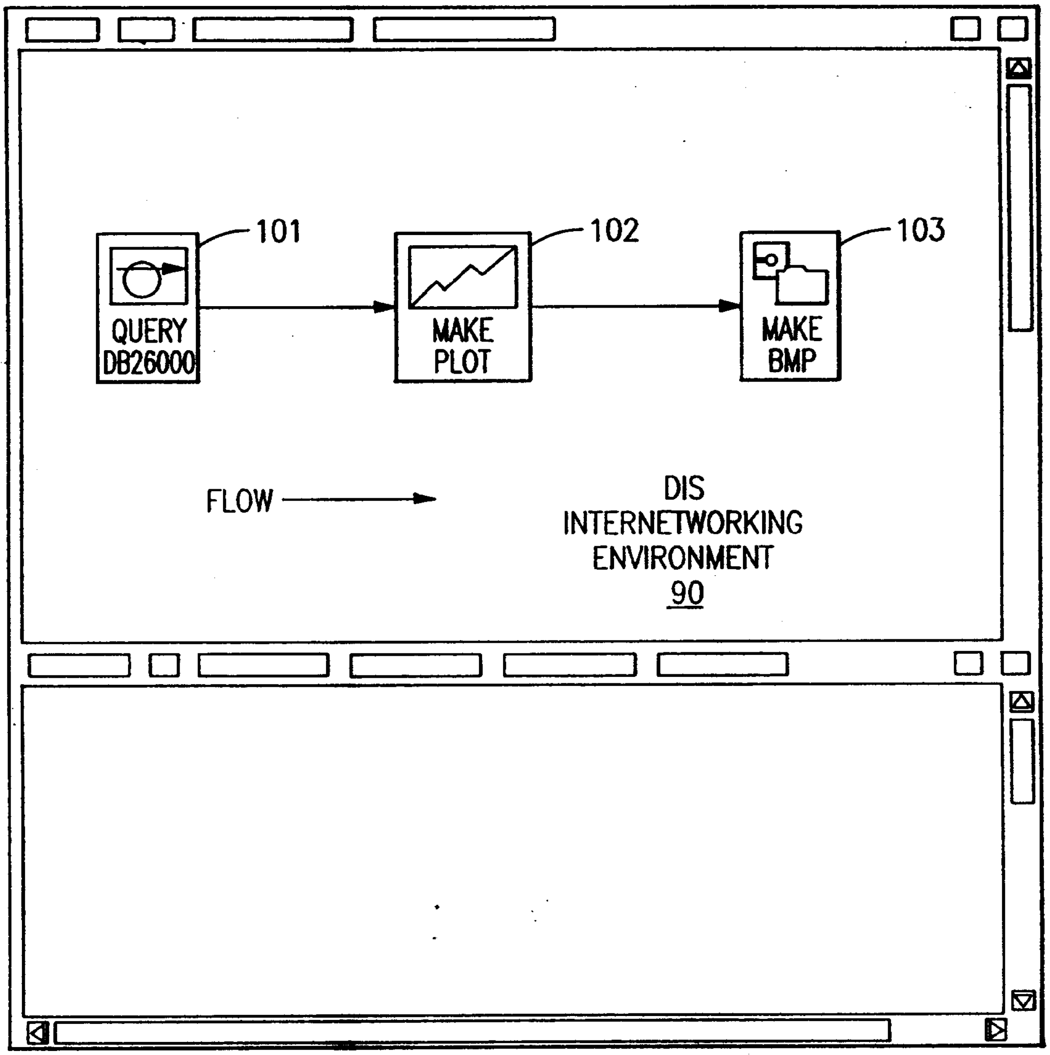

101
QUERY
DB26000
102
MAKE
PLOT
103
MAKE
BMP
FLOW
DIS
INTERNETWORKING
ENVIRONMENT
90

FIG.10

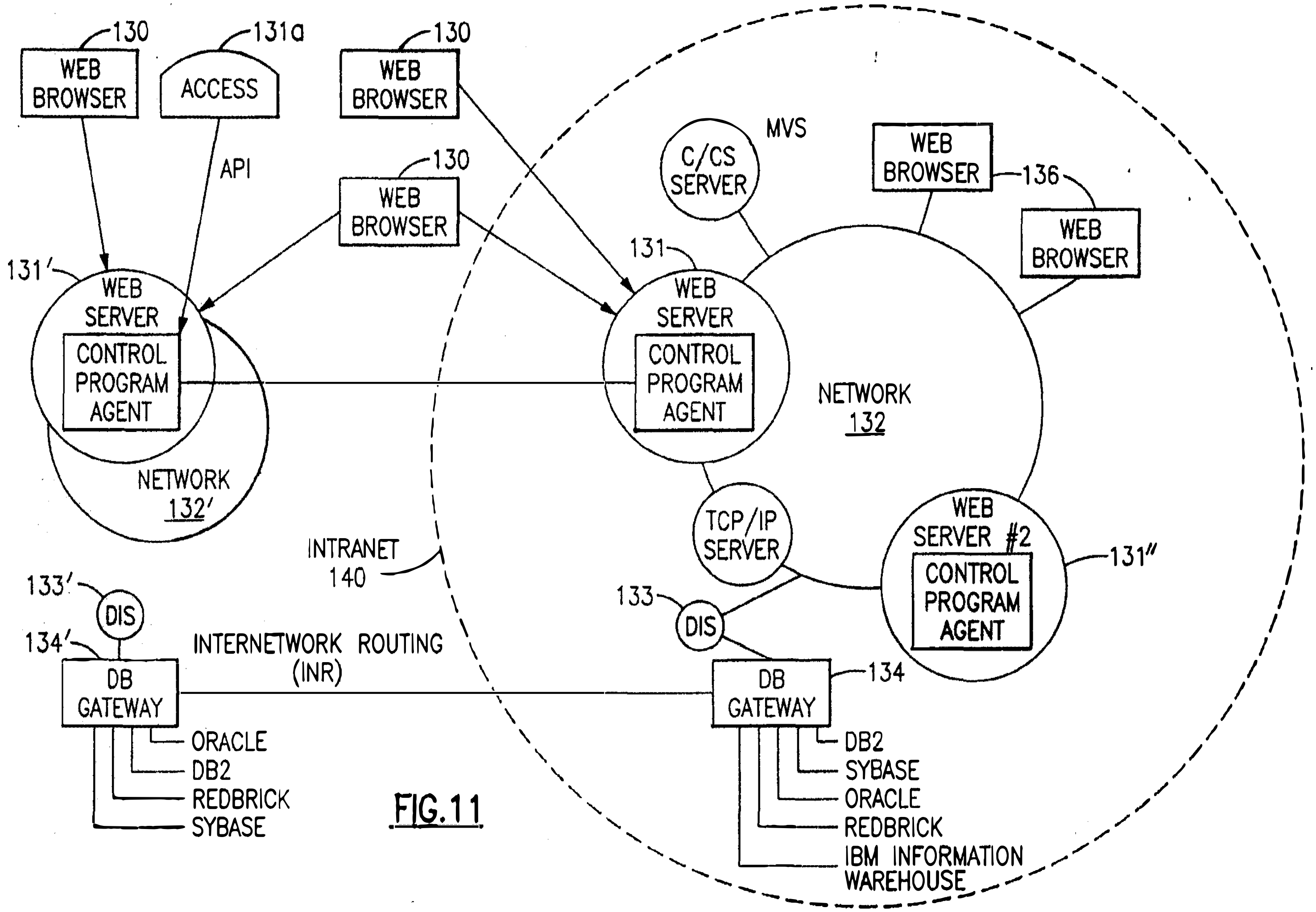

130
WEB BROWSER
131a
ACCESS
API
130
WEB BROWSER
130
WEB BROWSER
131′
WEB SERVER
CONTROL PROGRAM AGENT
NETWORK 132′
MVS
C/CS SERVER
WEB BROWSER
136
WEB BROWSER
131
WEB SERVER
CONTROL PROGRAM AGENT
NETWORK 132
TCP/IP SERVER
WEB SERVER #2
CONTROL PROGRAM AGENT
131″
INTRANET 140
133′
DIS
134′
DB GATEWAY
ORACLE
DB2
REDBRICK
SYBASE
INTERNETWORK ROUTING (INR)
133
DIS
DB GATEWAY
134
DB2
SYBASE
ORACLE
REDBRICK
IBM INFORMATION WAREHOUSE

FIG.11

European Patent Office

# EUROPEAN SEARCH REPORT

Application Number
EP 96 10 8976

## DOCUMENTS CONSIDERED TO BE RELEVANT

| Category | Citation of document with indication, where appropriate, of relevant passages | Relevant to claim | CLASSIFICATION OF THE APPLICATION (Int.Cl.6) |
|---|---|---|---|
| A | SCALING NEW HEIGHTS IN TECHNICAL COMMUNICATION, BANFF, SEPT. 28 - OCT. 1, 1994,<br>28 September 1994, INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, pages 192-197, XP000510688<br>LAU T: "BUILDING A HYPERMEDIA INFORMATION SYSTEM ON THE INTERNET"<br>* abstract *<br>* page 192, line 1 - page 193, line 3 *<br>--- | 1-24 | G06F17/30 |
| D,A | US-A-4 274 139 (HODGKINSON SUSAN D ET AL) 16 June 1981<br>* abstract *<br>* column 1, line 1 - column 3, line 18 *<br>----- | 1 | |
| | | | TECHNICAL FIELDS SEARCHED (Int.Cl.6) |
| | | | G06F |

The present search report has been drawn up for all claims

| Place of search | Date of completion of the search | Examiner |
|---|---|---|
| THE HAGUE | 11 September 1996 | Katerbau, R |

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 782 085 A1

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
**02.07.1997 Bulletin 1997/27**

(51) Int. Cl.$^6$: **G06F 17/30**

(21) Application number: **96308817.4**

(22) Date of filing: **05.12.1996**

(84) Designated Contracting States:
**DE FR GB**

(30) Priority: **28.12.1995 US 581300**

(71) Applicant: **International Business Machines Corporation**
**Armonk, N.Y. 10504 (US)**

(72) Inventor: **Steele, David Aaron**
**Cupertino, California 95014 (US)**

(74) Representative: **Ling, Christopher John**
**IBM United Kingdom Limited,**
**intellectual Property Department,**
**Hursley Park**
**Winchester, Hampshire SO21 2JN (GB)**

# (54) Video browsing on the world wide web

(57) A system and method are provided for supporting video browsing over a communication network such as the Internet/World Wide Web. A graphical user interface is provided through a client software tool such as a Web browser. A client/user selects a video data object stored at a remote server. A set of points (52) within the object are displayed at the client's graphical user interface display, as representations, preferably thumbnail images, of the points within the object. The user selects an interval (56) defined by the representations, preferably the interval between two temporally adjacent representations. Responsive to this selection, a new set of points, falling within the selected interval, are chosen, and representations of those points are generated and displayed. By doing so repeatedly, the user can easily browse through the video data object, and quickly and easily zero in on a desired portion of the video data object.

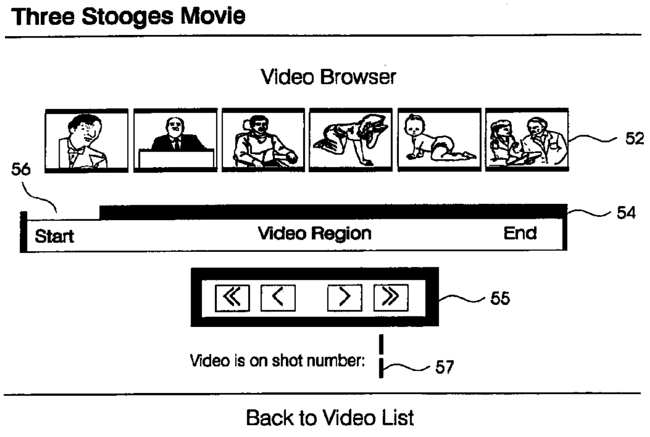


*FIG. 7*

Printed by Rank Xerox (UK) Business Services
2.14.9/3.4

## Description

### Field of the Invention

The invention generally relates to the fields of computer systems and multimedia communications, and more particularly relates to the fields of video storage and compression, and of interactive video playback and browsing.

### Glossary of Terms Used

While dictionary meanings are also implied by certain terms used here, the following glossary of some terms may be useful.

Internet ("the Net"): The connection system that links computers worldwide in a network.

TCP/IP: Transmission Control Protocol/Internet Protocol. A packet switching scheme the Internet uses to chop, route, and reconstruct the data it handles, from e-mail to video.

World Wide Web (WWW, "the Web") : The Internet's application that lets people seeking information on the Internet switch from server to server and database to database by clicking on highlighted words or phrases of interest. An Internet Web server supports clients and provides information.

Home page: A multi-media table of contents that guides a Web user to stored information on the Internet.

Server: A machine (computer) which performs a task at the command of another machine ("client"). In the context of the present invention, a server's primary function is to facilitate distribution of stored information over the Web.

Client: A machine which provides commands to a server, and is serviced by the server. Typically, a client machine is operated by an end user, and functions responsive to user commands.

Web Browser: A program running on a user-operated client computer. When a user "surfs" the Web using a browser, the browser acts as an Internet tour guide, allowing the client machine to display pictorial desktops, directories and search tools supported by the server.

URL: Universal Resource Locater, a Web document version of an e-mail address, in character string form, which uniquely identifies a document, application, or tool available over the Web.

Hyperlink: A network addressing tool embedded in a user-understandable displayed and/or highlighted item, such as a word, phrase, icon or picture. A URL can be accessed by means of its corresponding Hvperlink. When a user on a client machine selects the highlighted hvperlink through the user interface, the underlying item is then retrieved to the client supporting a Web browser.

HTTP Hypertext transfer protocol: Hypertext transfer protocol. The character string "http:" at the beginning of a URL indicates that the document or file designated by the URL contains hyperlinks defined according to the HTTP.

HyperText Markup Language (HTML): HTML is the language used by Web servers to create and connect documents that are viewed by Web clients. HTML uses Hypertext documents. Other uses of Hypertext documents are described in the following U.S. Patents:

Bernstein et al., 5,204,947, issued April 20, 1993;
Bernstein et al., 5,297,249, issued March 22, 1994; and
Lewis, 5,355,472, issued October 11, 1994;

all of which are assigned to International Business Machines Corporation, and which are referenced herein.

### Background Art

In recent years, the technologies of video data compression, storage, and interactive accessing have converged with network communications technologies, to present exciting prospects for users who seek access to remotely stored multimedia information.

In the area of network communications technologies, particularly exciting has been the recent prominence of the Internet and its progeny, the World Wide Web. The Internet and the Web have captured the public imagination as the so-called "information superhighway." Accessing information through the Web has become known by the metaphorical term "surfing the Web."

The Internet is not a single network, nor does it have any single owner or controller. Rather, the Internet is an unruly network of networks, a confederation of many different networks, public and private, big and small, whose human operators have agreed to connect to one another. The composite network represented by these networks relies on no single transmission medium. Bi-directional communication can occur via satellite links, fiber-optic trunk lines, phone lines, cable TV wires and local radio links.

To this point the World Wide Web (Web) provided by the Internet has been used in industry predominately as a means of communication, advertisement, and placement of orders. The World Wide Web facilitates user access to information resources by letting people jump from one server to another simply by selecting a highlighted word, picture or icon (a program object representation) about which they want more information, a manoeuvre known as a "hyperlink". In order to explore the WWW today, the user loads a special navigation program, called a "Web browser" onto his computer.

There are a number of browsers presently in existence and in use. Common examples are NetScape, Mosaic and IBM's Web Explorer. Browsers allow a user of a client to access servers located throughout the world for information which is stored therein. The information is then provided to the client by the server by sending files or data packets to the requesting client from the server's storage resources.

Part of the functionality of a browser is to provide image or video data. Web still image or video information can be provided, through a suitably designed Web page or interface, to a user on a client machine. Still images can also be used as Hypertext-type links, selectable by the user, for invoking other functions. For instance, a user may run a video clip by selecting a still image.

However, video data objects are very large, or, to put it more precisely, the quantity of data per unit time in a real-time viewing of a video data object is large. As a consequence, access by a user to a desired video data object is subject to data throughput constraints. The present state of the art makes it impracticable to provide more than a few tens of seconds of real-time video over the Internet with a response time that will be satisfactory to a user.

Therefore, multimedia and communication systems for providing users with access to video data objects, for browsing, searching, etc., must grapple with the problem of providing video data in a manner which best utilizes the available throughput to provide video data in a form which is most useful to the user.

With this design objective in mind, let us now consider the state of the art in the technologies of video data compression, storage, and interactive accessing. Recent work has been done to make video material more available and usable over the Web. For instance, an article in the August 1995 issue of ADVANCED IMAGING, by Amy T. Incremona, titled "Automatically Transcribing and Condensing Video: New Technology is Born", describes a method for providing video having an accompanying textual index, such as audio narration or closed caption text. Still images are presented, along with a transcription of audio text that accompanies the images (illustration on page 60). This information is provided in HTML format. Thus, a user can take advantage of the temporal correspondence between video shots and narration or closed caption text. To find a desired point in the video corresponding with a known point in the text, the user performs a key word search for the known point in the text. The result of this key word search is that the desired point in the video is reached.

Additionally in Shahraray et al., "Automatic Generation of Pictorial Transcripts of Video Programs", SPIE Vol. 2417, pp. 512-518, there is described an automatic authoring system for the generation of pictorial transcripts of video programs which are accompanied by closed caption information. The system employs a table having a series of rows, each row containing a pointer to a location of an image, and another pointer to the beginning of a text segment related to the image. A viewing window for a GUI display is shown in FIG. 4 of Shahraray et al., and reproduced herein in simplified form as FIG. 1 of the present patent application. FIG. 1 shows a video image 2, a closed caption text subtitle area 4, and a basic user control area 6. The basic user controls include a "Seek" slider 8.

Accordingly, the state of the art allows for user access to video information based on associated text. However, a more general method for accessing video, not provided by the prior art, would sever the tie between video images and accompanying audio narration or closed caption text.

## Disclosure of the Invention

It is therefore an objective of the present invention to provide a system and method for allowing convenient user access to a stored video object, for viewing and browsing, through a communication medium having a client-server architecture, such as the World Wide Web.

It is a further objective of the present invention to provide such convenient user access to a stored video object without requiring that access to the video object to be keyed with any text or other accompanying indicia, not actually part of the video itself.

To achieve these and other objects, there is provided, in accordance with the invention, a method for displaying, on a user terminal, video data object information pertaining to a stored video data object, the stored video object including a temporal beginning point, a temporal end point, and temporal intermediate points therebetween.

The method comprises the following steps:

First, a sequence of representations of points within the video data object, preferably still images, is provided to the user through a user interface. Each representation or still image that is provided corresponds with one of the temporal points in the video data object. In a typical environment, in which the user is an Internet/World Wide Web user, and the video data object is stored in a remote repository accessible through a server, the step of providing the representations includes performing suitable operations to identify points within the video data object, for which representations are to be provided. A preferred technique is to detect scene cuts.

Then, a user interface is provided, including means for allowing a user to select an interval between first and second ones of the temporal points. The user interface is preferably a graphical user interface (GUI), as is commonly made available through computer operating systems such as IBM's OS/2 and Microsoft's Windows operating systems. Also, suitable user interface equipment, such as a video screen and a mouse, are preferably used.

Finally, responsive to user selection of such an interval, a subsequence of representations is provided, each representation of the subsequence corresponding with a respective temporal point in the video data object, each of the respective temporal points falling between the first and second temporal points selected as discussed above.

Therefore, by providing this capability, the invention allows a user to iteratively home in on the portion of a lengthy, large video object that he/she is interested in. The invention advantageously eliminates the need to use text accompanying the video object as a crutch, as was done in the prior art discussed above.

While the invention is primarily disclosed as a method, it will be understood by a person of ordinary skill in the art that an apparatus, such as a conventional data processor, including a CPU, memory, I/O, program storage, a connecting bus, and other appropriate components, could be programmed or otherwise designed to facilitate the practice of the method of the invention. Such a processor would include appropriate program means for executing the method of the invention. Also, an article of manufacture, such as a pre-recorded disk or other similar computer program product, for use with a data processing system, could include a storage medium and program means recorded thereon for directing the data processing system to facilitate the practice of the method of the invention.

## Brief Description of the Drawings

The invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

FIG. 1 is an illustration of a prior art graphical user interface for video browsing;

FIG. 2 is a system block diagram of a typical distributed computer system which has internal and external networks, including the Internet, to connect clients to World Wide Web servers and other servers, the client systems being capable of incorporating the invention;

FIG. 3 is a high-level flowchart showing the method of the invention;

FIG. 4 is a flowchart showing a more detailed implementation of step 22 of the flowchart of FIG. 3;

FIG. 5 is a flowchart showing a still more detailed implementation of step 28 of the flowchart of FIG. 4;

FIG. 6 is an illustration of a graphical user interface according to the invention;

FIG. 7 is an illustration of a graphical user interface according to the invention; and

FIG. 8 is a flowchart showing a more detailed implementation of step 44 of the flowchart of FIG. 3.

## Description of the Preferred Embodiment

FIG. 2 illustrates a typical communication and processing system, including nodes (computers, processors, etc.) and communication media therebetween, making up an information delivery solution of a typical combination of resources including clients, servers, information repositories, and communication links or networks. Various nodes, shown typically as 9, are also coupled to one another by various networks, shown generally as 7, which may include local area networks (LANs), wide area networks (WANs), etc.

The discussion of FIG. 2 which follows will focus on certain individual components for clearer illustration and description of the novel and non-obvious features of the present invention. In the system of FIG. 2, a client machine 10 includes a computer or other device (as discussed above), running a Web browser program. The program run by the client 10, which incorporates the invention, may be provided to a general purpose computer by means of a commercial computer program product, such as a pre-recorded floppy disk 11 or other suitable computer-readable recording medium. In such a computer program product, the novel and non-obvious features of the invention, implemented as program code, are pre-programmed onto the disk 11 for convenient sales and marketing, and for directing operation of the client machine 10 in accordance with the invention.

The client 10 is coupled, through a communication network 12 such as the Internet, to a Web server 14. Physical access to the Internet is provided in conventional fashion. The general command protocols, etc., for exchanging messages between the client 10 and the server 12, are also conventional.

The client machine 10 may preferably be an intelligent computer system, such as an IBM PS/2 computer, an IBM ThinkPad laptop computer, or an IBM RISC System/6000 workstation. Communication coupling made to the network via a suitable communication interface software tool, such as IBM's OS/2 WARP Connect software product. (IBM, PS/2, ThinkPad, RISC System/6000 and OS/2 are trademarks of IBM Corporation.)

The Web browser in the client machine 10 may preferably be the IBM Web Explorer software product, or equivalent software tools such as the NetScape or Mosaic tools. This computer system 10 is bi-directionally coupled with the OS/2 WARP Connect facility over a line or via a wireless system to the server machine 14. The server machine 14 may preferably be another IBM PS/2 computer, an IBM RISC System/6000 workstation, or other similar system.

The program run by the server 14, which incorporates the invention, may be provided to a general purpose computer on a pre-recorded medium, as discussed above, such as a floppy disk 15, in which case the novel and non-obvious features of the program code are pre-programmed onto the disk 15 for convenient sales and marketing, and for directing operation of the server machine 14 in accordance with the invention.

FIG. 3 is a high level flowchart showing the operation of the method of the invention. A preferred embodiment of the invention employs a client machine supporting a user interface utilizing HTML format, and further employs the World Wide Web. Much of the functionality of the invention may be carried out at the server. However, the description which follows will be readily understandable from the vantage point of the user at the client machine.

Initially, the user selects a video object (step 20). This may be done in any suitable way. The Web provides numerous formats, techniques, etc., for selecting objects. In the preferred embodiment, selection of the video object by the user causes a request to be sent over the Web from the client machine to the server machine. In particular, a Web page, provided by the server 14 to the client 10, may include a graphical interface, such as Hypertext links, for allowing the user to select an object. A preferred way of facilitating this user selection is by means of a graphical interface such as the Web page shown in FIG. 6 (discussed below). The Web page of FIG. 6 may be designed and implemented using the known techniques such as those given in the references discussed above.

In step 22, a first set of representations of selected points within the video data object are displayed, responsive to the user request. In the preferred embodiment employing the World Wide Web, the selection of the selected points in step 22 takes place at the server. A more detailed description of the server activity implementing step 22, for this preferred embodiment, is given in FIG. 4.

Referring to FIG. 4, the server initially receives a user request for a video object (step 24). The server accesses the video object (step 26), and selects a set of temporal points in the object (step 28, described in still more detail in FIG. 5). The server then produces representations of the selected temporal points for display (step 30). In a preferred embodiment, employing the Web and HTML links, the representations of the temporal points are thumbnail images corresponding with the temporal points in the video object. Finally, this information is provided for display (step 32), preferably by transmitting, from the server to the client machine, a display such as a Web page. A preferred implementation is that of the Web page shown in FIG. 6 (discussed below).

Before the display is illustrated and described in detail, a more detailed description will be made of step 28 (selecting temporal points). This may be done in any suitable fashion. A few possible ways would be to select temporal points at random, or evenly spaced, within the video object.

However, it is preferred that the temporal points be selected at scene cut points in the video object. In a preferred embodiment of the invention, step 28 is implemented as shown in more detail in FIG. 5.

Referring to FIG. 5, a scene change measure is computed for adjacent frames (step 34). In a particular preferred embodiment, the measure is computed by calculating a normalized correlation between the two frames. (For discussion purposes, the frames will be referred to as the "image" I and the "model" M, without necessarily implying any temporal order or sequence between them.)

It is possible to compute the scene change measure using straight correlation, in which case, each pixel of the image I is multiplied with a corresponding pixel of the model M, and a running sum of the products is accumulated. In such a scheme, correspondence might be established by means of row-and-column coordinates or other suitable methods.

However, a particularly preferred approach is as follows: For each pixel of the image I, a neighbourhood of the corresponding point in the model M is considered. For calculating the frame change measure, a point within the neighbourhood of the model M whose value is closest to the value of the point in the image I, preferably closest in terms of the grey scale value, is multiplied by the value of the point in the image I.

It has been found that using the neighbour point having the nearest value, rather than the exact corresponding point, improves response to motion in areas of high frequency image data. The set of neighbour points in the Model, for a given point in the Image, may be established in any suitable manner, such as by identifying the Image point according to a system such as row-and-column coordinates, and defining the Model neighbourhood as the points whose row and column coordinates matched those of the Image point, or were either one above or one below the Image point coordinate values.

Also, a normalized version of correlation is used, in order to reduce sensitivity to changes in illumination between images of essentially the same scene.

A preferred formula, incorporating both the correlation function and the pixel-neighbourhood search described above, has been found to work well. Specifically, it is easy to threshold, because it is already normalized. That formula is as follows:

$$r = \frac{n\sum_i I_i M_i - \sum_i I_i \sum_i M_i}{\sqrt{(n\sum_i I_i^2 - \sum_i I_i^2)\,(n\sum_i M_i^2 - \sum_i M_i^2)}}$$

In this expression, r is the scene change measure, and n is the number of active pixels used in the measure calculation. Depending on the particular circumstances or the preference of the user, n can be the total number of pixels in the frame, or a subset of the total number used for the calculation.

$I_i$ is the value (such as the grey scale value) for the i-thy one of the pixels in the image I. $M_i$ is the value for one of the pixels in the model M which is to be paired up with $I_i$ for the above-discussed scene change calculation.

The subscript i, for distinguishing between pixels, is a simplified representation for the purpose of the present discussion. Any suitable arrangement for distinguishing between pixels, which would be understood or deemed appropriate to a particular implementation, may be used. For instance, if the pixels are a rectangular array, then the subscript might be an ordered pair, such as row and column counts. A polar coordinate system, or other systems which would be understood to be suitable for a particular implementation, could also be used.

Note that, for this purpose, $M_i$ is either the i-thy pixel, or a pixel in the neighbourhood of the i-thy pixel. That is, if, for instance, a row-and-column coordinate system were used, and a neighbourhood were defined as given above, then the Image point $I_i = I_{r,c}$ and the Model point $M_i = M_{(r+1),c}$ might be paired up, if $M_{(r+1),c}$ had the closest value to $I_{r,c}$ of any of the Model points in the defined neighbourhood.

Once the scene change values have been determined for the various pairs of adjacent frames in the video object, they are compared with a frame change threshold (step 36 in FIG. 4). As stated above, this expression produces a normalized scene change value, the value being 0 for two identical frames. It is convenient to work with a scene change threshold having a value between 0 and 1. However, any suitable measure of the threshold may be employed.

Then, a test is made (step 38) to determine whether the value is greater than the threshold (alternatively, greater than or equal to the threshold; whichever is considered suitable according to the particular implementation). The result of this test determines whether the two frames are treated as a scene change (steps 40, 42).

A preferred graphical implementation of the user interface is shown in FIGs. 6 and 7. In a preferred embodiment, to be used in an environment such as the World Wide Web, the graphical interfaces are implemented using HTML. Details of this implementation will be omitted, since they would be known to persons skilled in the fields of HTML and other graphical user interfaces.

Referring first to FIG. 6, a selection menu is provided for the user. The menu lists video objects which are available. The listing may include thumbnail images 46 which illustrate the content of the video objects, text captions 48 naming or describing the video objects, or other suitable descriptors. Also, the menu may include other information which may be of use to the user, such as the characteristics of the stored file. For instance, FIG. 6 shows a legend "Energy 1 mpeg file" 50, which informs the user that the video data object contains image information which is compressed according to the MPEG data compression standard. The listed video objects are preferably provided in HTML format, so that the user can select a video object (step 20 of FIG. 3) by a suitable means such as mouse-clicking on the thumbnail image or other identifier.

Responsive to the user command, the server 14 obtains the video object and, as per FIG. 4, responds to the client machine 10. In step 44 of FIG. 3, a user interface is provided for viewing and browsing the video object. A preferred embodiment, for use on the Web, is illustrated in FIG. 7.

FIG. 7 shows a graphical image that would be displayed on the client machine for the user's viewing. In accordance with the invention, representations of several temporal points in the video object are shown. Preferably, these representations are shown as thumbnail images 52. For ease of comprehension, the thumbnail images 52 are preferably shown in temporal order, in a user-intuitive arrangement such as from left to right, forming a "storyboard" that lets the user scan his/her eyes from left to right, and get a sense of the sequence of images or events in the video object.

To further assist the user in getting a sense of where the displayed thumbnail images occur in the video object, a representation of the video object in its entirety, and of the portion covered by the displayed thumbnail images, is given. In FIG. 7, this is shown as a slider bar 54. The slider bar 54 is shown horizontally, but can be in any suitable configuration, preferably a configuration which comports well with user intuition. The entire horizontal length of the slider bar 54 represents the total duration of the video object, and a marker 56 represents the relative location, within the video object, of the displayed thumbnail images. Preferably, the marker 56 has a width, relative to the width of the entire slider bar 54, which reflects the portion of the entire video data object covered by all of the displayed representations.

The marker 56 may be moved using the technique, familiar to users of slider bars in conventional GUI window applications, of moving a cursor onto the marker 56 with a mouse, holding the mouse button down, and dragging and dropping the marker 56 by moving the mouse. Alternatively, a GUI control panel 55 may be provided, having buttons such as the left, right, fast left, and fast right button shown. By positioning the cursor on one of these buttons, or other suitable control buttons, and pressing a mouse button, the user causes the marker 56 to scroll along the slider bar 54.

Regardless of what particular technique the user uses to move the marker 56 along the slider bar 54, representations of points within the video data object appear and disappear to represent the key frames in whichever part of the video data object the marker 56 is covering at that moment. Thus, by moving the marker 56, the user selects a portion of the video data object which he or she wishes to view in more detail.

Also in accordance with the invention, identifying indicia such as consecutive numbers may be assigned to the key frames, to assist the user in keeping track of which key frames have been examined, or where, within the video data object as a whole, a given image appears. This information may be provided in the user interface. A shot count 57 displays the indicia, in the form of consecutive numbers. For instance, in FIG. 7, the marker 56 is at the left end of the slider bar 54. Let us say, then, that the first six thumbnails in the video data object being viewed are displayed. Accordingly, the shot count 57 displays a value of 1, indicating that the left-most thumbnail image represents the first thumbnail (i.e., the earliest temporal point) of the video data object. If a user were then to use the controls 55 or the marker 56 to move through the video data object, then, as the marker 56 moved to the right and different thumbnails 52 appeared for later points in the video data object, the shot count 57 would likewise ascend in value. As a result, a user might identify a shot of particular interest, and remember it for easy future access by taking note of the value of the shot count.

Step 22 of FIG. 3 is illustrated, in a preferred embodiment, by the thumbnail images 52 and the slider bar 54, discussed above.

Also in accordance with the invention, the user interface includes means for selecting an interval within the data objects. The interval is selected in terms of the displayed representations. Preferably, the user selects one of the representations, and the interval is selected in terms of an interval between the selected representation and another one of the representations, such as the temporally subsequent one of the representations. The method step 44 of FIG. 3, in which a user interface is provided for facilitating viewing and browsing, is implemented by providing the means for selecting.

In accordance with the invention, the user interface provided in step 44 includes means for selecting an interval between two of the key frames. Preferably the means for selecting an interval operates in conjunction with the graphical user interface GUI of the system according to the invention. That is, a user uses a GUI input device, such as a mouse, to select a representation of a desired interval from several representations of intervals shown on the display.

A preferred way of performing step 44 is shown in FIG. 8.

Referring to FIG. 8, a user is initially presented with a set of representations of frames of the video object, such as the sequence 52 of thumbnails shown in FIG. 7. The user views the thumbnails and decides that a portion of the video object which he/she would like to examine in more detail falls between two of the thumbnails. The user selects the interval (step 58) through the graphical user interface. In the preferred embodiment of FIG. 7, for instance, the user selects one of the thumbnail images. The thumbnail image is implemented as a hypertext link, and the user's selection of the thumbnail indicates that he/she is interested in the interval between that thumbnail and an adjacent thumbnail (preferably the thumbnail which follows the selected thumbnail, temporally).

The user's selection of the thumbnail is processed in known manner as a hypertext selection, and a message is sent to the server. The server examines the portion of the video object between the selected frame and the next frame. In a manner similar to that of step 28 in FIG. 4, the server selects a new set of key frames falling between the selected frame and the next frame (step 60).

The selected new frames are then displayed (step 62), essentially in the same manner as before (steps 30 and 32 of FIG. 4). Also, the slider bar 56 reflects the selection of the interval by moving and/or changing in width.

In many cases, the user will want to perform several iterations of this process, in order to narrow down to the particular small area of interest within the video object. Accordingly, a repetition loop is shown in FIG. 8 (step 64 and the loop back to step 58).

## Claims

1. A method for facilitating user examination and browsing of video data object information pertaining to a stored video data object (Object), the Object including a beginning point, an end point, and points therebetween, the method comprising the steps of:

   providing (30), to a user interface, a sequence of representations, each representation corresponding with one of the points in the Object;

   a user utilizing a user interface to select (60) an interval of the Object in terms of a first one of the points; and responsive to user selection of such an interval, providing (62) a subsequence of representations, each representation of the subsequence corresponding with a respective point in the Object, each of the respective points falling within the selected interval.

**2.** A method as claimed in claim 1, wherein the step of providing (30) a sequence of representations includes communicating, through a network communication medium (7), with a video repository (14) having video data objects, including the Object, stored therein.

**3.** A method as claimed in claim 1, wherein the step of providing a sequence of representations includes displaying (32) still images representative of content of the Object at respective ones of the points in the Object.

**4.** A method as claimed in claim 1, wherein the step of providing a sequence of representations includes:

generating thumbnail images based on content of the Object at respective ones of the points in the Object; and

providing thumbnail images representing still frames of the video object at the respective points therein.

**5.** A method as claimed in claim 1, wherein the step of providing a user interface includes:

displaying a representation of the Object in its entirety; and
displaying a representation of a portion of the Object covered by the displayed representations of points in the Object.

**6.** A method as claimed in claim 5, wherein:
the step of displaying the representation in its entirety includes displaying a slider bar (54); and
the step of displaying a representation of a portion of the Object covered by the displayed representations of points in the Object includes displaying a marker (56) on the slider bar (54).

**7.** A method as claimed in claim 6, wherein:
the step of selecting (60) an interval includes moving the marker (56) along the slider bar (54) from a first position thereon to a second position thereon; and
the step of displaying the sequence of representations (52) includes the step (62), responsive to movement of the marker along the slider, of displaying representations of points within the Object corresponding with the second position of the marker, in place of previously displayed representations of points within the Object corresponding with the first position of the marker.

**8.** A method as claimed in claim 1, wherein the step of selecting an interval includes selecting one of the representations of still images which forms a boundary of the interval to be selected.

**9.** A method as claimed in claim 8, wherein the step of selecting an interval further includes selecting one of the representations of still images which forms a first boundary of the interval to be selected, a next one of the representations forming a second boundary of the interval.

**10.** A system for facilitating user examination and browsing of video data object information pertaining to a stored video data object (Object), the Object including a beginning point, an end point, and points therebetween, the system comprising:

a display unit;

an interface for coupling the display unit with a network communication medium (7), the medium being coupled to a video repository (14) having video data objects, including the Object, stored therein;

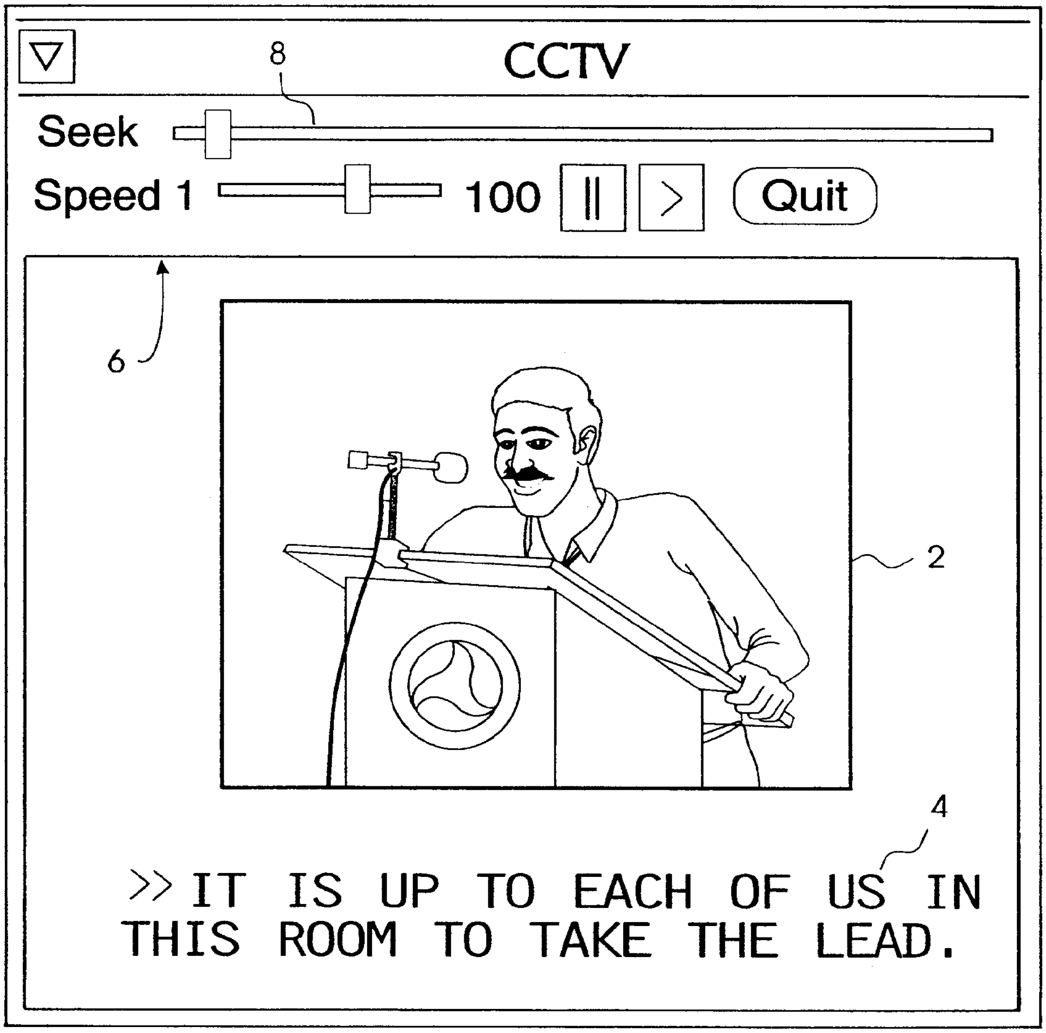
means for sending, through the interface and over the medium, a request to view the Object;

means, responsive to receipt of a response to the request to view, for displaying a sequence of representations (52), each representation corresponding with one of the points in the Object;

means (56) for selecting an interval in terms of a first one of the points; and means, operable responsive to operation of the means for selecting an interval, for providing a subsequence of representations, each representation of the subsequence corresponding with a respective point in the Object, each of the respective points falling within the selected interval.

**11.** A system as claimed in claim 10, wherein:
the network communication medium (7) includes the Internet;
the interface includes an Internet interface; and
the video repository (14) includes a server on the Internet.

**12.** A system as claimed in claim 10, wherein the beginning point, the end point, and the points are temporal points within the Object.



>> IT IS UP TO EACH OF US IN THIS ROOM TO TAKE THE LEAD.

*FIG. 1*
*(PRIOR ART)*

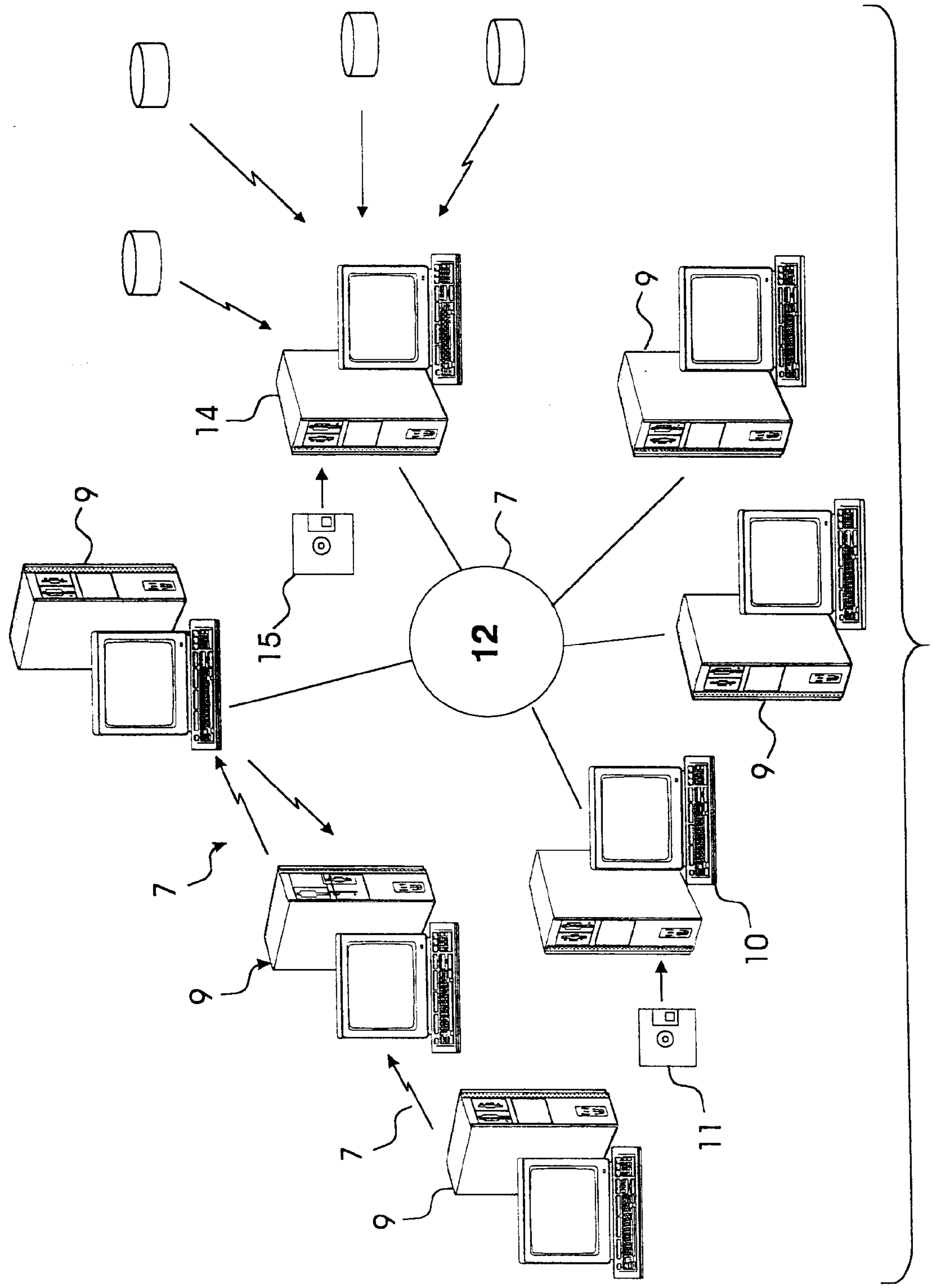


FIG. 2

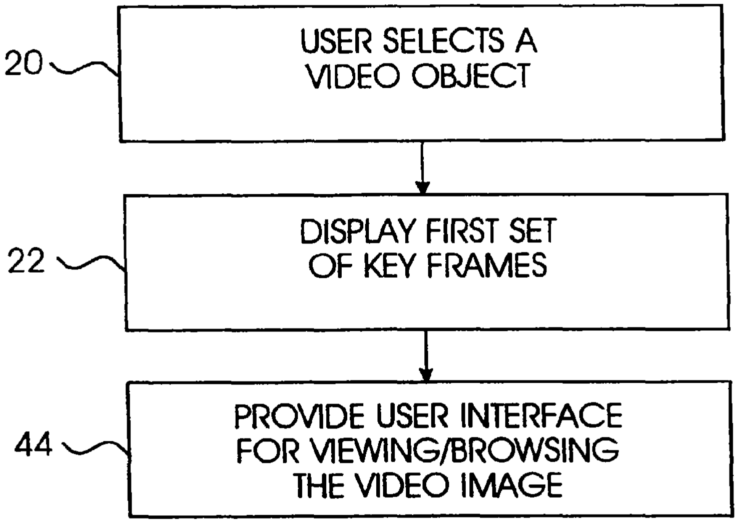

20
USER SELECTS A VIDEO OBJECT
22
DISPLAY FIRST SET OF KEY FRAMES
44
PROVIDE USER INTERFACE FOR VIEWING/BROWSING THE VIDEO IMAGE

*FIG. 3*

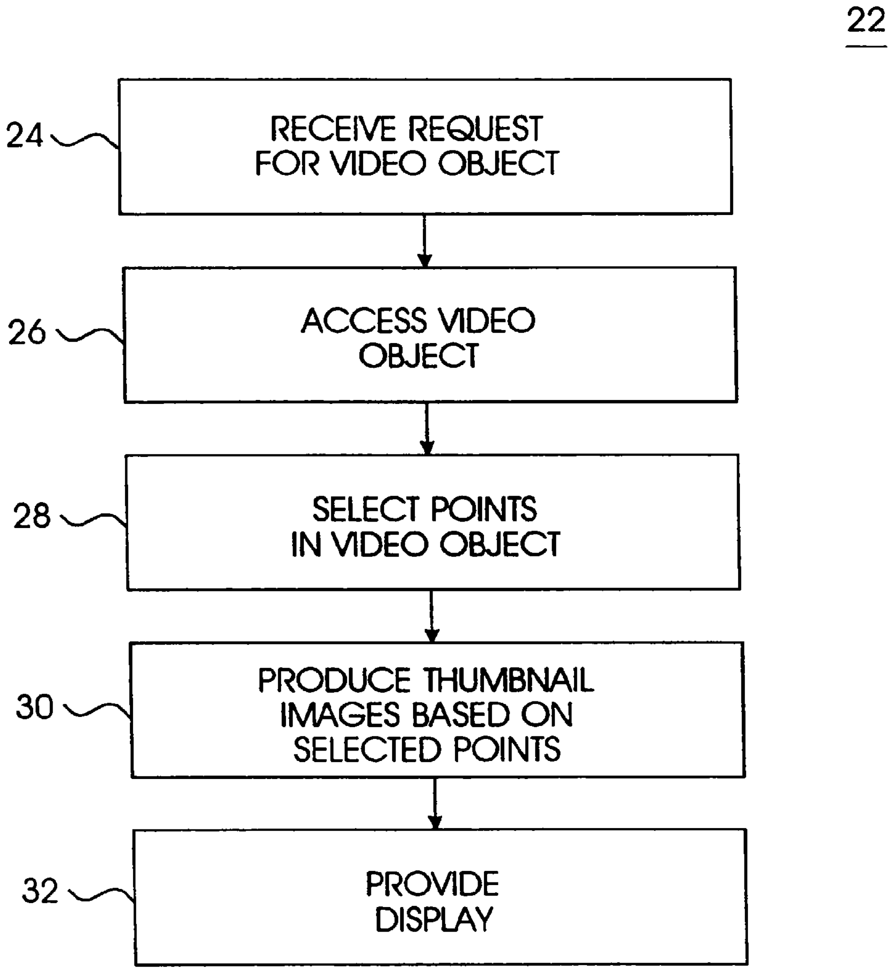

22
24
RECEIVE REQUEST FOR VIDEO OBJECT
26
ACCESS VIDEO OBJECT
28
SELECT POINTS IN VIDEO OBJECT
30
PRODUCE THUMBNAIL IMAGES BASED ON SELECTED POINTS
32
PROVIDE DISPLAY

*FIG. 4*

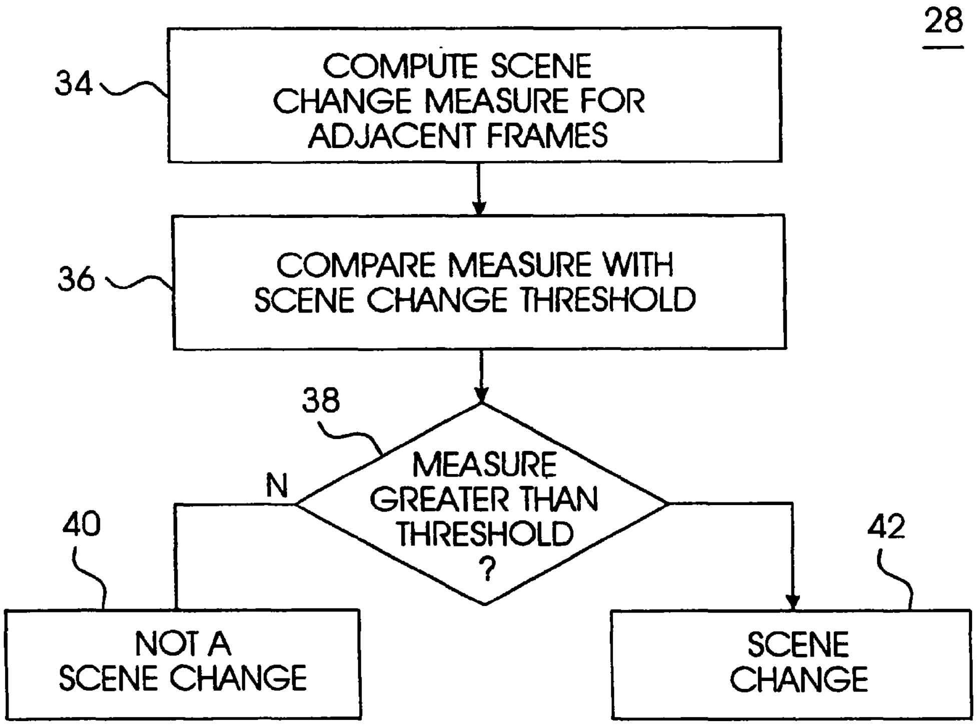

28
34
COMPUTE SCENE CHANGE MEASURE FOR ADJACENT FRAMES
36
COMPARE MEASURE WITH SCENE CHANGE THRESHOLD
38
MEASURE GREATER THAN THRESHOLD ?
N
40
NOT A SCENE CHANGE
42
SCENE CHANGE

*FIG. 5*

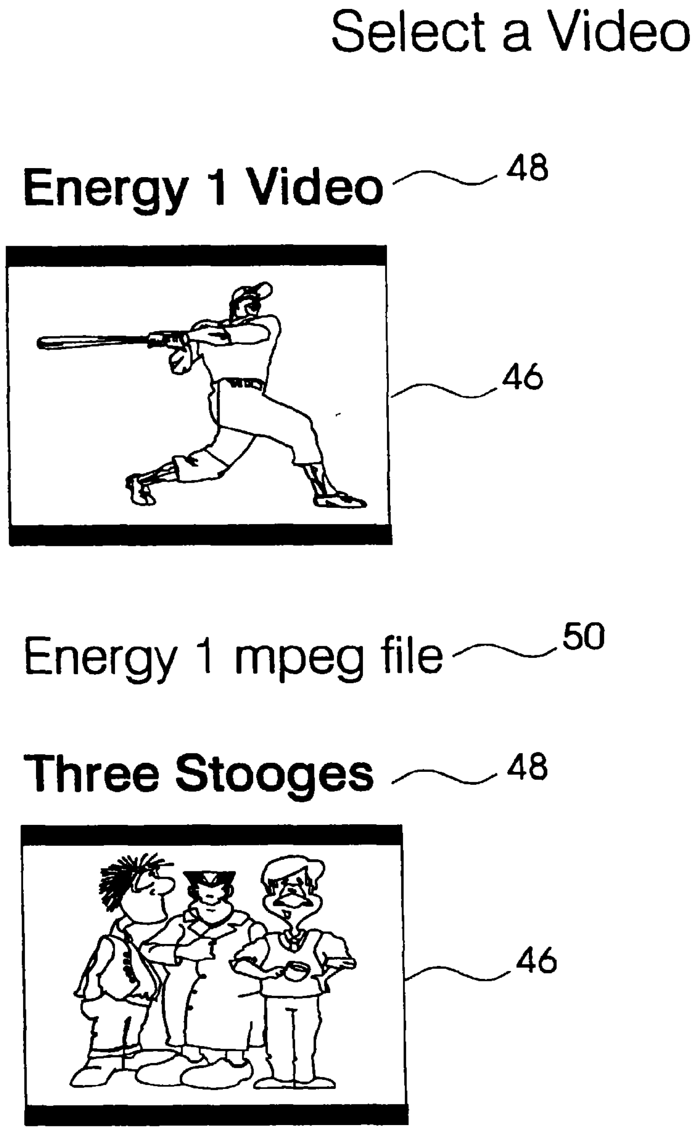

Select a Video
Energy 1 Video
48
46
Energy 1 mpeg file
50
Three Stooges
48
46

*FIG. 6*

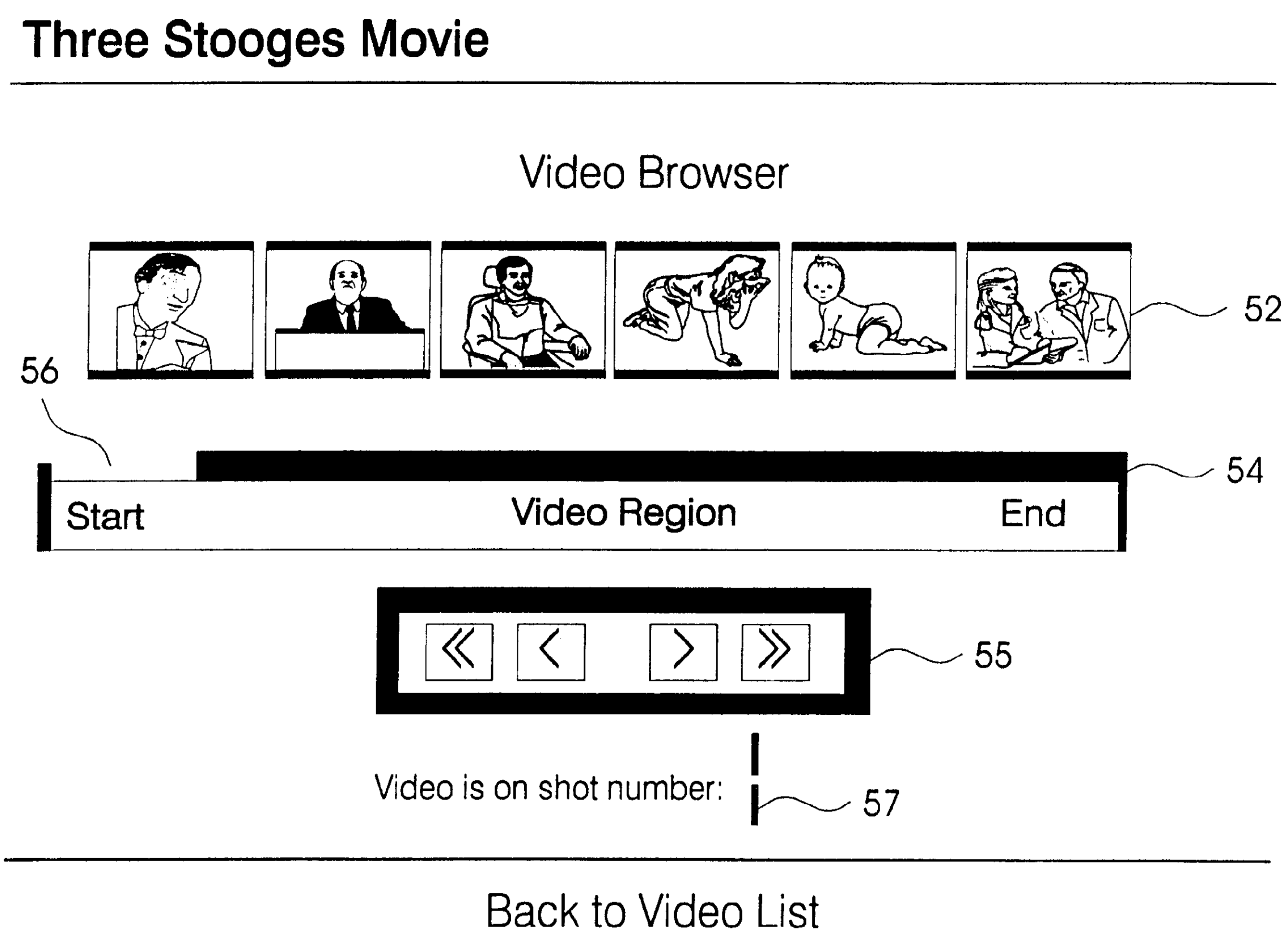

Three Stooges Movie
Video Browser
52
56
54
Start
Video Region
End
55
Video is on shot number:
57
Back to Video List

FIG. 7

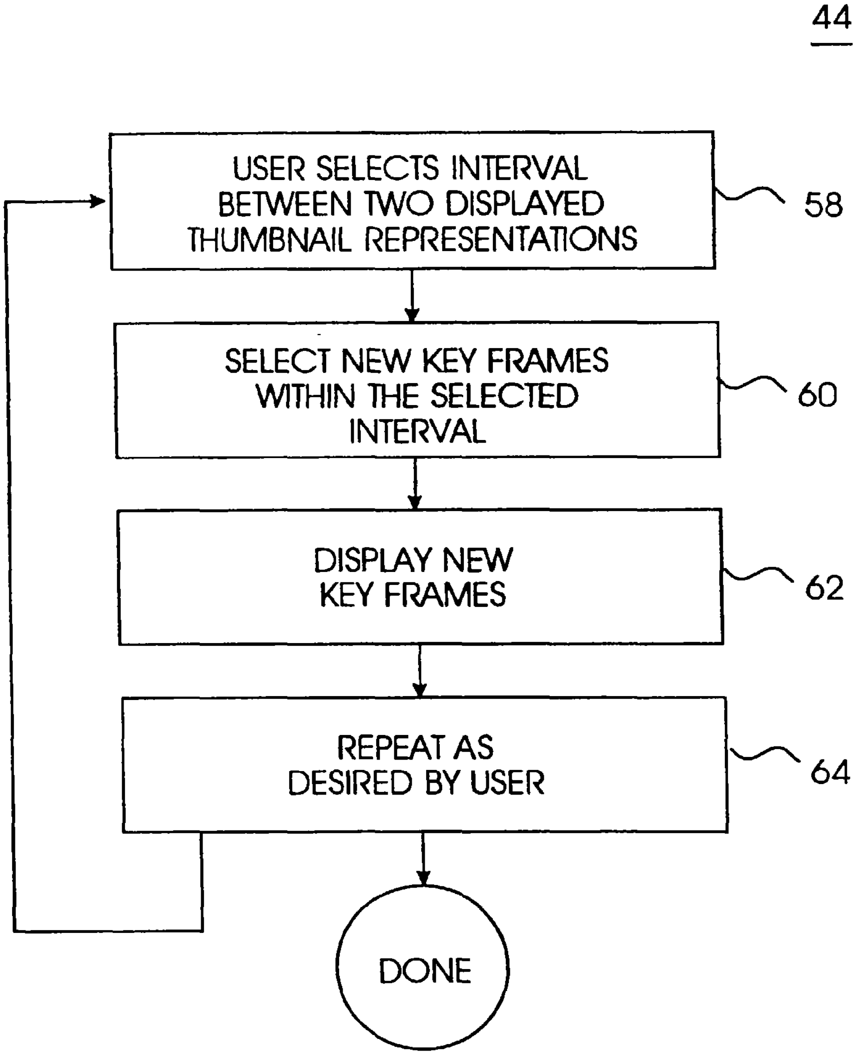

44
USER SELECTS INTERVAL BETWEEN TWO DISPLAYED THUMBNAIL REPRESENTATIONS
58
SELECT NEW KEY FRAMES WITHIN THE SELECTED INTERVAL
60
DISPLAY NEW KEY FRAMES
62
REPEAT AS DESIRED BY USER
64
DONE

*FIG. 8*

European Patent Office

# EUROPEAN SEARCH REPORT

Application Number
EP 96 30 8817

## DOCUMENTS CONSIDERED TO BE RELEVANT

| Category | Citation of document with indication, where appropriate, of relevant passages | Relevant to claim | CLASSIFICATION OF THE APPLICATION (Int.Cl.6) |
|---|---|---|---|
| X | CHI'92 CONFERENCE PROCEEDINGS , MONTEREY, MAY 3 - 7, 1992,<br>no. -, 3 May 1992, BAUERSFELD P;BENNETT J; LYNCH G,<br>pages 93-98, XP000426811<br>MILLS M ET AL: "A MAGNIFIER TOOL FOR VIDEO DATA"<br>* the whole document *<br>--- | 1-12 | G06F17/30 |
| X | EP 0 619 550 A (IBM) 12 October 1994<br>* abstract; claims; figures 3,5 *<br>--- | 1-10,12 | |
| A | PROCEEDINGS OF THE REGION 10 ANNUAL INTERNATIONAL CONFERENCE (TENCO, SINGAPORE, 22 - 26 AUG., 1994,<br>vol. 2, 22 August 1994, CHAN T K Y (ED ),<br>pages 852-856, XP000528240<br>SMOLIAR S W ET AL: "INTERACTING WITH DIGITAL VIDEO"<br>* page 854, right-hand column, paragraph 2.3 - page 855, right-hand column, paragraph 4 *<br>--- | 1-12 | |
| A | IEEE MULTIMEDIA,<br>vol. 2, no. 1, 21 March 1995,<br>pages 12-25, XP000500083<br>WEISS R ET AL: "COMPOSITION AND SEARCH WITH A VIDEO ALGEBRA"<br>* page 22, right-hand column, line 1 - page 24, left-hand column, line 6; figures 2,9,10 *<br>----- | 1,10,11 | TECHNICAL FIELDS SEARCHED (Int.Cl.6)<br>G06F |

The present search report has been drawn up for all claims

| Place of search | Date of completion of the search | Examiner |
|---|---|---|
| THE HAGUE | 9 April 1997 | Fournier, C |

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 818 907 A2

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
**14.01.1998 Bulletin 1998/03**

(51) Int. Cl.$^{6}$: **H04L 29/06**, H04N 7/173,
H04L 12/14, H04M 15/00

(21) Application number: **97111872.4**

(22) Date of filing: **11.07.1997**

(84) Designated Contracting States:
**AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE**
Designated Extension States:
**AL LT LV RO SI**

(30) Priority: **12.07.1996 US 678915**

(71) Applicant: **AT&T Corp.**
**New York, NY 10013-2412 (US)**

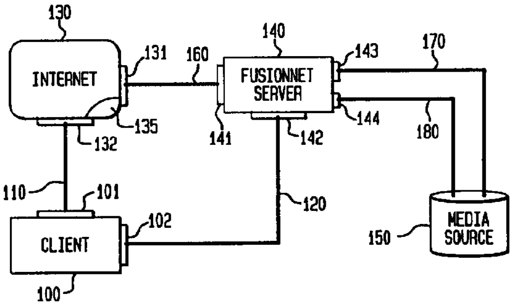
(72) Inventor: **Civanlar, Mehmet Reha**
**Red Bank, New Jersey 07701 (US)**

(74) Representative:
**KUHNEN, WACKER & PARTNER**
**Alois-Steinecker-Strasse 22**
**85354 Freising (DE)**

# (54) Improved client-server architecture using internet and guaranteed quality of service networks for accessing distributed media sources

(57) An improved client-server architecture of the present invention utilizes the advantages of known QOS networks to provide guaranteed quality of service, security, and a charge mechanism for handling requests initiated over a packet network, such as the Internet, for access to distributed media sources. Such media sources may be independent of the QOS network provider and may be located by browsing the Internet. A method of operating a client-server network enables the system level merger of the Internet and a guaranteed QOS network, such as the public switched telephone network, in order to provide the users with a complete information superhighway today. It will appear to the average user that the Internet and QOS network are fused together. Thus, when a user, connected to the Internet, selects an application that requires functionalities offered by the telephone network, such as guaranteed QOS delivery of media information or customized billing, the Internet-resident application will communicate information to a server, which will in turn initiate a session over the QOS network for delivery of the required information to the client using client information transmitted from the client (or from the application) to the server over the established Internet session. The client information may include a client account number, login and password, and/or phone number to enable the server to establish the switched network connection to the client. Accordingly, media sources which are separate and independent from the QOS network provider may be accessed using a secure, guaranteed QOS network in a manner providing for ease of identification and billing.

FIG. 1



Printed by Xerox (UK) Business Services
2.15.8/3.4

## Description

The present application is related to U.S. Application Serial No. 08/402,664 entitled Client-Server Architecture Using Internet and Public Switched Networks, filed Mary 13, 1995, and to U.S. Application Serial No. 08/648,260 entitled Multimedia Information Service Access, filed on May 15, 1996. These three applications are co-pending and commonly assigned.

## Technical Field

This invention relates to the Internet network and, more particularly, provide an improved client-server architecture utilizing a packet network, such as the Internet, and a guaranteed quality of service network, such as the public switched network, for access to distributed media sources.

## Background of the Invention

The Internet's global and exponential growth is common knowledge today. The Internet is implemented using a large variety of connections between millions of computers. Internet access is readily available to individuals across the globe. Various on-line service providers, such as America Online, CompuServe, Prodigy, Netcom, etc., use PSTN with modems or ISDN adapters for client connections. These on-line service providers maintain servers on the Internet providing client access to the Internet.

The recent developments on the World Wide Web user interfaces and information navigation software such as the Netscape web browser, coupled with a continuously growing number of public access providers, are making the Internet a fundamental component of the information age, if not the information super highway itself. World Wide Web sites on the Internet are typically accessible through a browser program which interprets scripts written in Hyper Text Markup Language (HTML). Users may browse the World Wide Web for virtually any kind of information, including information having content derived from one or more media, such as words, sounds or images. It is desired to enable the on-demand access to a variety of sources of media information using the Internet.

On the other hand, it is also well known that the current Internet is deficient, as compared to other existing networks, in the following three fundamental functionalities expected from a complete information network: 1) quality of service (QOS), 2) security, and 3) an easy and flexible mechanism to charge for the information and transmission services. These deficiencies discourage not only media providers from providing real-time on-demand access to media source information over the Internet, but also users who might otherwise desire to purchase real-time on-demand access to information held by such media sources. Although recent developments in the Internet Protocol (IP) and in Internet applications show promise for providing tools for secure transmission and real-time support, without a globally-agreed and implemented billing mechanism, guaranteed QOS over a global network is not possible. Consequently, none of these mechanisms can provide reliable, real-time global transmission employing the Internet today. One such service which may not be adequately provided by the Internet would be a request for delivery of a long video or audio segment. It may not be possible to have a guaranteed real-time delivery of such a long video and audio segment because one or more of the routers or computer links in the Internet may be busy handling other requests.

The existing telephone networks, on the other hand, have been offering the capabilities of guaranteed quality of service, security, and an easy and flexible mechanism to charge for the information and transmission services for a long time; the level of QOS, security and ease and flexibility of billing provided by existing telephone networks clearly surpasses the levels of these functions afforded by the Internet. What is desired is a practical way to utilize these advantages to provide guaranteed quality of service, security, and a charge mechanism for handling requests to access media sources that may be located by browsing the Internet.

Pending U.S. Application Serial No. 08/402,664 and U.S. Application Serial No. 08/648,260 have recognized these shortcoming of the Internet and the advantages of using a QOS network, such as the public switched telephone network, in tandem with the Internet to provide guaranteed quality of service, security and an easy and flexible billing mechanism. The present invention utilizes these advantages to enhance access to distributed media sources that are unbundled from or separate from the server architecture.

## Summary of the Invention

An object of the improved client-server architecture of the present invention is to utilize the advantages of known QOS networks to provide enhancements in functions, such as quality of service, security, and billing, for handling requests initiated over a packet network, such as the Internet, for on-demand access to distributed media sources. Such media sources may be independent of the QOS network provider.

In accordance with the present invention, a method of operating a client-server network enables a system level merger of a packet network, such as the Internet, and a guaranteed QOS network, such as the public switched network, in order to provide the users with a complete information superhighway today. As the present invention will make it appear to the average user that the packet network and the QOS network appear to be fused together, the present invention will be referred to from time-to-time as "FusionNet".

When a user engaged in an Internet session selects an application that requires enhanced functionalities offered by the telephone network, such as guaranteed QOS delivery of media information or customized billing, the Internet-resident application will communicate information to a FusionNet server, which will in turn initiate a communications session for on-demand delivery of the required information to the client over the QOS network using client information transmitted from the client (or from the application) to the server. The client information may include a client account number, login and password, and/or phone number to enable the server to establish the switched network connection to the client. Billing for service and transmission may proceed as in a normal or collect telephone call. There is no security problem or complicated identification mechanism. Also, because of the built-in security of the telephone networks, FusionNet's guaranteed QOS path does not require a firewall between the client an the outside world in order to avert unauthorized access.

Accordingly, media sources which are separate and independent from the QOS network provider may be accessed using a secure, guaranteed QOS network in a manner providing for ease of identification and billing.

## Brief Description of the Drawings

FIG. 1 shows an illustrative client-server architecture using both the Internet and a guaranteed QOS network in accordance with an embodiment of the present invention.

FIG. 2 shows another depiction of an illustrative client-server architecture using both the Internet and a guaranteed QOS network connected to a media source through a gateway in accordance with the present invention.

FIG. 3 shows a variation of the client-server architecture in which a client connects to the Internet and to the FusionNet server through a service provider in accordance with the present invention.

## Detailed Description

The Internet is a dynamic packet network consisting of millions of interconnected computers which could run several applications, such as the World Wide Web. Web browsers such as the one by Netscape are programs using a graphical user interface that provides a user easy access to various services over the Internet. The present invention enables the above-identified functionality of a QOS network, such as the public switched network, to be combined with the Internet functionality to enhance services provided by existing Internet applications and to create new ones.

The present invention provides a seamlessly integrated system that makes it possible to use a packet network, such as the Internet, together with a QOS network that offers functions such as 1) quality of service, 2) security and 3) an easy and flexible charging mechanism at an enhanced level (as compared with the packet network). The QOS network may be a known switched network, such as the public switched telephone network (PSTN), the integrated services digital network (ISDN), an asynchronous transmission network (ATM), a cable network used for video or television programs, etc., or it may also be private or corporate communication or data network as long as it provides enhancements in at least one of the three functionalities described above.

A high level block diagram for a FusionNet services architecture is shown in FIG. 1. With reference to FIG. 1, the system is based on a client-server architecture, where the client apparatus 100 may be either a personal computer (PC), a single workstation or two such systems which can be operated in a coordinated manner (e.g., physically close systems, users who talk over the telephone, etc.). The key factor is that each client station 100 requires two logical network connections or interfaces, e.g., 101 and 102. One of these interfaces 101 provides a connection to the Internet 130 which can be made through one of a number of known channels, e.g., a local area network (LAN) connection, a Serial Line Internet Protocol (SLIP) or Point-to-Point Protocol (PPP) connection over a modem or over an ISDN port. These connections are made through network 110 which may be either a private connection or the public switched telephone network. The second interface 102 provides a connection to guaranteed QOS network 120, e.g., a public or private switched telephone network, which may also be made through a modem, an ISDN port, or via a connection to a special LAN such as an ATM LAN or a LAN that offers bandwidth reservations.

It should be noted that these two interface connections 101 and 102 are defined at the logical level, that is, client apparatus 100 may have a single physical connection (i.e., modem, ISDN adapter, etc.) that can be used to attach it to both the Internet 130 and the guaranteed QOS network (e.g., public switched network) 120. For example, a modem with "Call Waiting" functionality, an ISDN line used as two separate B channels, an ATM LAN connection where the PSTN connection can be accomplished through a bridge, or a video telephony connection conforming to the CCITT H.320 standard which provides Internet connectivity over its data channel can provide the required two logical connections over a single physical connection.

Similarly, a FusionNet server 140 has access to both Internet 130 and the QOS network 120 via logical network interfaces 141 and 142. A FusionNet server 140 may be a computer or other software-driven machine having processing capability for carrying out the server functions herein described, or it may be a software package that includes those sever functions and that operates on a computer or other software-driven machine. Generally, the system of FIG. 1 enables a plurality of client computer-based apparatuses 100 to access a plurality of FusionNet servers 140 via Internet 130 and QOS network 120. Typically, access to the

Internet is made through a logical network interface 132. Typically, access by a client apparatus 100 to FusionNet server 140 via the Internet 130 utilizes the HyperText Transport Protocol (HTTP) which is accessed using a browser application program available at the client apparatus 100.

A general overview of the operation of the present invention is as follows. A user at a client apparatus 100 connects to Internet 130 and requests an access to media information requiring enhanced functionalities not available on Internet 130, but offered by QOS network 120. Illustratively, an enhanced service request would be one requesting a guaranteed QOS delivery, security, or customized charging. Providing such enhanced Internet services would enable the delivery over the switched network of 1) real-time video transmission, 2) real-time, high quality audio transmission, or 3) immediate access to sensitive data such as stock market data, as just three examples. The user may locate such desired media information by, for example, browsing the Internet and locating references to one or more media sources at a variety of World Wide Web sites. World Wide Web sites on the Internet are typically accessible through a browser program. As an example, a World Wide Web site could contain information about one or more media sources, such as the type of media (e.g., music, text, images, or movies), available selections (e.g., a list of movies by title and lead performers), and pricing information. Alternatively, other information sources available over the Internet (e.g., a USENET newsgroup posting) could provide information about a media source.

Upon locating a desired media source 150 by browsing a World Wide Web site 135, the user selects access to the media source 150 by clicking on a reference point contained in Web site 135, which in turn initiates a program to establish a communications session between Internet 130 and FusionNet server 140 using network 160 and logical network interfaces 131 and 141. During this process a program running from Web site 135 may request information about the client, for example the client's telephone number, a login name and password, or an account number. The client information along with information identifying the selected media source is then passed along to FusionNet server 140 over communications path 160. Functions such as these may be implemented using the well-known Common Gateway Interface (CGI) for which public software implementations are available for HTTP-compatible servers using common computer communications interfaces such as sockets. The CGI is a publicly-documented open interface with a specification that can be readily obtained from the Internet. For example, an Internet World Wide Web site maintained by the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign contains a description of CGI and its specification as well as a CGI program archive.

FusionNet server 140 then undertakes to establish a communications session with the selected media source 150, utilizing network 170 and logical network interface 143, for purposes of controlling the selection and delivery of media information available through the media source. An address for the media source, if necessary, may be provided by the Web site 135 or stored by the FusionNet server 140. Communications path 170 can be any one of a number of known communications channels, e.g., a private communications network, the PSTN, or even the Internet. Once a communications session is established between FusionNet server 140 and selected media source 150, FusionNet server 140 could then query the selected media source for availability of the desired media information.

After FusionNet server 140 establishes the communications session with selected media source 150 over communications path 170, the server 140 then initiates a communications session over a QOS network 120 with client 100. This communications session may be established, for example, by initiating a call connection over the public switched network using a telephone number provided by the client 100 as part of the service request. Several programs for automatic dialing of PSTN or ISDN numbers are available for various computers (e.g., PictureTel™ PCS series) and can be used for this purpose. Client apparatus 100 would include a program that accepts incoming phone calls. Such software is also available from many vendors (e.g., PictureTel™ PCS series). Such a call may be billed by any of the standard call billing methods, e.g., collect or third party billing, etc. The QOS network 120 chosen for the communications session provides the billing for the service and transmission, as well as enhanced security, and complicated identification mechanisms are not needed. In a typical application or service request between a client and server, both the Internet and QOS network connection may coexist, or the Internet session may be discontinued. Another possibility is to use an already established client account for billing purposes.

The actual transaction, e.g. security checks and sending a file or audio/video data over the QOS connection, can then be handled by the same process or yet another process. FusionNet server 140 completes the QOS connection between the client 100 and the selected media source 150 by establishing communications session over path 180 between the server 140 and media source 150. Communications path 180 may also be made using the PSTN, or ISDN network, or another QOS network. FusionNet server 140 then causes the media source to transfer media information to client 100 over the OQS connection, and server 140 can retain control over the transfer process by causing media source 150 to, e.g., stop, pause, or retransmit the requested media information.

Typically, QOS network connections 120 and 180 will be of a common type (e.g., each PSTN) such that FusionNet server 140 may simple implement a switched connection of the two QOS network connections providing therefore a single connection between client 100 and media source 150 to permit transfer of media information from the source 150 to client 100. It will be obvious to one skilled in the art that the media information could be transferred, subject to the control of FusionNet server 140, from media source 150 to client 100 over a QOS network directly connected between source 150 and client 100 without routing the media information through FusionNet server 140.

Where the transfer of information through network 120 and network 180 requires more than just a simple switching operation, e.g., data conversion or data reformatting is required, a FusionNet gateway 100 as shown in FIG. 2 operates to transfer media information from source 150 to client 100. All of the other elements in FIG. 2 are the same as those in FIG. 1, and therefore bear the same reference numerals. Once the QOS gateway connection is established, FusionNet server 140 instructs media source 150 to transmit the desired media information to client 100. The FusionNet gateway 200 may, if necessary, translate or reformat the information as it is transferred from source 150 to client 100. Gateway 200 may, alternatively, adjust to the client's information protocol. At the end of the transaction, the QOS gateway connection may be dropped. FusionNet server 140 may then complete any remaining billing procedures for the transfer.

A variation of the architecture described above may be derived by shifting the client connection functionality to a FusionNet service provider as shown in FIG. 3. The client makes a single connection to the FusionNet service provider which, in turn, would implement the two logical connections for the Internet and the QOS network connection to the FusionNet server, respectively. This has the advantage of providing a client access to the FusionNet architecture by making only a single physical connection which may reduce the complexity of client apparatus as well as remove the need for the client to coordinate two logical connections. Referring to FIG. 3, a user at client station 300 establishes a connection 310 with FusionNet service provider 320. Connection 310 is a point-to-point connection made using logical interfaces 301 and 321 through a QOS network that may be one of the known channels such as the PSTN, or ISDN or a private telephone network.

In this variation of the client-server architecture, FusionNet service provider 320 has the capability of establishing communications sessions with the Internet and with a FusionNet server that are functionally equivalent to the sessions among the client, the Internet and the FusionNet server described above with reference to FIGS. 1 and 2. In FIG. 3, a communications session is established between the FusionNet service provider and the Internet using communications path 330 and logical interface 322, which correspond to path 110 and logical interface 101 shown, respectively, in FIGS. 1 and 2. Similarly, in FIG. 3 a communications session between the FusionNet service provider and the FusionNet server is established using path 340 and logical interface 323, which correspond to path 120 and logical interface 102 shown, respectively, in FIGS. 1 and 2. FusionNet service provider 320 also has the capability of obtaining and supplying the client information, and transferring the desired media information obtained from the selected media source to the client. Since the user is connected to the FusionNet service provider through a point-to-point dedicated connection, the advantages of using a QOS network as described above can be preserved.

The remaining steps in obtaining access to the selected media source proceed as described above with reference to FIGS. 1 and 2, with the FusionNet service provider 320 shown in FIG. 3 substituting in place of client 100 that is shown in FIGS. 1 and 2. Thus, whereas the client is the requesting party in the embodiment shown in FIGS. 1 and 2, the FusionNet service provider becomes the requesting party (on behalf of the client) for access to a media source in the variation shown in FIG. 3.

In summary, an improved client-server architecture for access to distributed media sources locatable over the Internet has been described which utilizes the advantages of known QOS networks, such as the PSTN, to provide guaranteed quality of service, security, and a charge mechanism for handling such media access requests. The architecture has the advantage of allowing access to media sources that may be independent from the QOS network provider. Where necessary, the architecture may include a gateway for transferring the requested information from the media source to the client. A variation of the architecture provides for the client functionality to be included within a service provider so that a user may obtain the benefits of the FusionNet architecture while making only a single connection to the service provider.

## Claims

1. A method linking a requesting party (100) to a selected one of a plurality of media sources (150) locatable over a packet network (130) in response to a client request, characterized by the steps of:

   a. receiving information about the client and the selected media source at a server (140) over a first communications session established using the packet network (130);

   b. establishing a second communications session between the server and the selected media source (150); and

   c. transferring information over the second communications session to cause the selected media source to transfer media information, subject to the control of the server, to the requesting party over a third communications session established using a network (120) having at least one functional quality different from a functional quality of the packet network.

2. The method according to claim 1, characterized in that the requesting party is the client.

3. The method according to claim 1 or 2, characterized in that the packet network is the Internet.

4. The method according to claim 1, characterized in that the client information includes a communication number for use in establishing the third communications session with the requesting party.

5. The method according to claim 1, characterized in that the client information includes a login and password from which the server determines a communication number for use in establishing the third communications session with the requesting party or includes an account number from which the server determines a communication number for use in establishing the third communications session with the requesting party.

6. The method according to claim 1, characterized in that the client information includes a communication number for use in billing the client or includes a login and password from which the server determines a communication number for use in billing the client or includes an account number from which the server determines a communication number for use in billing the client.

7. The method according to claim 1, further characterized by the step of before transferring information over the second communications session to cause the media source to transfer media information to the requesting party over the third communications session, interrogating the selected media source using the second communications session to determine the availability of the requested media information.

8. The method according to claim 1 or 7, characterized in that the third communications session is established among the server, the requesting party, and the selected media source.

9. The method according to claim 8, characterized in that the third communications session is established using a modem or using an ISDN adapter or using the public switched telephone network or using a private telephone network or using a high speed data transmission link or using a cable network employed for video or television.

10. The method according to claim 8 or 9, further characterized by the step of transferring additional client information from the requesting party to the server over the third communications session for use in billing the client.

11. The method according to claim 10, characterized in that the additional client information includes an account number for use in billing the client.

12. The method according to claim 1, characterized in that the third communications session is established by linking each of the selected media source and the requesting party to a gateway.

13. A system for linking a requesting party (100) to a selected one of a plurality of media sources (150) locatable over a packet network (130) in response to a client request, characterized by:

a. a server (140) for communicating with a selected media source;

b. a first interface (141) for receiving client information at the server using the packet network;

c. a second interface (143) for establishing a communications session between the server and the selected media source; and

d. a controller causing the selected media source to transfer media information to the requesting party over a communications session established using a network (120) having at least one functional quality different from a functional quality of the packet network.

14. The system according to claim 13, characterized in that the requesting party is the client.

15. The system according to claim 13, characterized in that the packet network is the Internet.

16. The system according to claim 13, characterized in that the first interface includes a mechanism used to transmit information according to the Internet Protocol or includes means responsive to the client request for identifying the selected media source in order to establish the communications session between the server and the selected media source.

17. The system according to claim 13, further characterized by a third interface for establishing a communications session among the requesting party, the server, and the selected media source.

18. The system according to claim 17, characterized in that the third interface includes a mechanism used

to establish communications session employing the public switched telephone network or employing a private telephone network or

employing a high speed data transmission link or through a cable network employed for video or television.

**19.** The system according to claim 17, characterized in that the third interface includes a modem or an ISDN adapter.

**20.** The system according to claim 13, further characterized by a gateway for linking the requesting party and the selected media source.

FIG. 1

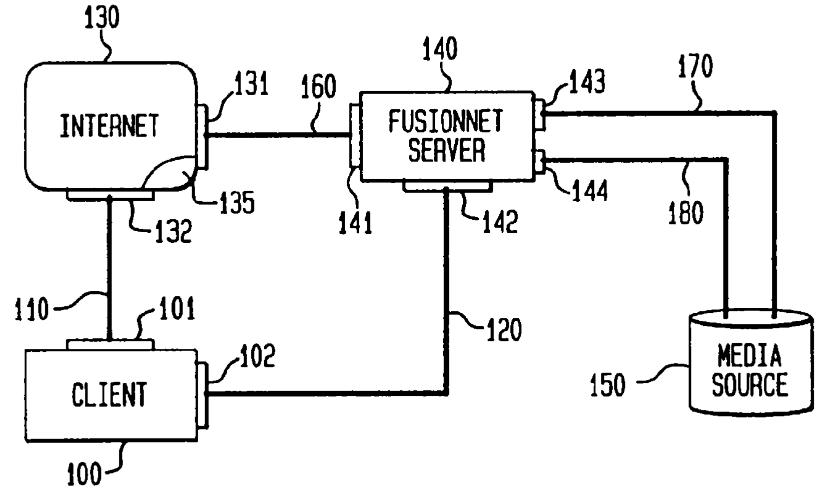


FIG. 2

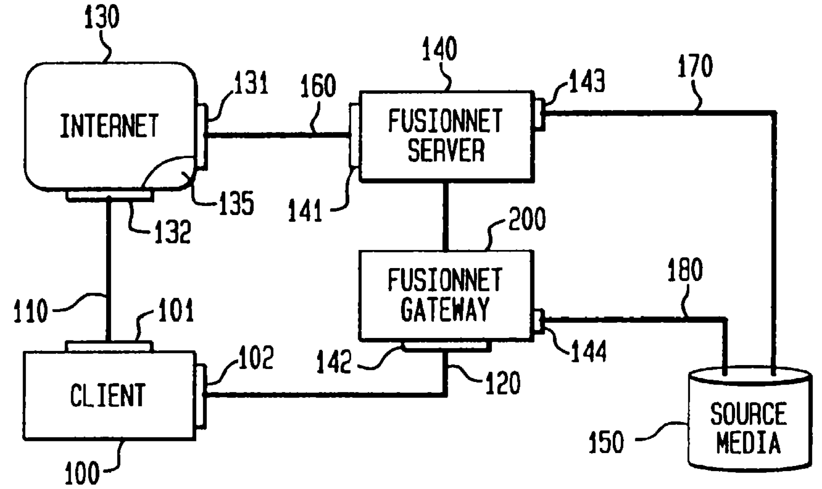
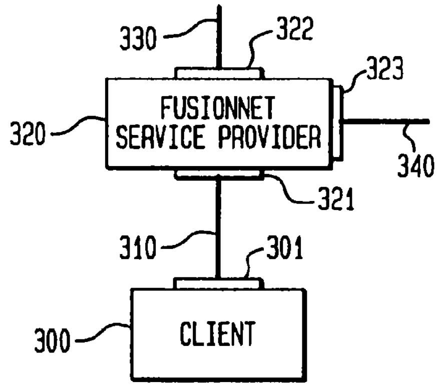


FIG. 3



(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 843 276 A1

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
**20.05.1998 Bulletin 1998/21**

(51) Int. Cl.$^6$: **G06K 9/20**, G06F 17/21

(21) Application number: **97304451.4**

(22) Date of filing: **24.06.1997**

(84) Designated Contracting States:
**AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE**
Designated Extension States:
**AL LT LV RO SI**

(30) Priority: **18.11.1996 US 751676**

(71) Applicant:
**Canon Information Systems, Inc.**
**Costa Mesa, CA 92626 (US)**

(72) Inventors:
- **Tyan, Ching-Yu**
  **Irvine, California 92612 (US)**
- **Huang, Hung Khei**
  **Lake Forest, California 92630 (US)**
- **Niki, Toru**
  **Irvine, California 92614 (US)**

(74) Representative:
**Beresford, Keith Denis Lewis et al**
**BERESFORD & Co.**
**2-5 Warwick Court**
**High Holborn**
**London WC1R 5DJ (GB)**

(54) **HTML generator**

(57) Automatic generation of HTML files based on bitmap image data, which faithfully preserves layout information of an original document from which the bitmap data was obtained. Generally, multi-column document layouts result in automatic generation of HTML files that use HTML "table tags" to display each of the different columns. More particularly, a bitmap image is obtained such as by scanning or retrieval of a pre-existing image, and the bitmap image is segmented into blocks. The location of each block is determined, each block is analyzed in preparation for insertion of appropriate data into an HTML file, and layout analysis is performed to identify layout relationships between the blocks based on the relative locations of the blocks in the bitmap image. Based on the layout relationships, a block type is determined for each block, column span and row span data for each block is determined, blocks are re-ordered if needed, and an HTML file is generated in which blocks are tagged as data elements in a row of an HTML "table tag" based on block type and based on column and row span information for the block.

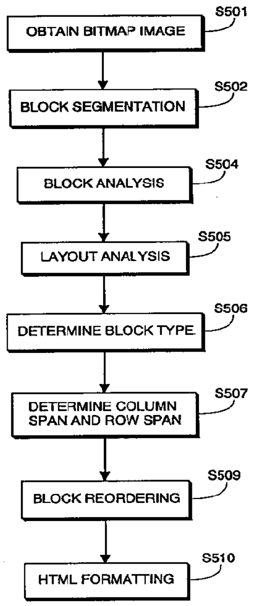


**FIG. 5**

EP 0 843 276 A1

Printed by Xerox (UK) Business Services
2.16.3/3.4

## Description

The present invention concerns generating an HTML file based on an input bitmap image, and is particularly directed to automatic generation of an HTML file, based on a scanned-in document image, with the HTML file in turn being used to generate a Web page that accurately reproduces the layout of the original input bitmap image.

In recent years, the popularity of the internet has grown dramatically. One reason for such growth has been the widespread adoption of HTML (HyperText Markup Language), which is a language for describing document appearance, document layout and hyperlink specifications. It defines the syntax that describe the structure and the content of a document including text, images, and other supported media. The language also provides connections among documents and other Internet resources through hypertext links and other hyperlinks. Using HTML, a Web page can be created which contains, in addition to bitmap images, graphic images, and text of various styles and sizes, hyperlinks which permit a viewer of the Web page to easily jump to another point within the page or to a completely different Web page, even one that is provided by a different server.

Once an HTML file is made available on the World Wide Web via a server, any client connected to the World Wide Web can access the page merely by typing the page address in the appropriate field of his browser. After the address has been entered, the browser requests the server to send the HTML file, which can contain text, references to graphic and bitmap image files, and formatting and hyperlink information for the entire page. Upon receipt of the HTML file, the browser automatically requests the graphic and bitmap image files referenced in the HTML file from the identified source.

To display the HTML file and the downloaded image files, the browser relies on HTML commands embedded in the HTML file. These commands are referred to as "tags". The tags indicate features or elements of a page and cause the browser to perform various functions, such as a particular type of formatting. HTML tags can be identified in HTML files by their syntax. That is, the tags are surrounded by left and right angle brackets, such as "⟨P⟩". In this case, "⟨" indicates the start of the HTML tag, "P" is the tag itself (here a tag indicating a new text paragraph), and "⟩" indicates the end of the tag. Often, tags come in pairs so as to indicate the start and end of a special function. The beginning tag initiates a feature (such as heading, bold, and so on), and the ending tag turns it off. Ending tags typically consist of the initiating tag name preceded by a forward slash (/). For example, ⟨strong⟩and ⟨/strong⟩, surrounding text, will display the surrounded text more strongly that other text. Any additional words in a tag are attributes, sometimes with an associated value after an equal sign (=), which further define or modify the tag's actions.

HTML 3.0 is presently the de-facto World Wide Web standard that defines permissible tags and nesting of tags. Approximately 100 different tags are permitted and defined.

Because of the complexity of HTML 3.0, as well as its cumbersome usage requirements, considerable effort is expended by the Web designer when authoring visually appealing and useful Web pages. For example, assume that an organization had good existing written marketing materials which it wanted to reproduce identically on a Web page. Even this seemingly simple task has typically required that a specialist spend a significant amount of time authoring HTML instructions by hand in an attempt to reproduce the layout and appearance of the written materials.

Several systems have been proposed that would automate this job of authoring HTML files from written documents. Xerox Text Bridge Pro and Caere Omni Page Pro are examples of systems which scan in written documents and generate HTML files based on the scanned-in document image. Where these systems fail is in producing HTML files that accurately represent the layout, tables and images of the original written document. In particular, a major problem has been automatically generating HTML instructions for the case where the written document is arranged in columns, or, more generally, when regions in the original document are horizontally adjacent. The term "horizontally adjacent," when used with respect to two image blocks, means a situation where the vertical extent of the two blocks overlap, or, equivalently, where a horizontal line can be found which will intersect both blocks. Similarly, the term "vertically adjacent," when used with respect to two image blocks, means a situation where the horizontal extent of the two blocks overlap, or, equivalently, where a vertical line can be found which will intersect both blocks.

A typical example of the problems associated with such systems is illustrated by reference to Figures 1 and 2. Figure 1 depicts an original printed document 10 to be converted into HTML format. As shown in Figure 1, the original document, has, among other features: title 1 in the upper left corner, subtitle 2 in the upper right corner, text columns 4, 5 and 6, picture 7 in the lower left corner, and footer 9 in the lower right corner.

Figure 2 illustrates how a Web page 20 would be displayed on display 23 by a Web browser based on the HTML file generated by an existing system for converting bitmap images into HTML. Elements in Figure 2 corresponding to those in Figure 1 are numbered similarly to those in Figure 1. Thus, after processing by the existing system, title 1 is reproduced as title 11. However, subtitle 2, rather than being reproduced in the upper right corner, is instead reproduced as element 12 in the upper left corner, just below title 11. Similarly, the entire text column structure of the original document has been eliminated, and picture 17, rather than being in the lower left corner, occupies the entire width of the page and is interposed between lines of text. Finally, footer 19 is reproduced in the lower left corner, instead of the lower right corner.

The above comparison shows the complete failure of commercially available systems to capture many layout and stylistic elements of the original document. Other known available systems also too frequently miss important layout features. Accordingly, the problems caused by the complex and cumbersome nature of HTML are not adequately addressed by commercially available systems.

It is therefore an objective of the present invention to address the foregoing problems by providing a means by which an HTML file can be automatically generated based on a bitmap image, which HTML file can be used to display a Web page which preserves layout information of the original bitmap image. In particular, according to the invention, multi-column layouts are faithfully preserved by automatic generation of HTML files that use HTML "table tags" to display columns.

According to one aspect of the invention, an HTML file is generated based on a bitmap image by obtaining two horizontally adjacent blocks in separate vertical columns of the bitmap image, and then generating an HTML file in which the blocks are placed inside table cells by being tagged as data elements in a row of an HTML tagged table.

According to another aspect of the invention, an HTML file is generated based on a bitmap image. The bitmap image is first segmented to obtain image blocks. It is then determined where in the bitmap image each of the blocks is located. Next, positional relationships between the blocks are identified based on their relative locations. Finally, an HTML file is generated. In the HTML file, the blocks are tagged as data elements in an HTML tagged table, the tags being determined in accordance with the identified positional relationships.

This brief summary has been provided so that the nature of the invention may be understood quickly. A more complete understanding of the invention can be obtained by reference to the following detailed description of the preferred embodiments thereof in connection with the attached drawings.

## BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 depicts an original printed document.

Figure 2 depicts an HTML page displayed by a Web browser based on an HTML file generated by a commercially available system.

Figure 3 is a perspective view showing the outward appearance of a workstation.

Figure 4 is a block diagram of the workstation depicted in Figure 3.

Figure 5 is a flow diagram illustrating a method for generating an HTML file based on a bitmap image.

Figure 6 is a diagram illustrating output of block selection analysis performed on the printed document of Figure 1.

Figure 7 is a view for explaining data output by block segmentation analysis. Figure 8 is a view for explaining output of block analysis. Figure 9 is a flow diagram for explaining layout analysis and generation of layout data based on block analysis.

Figures 10A-1 through 10A-14 are views for explaining iterative processing of layout analysis on the blocks depicted in Figure 6. Figures 10B-1 through 10B-14 are views for explaining generation of layout data for each corresponding iteration of Figures 10A-1 through 10A-14.

Figure 11A is a flow diagram illustrating identification of block type according to the present invention.

Figure 11B illustrates two cases where block reordering is required.

Figure 12 is a view for explaining output of block type determination based on layout data.

Figures 13A-13D are flow diagrams illustrating the generation of HTML instructions for each block in a bitmap image.

Figure 14 is a view showing an HTML page displayed by a Web browser based on an HTML document automatically generated in accordance with the invention.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figure 3 is a view showing the outward appearance of a computer workstation for implementing the present invention. Shown in Figure 3 is a workstation 30, such as a Macintosh or an IBM PC or PC-compatible computer having a windowing environment, such as Microsoft Windows. Provided with the workstation 30 is a display screen 31, such as a color monitor, a keyboard 32 for entering user commands, and a pointing device 34, such as a mouse, for pointing to and for manipulating objects displayed on the screen 31. The workstation 30 includes a mass storage device such as a computer disk 35 for storing data files. A scanner 37 is provided for generating a bitmap image from an input printed document.

Figure 4 is a detailed block diagram showing the internal construction of the workstation 30. As shown in Figure 4, the workstation 30 includes a central processing unit (CPU) 41 interfaced with a computer bus 42. Also interfaced with the computer bus 42 is a scanner interface 44, a display interface 45, a keyboard interface 46, a mouse interface 47, a main memory 49, and a fixed disk 35. Disk 35 stores a windowing operating system, such as Microsoft Windows, various Windows applications, an HTML Conversion Program for generating an HTML file based on a bitmap image according to the present invention, previously scanned bitmap images, previously generated HTML files, and a Web browser for displaying HTML files. The main memory 49 interfaces with the computer bus 42 so as to provide random access memory storage for use by the CPU 41 while executing stored process steps such as those of the HTML Conversion Program. More specifically, the CPU 41 loads those process steps from the disk 35 into the main memory 49 and executes those stored process steps out of the main memory 49.

[Generation of HTML]

Figure 5 is a flow diagram comprised of process steps stored on disk 35 for generating an HTML file based on a bitmap image according to the present invention. Briefly, according to Figure 5, a bitmap image is obtained, such as by scanning or retrieval, and segmented into blocks. The location of each block is determined, each block is analyzed in preparation for insertion into an HTML file, and layout analysis is performed to identify layout relationships between the blocks based on the relative locations in the bitmap image. Based on the layout relationships, a block type is determined for each block, column span and row span for each block is determined, blocks are reordered if needed, and an HTML file is generated based on block type and column and row span information for the blocks.

[Obtaining A Bitmap Image]

More particularly, in step S501 a bitmap image is obtained. Typically, this is accomplished by scanning a printed document using scanner 37. However, the bitmap image may also be retrieved from a file stored on disk 35 or in any other manner.

[Block Identification and Segmentation]

Step S502 comprises process steps by which blocks in the image data are automatically detected and their locations with respect to the image are automatically identified. A "block" is a logically-related group of image data, such as a region of consecutive paragraphs of text image data, a region comprising title image data, or regions comprising non-text image data such as graphical image data, line drawing image data, picture data, or tabular image data. Preferably, in addition to automatic detection and identification of blocks and their locations, block segmentation according to step S502 also generates hierarchical tree data by which the logical relationship of each block with respect to other blocks is identified. For example, text image data is often found within non-text image data, such as in the case of text labels within a line-art graphic. In such a situation, the text labels would be identified in the hierarchical tree as a child node of the parent block that contains the graphic. Hierarchical Tree data such as this is useful in subsequent processing steps so as to determine the relationship of each of the blocks.

Suitable block segmentation techniques are described in the literature, and also in co-pending patent application serial number 07/873,012 filed April 24, 1992 and entitled "Method And Apparatus For Character Recognition"; and co-pending patent application serial number 08/338,781 filed November 10, 1994 entitled "Page Analysis System." The contents of these two patent applications are incorporated herein by reference as if set forth in full. Generally speaking, such block segmentation techniques work by analyzing individual pixel data within the image so as to identify connected components of the image, analyzing the connected components so as to determine whether a connected component is a text connected component or a non-text connected component, and thereafter grouping the connected components into blocks. The text and non-text blocks are then analyzed so as to determine attributes of each block. For example, in the case of text blocks, the blocks are analyzed to determine whether they are titles, plain text, captions to figures, and the like. In the case of non-text blocks, the blocks are analyzed to determine whether they are line-art images, half tone images, color images, tabularly arranged data, and the like. At the same time, a hierarchical tree is generated showing, through parent-child nodal relationships, the relationships of the text and non-text blocks within the image data.

Figure 6 is an illustration of the results of block segmentation according to step S502. As shown in Figure 6, and with respect to original image data illustrated in Figure 1, a first block 51 (BLK1) corresponds to the first line of title data of Figure 1. Likewise, a second block 52 (BLK2) corresponds to the second line of title-data of Figure 1. A fifth block 55 (BLK5) corresponds to the first line of the subtitle. An eighth block 58 (BLK8) corresponds to the large "S" in the article's caption, whereas a ninth block 59 (BLK9) corresponds to the remainder of the text in the article's caption. Blocks 53 (BLK10, BLK11, BLK14 and BLK15) correspond to text in the article itself. Block 54 (BLK12) corresponds to the half-tone image in the article whereas block 56 (BLK13) corresponds to the textual caption for that image. Other blocks shown in Figure 6 have meanings that can readily be appreciated from the foregoing.

Figure 7 is a view for explaining data that is stored for each block, including attribute data and hierarchical tree structure data. Specifically, and as shown in Figure 7, for each block, block data 150 is stored, and includes at least block number identification 151, coordinate information 152 (such as upper-left and lower-right coordinates of a circumscribing rectangle for the block), attribute data 153 which stores whether the block is text or non-text, together with any other attributes determined during block segmentation, and hierarchical tree structure 154 which includes pointers to parent and child nodes for the block.

Summarizing, at the end of block segmentation in step S502, image data has been analyzed so as to automatically detect blocks of logically-related image data and to automatically identify locations of those blocks. In addition, hierarchical tree structure showing the logical relation from one block to another has been stored for each such block.

[Block Analysis]

Block analysis in step S504 comprises process steps by which each of the blocks identified in step S502 is analyzed so as to determine whether it is a block that affects document layout, and, if so, to extract needed information for HTML generation.

Specifically, for each block identified in step S502, block analysis according to step S504 determines whether the block affects document layout by inspecting the hierarchical tree to determine whether the block is a text block that is not a child of a non-text block. The purpose of this test is to eliminate from further processing any text block that is within a non-text block, inasmuch as such a text block does not affect document layout. For example, a textual label within a line-art graphic does not affect document layout; rather, it is the histogram that affects document layout. Accordingly, by inspecting the hierarchical tree generated in step S502, it is possible to distinguish between blocks that affect document layout from blocks that do not affect document layout.

On the other hand, non-text blocks, such as line-art drawings, half-tone images and tabular data, presumptively affect document layout and are therefore targeted for further analysis in step S504.

In the context of the Figure 6 example, it will be appreciated that all blocks shown there affect document layout. Accordingly, no blocks are eliminated from further processing. This would not be the case, for example, in the situation where a document image included textual data within an image or a table. In those situations, the block containing such textual data would be eliminated from further processing, since such a text block does not affect document layout but rather the image block affects document layout.

All blocks so identified as affecting document layout are thereafter processed in accordance with image attributes 153 in preparation for HTML generation. Specifically, if image attribute 153 identifies the block as non-text data such as a half-tone image, then block analysis in step S504 generates a suitable image file for the image. Preferably, the image file is stored in a commonly used Web browser format, such as .GIF or .JPEG format. On the other hand, for text-type blocks, the block is subjected to optical character recognition analysis so as to obtain a file of computer readable character codes such as ASCII character codes for subsequent use in HTML generation, and also to determine the number of text lines in each text block.

Figure 8 is a view for explaining the output of block analysis in step S504. Particularly, as shown in Figure 8, for each block that affects document layout, data 160 is stored containing at least block number 161, coordinate data 162, attribute data 163 which is the same attribute data as attribute data 153 of Figure 7, hierarchical tree data 164 which is the same as hierarchical tree data 154 of Figure 7 with the exception that child node information may be omitted, and block analysis results 165. Block analysis results 165 may include, for example, an image file name in the case where attribute data 163 indicates that the block is non-text data, or may include optical character recognition results and number of text lines in each text block in the case where attribute data 163 indicates that the block is text-type data.

Summarizing, block analysis according to step S504 results in identification of those blocks that affect document layout, and for each of those blocks, an image file or optical character recognition results.

[Layout Analysis]

Step S505 of Figure 5 comprises process steps by which the blocks identified in step S504 are analyzed so as to provide layout data for the document. These process steps are illustrated in Figure 9. Generally speaking, the process steps of Figure 9 test each possible combination of pairs of vertically arranged blocks or groupings of blocks to determine whether the smallest rectangle which circumscribes the pair of blocks overlaps onto any other blocks, and if the combination does not overlap, the pair is combined into a single grouping. Likewise, each possible combination of pairs of horizontally arranged blocks or groupings of blocks are tested to determine whether the combination overlaps onto any other blocks, and if there is no overlap the pair is combined into a horizontal grouping. Vertical combinations are tested first since most english language documents are arranged into vertical layouts; accordingly, in situations where it can be expected for documents to be arranged in horizontal layouts, such as in some Japanese language documents, horizontal groupings should be tested first. The process of testing vertical groupings and then horizontal groupings is repeated until there are no further groupings that can be made and the entire document has been combined into a single grouping, whereupon the layout data is complete.

Figure 9 will now be explained in more detail, in connection with the illustrative example of Figure 10 which is based on the document of Figure 1. Thus, in step S901, a combination of two vertically arranged "groupings" is first tested.

Groupings are either individual blocks or blocks that have previously been combined into a single grouping. "Vertically-arranged" groupings are groupings that differ in location vertically; thus, groupings that do not differ in the vertical direction, as measured based on coordinate data 162, are not tested in this step S901.

An illustrative example of a first such combination is shown in Figure 10A-1. As shown in Figure 10A-1, blocks BLK1 and BLK2 are vertically-arranged in the sense that they differ in vertical locations. A dotted line 171 represents a circumscribing rectangle that encloses these two blocks. It is this combination as represented by the circumscribing rectangle that is tested in step S902. For illustrative purposes, a small gap is shown between the rectangle and the enclosed blocks. However, it should be noted that the actual grouping border will generally coincide with the block borders.

Specifically, step S902 tests to determine whether the combination of vertically-arranged groupings overlaps onto any other grouping. A combination would overlap onto another grouping if, when making the combination, the combination would include all or any part of any other block or grouping. If the combination does not overlap onto any other grouping, then flow branches to step S903 in which the groupings are combined vertically into a vertical grouping. On the other hand, if the combination does overlap onto any other grouping, then flow advances directly to step S904 which determines whether there are any more vertical combinations that need testing. If more vertical combinations need testing, then flow returns to step S901 until all vertically-arranged groupings are suitably tested.

Thus, reverting to Figure 10A-1, since circumscribing rectangle 171 does not overlap onto any other groupings (step S902), blocks BLK1 and BLK2 are combined vertically into a single vertical grouping (step S903). This is illustrated graphically in Figure 10B-1 which shows layout data depicting the combination of blocks BLK1 and BLK2 into a single vertical grouping.

Referring now to Figure 10A-2, since more vertical combinations exist (step S904), those vertical combinations are tested. Thus, in Figure 10A-2, a circumscribing rectangle 172 represents the combination of the vertical grouping that is comprised of blocks BLK1 and BLK2 as well as vertically-arranged block BLK3. Since the combination does not overlap onto any other grouping, the combination is combined vertically into a new grouping as illustrated in the layout data of Figure 10B-2.

In Figure 10A-3, block BLK4 is not tested since, based on coordinate data 163, it is not a vertically-arranged grouping. Rather, the next block that is tested is block BLK5 since it is a vertically-arranged grouping. The circumscribing rectangle for the combination is illustrated with dotted line 173. As can be appreciated, the combination includes block BLK4. Thus, the combination overlaps onto another grouping, and a vertical combination is not made. Thus, as illustrated in Figure 10B-3, the layout remains the same.

As can readily be appreciated, any other combination of the vertical grouping of blocks BLK1, BLK2 and BLK3, when combined with any other block of the document, will result in an overlap onto some other grouping. Accordingly, when each test is made in step S901, step S902 will determine that the combination overlaps onto some other grouping, and no further vertical combinations will be made with this grouping.

Referring now to Figure 10A-4, combinations of vertical groupings with the next permissible block or grouping, here block BLK4, are now made. Circumscribing rectangle 174, which shows a combination of block BLK4 with block BLK5, clearly shows that an overlap exists with block BLK3. Thus, since the combination overlaps onto another grouping, a vertical combination is not made and the groupings remain as shown in the layout data of Figure 10B-4. Likewise, it can be understood that combinations with any other block with BLK4 will result in an overlap, and no further vertical combinations are made.

Referring now to Figure 10A-5, combinations of vertically-arranged groupings with block BLK5 are tested. Dotted line 175 represents a circumscribing rectangle for the combination of blocks BLK5 and BLK6. Since no overlaps onto any other grouping exist, blocks BLK5 and BLK6 are combined vertically as illustrated in the layout data of Figure 10B-5.

Likewise, in Figure 10A-6, the grouping of blocks BLK5 and BLK6 are then tested in combination with block BLK7, as illustrated by circumscribing rectangle 176. Since the combination does not overlap onto any other grouping, the groupings are combined vertically as illustrated in the layout data of Figure 10B-6.

The process of testing vertical combinations of groupings continues until there are no more permissible vertical combinations that can be tested. In connection with this testing, it will be appreciated that the only further groupings that can be combined are blocks BLK12 and BLK13 and blocks BLK14 and BLK15. This situation is illustrated in Figures 10A-7 and 10B-7.

Since no further vertical combinations are available for testing, flow then advances to step S905, in which similar combinations and testing of combinations are made in the horizontal direction. Thus, step S905 tests combinations of two horizontally-arranged groupings, step S906 determines whether the combination overlaps onto any other groupings, step S907 combines groupings horizontally in a case where step S906 determines that the combination does not overlap them to any other grouping, and step S908 determines whether there are any more horizontal combinations that need testing.

Specifically, reverting to Figure 10A-8, the grouping of blocks BLK1, BLK2 and BLK3 is tested in combination with horizontally-arranged block BLK4. As seen in Figure 10A-8, circumscribing rectangle 177 does not overlap onto any other groupings and the combination is therefore made. This is illustrated graphically in the layout data of Figure 10B-8.

No other tests are made with respect to the new horizontal grouping of blocks BLK1 through BLK4 since, as will be appreciated, no other blocks or grouping of blocks is horizontally arranged. Likewise, no block or grouping of blocks is horizontally arranged with the vertical grouping of blocks BLK5 through BLK7, and no testing for these blocks is made.

Accordingly, the next block tested for combinations of horizontally arranged grouping is block BLK8 and the first such combination is with block BLK9. Inasmuch as circumscribing rectangle 178 does not overlap onto any other block, the horizontal combination is made as illustrated in the layout data of Figure 10B-9.

The new grouping of blocks BLK8 and BLK9 is not tested with block BLK10 because block BLK10 is not horizontally arranged. Block BLK11 is horizontally arranged, but as shown in Figure 10A-10, horizontal combination of the two overlaps onto BLK10 and accordingly no combination is made. Accordingly, the groupings remain as shown in the layout data of Figure 10B-10.

All other combinations of horizontally-arranged groupings also overlap onto other blocks and/or groupings. Accordingly, when formed in step S905 and when tested in step S906, no further horizontal combinations are made.

Flow then advances to step S909 to determine whether all blocks in the image have been incorporated into a single overall group. So long as blocks remain not grouped into one overall grouping, the process steps of S901 through S908 are iterated until all blocks are contained within one grouping.

Thus, as illustrated with respect to Figure 10A-11, the horizontal grouping containing blocks BLK1 through BLK4 is tested in combination with the vertical grouping containing the blocks BLK4 through BLK6, as illustrated by circumscribing rectangle 179. Since no blocks overlap, a combination is made vertically as illustrated in the layout data of Figure 10B-11.

Likewise, in Figure 10A-12, the horizontal grouping containing blocks BLK8 and BLK9 is tested with respect to block BLK10, as illustrated by circumscribing rectangle 180. Since no blocks are overlapped, a vertical grouping is made as illustrated in the layout data of Figure 10B-12.

In this iteration of steps S901 through S904, no further vertical combinations are overlap-free. Accordingly, no further vertical combinations are made.

Figure 10A-13 illustrates that, with respect to this iteration of horizontal steps S905 through S908, one further horizontal grouping can be made as illustrated at circumscribing rectangle 181. Accordingly, the groupings are as shown at Figure 10A-13 and in the layout data of Figure 10B-13.

Further vertical and horizontal iterations are made of steps S901 through S908, resulting in a final situation that is shown at Figures 10A-14 and 10B-14. It will be appreciated that the document has been arranged into one overall grouping, in this case a vertical grouping, which encompasses the entire document. Layout analysis thereupon concludes with storage of the layout data, together with storage of coordinates that identify circumscribing rectangles for each and every block or grouping of blocks and preferably also storage of width and height of each block. The layout data that is stored for the document of Figure 1 is reproduced below as an illustrative example:

```
VRT: (l, t, r, b, w, h) = (48, 72, 2247, 3167, 2200, 3096)
  VRT: (l, t, r, b, w, h) = (48, 72, 2247, 763, 2200, 692)
    HOZ: (l, t, r, b, w, h) = (52, 72, 2247, 423, 2196, 352)
      VRT: (l, t, r, b, w, h) = (52, 72, 1311, 415, 1260, 344)
        BLK1: (l, t, r, b, w, h) = (52, 72, 1311, 187, 1260, 116)
        BLK2: (l, t, r, b, w, h) = (56, 192, 827, 295, 772, 104)
        BLK3: (l, t, r, b, w, h) = (52, 316, 679, 415, 628, 100)
      BLK4: (l, t, r, b, w, h) = (1520, 380, 2247, 423, 728, 44)
    VRT: (l, t, r, b, w, h) = (48, 524, 2247, 763, 2200, 240)
      BLK5: (l, t, r, b, w, h) = (52, 524, 2247, 587, 2196, 64)
      BLK6: (l, t, r, b, w, h) = (48, 612, 2247, 675, 2200, 64)
      BLK7: (l, t, r, b, w, h) = (52, 696, 2091, 763, 2040, 68)
  HOZ: (l, t, r, b, w, h) = (48, 896, 2247, 3167, 2200, 2272)
    VRT: (l, t, r, b, w, h) = (48, 896, 1487, 3159, 1440, 2264)
      HOZ: (l, t, r, b, w, h) = (48, 896, 1487, 1819, 1440, 924)
        VRT: (l, t, r, b, w, h) = (48, 896, 727, 1815, 680, 920)
          HOZ: (l, t, r, b, w, h) = (56, 896, 551, 1059, 496, 164)
            BLK8: (l, t, r, b, w, h) = (56, 908, 131, 1035, 76, 128)
            BLK9: (l, t, r, b, w, h) = (144, 896, 551, 1059, 408, 164)
          BLK10: (l, t, r, b, w, h) = (48, 1084, 727, 1815, 680, 732)
        BLK11: (l, t, r, b, w, h) = (812, 900, 1487, 1819, 676, 920)
      VRT: (l, t, r, b, w, h) = (48, 1888, 1487, 3159, 1440, 1272)
        BLK12: (l, t, r, b, w, h) = (48, 1888, 1487, 2919, 1440, 1032)
        BLK13: (l, t, r, b, w, h) = (48, 2940, 1427, 3159, 1380, 220)
    VRT: (l, t, r, b, w, h) = (1568, 904, 2247, 3167, 680, 2264)
      BLK14: (l, t, r, b, w, h) = (1572, 904, 2247, 2499, 676, 1596)
      BLK15: (l, t, r, b, w, h) = (1568, 2524, 2247, 3167, 680, 644)
```

wherein "l", "t", "r" and "b" refer to left, top, right and bottom block coordinates, and wherein "w" and "h" refer to the width and height of the corresponding block.

Also in this step the block order is determined according to the order of the blocks in the layout data when read from top to bottom. Thus, in the present example the block order is: BLK1, BLK2, BLK3, BLK4, BLK5, BLK6, BLK7, BLK8, BLK9, BLK10, BLK11, BLK12, BLK13, BLK14, BLK15.

[Determining Block Type]

Step S506 shown in Figure 5 will now be discussed in more detail with reference to Figure 11. Generally according to Figure 11A, a type is determined for each block by successively performing a series of tests and either designating the block type when a test is first passed (the designation depending upon which test was passed) or designating the default block type in the event that none of the tests is passed.

More particularly, in step S1101 it is determined whether the current block is the first block encountered. It is also determined whether the current block is included, somewhere up the tree structure, in a horizontal grouping, and if so, whether it is the first block under its highest-level horizontal grouping. If either of the foregoing are answered affirmatively, then in step S1102 the block is designated "single column". If not, processing proceeds to step S1104.

In step S1104, a four-part test is applied. First, it is determined whether the previous block is a non-text image. Second, a determination is made whether the previous block's left edge is to the left of the current block's left edge. For purposes of describing these tests, the "current" block refers to the block for which a type presently is being determined, and the "previous" block refers to the block immediately preceding the current block in the ordered list identified in step S505.

The third part of the step S1104 test is to determine whether the right edge of the current block is close to the left edge of the next block. In the preferred embodiment, the answer to this part is yes only if the distance between the two is less than 0.01 times the width of the entire image (i.e., including all blocks). Fourth, it is determined whether the sum of the width of the current block and the next block is less than a fixed threshold. In the preferred embodiment, the threshold is equal to 0.6 times the width of the entire image. If each of the foregoing tests is answered affirmatively, then in step S1105 the block is designated as "combined column". If not, processing proceeds to step S1106.

In step S1106, a three-part test is performed. In the first part, it is determined whether the previous block's left edge is left of the current block's left edge. The second part determines whether the previous block is horizontally separated from the current block, that is, whether there is a non-zero horizontal distance between the right edge of the previous block and the left edge of the current block. The third part determines whether the current block is at least a second child of a horizontal grouping. This last part determines whether the current block is part of a horizontal grouping and is listed at least second in the hierarchical chart behind an element with which it is horizontally grouped, for example, (or is the first block in a grouping that is at least the second child of the horizontal grouping). If the foregoing tests are each answered affirmatively, then in step S1107 block is designated as multiple column. Otherwise, processing proceeds to step S1109.

In step S1109, a three-part test is applied. First, it is determined whether the previous block's top edge is vertically higher than the current block's top edge, that is, whether the y-coordinate of the top edge of the previous block is greater than the y-coordinate of the top edge of the current block. Second, it is determined whether the current block and the next block are vertically separated, that is, whether there is a non-zero vertical distance between the bottom of the previous block and the top of the current block. Third, it is determined whether the current block is included, somewhere up the tree structure, in a horizontal grouping, or whether the current block is vertically grouped with the previous block. If all parts of the foregoing test are answered affirmatively, then in step S1110 the block is designated as "joint column". Otherwise, in step S111 the block is designated as "single column".

The application of the processing steps shown in Figure 11 to the present example will now be discussed. Specifically, BLK1 is the first block, and therefore in steps S1101 and S1102 is designated as single column.

BLK2 is not the first block, nor is it the first block under its highest-level horizontal grouping, so the test specified in step S1101 fails, and the test specified is step S1104 is applied. In this case, the first part fails because BLK1 is not a non-text image. The test set forth at step S1106 is applied next. Here, the first part of the test also fails because the left edge of the previous block is not to the left of the current block's left edge. Accordingly, the test described in step S1109 is applied next. Here, BLK1's top edge is higher than BLK2's edge, vertical separation exists between the bottom edge of BLK1 and the top edge of BLK2, and BLK2 is vertically grouped with BLK1. Accordingly, the test in S1109 is satisfied and BLK2 is designated as joint column.

The analysis of BLK3 is identical to that of BLK2 and accordingly BLK3 will be designated as joint column also.

BLK4 is not the first block, nor is it the first block under its highest-level horizontal grouping, and BLK3 is not a non-text image. Accordingly, the tests in steps S1101 and S1104 fail, and the test in step S1106 is applied. Here, BLK3's left edge is left of BLK4's left edge, horizontal separation exists between the right edge of BLK3 and the left edge of BLK4, and by reference to the layout data set forth above, it can be seen that BLK4 is the second child of a horizontal grouping. Specifically, BLK4 is horizontally grouped with the vertical group consisting of blocks 1, 2 and 3. Moreover, BLK4 is shown in the layout data as being below that vertical grouping. Accordingly, the test in step S1006 is satisfied, and BLK4 is designated as multiple column.

BLK5 is not the first block, nor is it the first block under its highest-level horizontal grouping, and BLK4 is not a non-text image. Therefore, processing proceeds to step S1106. The left edge of BLK4 is not left of the left edge of BLK5. Accordingly, this test fails. With respect to the test at step S1109, the top edge of BLK4 is higher than the top edge of BLK5 and vertical separation exists between the two blocks. However, BLK5 is not under a horizontal grouping and is not vertically grouped with BLK4. Therefore, this test fails as well. Accordingly, BLK5 is designated as single column.

The analysis of blocks 6 and 7 is the same as that of BLK2, and accordingly, each such block is designated as joint column.

BLK8 is not the first block, but it is the first block of its highest-level horizontal grouping. Therefore, BLK8 is designated as single column.

BLK9 is not the first block, nor is it the first block under its first horizontal grouping. However, in step S1104, the previous block is a non-text image, the previous block's left edge is left of the current block's left edge, the two blocks are horizontally close, and the sum of the widths of the two blocks is less than 0.60 times the width of the entire image. Accordingly, BLK9 is designated combined column.

Similarly, BLK10, BLK12, BLK13 and BLK15 are designated as joint column, since the test in step S1109 is satisfied. BLK11 and BLK14 are designated as multiple column, since the test in step S1106 is satisfied.

Thus, each of the blocks is designated as set forth below:

BLK1: (SINGLE_COL)
BLK2: (JOINT_COL)
BLK3: (JOINT_COL)
BLK4: (MULTIPLE_COL)
BLK5: (SINGLE_COL)
BLK6: (JOINT_COL)
BLK7: (JOINT_COL)
BLK8: (SINGLE_COL)
BLK9: (COMBINED_COL)
BLK10: (JOINT_COL)
BLK11: (MULTIPLE_COL)

BLK12: (JOINT_COL)
BLK13: (JOINT_COL)
BLK14: (MULTIPLE_COL)
BLK15: (JOINT_COL)

An illustration of these classifications is shown in Figure 12.

[Determining Column Span and Row Span]

Step S507 will now be discussed in detail. Blocks having column span and row span are located by finding a nested horizontal-vertical-horizontal (parent-child-grandchild) structure that does not include any great-grandchild having been designated combined column in the layout analysis generated in step S505. To facilitate the following discussion, each element in the layout analysis will be referenced in relation to the higher level horizontal grouping in the nested horizontal-vertical-horizontal (parent-child-grandchild) genetic hierarchical structure, using tree structure terminology.

Once a nested horizontal-vertical-horizontal hierarchical structure without a combined column great-grandchild is located, the determination of row span and column span proceeds as follows. The column span number is determined to be the number of children of the horizontal grandchild. The block to which the column span number is assigned is selected to be the first non-horizontal grandchild that has a horizontal sibling, or the first great-grandchild whose parent has a horizontal sibling. The row span number is determined to be the number of horizontal grandchildren plus one. The block to which the row span number is assigned is selected to be the first non-vertical child, or the first grandchild that does not have a horizontal sibling. If a grouping is selected to be assigned the column span number or the row span number, the assignment passes to the first block in that grouping.

The above process is best illustrated by way of example. The structural portion of the layout analysis for the document shown in Figure 1 is reproduced below, with groupings and blocks numbered (beginning with number 101) for ease of reference.

```
101  VRT
102    VRT
103      HOZ
104        VRT
105          BLK1
106          BLK2
107          BLK3
108        BLK4
109      VRT
110        BLK5
111        BLK6
112        BLK7
113    HOZ
114      VRT
115        HOZ
116          VRT
117            HOZ
118              BLK8
119              BLK9
120            BLK10
121          BLK11
122        VRT
123          BLK12
124          BLK13
125      VRT
126        BLK14
127        BLK15
```

Since horizontal-vertical-horizontal (parent-child-grandchild) structure numbers 115-117 has a great-grandchild with a combined column designation (BLK9), the only horizontal-vertical-horizontal structure without a combined column great-grandchild in the layout analysis consists of numbers 113-115. The horizontal grandchild is numbered 115, and its only children are 116 and 121. Therefore, the column span number is two. The grandchildren of horizontal grouping 113 are horizontal grouping 115 and vertical grouping 122. Because BLK12 (123) is the first great-grandchild whose parent (122) has a horizontal sibling (115), BLK12 is assigned the column span number.

With respect to row span, there is only one horizontal grandchild, grouping 115. Therefore, the row span number is equal to two. Because BLK14 (126) is the first grandchild that does not have a horizontal sibling, BLK14 is assigned the row span number.

Consequently, BLK12 is assigned a column span number of 2 and BLK14 is assigned a row span number of 2.

[Block Reordering]

Step S509 will now be discussed with reference to Figure 11B. As shown in Figure 11B, two general situations might occur in which the block order obtained in step S505 will have to be rearranged to accommodate the way HTML processes table data (i.e., down row by row, in sequence). In each, a row span block is to the right of and horizontally adjacent to a column span block.

Referring to Figure 11B, the first case exists when blocks are arranged such as blocks 141-144, with the top edge of row span block 144 above the top edge of column span block 143, and step S505 having ordered the group of blocks sequentially as: 141, 142, 143 and 144. In this case, the order of blocks 143 and 144 are switched, resulting in a final order of: 141, 142, 144 and 143.

Similarly, the second case exists when blocks are arranged such as blocks 146-149, with the top edge of row span block 149 above the top edges of blocks 147 and 148, and step S505 having ordered the group of blocks sequentially as: 146, 147, 148 and 149. In this case, the order of block 149 is moved up in position to be placed before block 147, resulting in a final order of: 146, 149, 147 and 148.

The foregoing are merely illustrative of the two generic cases. It should be noted that the number of blocks might vary and the blocks shown might be representative of more than one block. The key elements of this reordering step are: (1) the existence of a row span block to the right of and horizontally adjacent to a column span block, and (2) whether the column span block is on the first or second row that the row span block spans. Moreover, although the foregoing describes the two cases where the row span is two rows, the concept can be easily extended to row spans of more than two.

Applying the foregoing to the present example concerning the document shown in Figure 1, a row break occurs between BLK10 and BLK12 and then another after BLK12. Accordingly, this presents an example of the first case described above, with BLK10 being the equivalent of block 141, BLK11 being the equivalent of block 142, BLK12 (the column span block) and BLK13 together being the equivalent of block 143, and BLK14 (the row span block) and BLK15 together being the equivalent of block 144. Accordingly, BLK14 and BLK15 are moved up in order before BLK12 and BLK13. The final order is thus: BLK1, BLK2, BLK3, BLK4, BLK5, BLK6, BLK7, BLK8, BLK9, BLK10, BLK11, BLK14, BLK15, BLK12, BLK13.

[HTML Formatting]

Step S510, shown in Figure 5, will now be discussed in detail with reference to Figures 13A through 13D. Generally according to Figure 13, first an HTML header is inserted. Then for each block, processing is performed in accordance with the block type determined in step S506, a different set of processing steps being assigned to each block type. Finally, after all blocks have been processed, end-of-file codes are inserted.

More particularly, in step S1301 the following HTML header codes are inserted: "⟨HTML⟩⟨HEAD⟩⟨TITLE⟩title⟨/TITLE⟩⟨/HEAD⟩⟨BODY⟩⟨BODY GCOLOR="#FFFFFF"⟩⟨CENTER⟩", where "title" is the title of the file.

In step S1302, the first block is selected.

In step S1304, it is determined whether the block type for the current block is single column. If it is, then single column processing is performed in step S1305 in accordance with steps shown in Figure 13B, and then processing proceeds to step S1313. If not, processing proceeds to step S1306.

In step S1306, it is determined whether the block type is multiple column. If it is, then multiple column processing is performed in step S1307 in accordance with the steps shown in Figure 13C, and then processing proceeds to step S1313. If not, processing proceeds to step S1309.

In step S1309, it is determined whether the block type is joint column. If it is, then joint column processing is performed in step S1310 in accordance with the steps shown in Figure 13D, and then processing proceeds to step S1313. If not, then the block type must be combined column. Because no additional codes are inserted for combined column, processing simply proceeds immediately to step S1313.

In step S1313 the block is inserted into the HTML file. Typically, either the contents of the block are inserted into the HTML file (for text blocks) or a reference to file containing the relevant data (for graphics and bitmap images) is inserted.

In step S1311, it is determined whether the current block is the last block. If it is not, the next block is selected in step S1312 and processing proceeds to step S1304. If the current block is the last block, then in step S1314 the following end-of-file HTML codes are inserted: "⟨/TABLE⟩⟨/CENTER⟩⟨/BODY⟩⟨/HTML⟩".

Single column processing of step S1305 will now be discussed in more detail with reference to Figure 13B. Specifically, in step S1315 it is determined whether the file is currently in a table. If it is, then in step S1316 the end HTML table code ("⟨/TABLE⟩") is inserted, and processing proceeds to step S1317. Otherwise, processing proceeds to step S1317 without inserting that code.

In step S1317, the following table definition code is inserted into the file: "⟨TABLE BORDER=0 CELLSPACING=0 CELLPADDING=7 WIDTH=w⟩", where BORDER, CELLSPACING, CELLPADDING, and WIDTH are attributes of the ⟨TABLE⟩ tag, and where w is equal to the width of the entire image times 72 and divided by the number of dots per inch in the image.

In step S1319, it is determined whether row span for the block (which was determined in step S507) is greater than one. If it is, the following code is inserted into the HTML file: "⟨TR⟩⟨TD VALIGN=TOP ROWSPAN=n⟩", where n is the row span number determined in step S507.
Otherwise, it is determined in step S1321 whether column span is greater than one. If it is, then in step S1322 the following HTML code is inserted into the HTML file: "⟨TR⟩⟨TD VALIGN=TOP COLSPAN=m⟩", where m is the column span number determined in step S507. Otherwise, in step S1324 the following code is inserted into the HTML file: "⟨TR⟩⟨TD VALIGN=TOP⟩".

Multiple column processing of step S1307 will now be discussed in more detail with reference to Figure 13C. Specifically, in step S1325 it is determined whether row span is greater than one. If it is, then the following HTML code is inserted: "⟨TD VALIGN=TOP ROWSPAN=n⟩". Otherwise, in step S1327 it is determined whether column span for the current block is greater than one. If it is, then in step S1329 the following HTML code is inserted into the HTML file: "⟨TD VALIGN=TOP COLSPAN=m⟩". Otherwise, in step S1330 the following HTML code is inserted: "⟨TD VALIGN=TOP⟩".

Joint column processing of step S1310 will now be discussed in detail with reference to Figure 13D. Specifically, in step S1331 it is determined whether the row span of the current block is greater than one. If it is, then in step S1332 the following HTML code is inserted: "⟨TR⟩⟨TD VALIGN=TOP ROWSPAN=n⟩". Otherwise, it is determined in step S1334 whether the column span for the current block is greater than one. If it is, then in step S1335 the following HTML code is inserted: "⟨TR⟩⟨TD VALIGN=TOP COLSPAN=m⟩". Otherwise, in step S1336 it is determined whether the number of lines in the block (which was determined in step S504 is one. If it is, then in step S1337 the following HTML code is inserted: "⟨BR⟩". Otherwise, in step S1339 the following HTML code is inserted: "⟨P⟩".

Step S510 is now applied to the present example. First, the HTML header is inserted as shown in step S1301. Next, each of blocks 1 through 15 is consecutively processed to construct the body of the HTML file.

Specifically, BLK1 is single column with no row span or column span. Accordingly, the HTML code is specified in steps S1317 and S1324. Immediately after the insertion of that code, BLK1 (or a reference thereto) is inserted into the HTML file. BLK2 is joint column with no column span or row span and containing one line of text. Accordingly, the ⟨BR⟩ code is inserted and immediately followed by BLK2. BLK3 has the same classification as BLK2 and thus is preceded by the same HTML code. BLK4 is multiple column with no row span or column span. Accordingly, the HTML code is specified by step

S1330. Proceeding in this manner, combined column BLK9 is eventually reached. Accordingly, no HTML code precedes BLK9. Proceeding even further, BLK14, which is multiple column with a row span of 2, is reached. Accordingly, the HTML code is specified by step S1326. Similarly, BLK12 is joint column with a column span of 2. Accordingly, the HTML code is specified by step S1335.

When all blocks have been processed, the end of file codes specified in step S1314 are inserted, and the HTML file is complete. The resulting code is set forth below:

```
<HTML>
<HEAD>
<TITLE>Earth1</TITLE>
</HEAD>
<BODY BGCOLOR = "#FFFFFF">
<CENTER>
<TABLE BORDER = 0 CELLSPACING = 0 CELLPADDING = 7 WIDTH = 738>
<TR><TD VALIGN = TOP> BLK1
                  <BR> BLK2
                  <BR> BLK3
    <TD VALIGN = TOP> BLK4
</TABLE>
<TABLE BORDER = 0 CELLSPACING = 0 CELLPADDING = 7 WIDTH = 738>
<TR><TD VALIGN = TOP> BLK5
                  <BR> BLK6
                  <BR> BLK7
</TABLE>
<TABLE BORDER = 0 CELLSPACING = 0 CELLPADDING = 7 WIDTH = 738>
<TR><TD VALIGN = TOP> BLK8
                      BLK9
                   <P> BLK10
    <TD VALIGN = TOP> BLK11
    <TD VALIGN = TOP ROWSPAN = 2> BLK14
                   <P> BLK15
<TR><TD VALIGN = TOP COLSPAN = 2> BLK12
                   <P> BLK13
</TABLE>
</CENTER>
</BODY>
</HTML>
```

Figure 14 illustrates the appearance of a Web page generated in accordance with the above HTML code. As can be seen from Figure 14, title 201 is properly placed in the upper-left corner, and article type 202 is placed very close to the upper-right corner. Columns 204, 205 and 206 have been retained and appear very similar to the layout of the original document. Also, footer 209 is in the lower-right corner and picture 207 is in the lower-left corner underneath columns 204 and 205 and to the left of column 206.

Thus, the HTML file generated can be used to generate a page which closely follows the layout presented in the original document.

The invention has been described with respect to particular illustrative embodiments. It is to be understood that the invention is not limited to the above described embodiments and modifications thereto, and that various changes and modifications may be made by those of ordinary skill in the art without departing from the spirit and scope of the appended claims.

## Claims

1. A method for generating an HTML file based on a bitmap image, comprising:

   obtaining two horizontally adjacent image blocks in separate vertical columns of the bitmap image; and
   generating an HTML file in which the blocks are tagged as data elements in a row of an HTML tagged table.

2. A method according to Claim 1, wherein an HTML code is assigned to each block in accordance with a position of the block relative to other elements within the bitmap image.

3. A method for generating an HTML file based on a bitmap image, comprising:

   a segmenting step of segmenting the bitmap image to obtain image blocks;
   a determining step of determining where in the bitmap image each of the blocks is located;
   an identifying step of identifying positional relationships between the blocks based on their relative locations; and

a generating step of generating an HTML file in which the blocks are tagged as being data elements in an HTML tagged table, the tags being determined in accordance with the identified positional relationships.

**4.** A method according to Claim 3, wherein the identifying step includes forming vertical and horizontal groupings of the blocks.

**5.** A method according to claim 3, wherein the identifying step includes forming a hierarchical structure.

**6.** A method according to Claim 5, wherein the step of forming the hierarchical structure includes forming horizontal and vertical groupings of elements, the elements consisting of previously formed groupings and ungrouped blocks.

**7.** A method according to Claim 6, wherein horizontal and vertical groupings are formed by testing two elements at a time, a new grouping being formed when a rectangle can be formed which encloses the two tested elements and no other elements.

**8.** A method according to Claim 7, wherein the step of forming vertical and horizontal groupings includes alternating between forming all possible vertical groupings and forming all possible horizontal groupings.

**9.** A method according to Claim 7, wherein two identified elements are tested for vertical grouping only if the two identified elements are not horizontally adjacent to each other.

**10.** A method according to Claim 7, wherein two identified elements are tested for horizontal grouping only if the two identified elements are not vertically adjacent to each other.

**11.** A method according to Claim 6, wherein the identifying step includes ordering the blocks based on a sequence in which the vertical and horizontal groupings are formed.

**12.** A method according to Claim 3, wherein the identifying step includes a step of ordering the blocks based on the identified positional relationships.

**13.** A method for generating an HTML file based on bitmap image data, comprising the steps of:

segmenting the bitmap image into blocks;
analyzing layout relationships between the blocks based on the relative locations of the blocks in the bitmap image;
a block-type determination step of determining block type of each block based on the layout relationships obtained in said analyzing step;
a span determination step of determining column span and row span information for each block that spans more than one column and/or row; and
generating an HTML file based on the block type obtained in said block-type determination step and based on the column span and row span information obtained in said span determination step.

**14.** A method according to Claim 13, wherein said analyzing step includes iterative steps of combining blocks into vertical and horizontal groupings.

**15.** A method according to Claim 13, wherein said block-type determination step includes the step of assigning to each block a type selected from the group consisting of: a single block, a combined block, a joint block, and a multiple block.

**16.** A method according to Claim 15, wherein said step of assigning is based on proximity of each block to adjacent blocks and also based on layout information obtained is said analyzing step.

**17.** A method according to Claim 13, wherein analyzing step groups blocks into vertical and horizontal groupings, and wherein said span determination step includes the step of determining column span and row span by counting the number of blocks within vertical groupings and the number of blocks within horizontal groupings.

**18.** A method of generating layout data from block data comprised by horizontally and vertically arranged blocks, the method comprising:

a vertical determination step of determining whether a combination of vertically-arranged blocks overlaps onto any other of the horizontally and vertically arranged blocks;

a vertical combination step of combining the combination of vertically-arranged blocks in a case where said vertical determination step determines that there is no overlap;

a horizontal determination step of determining whether a combination of horizontally-arranged blocks overlaps onto any other of the horizontally and vertically arranged blocks;

a horizontal combination step of combining the combination of horizontally-arranged blocks in a case where said horizontal determination step determines that no overlap exists; and

a generation step of generating layout data based on the combinations made in said vertical combination step and said horizontal combination step.

**19.** A method according to Claim 18, wherein said vertical determination step is performed for all permissible combinations of vertically-arranged blocks before any determinations are made in said horizontal determination step.

**20.** A method according to Claim 19, wherein said block data includes coordinate data for each block, and wherein permissibility of combination is determined based on a comparison of coordinate data for respective blocks.

**21.** A method according to Claim 19, further comprising the step of repeating said vertical determination step after all permissible combinations of horizontally-arranged blocks are tested in said horizontal determination step.

**22.** A method according to Claim 21, wherein said block data includes coordinate data for each block, and wherein permissibility of combination is determined based on a comparison of coordinate data for respective blocks.

**23.** An apparatus for generating an HTML file based on a bitmap image, comprising:

obtaining means which obtains two horizontally adjacent image blocks in separate vertical columns of the bitmap image;

a memory which stores process steps that are executable by a processor; and

a processor which executes the process steps stored in the memory to generate an HTML file in which the blocks obtained by the obtaining means are tagged as data elements in a row of an HTML tagged table.

**24.** An apparatus according to Claim 23, wherein the processor assigns an HTML code to each block in accordance with a position of the block relative to other elements within the bitmap image.

**25.** An apparatus for generating an HTML file based on a bitmap image, comprising:

obtaining means for obtaining the bitmap image;

a memory which stores process steps that are executable by a processor; and

a processor which executes the process steps stored in the memory (1) to segment the bitmap image to obtain image blocks, (2) to determine where in the bitmap image each of the blocks is located, (3) to identify positional relationships between the blocks based on their relative locations, and (4) to generate an HTML file in which the blocks are tagged as being data elements in an HTML tagged table, the tags being determined in accordance with the identified positional relationships.

**26.** An apparatus according to Claim 25, wherein when the processor identifies positional relationships between the blocks, the processor forms vertical and horizontal groupings of the blocks.

**27.** An apparatus according to claim 25, wherein when the processor identifies positional relationships between the blocks, the processor forms a hierarchical structure.

**28.** An apparatus according to Claim 27, wherein the processor forms the hierarchical structure by forming horizontal and vertical groupings of elements, the elements consisting of previously formed groupings and ungrouped blocks.

**29.** An apparatus according to claim 28, wherein the processor forms the horizontal and vertical groupings by testing two elements at a time, a new grouping being formed when a rectangle can be formed which encloses the two tested elements and no other elements.

**30.** An apparatus according to claim 29, wherein the processor forms the vertical and horizontal groupings by alternating between forming all possible vertical groupings and forming all possible horizontal groupings.

**31.** An apparatus according to Claim 28, wherein the processor tests two identified elements for vertical grouping only if the two identified elements are not horizontally adjacent to each other.

**32.** An apparatus according to Claim 28, wherein the processor tests two identified elements for horizontal grouping only if the two identified elements are not vertically adjacent to each other.

**33.** An apparatus according to Claim 27, wherein the processor identifies the positional relationships between the blocks by ordering the blocks based on a sequence in which the vertical and horizontal groupings are formed.

**34.** An apparatus according to Claim 25, wherein the processor identifies the positional relationships between the blocks by ordering the blocks based on the identified positional relationships.

**35.** An apparatus for generating an HTML file based on bitmap image data, comprising:

obtaining means for obtaining the bitmap image data;
a memory for storing the bitmap image data and process steps executable by a processor; and
a processor which executes the process steps stored in the memory (1) to segment the bitmap image into blocks, (2) to analyze layout relationships between the blocks based on the relative locations of the blocks in the bitmap image, (3) to determine a block type of each block based on the layout relationships obtained by the processor, (4) to determine column span and row span information for each block that spans more than one column and/or row, and (5) to generate an HTML file based on the block type determined by the processor and the column span and row span information determined by the processor.

**36.** An apparatus according to Claim 35, wherein the processor analyzes layout relationships between the blocks by combining the blocks into vertical and horizontal groupings.

**37.** An apparatus according to Claim 35, wherein the processor determines a block type of each block by assigning to each block a type selected from the group consisting of: a single block, a combined block, a joint block, and a multiple block.

**38.** An apparatus according to Claim 37, wherein the processor assigns a block type to each block based on proximity of each block to adjacent blocks and also based on layout information obtained by the processor.

**39.** An apparatus according to Claim 35, wherein the layout relationship analysis step executed by the processor includes grouping the blocks into vertical and horizontal groupings, and wherein the processor determines column span and row span information for each block by counting the number of blocks within vertical groupings and the number of blocks within horizontal groupings.

**40.** An apparatus for generating layout data from block data comprised by horizontally and vertically arranged blocks, the apparatus comprising:

an obtaining means for obtaining the block data;
a memory for storing process steps executable by a processor; and
a processor which executes the process steps stored in the processor (1) to determine whether a combination of vertically-arranged blocks overlaps onto any other of the horizontally and vertically arranged blocks, (2) to combine the combination of vertically-arranged blocks in a case where said vertical determination step determines that there is no overlap, (3) to determine whether a combination of horizontally-arranged blocks overlaps onto any other of the horizontally and vertically arranged blocks, (4) to combine the combination of horizontally-arranged blocks in a case where said horizontal determination step determines that no overlap exists, and (5) to generate layout data based on the combinations made in said vertical combination step and said horizontal combination step.

**41.** An apparatus according to Claim 40, wherein the processor determine whether a combination of vertically-arranged blocks overlaps onto any other of the horizontally and vertically arranged blocks for all permissible combinations of vertically-arranged blocks before any determinations are made by the processor regarding overlapping of horizontally-arranged blocks.

**42.** An apparatus according to Claim 41, wherein said block data includes coordinate data for each block, and wherein permissibility of combination is determined based on a comparison of coordinate data for respective blocks.

**43.** An apparatus according to Claim 41, wherein the processor determines whether a combination of vertically-arranged blocks overlaps onto any other of the horizontally and vertically arranged blocks after all permissible combinations of horizontally-arranged blocks are tested.

**44.** An apparatus according to Claim 43, wherein said block data includes coordinate data for each block, and wherein permissibility of combination is determined based on a comparison of coordinate data for respective blocks.

**45.** Computer-executable process steps stored on a computer-readable medium, the computer executable process steps to generate an HTML file based on a bitmap image, the process steps comprising:

an obtaining step to obtain two horizontally adjacent image blocks in separate vertical columns of the bitmap image; and
a generating step to generate an HTML file in which the blocks are tagged as data elements in a row of an HTML tagged table.

**46.** Computer-executable process steps according to Claim 45, wherein an HTML code is assigned to each block in accordance with a position of the block relative to other elements within the bitmap image.

**47.** Computer-executable process steps stored on a computer-readable medium, the computer executable process steps to generate an HTML file based on a bitmap image, the computer-executable process steps comprising:

a segmenting step to segment the bitmap image to obtain image blocks;
a determining step to determine where in the bitmap image each of the blocks is located;
an identifying step to identify positional relationships between the blocks based on their relative locations; and
a generating step to generate an HTML file in which the blocks are tagged as being data elements in an HTML tagged table, the tags being determined in accordance with the identified positional relationships.

**48.** Computer-executable process steps according to Claim 47, wherein the identifying step includes forming vertical and horizontal groupings of the blocks.

**49.** Computer-executable process steps according to claim 37, wherein the identifying step includes forming a hierarchical structure.

**50.** Computer-executable process steps according to Claim 49, wherein the step of forming the hierarchical structure includes forming horizontal and vertical groupings of elements, the elements consisting of previously formed groupings and ungrouped blocks.

**51.** Computer-executable process steps according to Claim 50, wherein horizontal and vertical groupings are formed by testing two elements at a time, a new grouping being formed when a rectangle can be formed which encloses the two tested elements and no other elements.

**52.** Computer-executable process steps according to Claim 51, wherein the step of forming vertical and horizontal groupings includes alternating between forming all possible vertical groupings and forming all possible horizontal groupings.

**53.** Computer-executable process steps according to Claim 51, wherein two identified elements are tested for vertical grouping only if the two identified elements are not horizontally adjacent to each other.

**54.** Computer-executable process steps according to Claim 51, wherein two identified elements are tested for horizontal grouping only if the two identified elements are not vertically adjacent to each other.

**55.** Computer-executable process steps according to Claim 50, wherein the identifying step includes ordering the blocks based on a sequence in which the vertical and horizontal groupings are formed.

**56.** Computer-executable process steps according to Claim 47, wherein the identifying step includes a step to order the blocks based on the identified positional relationships.

**57.** Computer-executable process steps stored on a computer-readable medium, the computer executable process steps to generate an HTML file based on bitmap image data, the computer-executable process steps comprising:

a segmenting step to segment the bitmap image into blocks;
an analyzing step to analyze layout relationships between the blocks based on the relative locations of the blocks in the bitmap image;
a block-type determination step to determine a block type of each block based on the layout relationships obtained in said analyzing step;
a span determination step to determine column span and row span information for each block that spans more than one column and/or row; and
a generating step to generate an HTML file based on the block type obtained in said block-type determination step and based on the column span and row span information obtained in said span determination step.

**58.** Computer-executable process steps according to Claim 57, wherein said analyzing step includes iterative steps to combine blocks into vertical and horizontal groupings.

**59.** Computer-executable process steps according to Claim 57, wherein said block-type determination step includes a step to assign to each block a type selected from the group consisting of: a single block, a combined block, a joint block, and a multiple block.

**60.** Computer-executable process steps according to Claim 59, wherein said step of assigning is based on proximity of each block to adjacent blocks and also based on layout information obtained is said analyzing step.

**61.** Computer-executable process steps according to Claim 57, wherein analyzing step groups blocks into vertical and horizontal groupings, and wherein said span determination step includes a step to determine column span and row span by counting the number of blocks within vertical groupings and the number of blocks within horizontal groupings.

**62.** Computer-executable process steps stored on a computer-readable medium, the computer executable process steps to generate layout data from block data comprised by horizontally and vertically arranged blocks, the computer-executable process steps comprising:

a vertical determination step to determine whether a combination of vertically-arranged blocks overlaps onto any other of the horizontally and vertically arranged blocks;
a vertical combination step to combine the combination of vertically-arranged blocks in a case where said vertical determination step determines that there is no overlap;
a horizontal determination step to determine whether a combination of horizontally-arranged blocks overlaps onto any other of the horizontally and vertically arranged blocks;
a horizontal combination step to combine the combination of horizontally-arranged blocks in a case where said horizontal determination step determines that no overlap exists; and
a generation step to generate layout data based on the combinations made in said vertical combination step and said horizontal combination step.

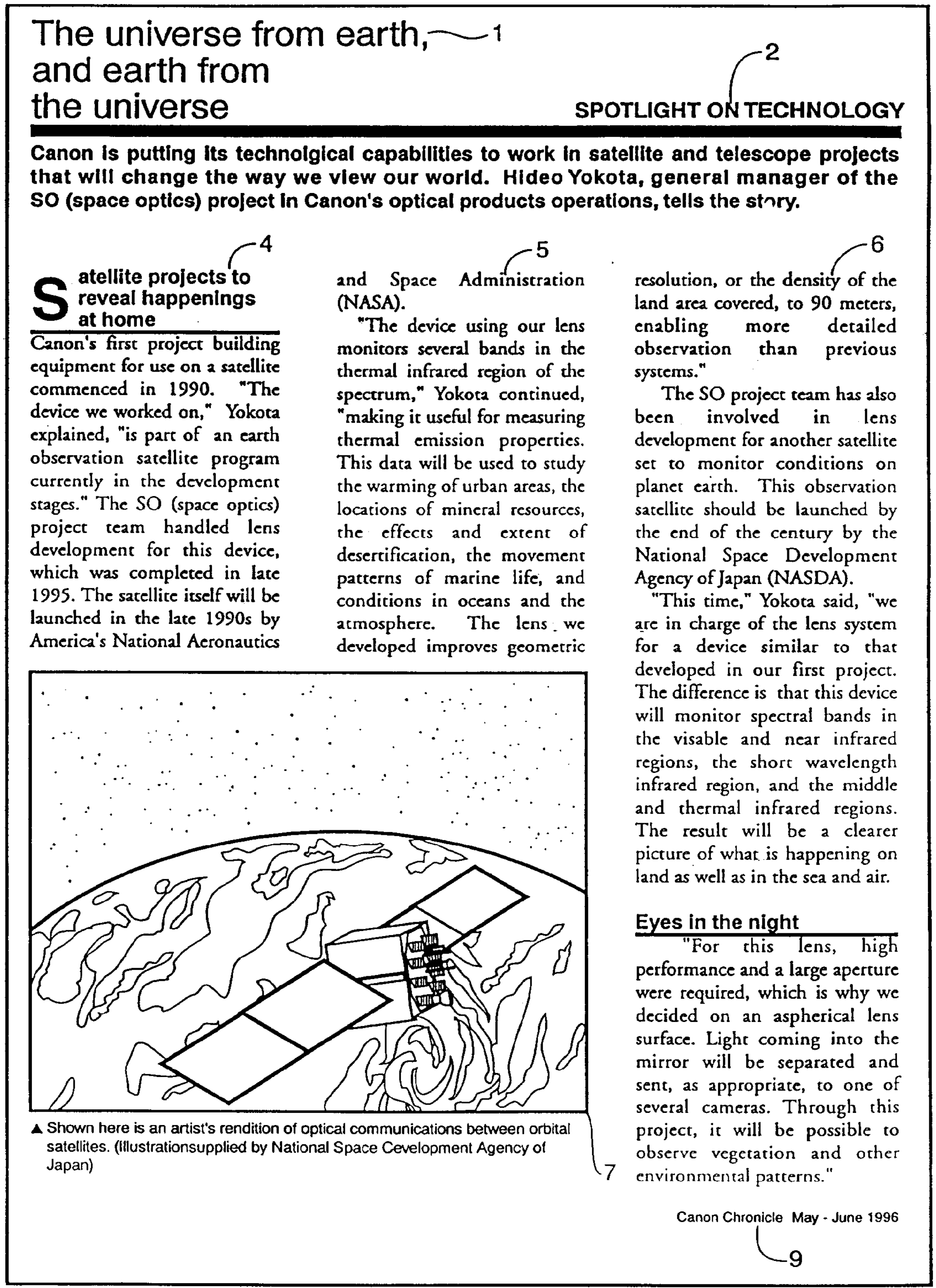
**63.** Computer-executable process steps according to Claim 62, wherein said vertical determination step is performed for all permissible combinations of vertically-arranged blocks before any determinations are made in said horizontal determination step.

**64.** Computer-executable process steps according to Claim 63, wherein said block data includes coordinate data for each block, and wherein permissibility of combination is determined based on a comparison of coordinate data for respective blocks.

**65.** Computer-executable process steps according to Claim 63, further comprising a repeating step to repeat said vertical determination step after all permissible combinations of horizontally-arranged blocks are tested in said horizontal determination step.

**66.** Computer-executable process steps according to Claim 65, wherein said block data includes coordinate data for each block, and wherein permissibility of combination is determined based on a comparison of coordinate data for respective blocks.

**67.** A method and apparatus having the features of any combination of the preceding claims.



# The universe from earth, and earth from the universe

SPOTLIGHT ON TECHNOLOGY

**Canon is putting its technolgical capabilities to work in satellite and telescope projects that will change the way we view our world. Hideo Yokota, general manager of the SO (space optics) project in Canon's optical products operations, tells the story.**

## Satellite projects to reveal happenings at home

Canon's first project building equipment for use on a satellite commenced in 1990. "The device we worked on," Yokota explained, "is part of an earth observation satellite program currently in the development stages." The SO (space optics) project team handled lens development for this device, which was completed in late 1995. The satellite itself will be launched in the late 1990s by America's National Aeronautics and Space Administration (NASA).

"The device using our lens monitors several bands in the thermal infrared region of the spectrum," Yokota continued, "making it useful for measuring thermal emission properties. This data will be used to study the warming of urban areas, the locations of mineral resources, the effects and extent of desertification, the movement patterns of marine life, and conditions in oceans and the atmosphere. The lens we developed improves geometric resolution, or the density of the land area covered, to 90 meters, enabling more detailed observation than previous systems."

The SO project team has also been involved in lens development for another satellite set to monitor conditions on planet earth. This observation satellite should be launched by the end of the century by the National Space Development Agency of Japan (NASDA).

"This time," Yokota said, "we are in charge of the lens system for a device similar to that developed in our first project. The difference is that this device will monitor spectral bands in the visable and near infrared regions, the short wavelength infrared region, and the middle and thermal infrared regions. The result will be a clearer picture of what is happening on land as well as in the sea and air.

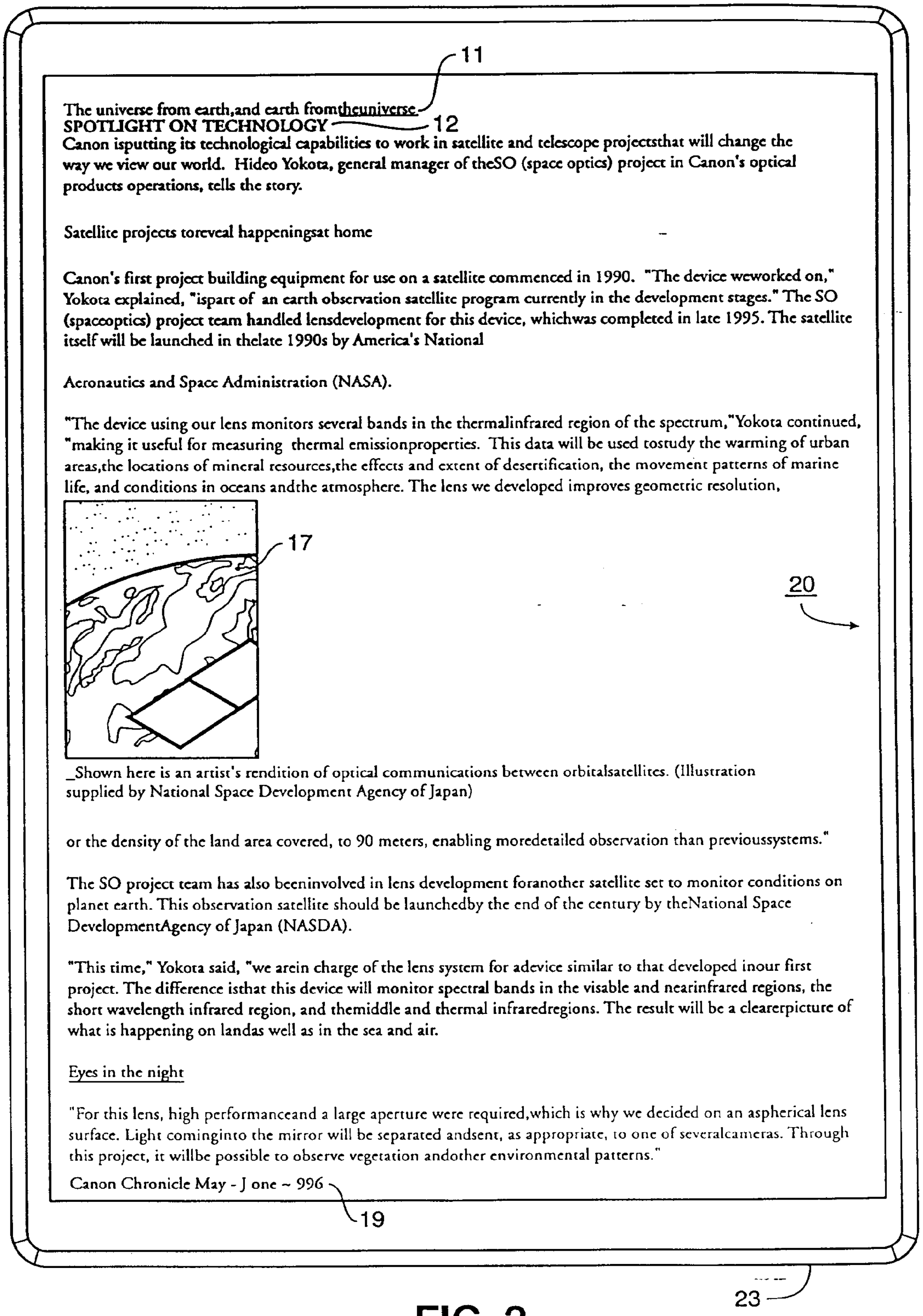
## Eyes in the night

"For this lens, high performance and a large aperture were required, which is why we decided on an aspherical lens surface. Light coming into the mirror will be separated and sent, as appropriate, to one of several cameras. Through this project, it will be possible to observe vegetation and other environmental patterns."

▲ Shown here is an artist's rendition of optical communications between orbital satellites. (Illustrationsupplied by National Space Cevelopment Agency of Japan)

Canon Chronicle May - June 1996

FIG. 1



The universe from earth,and earth fromtheuniverse

SPOTLIGHT ON TECHNOLOGY

Canon isputting its technological capabilities to work in satellite and telescope projectsthat will change the way we view our world. Hideo Yokota, general manager of theSO (space optics) project in Canon's optical products operations, tells the story.

Satellite projects toreveal happeningsat home

Canon's first project building equipment for use on a satellite commenced in 1990. "The device weworked on," Yokota explained, "ispart of an earth observation satellite program currently in the development stages." The SO (spaceoptics) project team handled lensdevelopment for this device, whichwas completed in late 1995. The satellite itself will be launched in thelate 1990s by America's National

Aeronautics and Space Administration (NASA).

"The device using our lens monitors several bands in the thermalinfrared region of the spectrum,"Yokota continued, "making it useful for measuring thermal emissionproperties. This data will be used tostudy the warming of urban areas,the locations of mineral resources,the effects and extent of desertification, the movement patterns of marine life, and conditions in oceans andthe atmosphere. The lens we developed improves geometric resolution,

_Shown here is an artist's rendition of optical communications between orbitalsatellites. (Illustration supplied by National Space Development Agency of Japan)

or the density of the land area covered, to 90 meters, enabling moredetailed observation than previoussystems."

The SO project team has also beeninvolved in lens development foranother satellite set to monitor conditions on planet earth. This observation satellite should be launchedby the end of the century by theNational Space DevelopmentAgency of Japan (NASDA).

"This time," Yokota said, "we arein charge of the lens system for adevice similar to that developed inour first project. The difference isthat this device will monitor spectral bands in the visable and nearinfrared regions, the short wavelength infrared region, and themiddle and thermal infraredregions. The result will be a clearerpicture of what is happening on landas well as in the sea and air.

Eyes in the night

"For this lens, high performanceand a large aperture were required,which is why we decided on an aspherical lens surface. Light cominginto the mirror will be separated andsent, as appropriate, to one of severalcameras. Through this project, it willbe possible to observe vegetation andother environmental patterns."

Canon Chronicle May - J one ~ 996

FIG. 2

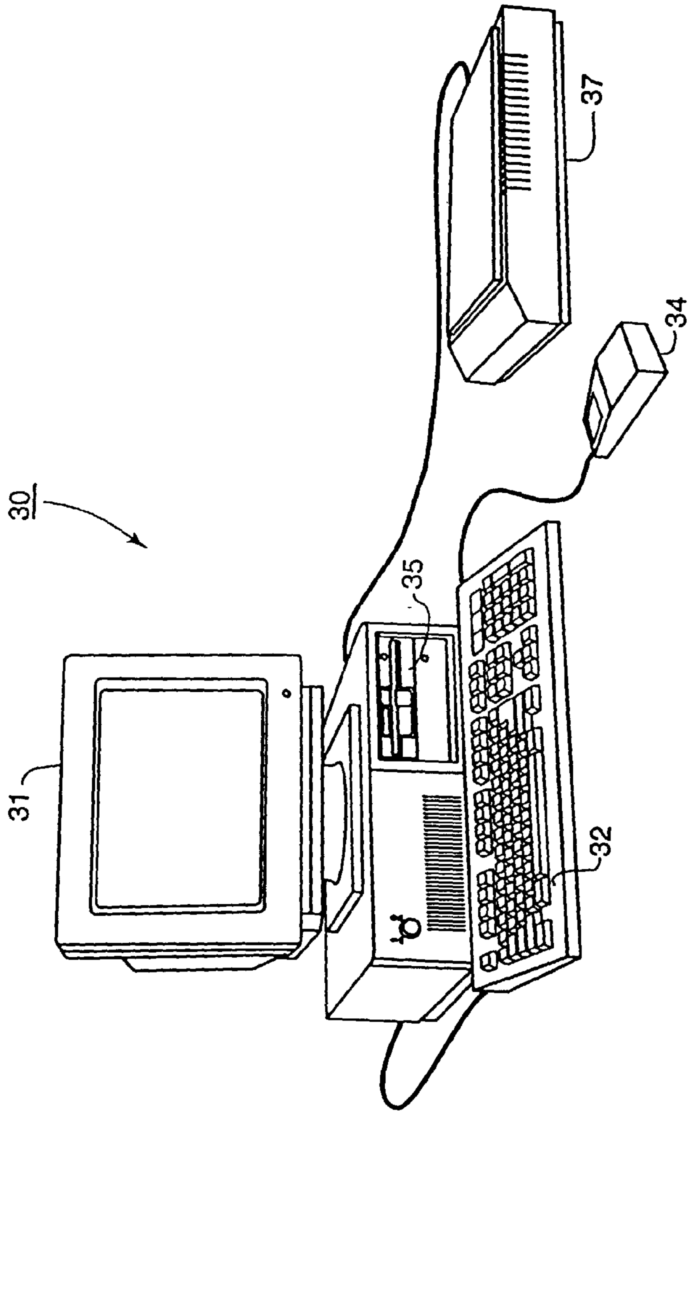


FIG. 3

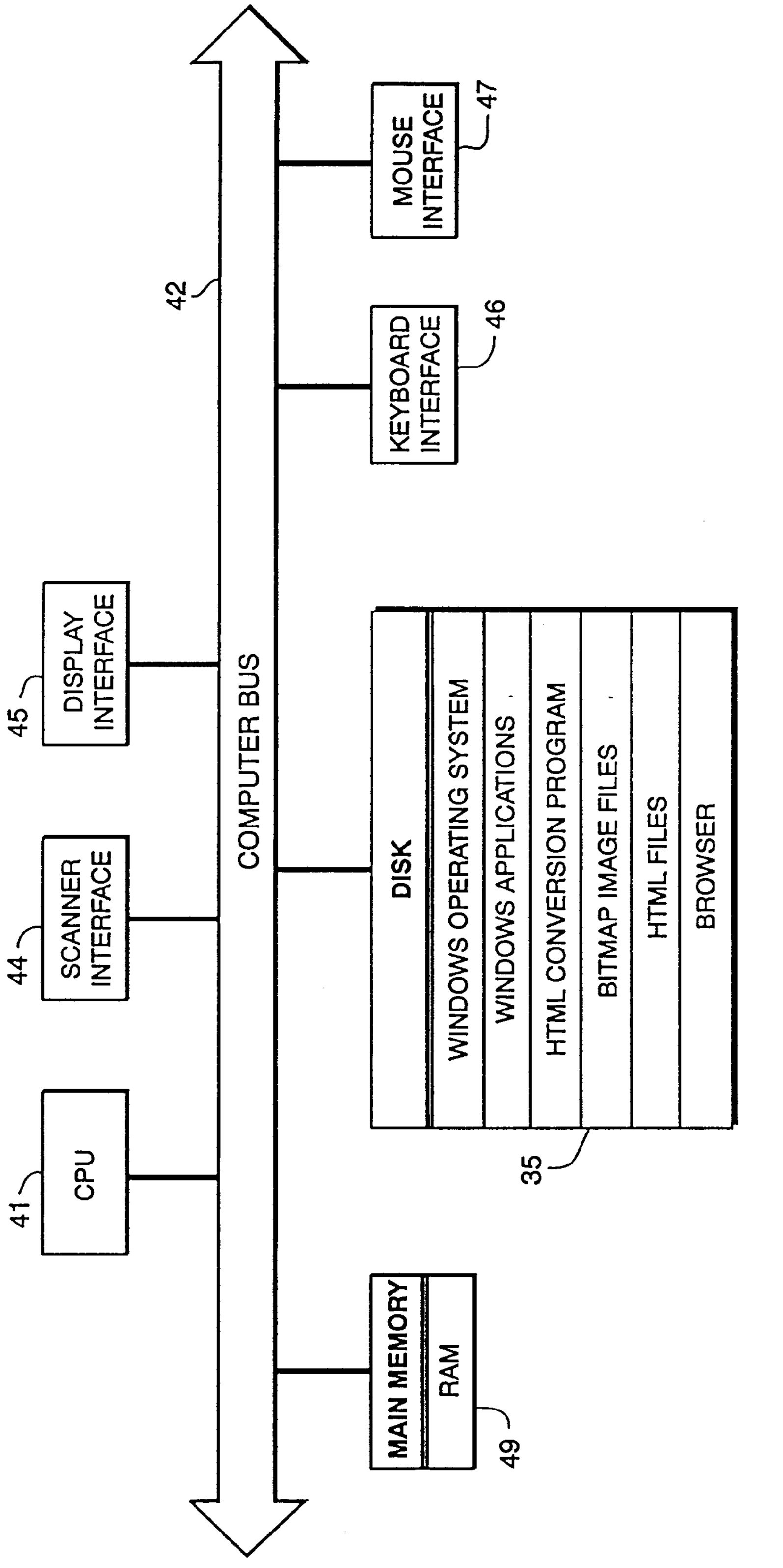
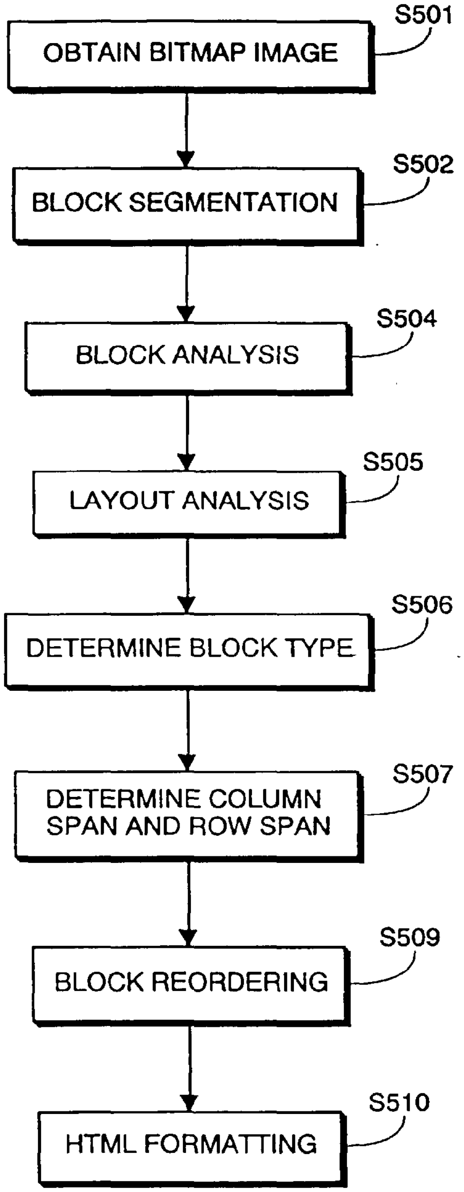

42
COMPUTER BUS
41
CPU
44
SCANNER INTERFACE
45
DISPLAY INTERFACE
MAIN MEMORY
RAM
49
DISK
WINDOWS OPERATING SYSTEM
WINDOWS APPLICATIONS
HTML CONVERSION PROGRAM
BITMAP IMAGE FILES
HTML FILES
BROWSER
35
KEYBOARD INTERFACE
46
MOUSE INTERFACE
47

FIG. 4



S501
OBTAIN BITMAP IMAGE
S502
BLOCK SEGMENTATION
S504
BLOCK ANALYSIS
S505
LAYOUT ANALYSIS
S506
DETERMINE BLOCK TYPE
S507
DETERMINE COLUMN SPAN AND ROW SPAN
S509
BLOCK REORDERING
S510
HTML FORMATTING

FIG. 5

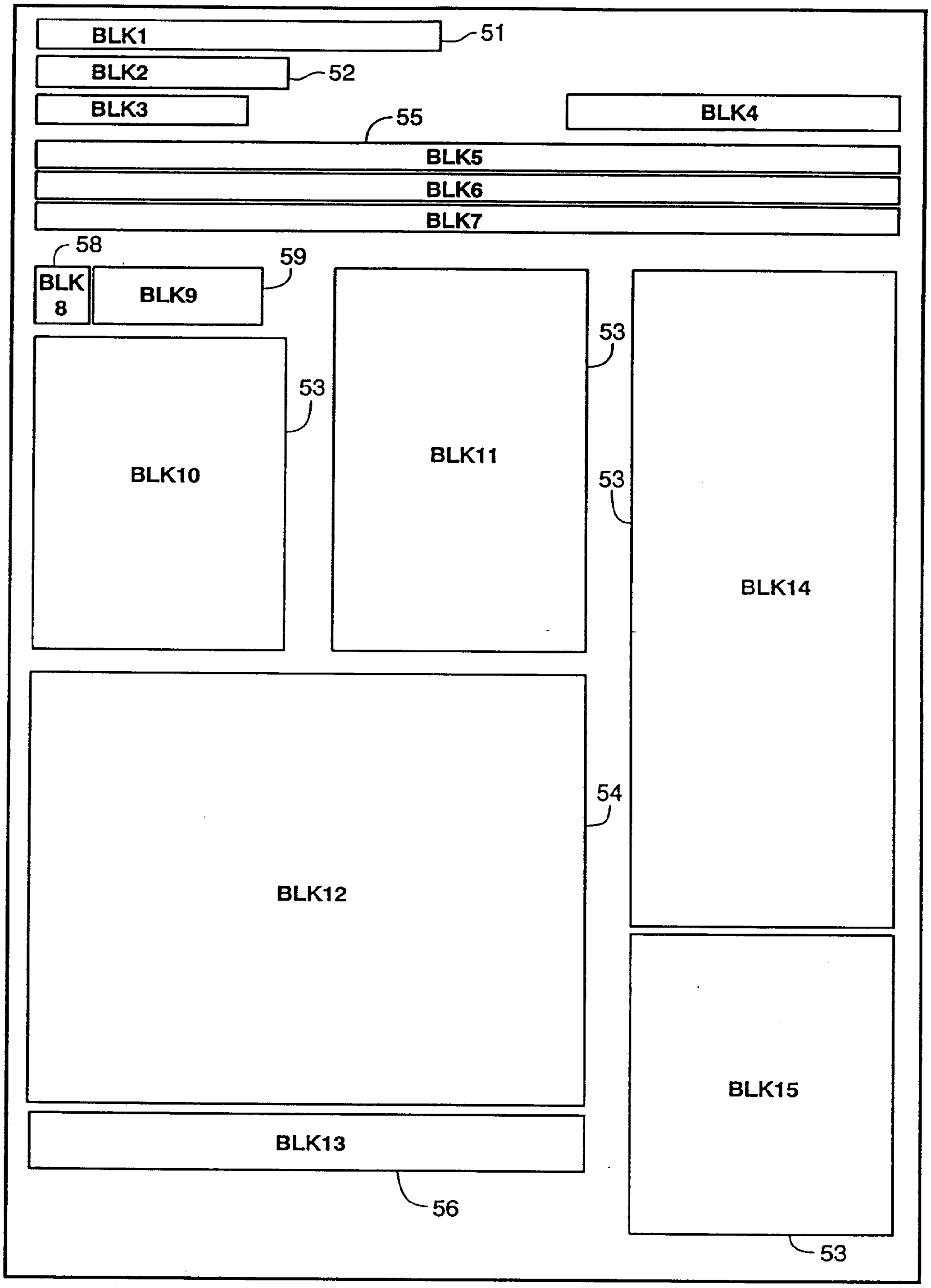

BLK1
51
BLK2
52
BLK3
BLK4
55
BLK5
BLK6
BLK7
58
59
BLK 8
BLK9
53
BLK10
53
BLK11
53
BLK14
54
BLK12
BLK13
56
BLK15
53

FIG. 6

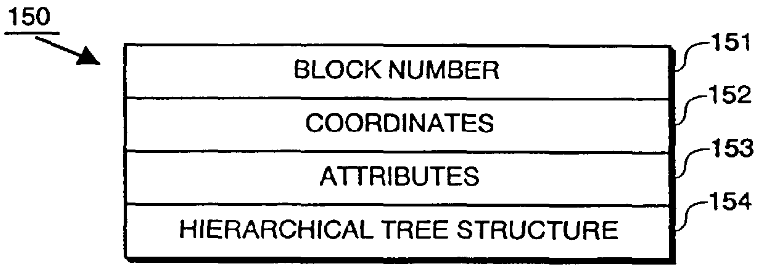

150
BLOCK NUMBER
COORDINATES
ATTRIBUTES
HIERARCHICAL TREE STRUCTURE
151
152
153
154

FIG. 7

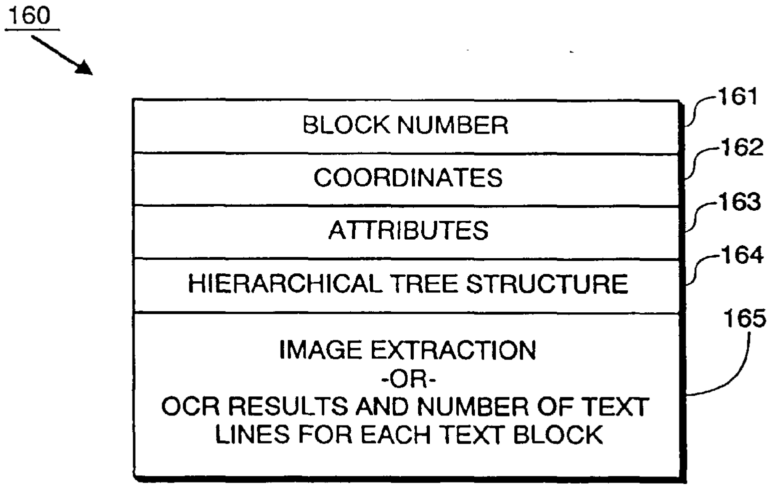

160
BLOCK NUMBER
COORDINATES
ATTRIBUTES
HIERARCHICAL TREE STRUCTURE
IMAGE EXTRACTION
-OR-
OCR RESULTS AND NUMBER OF TEXT
LINES FOR EACH TEXT BLOCK
161
162
163
164
165

FIG. 8

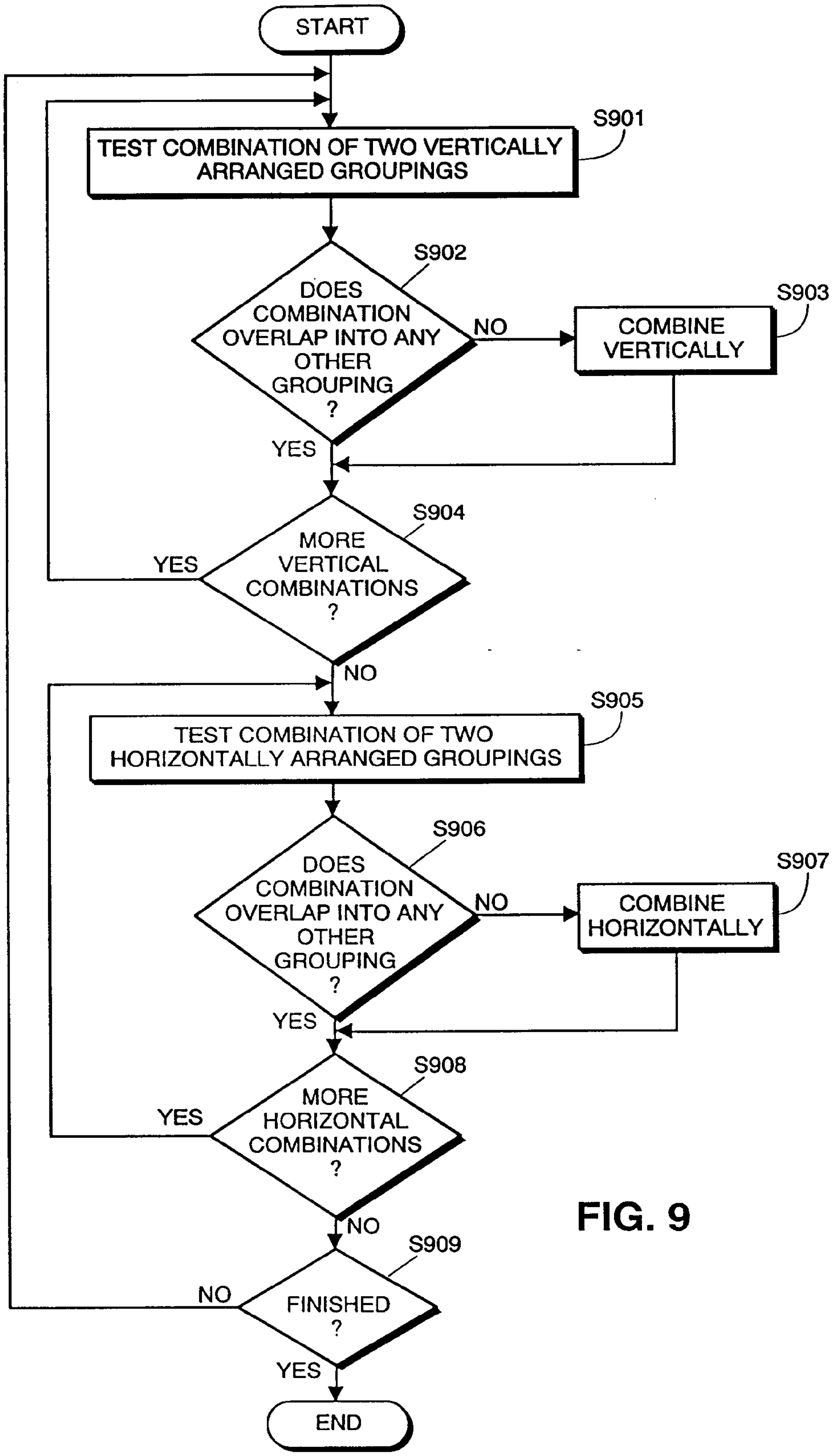

START
TEST COMBINATION OF TWO VERTICALLY ARRANGED GROUPINGS
S901
DOES COMBINATION OVERLAP INTO ANY OTHER GROUPING ?
S902
NO
COMBINE VERTICALLY
S903
YES
MORE VERTICAL COMBINATIONS ?
S904
YES
NO
TEST COMBINATION OF TWO HORIZONTALLY ARRANGED GROUPINGS
S905
DOES COMBINATION OVERLAP INTO ANY OTHER GROUPING ?
S906
NO
COMBINE HORIZONTALLY
S907
YES
MORE HORIZONTAL COMBINATIONS ?
S908
YES
NO
FINISHED ?
S909
NO
YES
END

FIG. 9

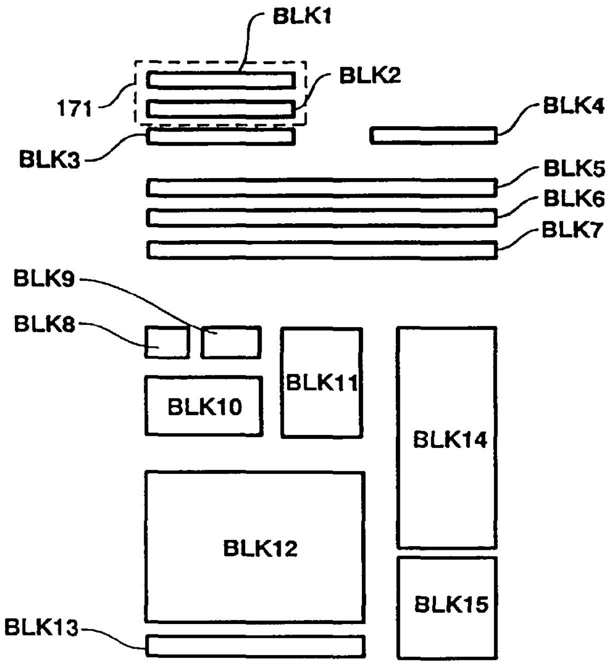


FIG. 10A-1

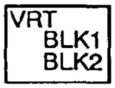


FIG. 10B-1

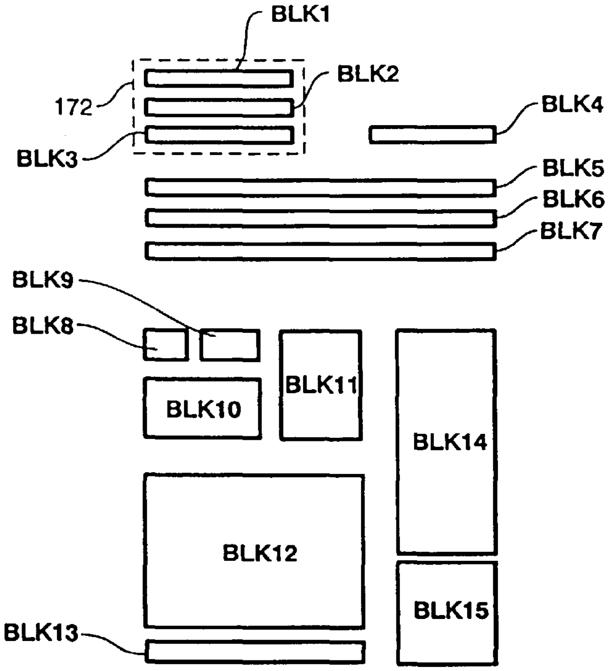


FIG. 10A-2

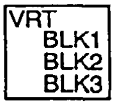


FIG. 10B-2

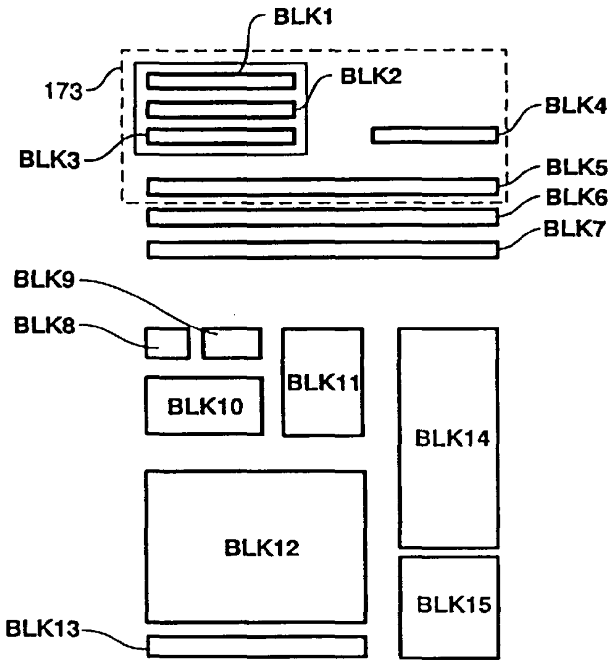


FIG. 10A-3

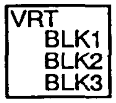


FIG. 10B-3

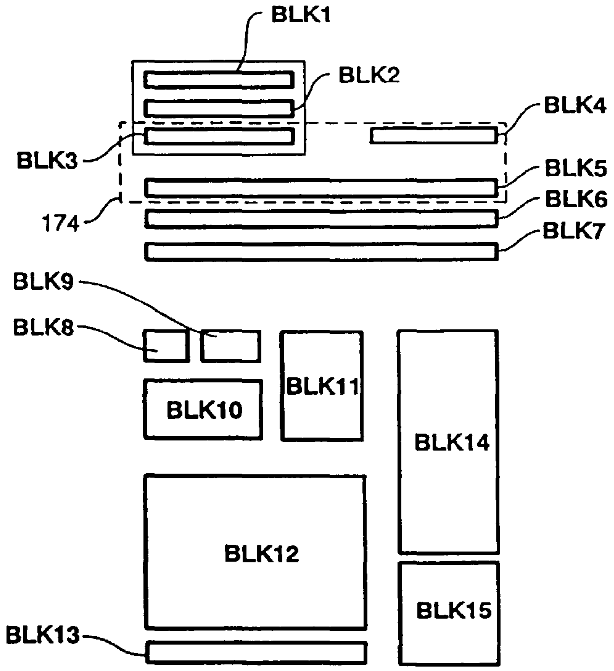


FIG. 10A-4

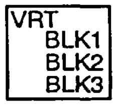


FIG. 10B-4

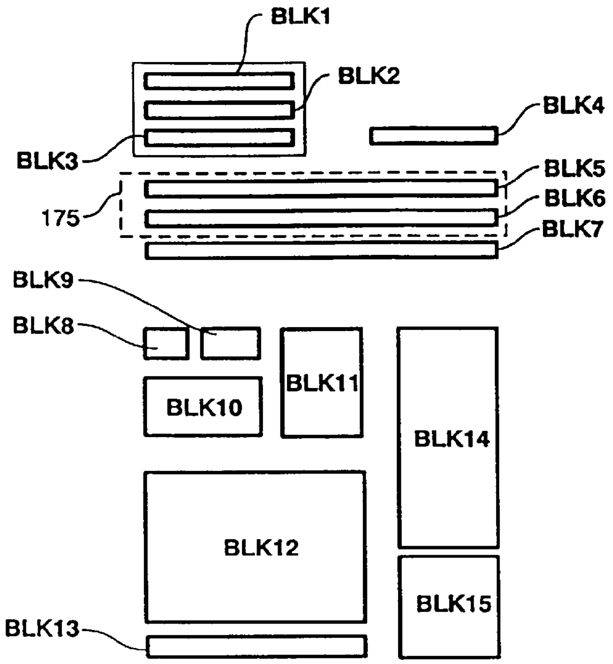


FIG. 10A-5

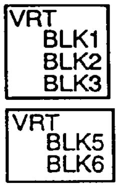


FIG. 10B-5

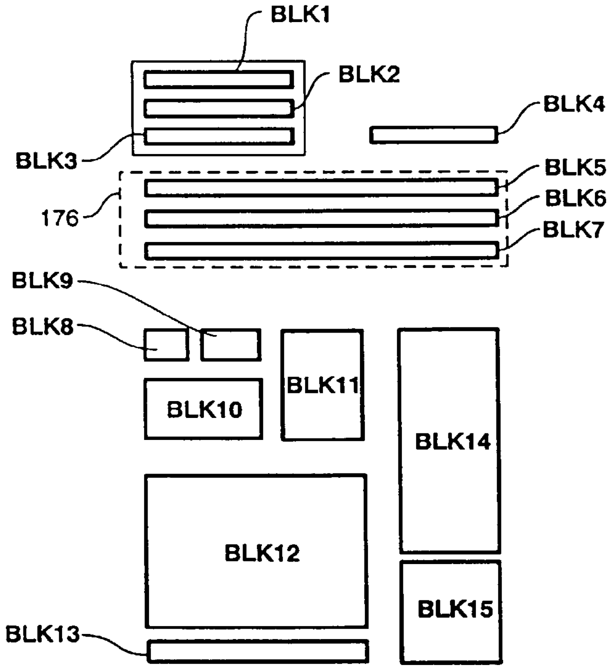


FIG. 10A-6

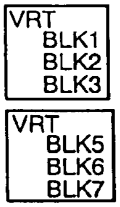


FIG. 10B-6

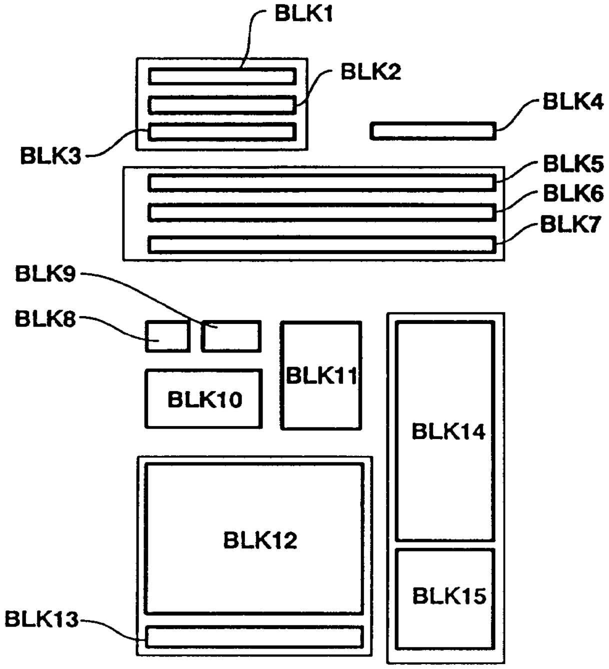


FIG. 10A-7

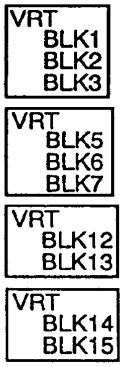


FIG. 10B-7

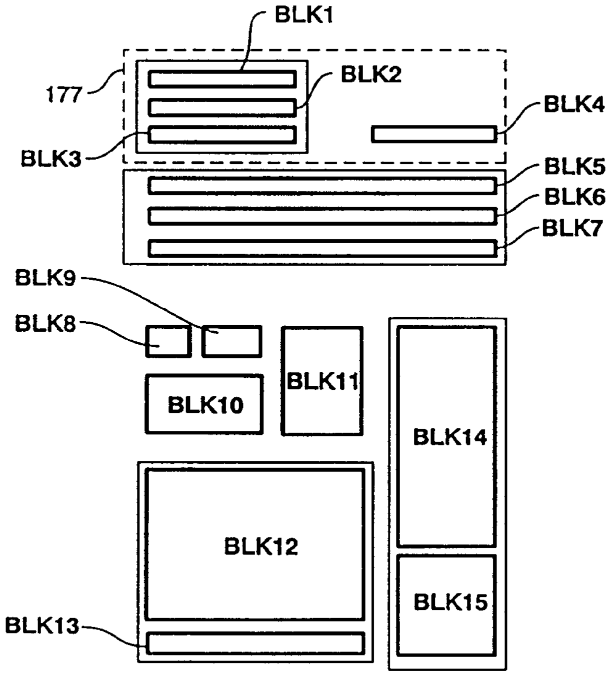


FIG. 10A-8

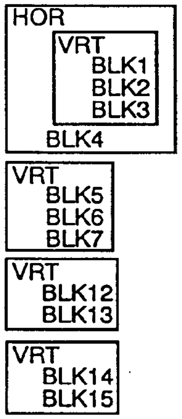


FIG. 10B-8

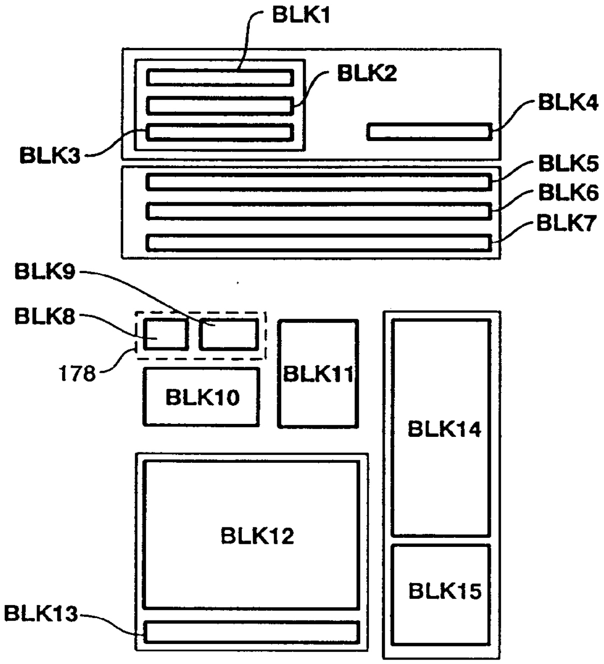


FIG. 10A-9

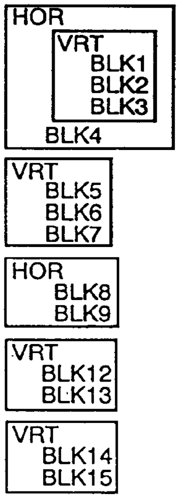


FIG. 10B-9

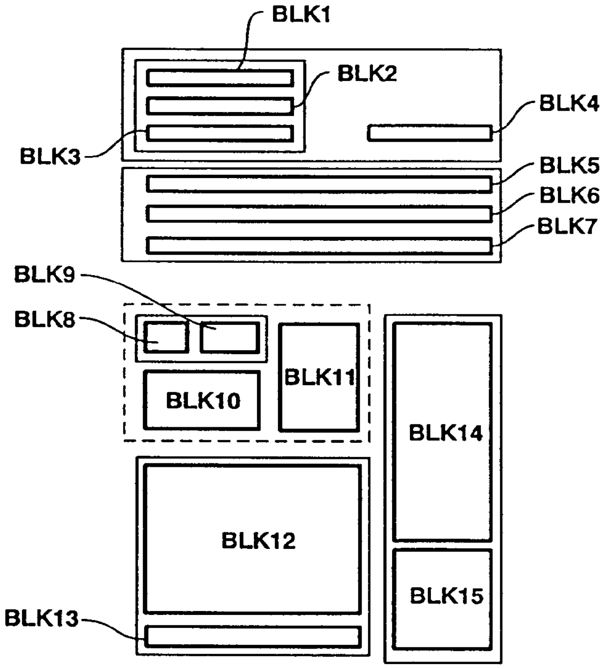


FIG. 10A-10

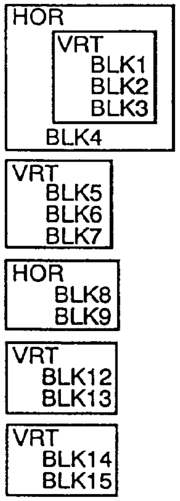


FIG. 10B-10

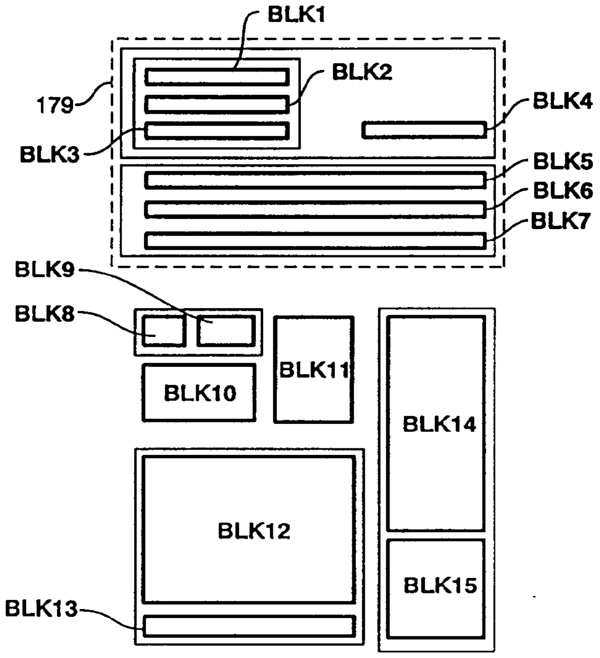


FIG. 10A-11

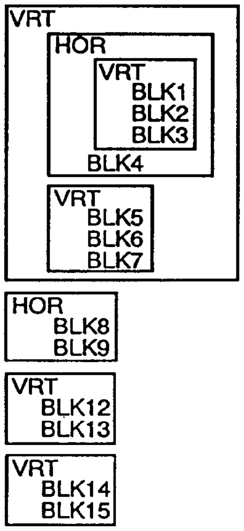


FIG. 10B-11

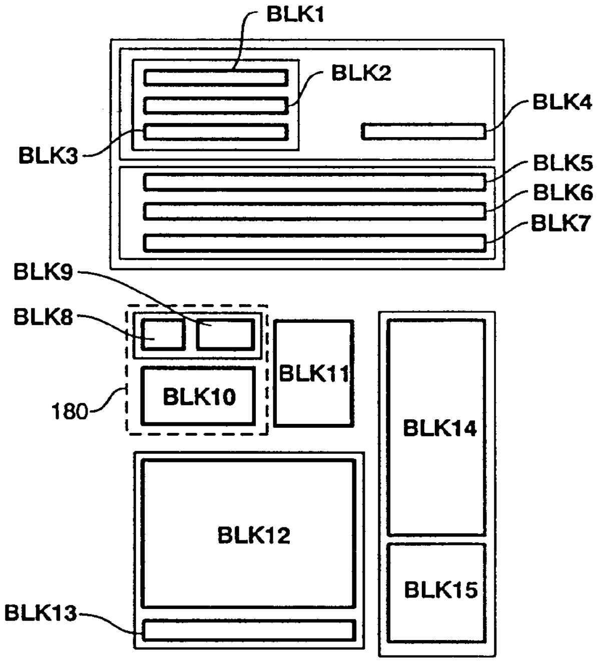


FIG. 10A-12

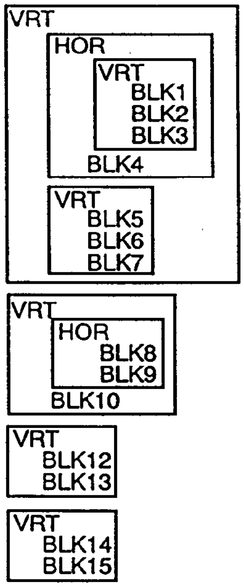


FIG. 10B-12

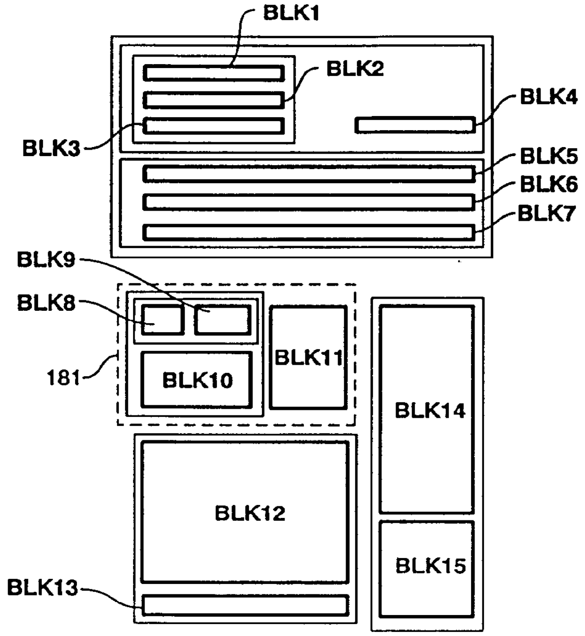


FIG. 10A-13

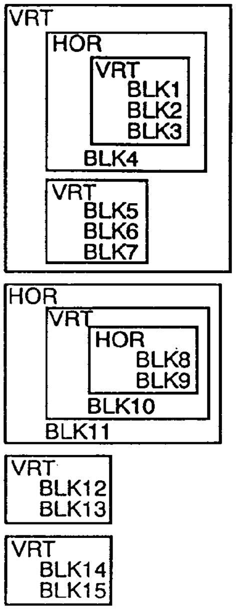


FIG. 10B-13

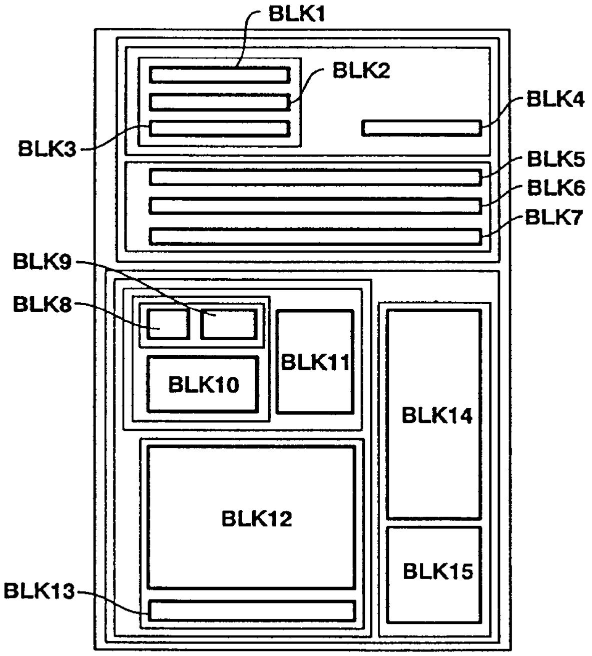


FIG. 10A-14

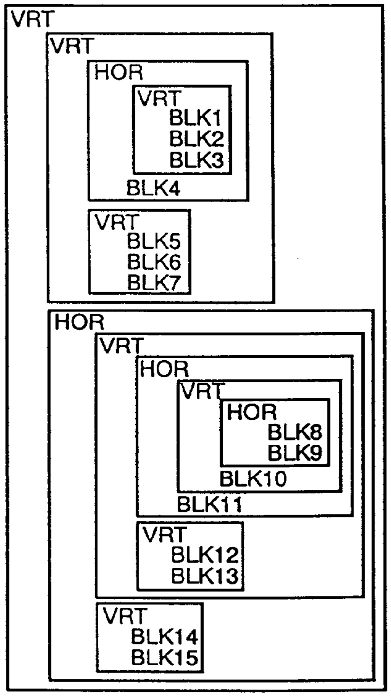


FIG. 10B-14

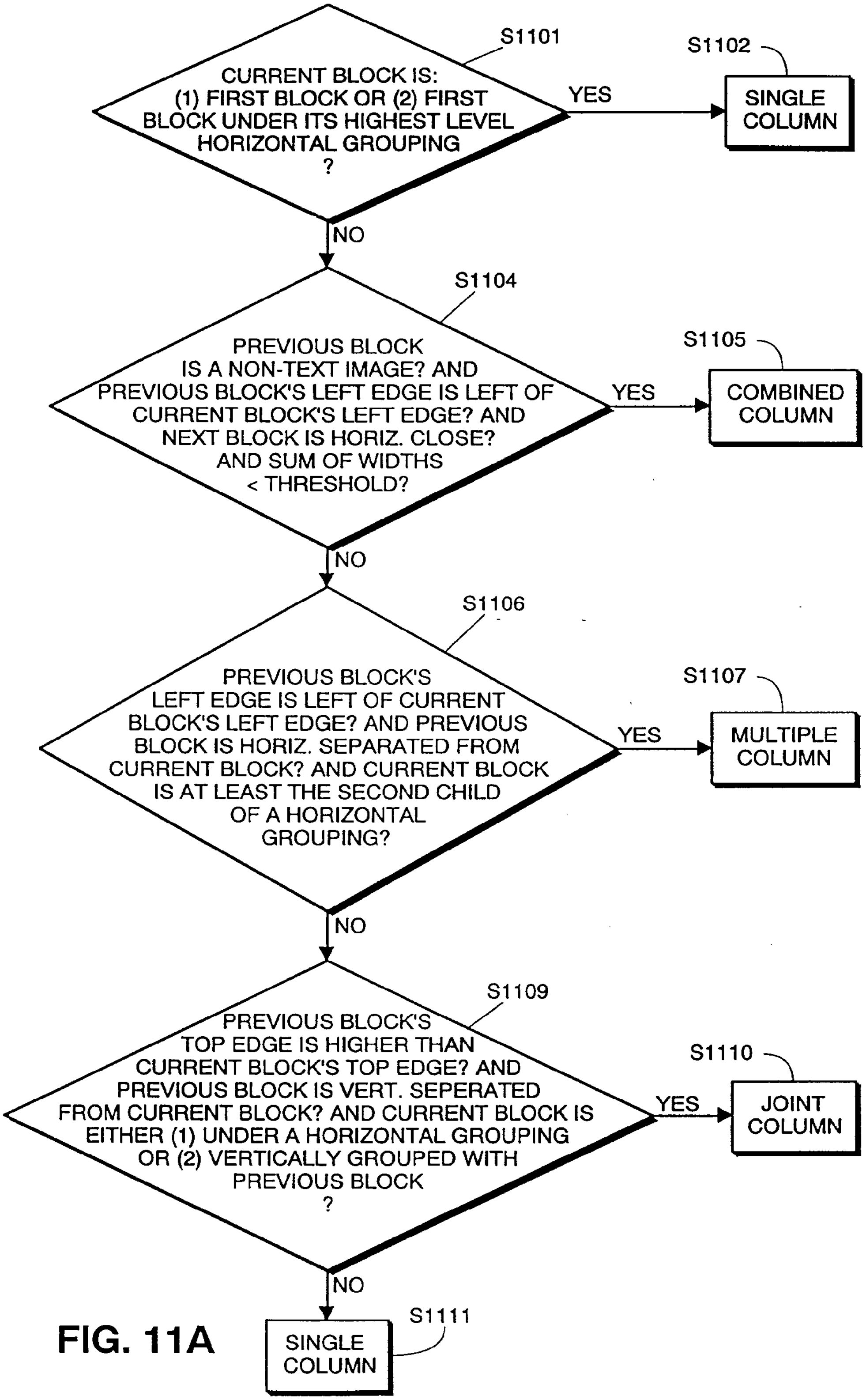

S1101
CURRENT BLOCK IS:
(1) FIRST BLOCK OR (2) FIRST
BLOCK UNDER ITS HIGHEST LEVEL
HORIZONTAL GROUPING
?
YES
S1102
SINGLE
COLUMN
NO
S1104
PREVIOUS BLOCK
IS A NON-TEXT IMAGE? AND
PREVIOUS BLOCK'S LEFT EDGE IS LEFT OF
CURRENT BLOCK'S LEFT EDGE? AND
NEXT BLOCK IS HORIZ. CLOSE?
AND SUM OF WIDTHS
< THRESHOLD?
YES
S1105
COMBINED
COLUMN
NO
S1106
PREVIOUS BLOCK'S
LEFT EDGE IS LEFT OF CURRENT
BLOCK'S LEFT EDGE? AND PREVIOUS
BLOCK IS HORIZ. SEPARATED FROM
CURRENT BLOCK? AND CURRENT BLOCK
IS AT LEAST THE SECOND CHILD
OF A HORIZONTAL
GROUPING?
YES
S1107
MULTIPLE
COLUMN
NO
S1109
PREVIOUS BLOCK'S
TOP EDGE IS HIGHER THAN
CURRENT BLOCK'S TOP EDGE? AND
PREVIOUS BLOCK IS VERT. SEPERATED
FROM CURRENT BLOCK? AND CURRENT BLOCK IS
EITHER (1) UNDER A HORIZONTAL GROUPING
OR (2) VERTICALLY GROUPED WITH
PREVIOUS BLOCK
?
YES
S1110
JOINT
COLUMN
NO
S1111
SINGLE
COLUMN

FIG. 11A

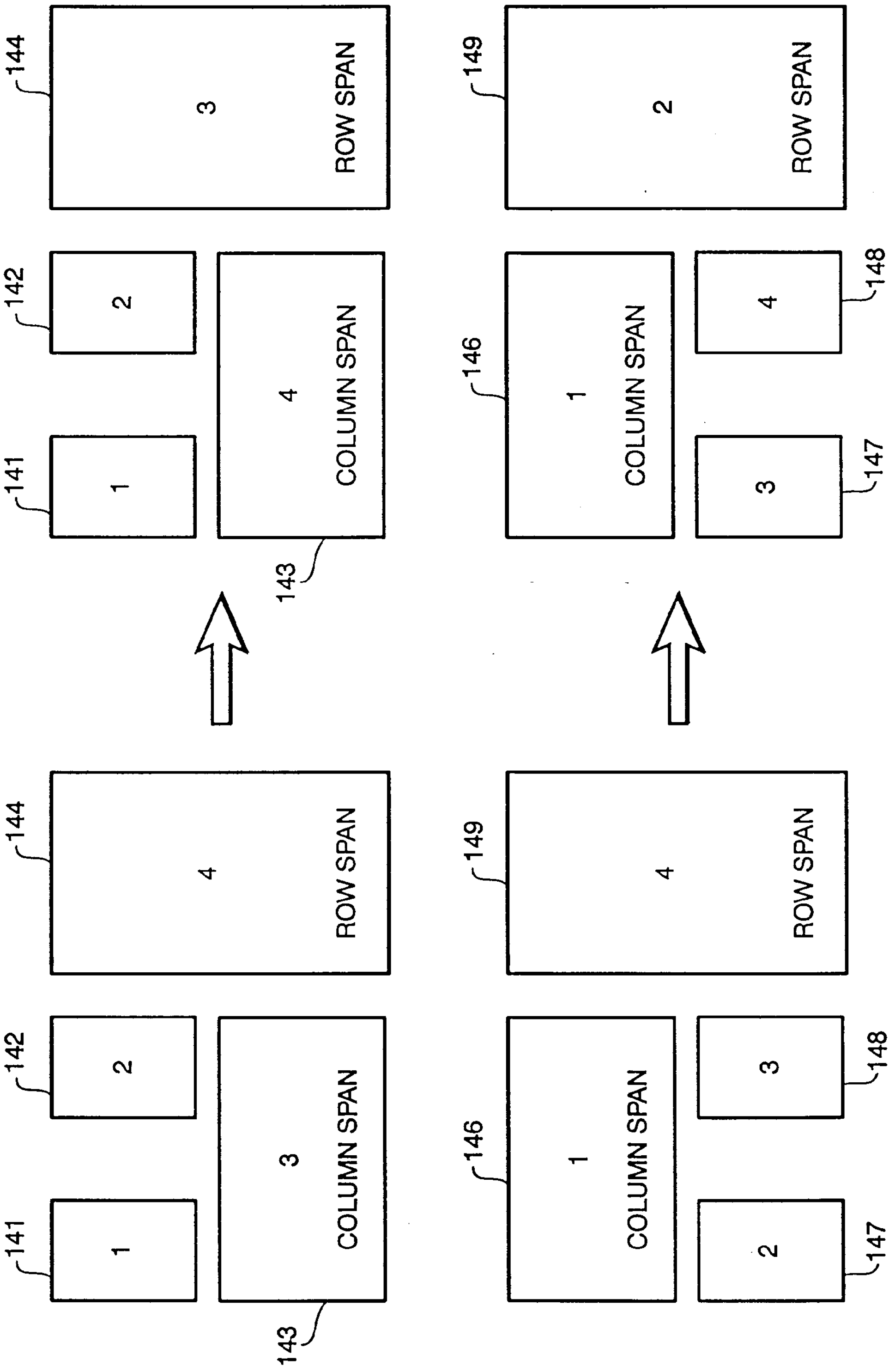

141
1
142
2
143
3
COLUMN SPAN
144
4
ROW SPAN
141
1
142
2
143
4
COLUMN SPAN
144
3
ROW SPAN
146
1
COLUMN SPAN
147
2
148
3
149
4
ROW SPAN
146
1
COLUMN SPAN
147
3
148
4
149
2
ROW SPAN

FIG. 11B

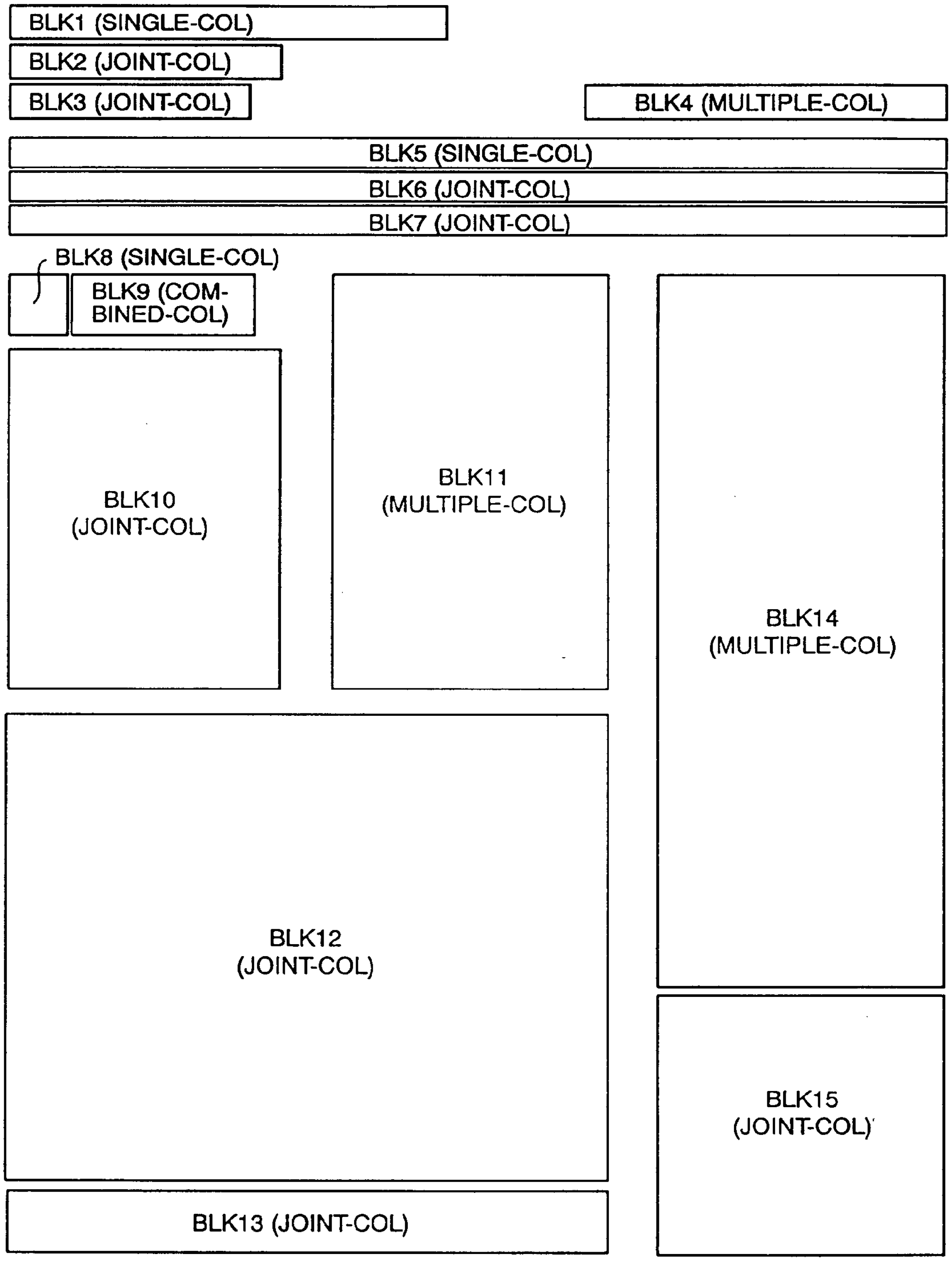

BLK1 (SINGLE-COL)
BLK2 (JOINT-COL)
BLK3 (JOINT-COL)
BLK4 (MULTIPLE-COL)
BLK5 (SINGLE-COL)
BLK6 (JOINT-COL)
BLK7 (JOINT-COL)
BLK8 (SINGLE-COL)
BLK9 (COMBINED-COL)
BLK10 (JOINT-COL)
BLK11 (MULTIPLE-COL)
BLK14 (MULTIPLE-COL)
BLK12 (JOINT-COL)
BLK15 (JOINT-COL)
BLK13 (JOINT-COL)

FIG. 12

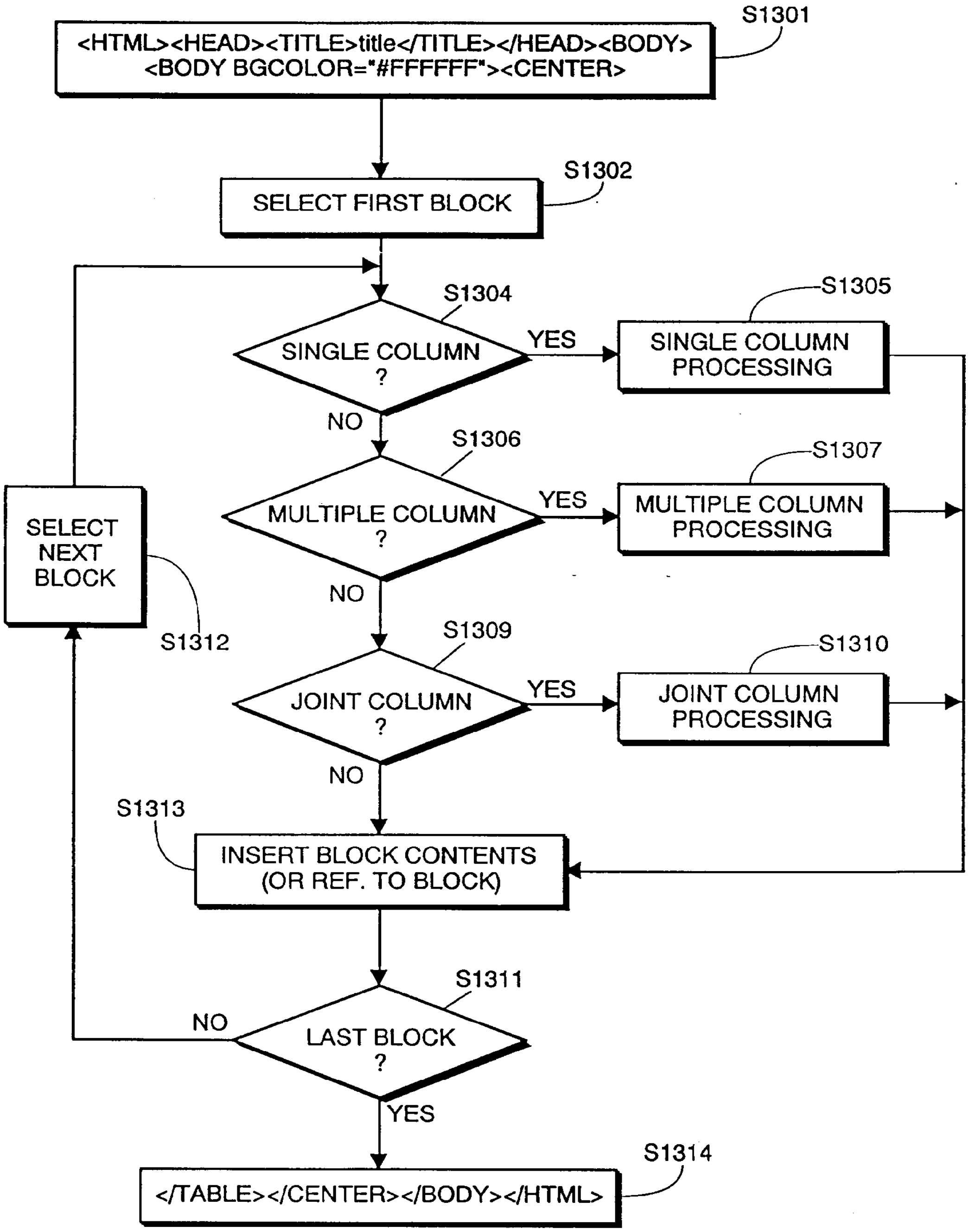

S1301
<HTML><HEAD><TITLE>title</TITLE></HEAD><BODY>
<BODY BGCOLOR="#FFFFFF"><CENTER>
S1302
SELECT FIRST BLOCK
S1304
SINGLE COLUMN ?
YES
S1305
SINGLE COLUMN PROCESSING
NO
S1306
MULTIPLE COLUMN ?
YES
S1307
MULTIPLE COLUMN PROCESSING
SELECT NEXT BLOCK
S1312
NO
S1309
JOINT COLUMN ?
YES
S1310
JOINT COLUMN PROCESSING
NO
S1313
INSERT BLOCK CONTENTS (OR REF. TO BLOCK)
S1311
NO
LAST BLOCK ?
YES
S1314
</TABLE></CENTER></BODY></HTML>

FIG. 13A

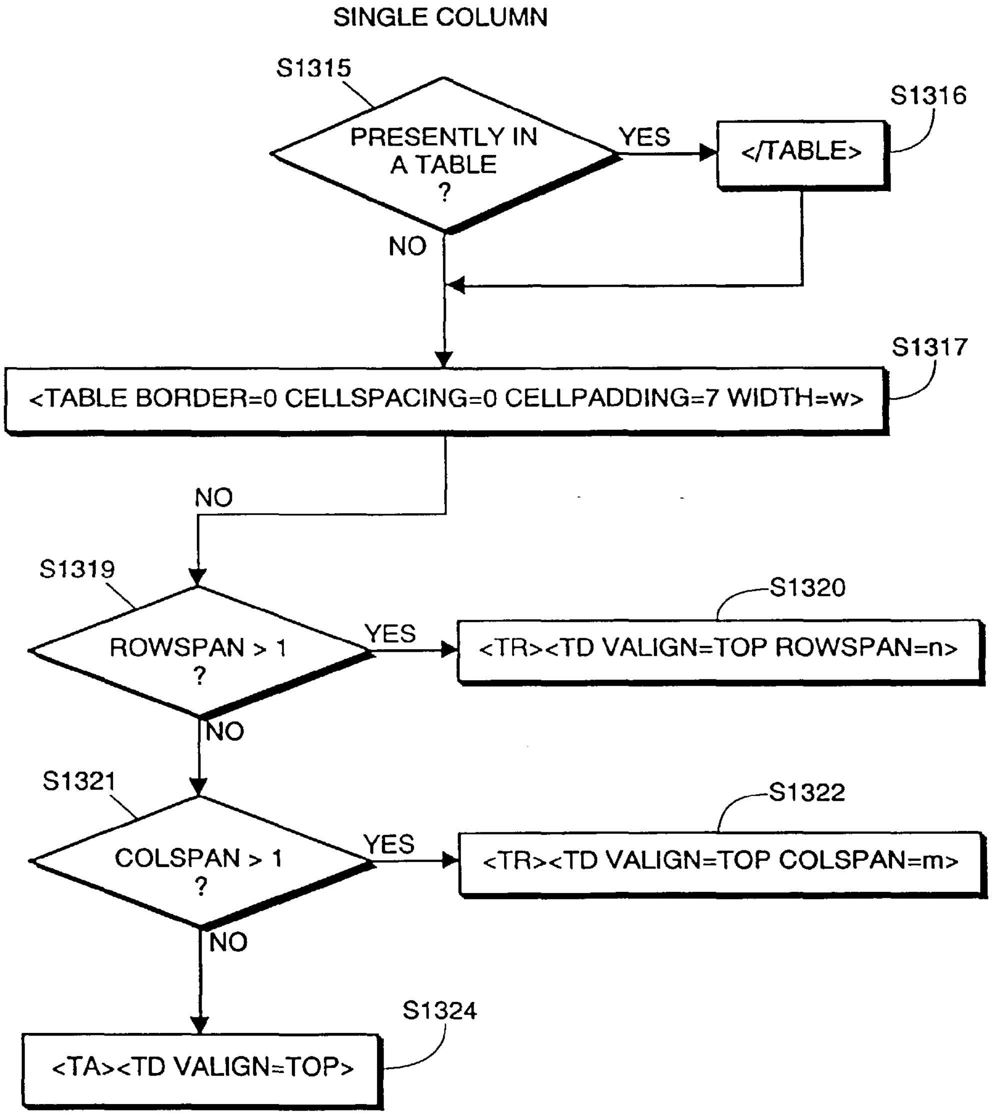

SINGLE COLUMN
S1315
PRESENTLY IN A TABLE ?
YES
NO
S1316
</TABLE>
S1317
<TABLE BORDER=0 CELLSPACING=0 CELLPADDING=7 WIDTH=w>
NO
S1319
ROWSPAN > 1 ?
YES
NO
S1320
<TR><TD VALIGN=TOP ROWSPAN=n>
S1321
COLSPAN > 1 ?
YES
NO
S1322
<TR><TD VALIGN=TOP COLSPAN=m>
S1324
<TA><TD VALIGN=TOP>

FIG. 13B

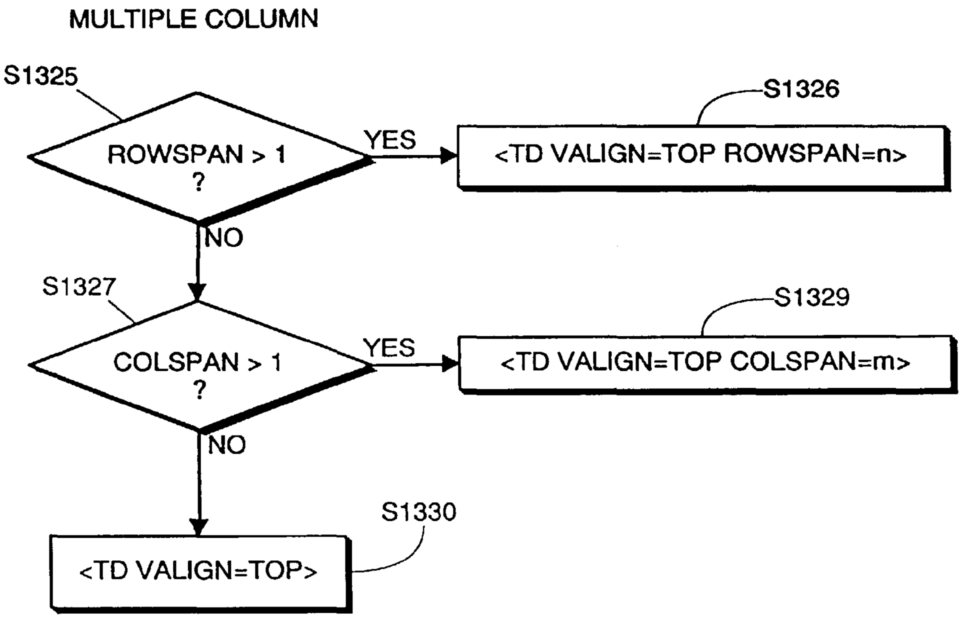


FIG. 13C

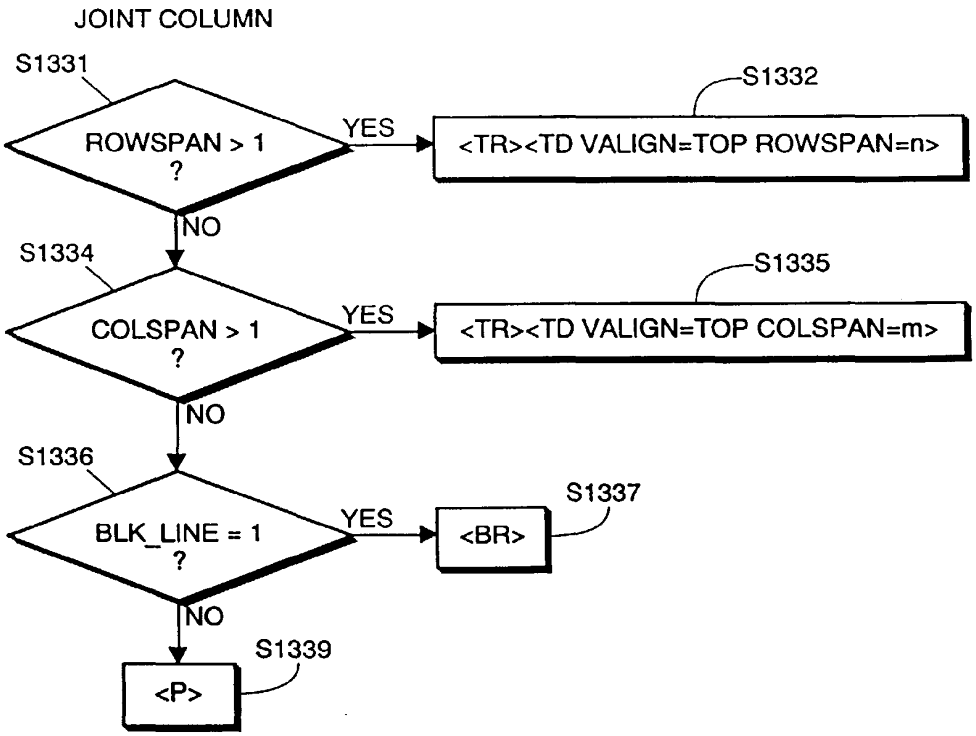
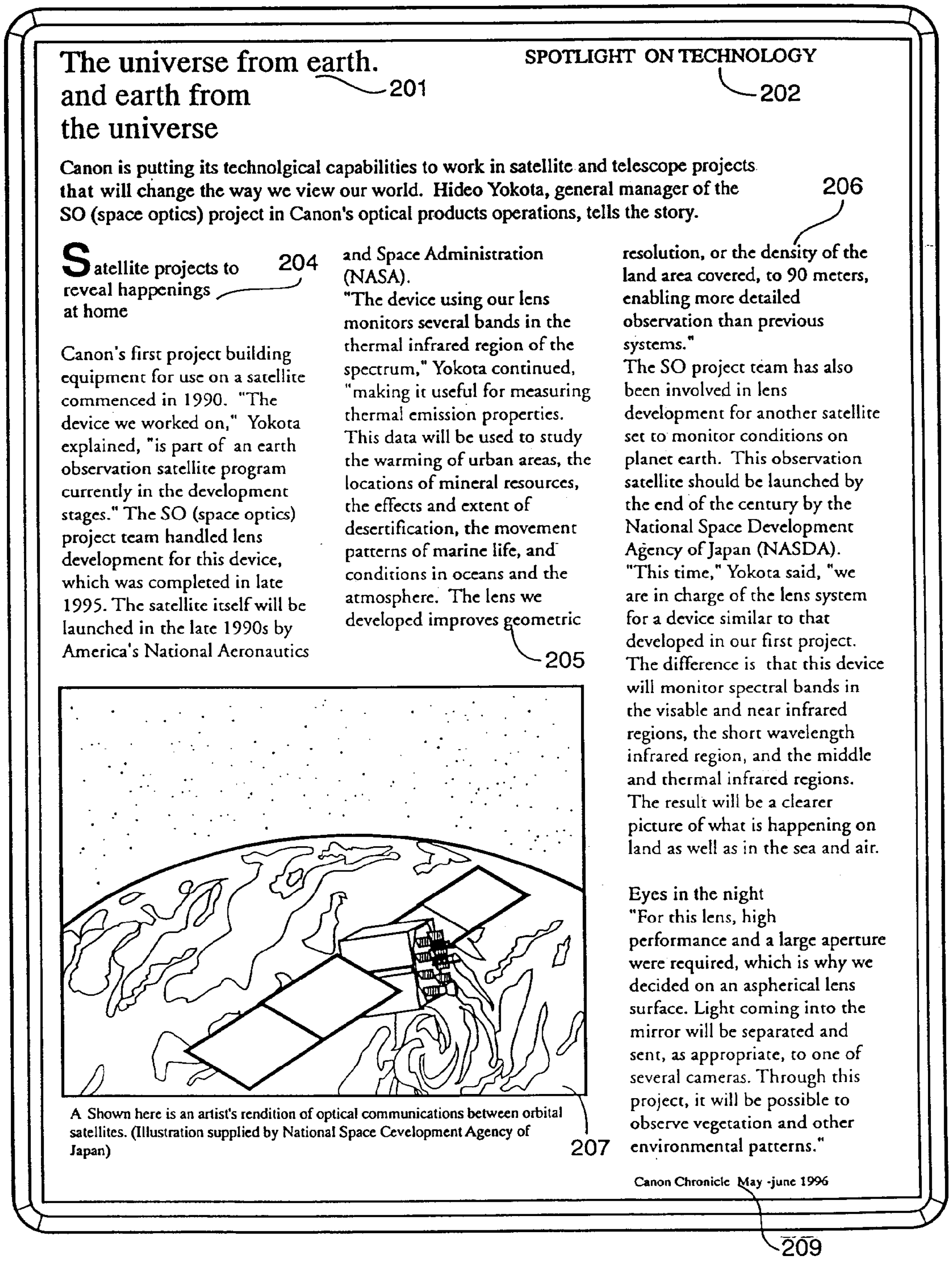


FIG. 13D



# The universe from earth. and earth from the universe

SPOTLIGHT ON TECHNOLOGY

**Canon is putting its technolgical capabilities to work in satellite and telescope projects that will change the way we view our world. Hideo Yokota, general manager of the SO (space optics) project in Canon's optical products operations, tells the story.**

## Satellite projects to reveal happenings at home

Canon's first project building equipment for use on a satellite commenced in 1990. "The device we worked on," Yokota explained, "is part of an earth observation satellite program currently in the development stages." The SO (space optics) project team handled lens development for this device, which was completed in late 1995. The satellite itself will be launched in the late 1990s by America's National Aeronautics and Space Administration (NASA).

"The device using our lens monitors several bands in the thermal infrared region of the spectrum," Yokota continued, "making it useful for measuring thermal emission properties. This data will be used to study the warming of urban areas, the locations of mineral resources, the effects and extent of desertification, the movement patterns of marine life, and conditions in oceans and the atmosphere. The lens we developed improves geometric resolution, or the density of the land area covered, to 90 meters, enabling more detailed observation than previous systems."

The SO project team has also been involved in lens development for another satellite set to monitor conditions on planet earth. This observation satellite should be launched by the end of the century by the National Space Development Agency of Japan (NASDA). "This time," Yokota said, "we are in charge of the lens system for a device similar to that developed in our first project. The difference is that this device will monitor spectral bands in the visable and near infrared regions, the short wavelength infrared region, and the middle and thermal infrared regions. The result will be a clearer picture of what is happening on land as well as in the sea and air.

## Eyes in the night

"For this lens, high performance and a large aperture were required, which is why we decided on an aspherical lens surface. Light coming into the mirror will be separated and sent, as appropriate, to one of several cameras. Through this project, it will be possible to observe vegetation and other environmental patterns."

A Shown here is an artist's rendition of optical communications between orbital satellites. (Illustration supplied by National Space Cevelopment Agency of Japan)

Canon Chronicle May -june 1996

FIG. 14

European Patent Office

## EUROPEAN SEARCH REPORT

Application Number
EP 97 30 4451

### DOCUMENTS CONSIDERED TO BE RELEVANT

| Category | Citation of document with indication, where appropriate, of relevant passages | Relevant to claim | CLASSIFICATION OF THE APPLICATION (Int.Cl.6) |
|---|---|---|---|
| Y | R. GANN: "Caere inproves its OCR interface"<br>PC USER,<br>no. 289, 21 August 1996 - 3 September 1996, GB,<br>page 44 XP002055985<br>* the whole document *<br>--- | 1-67 | G06K9/20<br>G06F17/21 |
| Y | R. GANN: "Accurate OCR for complex pages"<br>PC USER,<br>no. 292, 2 - 15 October 1996, GB,<br>page 50 XP002055986<br>* the whole document *<br>--- | 1-67 | |
| P,Y | EP 0 758 775 A (CANON KABUSHIKI KAISHA)<br>* the whole document *<br>--- | 1-67 | |
| Y | EP 0 660 256 A (CANON KABUSHIKI KAISHA)<br>* page 2, line 7 - page 18, line 6 *<br>--- | 1-67 | |
| A | YOUNG SEAK PARK ET AL: "A Hierarchical Method for Block Segmentation and Classification of General Document Images"<br>SYSTEMS AND COMPUTERS IN JAPAN,<br>vol. 24, no. 9, 1993, NEW YORK, US,<br>pages 84-96, XP000433079<br>* the whole document *<br>--- | 1-67 | TECHNICAL FIELDS SEARCHED (Int.Cl.6)<br>G06K<br>G06F |
| P,A | YUAN T. TANG ET AL: "AUTOMATIC DOCUMENT PROCESSING: A SURVEY"<br>PATTERN RECOGNITION,<br>vol. 29, no. 12, December 1996, GB,<br>pages 1931-1952, XP000639709<br>* the whole document *<br>----- | 1-67 | |

The present search report has been drawn up for all claims

| Place of search | Date of completion of the search | Examiner |
|---|---|---|
| BERLIN | 18 February 1998 | Abram, R |

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 876 034 A2

# (12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
04.11.1998 Bulletin 1998/45

(51) Int Cl.6: **H04L 29/06**, G06F 17/30

(21) Application number: **98300847.5**

(22) Date of filing: **05.02.1998**

(84) Designated Contracting States:
**AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE**
Designated Extension States:
**AL LT LV MK RO SI**

(30) Priority: **10.02.1997 US 797902**

(71) Applicant: **International Business Machines Corporation**
**Armonk, N.Y. 10504 (US)**

(72) Inventors:
- **Thompson, Joseph Raymond**
  **Round Rock, Texas 78681 (US)**
- **Berstis, Viktors**
  **Austin, Texas 78746 (US)**

(74) Representative: **Davies, Simon Robert**
**IBM,**
**United Kingdom Limited,**
**Intellectual Property Law,**
**Hursley Park**
**Winchester, Hampshire SO21 2JN (GB)**

## (54) Method for content retrieval over a network

(57) A method is provided for retrieving Web content from a plurality of Web servers for delivery to a Web client connectable to the World Wide Web via a communication link 227. The Web client is preferably a data processing system connectable to a television 104 or other conventional monitor to provide low cost Internet access. The method begins by having the user define a set of one or more servers from which content is desired to be retrieved and stored in the cache. These servers are preferably identified by a list of favorite Web sites. A test is then made to determine whether a given download period has terminated 250. Typically, this download period occurs during an "off" period, such as in the middle of the night, to avoid traffic congestion at the Web server sites. If the given download period has not terminated, a determination is then made of an activity level for the communication link as content is being downloaded to the cache from the one or more servers 254. If the activity level for the communication link is less than a given threshold level, additional requests for content are issued to the cache 260 according to a so-called "fairness policy" that ensures that content from as many sites as possible is downloaded during the download period.

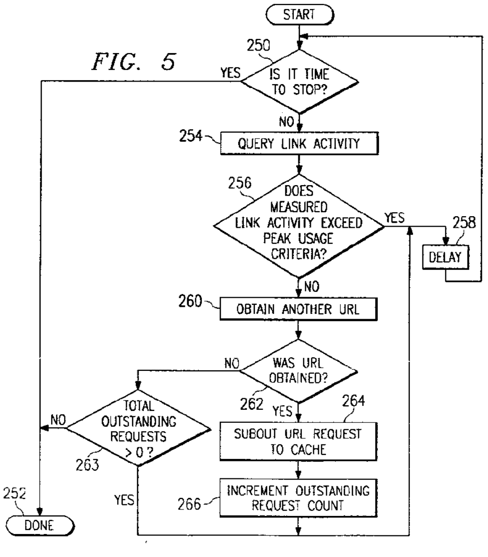


FIG. 5
START
250
YES
IS IT TIME TO STOP?
NO
254
QUERY LINK ACTIVITY
256
DOES MEASURED LINK ACTIVITY EXCEED PEAK USAGE CRITERIA?
YES
258
DELAY
NO
260
OBTAIN ANOTHER URL
NO
WAS URL OBTAINED?
262
YES
264
SUBOUT URL REQUEST TO CACHE
NO
TOTAL OUTSTANDING REQUESTS > 0 ?
263
252
YES
266
INCREMENT OUTSTANDING REQUEST COUNT
DONE

Printed by Jouve, 75001 PARIS (FR)

## Description

The present invention relates generally to information retrieval over the World Wide Web or such like, and more particularly to retrieving content for delivery to a client connectable to a network via a communication link, the client including a cache.

The World Wide Web of the Internet is the most successful distributed application in the history of computing. In the Web environment, client machines effect transactions to Web servers using the Hypertext Transfer Protocol (HTTP), which is a known application protocol providing users access to files (e.g., text, graphics, images, sound, video, etc.) using a standard page description language known as Hypertext Markup Language (HTML). HTML provides basic document formatting and allows the developer to specify "links" to other servers and files. In the Internet paradigm, a network path to a server is identified by a so-called Uniform Resource Locator (URL) having a special syntax for defining a network connection. Use of an HTML-compatible browser (e.g., Netscape Navigator or Microsoft Internet Explorer) at a client machine involves specification of a link via the URL. In response, the client makes a request to the server identified in the link and receives in return a document formatted according to HTML.

There has been great interest in providing Internet access at minimal economic cost. While most computers now are pre-configured for Internet access, a significant percentage of households still do not have a personal computer. Thus, it has now been proposed to provide a data processing system that, much like a VCR, may be connected to a television set and used in lieu of a personal computer to provide Web access through a conventional remote control device associated with the system unit. Such a system enables the television to become, in effect, a "Web appliance". The viewer can rapidly switch between conventional television and Internet access using the remote control unit. All of the conventional Internet access tools and navigational functions are preferably built-in to the system and thus hidden to the user.

One such tool is so-called "off-line" browsing. As any casual user of the Internet can attest, interesting or attractive web sites are sometimes difficult to access due to large traffic demands. As a result, several companies have developed so-called "off-line" browser programs that are designed to deliver web pages from favorite Web servers to a user's hard drive for browsing at the user's convenience. Typically, such programs include some form of scheduling feature that enables the user to fetch identifiable pages at off-peak hours, saving time and connection charges. The user may then browse the pages at his or her convenience without a modem and even without an active connection to the Internet.

While off-line browser programs offer certain advantages, they do not have the capability to optimize utilization of the communication link between the client and the World Wide Web during the off-peak information retrieval process. This problem becomes more acute if there are constraints on the amount of time that off-peak information retrieval may be accomplished. In the future, it is anticipated that Web appliances of the type described above will be provided by computer network or other service providers, who will only allow their subscribers limited periods of time during which off-line browsing will be permitted. Thus, for example, a network operator may restrict subscribers to off-line browsing for just one hour per night. During this hour, a user may desire to obtain content from numerous Web sites. Thus, it would be desirable to provide some mechanism that could optimize retrieval of Web content during this limited period of time.

Accordingly, the present invention provides a method of retrieving content for delivery to a client connectable to a network via a communication link, the client including a cache, said method comprising the steps of:

(a) defining a set of servers from which content is desired to be retrieved and stored in the cache;
(b) determining an activity level for the communication link as content is being downloaded to the cache from the servers; and
(c) if the activity level for the communication link is less than a threshold level, issuing to the cache additional requests for content such that each of the servers of the set has an opportunity to deliver content to the client.

In the preferred embodiment the network is the World Wide Web, and said client and servers are a Web client and Web servers respectively. The additional requests for content may be issued to the set of servers as one request per server in an ordered sequence, or up to a predetermined number of requests per server in an ordered sequence, or based on any suitable predetermined parameters, for example the link depth of a document located on a particular server, or the number of bytes received in the cache from a particular server. The threshold level of the communication link may be a given number of outstanding requests for content, or related to an average link utilization rate for a given monitored interval, or any other suitable criterion. In the preferred embodiment the content received from a server is processed to remove duplicative links and non-local links.

Typically the content is downloaded to the cache during a predetermined time period, for example, approximately one hour. This period may be set by the network access provider, or is possibly selectable by the user. In such circumstances, it is determined whether a given download period has terminated, and if not, said step of determining an activity level for the communication link is performed. The additional requests for content are issued to the cache such that each of the servers at least has an opportunity to deliver content to the client during the given download period.

The invention further provides a computer program product for retrieving content for delivery to a client connectable to a network via a communication link, the client including a cache, the computer program product comprising: a computer-readable storage medium having a substrate; and a program encoded in the substrate of the computer-readable storage medium, wherein the program comprises means for performing the methods described herein.

The invention further provides a computer comprising a modem connected to a communication link; a processor; a memory including a cache; a browser program running on the processor for providing World Wide Web information retrieval and including means for generating a list of Web sites to be downloaded; a cache control program running on the processor and including means for initiating download requests to the communication link based on the list of Web sites; and a policy control program running on the processor and including means for issuing multiple HTTP GET requests to the cache control program to ensure that each of the Web sites on the list has an opportunity to contribute content during a given download session. Such a computer may function as a base unit in a data processing system having a remote control unit, in which the base unit is connectable to a monitor for providing Internet access under the control of the remote control unit.

Preferably the Web browser program includes menu means for generating the list of Web sites to be downloaded during an off-peak download session; the cache control program includes means for receiving downloads that are stored in the cache for off-line browsing; and the cache control program ensures that each of the Web sites on the list has an opportunity to contribute content during a given download session.

Viewed from another aspect, the invention provides a method of retrieving Web content for delivery to a Web client connectable to the World Wide Web via a communication link, the Web client including a cache, comprising the steps of:

(a) defining a set of Web servers from which content is desired to be retrieved and stored in the cache;
(b) determining an activity level for the communication link as content is being downloaded to the cache from the servers; and
(c) if the activity level for the communication link is less than a threshold level, issuing to the cache additional requests for content according to a policy that ensures that each of the servers of the set has an opportunity to deliver content to the Web client.

Viewed from a further aspect, the invention provides a computer program product for retrieving Web content for delivery to a Web client connectable to the World Wide Web via a communication link, the Web client including a cache, the computer program product comprising:

a computer-readable storage medium having a substrate; and
program data encoded in the substrate of the computer-readable storage medium, wherein the program data comprises: means for defining a set of servers from which content is desired to be retrieved and stored in the cache; means for monitoring an activity level for the communication link as content is being downloaded to the cache from the servers; and means responsive to the monitoring means for issuing to the cache additional requests for content according to a fairness policy that ensures that each of the servers has an opportunity to contribute content to the Web client during a download session.

Viewed from a further aspect, the invention provides a computer, comprising: a processor; a memory including a cache; a modem connected to a communication link; a Web browser program run by the processor for providing World Wide Web information retrieval and including menu means for generating a list of Web sites to be downloaded during an off-peak download session; a cache control program run by the processor for initiating download requests to the communication link based on the list of Web sites and for receiving downloads that are stored in the cache for off-line browsing; and a fairness policy control program run by the processor issuing multiple HTTP GET requests to the cache control program to ensure that each of the Web sites on the list has an opportunity to contribute content during a given download session.

Thus utilization of the communication link between a Web appliance and World Wide Web servers may be optimised by enhancing the off-peak caching of Web data when client access to the network is restricted, by ensuring that the link between a Web client and one or more Web servers is used to its maximum bandwidth during such a time-restricted off-peak browsing session. This represents an improvement in the functionality of off-line browsing programs to make more efficient use of limited communication resources, and helps to ensure maximum utilization of the Web client modem during off-peak caching of Web data from the World Wide Web of the Internet. The approach described herein allows equitable caching of content from a plurality of Web sites during an automatic download session so that a user obtains a significant percentage of the downloads that he or she desires, with each of a plurality of Web sites having an opportunity to deliver content to a client during an automatic download session.

The methods described herein are particularly suited to providing a Web appliance with an off-line browsing capability, and more particularly to allow efficient off-peak Web browsing for such a Web client appliance, for example, a data processing system connected to a conventional television.

Thus as describe herein Web content may be retrieved from a plurality of Web servers for delivery to a Web client connectable to the World Wide Web via a communication link. The Web client is preferably a data processing system connectable to a television or other conventional monitor to provide low cost Internet access. Initially a user defines a set of one or more servers from which content is desired to be retrieved and stored in the cache. These servers are preferably identified by a "list" of favorite Web sites. A test is then made to determine whether a given download period has terminated. Typically, this download period occurs during an "off" period, such as in the middle of the night, to avoid traffic congestion at the Web server sites. If the given download period has not terminated, a determination is then made of an activity level for the communication link as content is being downloaded to the cache from the one or more servers. If the activity level for the communication link is less than a given threshold level, additional requests for content are issued to the cache according to a so-called "fairness policy" that ensures that content from as many sites as possible is downloaded during the download period.

Thus, for example, according to the fairness policy the additional requests for content are issued to the set of one or more servers as one request per server in an ordered sequence. Alternatively, the additional requests for content are issued to the set of one or more servers up to a predetermined number of requests per server in an ordered sequence. Or, the additional requests are issued to the set of one or more servers based on the number of bytes received in the cache from a particular server. The fairness policy ensures that no one server dominates the download session to the exclusion of the other servers from which the user desires to download content. When content (i.e. a Web document) is received from a particular server, it is stored in the cache for off-line browsing. Prior to storage, the routine preferably removes duplicative and/or non-local HTML links so that subsequent access requests to the same document are handled more expediently.

A data processing system is provided to facilitate low cost Internet access. The system comprises two major parts: a remote control unit, and a base unit connectable to a monitor for providing Internet access under the control of the remote control unit. The base unit is, in effect, a computer, and includes a modem connected to a communication link, a processor, a memory, and various embedded control programs. These programs include a browser program including means responsive to commands from the remote control unit for generating a list of Web sites, and a cache control program for initiating download requests to the communication link based on the list of Web sites. An optimization routine maintains maximum utilization of the modem during a download session by issuing multiple HTTP GET requests to the cache control program based on the fairness policy.

Various embodiments of the invention will now be described in detail by way of example only with reference to the following drawings:

FIGURE 1A is pictorial representation of a data processing system unit connected to a conventional television set to form a "Web appliance";

FIGURE 1B is a pictorial representation of a front panel of the data processing system unit of Figure 1A;

FIGURE 1C is a pictorial representation of a rear panel of the data processing system unit of Figure 1A;

FIGURE 1D is a pictorial representation of a remote control unit associated with the data processing system unit of Figure 1A;

FIGURE 2 is a block diagram of the major components of the data processing system unit;

FIGURE 3 is a representative "favorites" list created by a user as a result of browsing the World Wide Web;

FIGURE 4 is a representative server URL queue list for the favorites list of FIGURE 3;

FIGURE 5 is a flowchart of a preferred method of the present invention for optimizing communication link activity during an off-peak caching session;

FIGURE 6 is a flowchart of the process response routine that is executed for each Web document received in response to a URL request; FIGURE 7 is a block diagram of a representative Web server platform or Web site; and

FIGURE 8 is a flowchart of the methods that are carried out by a Web server in response to receipt of a request from an Internet client such as the Web appliance described herein.

With reference now to the figures, and in particular with reference to FIGURES 1A through 1D, various pictorial representations of a data processing system are depicted. FIGURE 1A is a pictorial representation of the data processing system as a whole. Data processing system 100 in the depicted example provides, with minimal economic costs for hardware to the user, access to the Internet. Data processing system 100 includes a data processing unit 102. Data processing unit 102 is preferably sized to fit in typical entertainment centers and provides all required functionality, which is conventionally found in personal computers, to enable a user to browse the Internet. Additionally, data processing unit 102 may provide other common functions such as serving as an answering machine or receiving facsimile transmissions.

Data processing unit 102 is connected to television 104 for display of graphical information. Television 104 may be any suitable television, although color televisions with an S-Video input will provide better presentations of the graphical information. Data processing unit 102 may be connected to television 104 through a standard coaxial cable connection. A remote control unit 106 allows a user to interact with and control data processing unit 102. Remote control unit 106 emits infrared (IR) signals, preferably modulated at a different frequency from the normal television, stereo, and VCR infrared remote control frequencies in order to avoid interference. Remote control unit 106 provides the functionality of a pointing device (such as a mouse, glidepoint, trackball or the like) in conventional personal computers, including the ability to move a cursor on a display and select items.

FIGURE 1B is a pictorial representation of the front panel of data processing unit 102 in accordance with a preferred embodiment of the present invention. The front panel includes an infrared window 108 for receiving signals from remote control unit 106 and for transmitting infrared signals. Data processing unit 102 may transmit infrared signals to be reflected off objects or surfaces, allowing data processing unit 102 to automatically control television 104 and other infrared remote controlled devices. Volume control 110 permits adjustment of the sound level emanating from a speaker within data processing unit 102 or from television 104. A plurality of light-emitting diode (LED) indicators 112 provide an indication to the user of when data processing unit 102 is on, whether the user has messages, whether the modem/phone line is in use, or whether data processing unit 102 requires service.

FIGURE 1C is a pictorial representation of the rear panel of data processing unit 102 in accordance with a preferred embodiment of the present invention. A three wire (ground included) insulated power cord 114 passes through the rear panel. Standard telephone jacks 116 and 118 on the rear panel provide an input to a modem from the phone line and an output to a handset (not shown). The real panel also provides a standard computer keyboard connection 120, mouse port 122, computer monitor port 124, printer port 126, and an additional serial port 128. These connections may be employed to allow data processing unit 102 to operate in the manner of a conventional personal computer. Game port 130 on the rear panel provides a connection for a joystick or other gaming control device (glove, etc.). Infrared extension jack 132 allows a cabled infrared LED to be utilized to transmit infrared signals. Microphone jack 134 allows an external microphone to be connected to data processing unit 102.

Video connection 136, a standard coaxial cable connector, connects to the video-in terminal of television 104 or a video cassette recorder (not shown). Left and right audio jacks 138 connect to the corresponding audio-in connectors on television 104 or to a stereo (not shown). If the user has S-Video input, then S-Video connection 140 may be used to connect to television 104 to provide a better picture than the composite signal. If television 104 has no video inputs, an external channel 3/4 modulator (not shown) may be connected in-line with the antenna connection.

FIGURE 1D is a pictorial representation of remote control unit 106 in accordance with a preferred embodiment of the present invention. Similar to a standard telephone keypad, remote control unit 106 includes buttons 142 for Arabic numerals 0 through 9, the asterisk or "star" symbol (*), and the pound sign (#). Remote control unit also includes "TV" button 144 for selectively viewing television broadcasts and "Web" button 146 for initiating browsing of the Internet. Pressing "Web" button 146 will cause data processing unit 102 to initiate modem dial-up of the user's Internet service provider and display the start-up screen for an Internet browser. The browser includes a "Favorites" or "Bookmarks" feature that enables the viewer to record the Uniform Resource Locator (URL) for those Web sites that the user desires to revisit.

A pointing device 147, which is preferably a trackpoint or button pointing device, is included on remote control unit 106 and allows a user to manipulate a cursor on the display of television 104. "Go" and "Back" buttons 148 and 150, respectively, allow a user to select an option or return to a previous selection. "Help" button 151 causes context-sensitive help to be displayed or otherwise provided. "Menu" button 152 causes a context-sensitive menu of options to be displayed, and "Update" button 153 will update the options displayed based on the user's input, while home button 154 allows the user to return to a default display of options. One of the options is the Favorites or Bookmarks list. A representative list is shown in FIGURE 3 as a pull-down menu 155 on the television screen. "PgUp" and "PgDn" buttons 156 and 158 allow the user to change the context of the display in display-sized blocks rather than by scrolling. The message button 160 allows the user to retrieve messages.

In addition to, or in lieu of, remote control unit 106, an infrared keyboard (not shown) with an integral pointing device may be used to control data processing unit 102. The integral pointing device is preferably a trackpoint or button type of pointing device. A wired keyboard (also not shown) may also be used through keyboard connection 120, and a wired pointing device such as a mouse or trackball may be used through mouse port 122. When a user has one or more of the remote control unit 106, infrared keyboard, wired keyboard and/or wired pointing device operable, the active device locks out all others until a prescribed period of inactivity has passed.

Referring now to FIGURE 2, a block diagram for the major components of data processing unit 102 in accordance with a preferred embodiment of the present invention is portrayed. As with conventional personal computers, data processing unit 102 includes a motherboard 202 containing a processor 204 and memory 206 connected to system bus 280. Processor 205 is preferably at least a 486 class processor operating at or above 100 MHz. Memory 206 may include cache memory and/or video RAM. Processor 205, memory 206, and system bus 208 operate in the same manner as corresponding components in a conventional data processing system.

Video/TV converter 210, located on motherboard 202 and connected to system bus 208, generates computer video signals for computer monitors, a composite television signal, and an S-Video signal. The functionality of Video/TV converter 210 may be achieved through a Trident TVG9685 video chip in conjunction with an Analog Devices AD722 converter chip. Video/TV converter 210 may require loading of special operating system device drivers.

Keyboard/remote control interface unit 212 on motherboard 202 receives keyboard codes through controller 214, regardless of whether a wired keyboard/pointing device or an infrared keyboard/remote control is being employed. Infrared remote control unit 106 transmits signals which are ultimately sent to the serial port as control signals generated by conventional mouse or pointing device movements. Two buttons on remote control unit 106 are interpreted identically to the two buttons on a conventional mouse, while the remainder of the buttons transmit signals corresponding to keystrokes on an infrared keyboard. Thus, remote control unit 106 has a subset of the function provided by an infrared keyboard.

Connectors/indicators 216 on motherboard 202 provide some of the connections and indicators on data processing unit 102 described above. Other connections are associated with and found on other components. For example, telephone jacks 116 and 118 are located on modem 222. The power indicator within connectors/indicators 216 is controlled by controller 214.

External to motherboard 202 in the depicted example are power supply 218, hard drive 220, modem 222 and speaker 224. Power supply 218 is a conventional power supply except that it receives a control signal from controller 214 which effects shut down of all power to motherboard 202, hard drive 220 and modem 222. In some recovery situations, removing power and rebooting is the only guaranteed method of resetting all of these devices to a known state. Thus, power supply 218, in response to a signal from controller 214, is capable of powering down and restarting data processing unit 102.

Controller 214 is preferably one or more of the 805x family controllers. Controller 214 receives and processes input from infrared remote control 106, infrared keyboard, wired keyboard, or wired mouse. When one keyboard or pointing device is used, all others are locked out (ignored) until none has been active for a prescribed period. Then the first keyboard or pointing device to generate activity locks out all others. Controller 214 also directly controls all LED indicators except that indicating modem use. As part of the failure recovery system, controller 214 specifies the boot sector selection during any power off-on cycle.

Hard drive 220 contains operating system and applications software for data processing unit 102, which preferably includes IBM DOS 7.0, a product of International Business Machines Corporation in Armonk, New York; an operating system such as Windows 3.1 (or higher), a product of Microsoft Corporation in Redmond, Washington; and Netscape Navigator (Version 1.0 or higher), a product of Netscape Communications Corporation in Mountain View, California. Minor modifications of these software packages may be desirable to optimize performance of data processing unit 102. Also, it is highly desirable to update one or more of these "off-the-shelf" programs as well as the other software used by the present invention by downloading new versions of the code via the Internet. Web appliance includes appropriate control software to facilitate such downloading. Hard drive 220 also stores data, such as the list of favorite Internet sites or unviewed downloads from one or more Internet site(s). A cache controller program 225 run by the processor is used to administer and manage these downloads as will be described below.

Modem 222 may be any suitable modem used in conventional data processing systems, but is preferably a 33.6 kbps modem supporting the V.42bis, V.34, V.17 Fax, MNP 1-5, and AT command sets. To maintain the slim height of data processing system 102, modem 222 is preferably inserted into a slot mounted sideways on motherboard 202. Modem 222 is connected to a physical communication link 227, which, in turn, is connected or connectable to the World Wide Web of the Internet (not shown). As is well-known, the World Wide Web is the Internet's multimedia information retrieval system, based on the Hypertext Transfer Protocol (HTTP), which provides users access to files using Hypertext Markup Language (HTML). A link activity monitor 229 determines the extent to which the communication link 227 is being utilized at a given point in time. The link activity monitor may be a hardware-based controller or a software application run by the processor.

A Web server, sometimes referred to as a Web site, supports hypertext documents in directories and files accessible through the Uniform Resource Locator. Typically, all hypertext documents available at a particular Web site are considered part of the same domain (e.g., www.domainname.com). Pages that are local to the domain usually are identified by a relative link, which is a reference to a path and/or filename within the domain, e.g., www.domainname.com/path/html1. A representative Web server is illustrated in FIGURE 7 below.

Those skilled in the art will recognize that the components depicted in FIGURES 1A-1D and 2 and described above may be varied for specific applications or embodiments.

It is desired to enable a user of the data processing system to browse the Web "off-line". This function is provided by the cache controller 225. The cache controller may be a piece of dedicated hardware, or it may be an application program run by the processor. In the preferred embodiment, cache controller 225 is implemented as a software program upgradable through Internet downloads.

As noted above, during one or more on-line browsing sessions, a viewer may compile a list of "favorite" or "bookmark" Web sites that he or she desires to revisit. All or any portion of this list may also be designated for access off-line so that the content of such sites may be downloaded and stored in a dedicated cache of the hard drive for later viewing, preferably off-line. The cache controller program thus includes a control engine 231 (preferably implemented in software run by the processor) for controlling the modem 222 to dial up and connect to the Internet site(s) automatically (e.g., each night while the appliance is unattended). As seen in FIGURE 4, each favorite Web site is associated with a server URL queue 235. A server URL queue 235 is a data structure that identifies the URL of the Web site as well as one or more HTML links spawned from (i.e. located within) the page. Preferably, the server URL queue 235 includes only relative links, although this does not have to be the case in all embodiments. Moreover, although the queue 235 is shown as a dedicated portion of the memory 206, this is not a requirement, as the queue may be a linked list or any other convenient data structure.

In a representative embodiment, the user will not have the ability to set the time period during which the engine will cache Internet site content; rather, this time period is predetermined by the network service provider. Generally, this time period will be restricted, e.g., one (1) hour per night. Therefore, the cache controller program 225 also includes an optimization routine to ensure that the modem 222 is used to its maximum capability during the restricted period of time that Internet sites are cachable to the hard drive 220. Moreover, the optimization routine includes a "load balancing" function to ensure that content identified by the server URL queues is equitably cached during the download period. As will be seen, this enables the viewer to obtain a significant percentage of the downloads that he or she desires.

This optimization routine is now described with reference to the flowchart of FIGURE 5. The primary processing of the routine begins at step 250 with a test to determine whether it is time to stop the process, i.e. whether the predetermined download period has expired. As discussed above, in an exemplary embodiment, this download period is one (1) hour, although it should be appreciated that other time period(s) may be used as well. This period may also be selectively adjusted if desired, but typically not by the user. If the outcome of the test at step 250 is positive, the primary processing routine is complete at step 252. If more time is available, then the routine continues at step 254 to query the activity on the communication link 227 to which the modem 222 is connected. Step 254 determines how much "bandwidth" is being used since a last iteration or cycle (of the routine) by receiving information from the link activity monitor 229. The routine then continues at step 256 to test whether the measured link activity meets some peak usage criteria.

As discussed above, goal of the present system is to maximize download throughput to the cache during the download period. The peak usage criteria generally is dependent on conditions on the communication link, the modem type, or such other criteria as may be predetermined or defined. Thus, for example, peak usage criteria may be defined by an average link utilization for a monitored interval exceeded by some preset limit between 0-100%. Or, the peak usage criteria may be based on some predefined limit on the number of outstanding HTTP GET requests that are issued from the cache manager to the network. A given HTTP GET request is used to request download of the content from a given Web site. Thus, for example, the peak usage criteria may be defined to include: not less than N total outstanding HTTP GET requests, not more than M total outstanding GET requests (M>N), and so on. It will be appreciated that any other convenient "peak usage" criteria may also be used in the comparison at step 256.

If the link activity meets the peak usage criteria, then the modem 222 is being used to its maximum capacity. As a result, the outcome of the test at step 256 is positive and the routine loops back to step 258, which is indicated as a delay. This box reflects that no more content requests are submitted. The routine then returns to step 250, as previously described.

If, however, the link activity does not meet the peak usage criteria, then, in effect, the modem is not being used to its maximum capacity. This is a negative outcome of the test at step 256. As a result, the routine continues at step 260 to obtain another URL from a server URL queue. The particular way in which this is accomplished will be described below, but it will be appreciated that this includes a balancing function to ensure that content identified by the server URL queue(s) 235 is cached equitably during the session. At step 262, a test is made to determine whether a URL was obtained from a queue. If not, the routine continues by testing at step 263 whether the total number of outstanding requests is greater than 0. If the result of the test at step 263 is positive, then the routine returns to the path of the delay 258 and returns. If the result of the test at step 263 is negative, meaning that no more outstanding requests exist, the routine is done and terminates. This outcome would occur, for example, if there were no more unserviced URLs on any server URL queue. If a URL was obtained at step 260, the outcome of the test at step 262 is positive, and the routine continues at step 264 to submit the URL request to the cache controller. Although not described in detail here, it should be appreciated that the cache controller then processes the request in a known manner to initiate the download process. The routine then continues at 266 to increment a count of the number of outstanding requests. This number may be a total for the overall sessions, or a per server URL queue count, or both. After step 266, the routine returns again through the delay loop and recycles until the outcome of the test at step 250 indicates that the download session is complete.

It should be appreciated that the flowchart shown in FIGURE 5 is merely representative of the processing flow. The precise sequence flow illustrated is not meant to be taken by way of limitation. Thus, for example, the step of obtaining a URL (shown as step 260) could be carried out before querying the link activity (at step 254), and so on.

According to a feature of the preferred embodiment, it is desired to "load balance" the content downloads from the "favorite" Web sites so that the user obtains as broad a range of content as is possible during the restricted download period. As a result, Web servers that are slow or busy (even during the off-peak hour) do not impact adversely the downloading process. This is achieved by implementing a so-called "fairness policy" during the download process. In the preferred embodiment, this means that each server on the favorites list has an opportunity to deliver content to the client during the caching session. In particular, some policy designed to achieve equitable caching of the content identified by the one or more server URL queues 235 is implemented for this purpose. The specific policy may be as simple as a "round robin" policy under which a particular URL is taken off each server URL queue (in first in, first out order) irrespective of the number of relative links within a particular queue. Thus, for example, during a first iteration of step 260 discussed above, the first URL is taken from the first server URL queue. At the next iteration, the first URL is taken from the second server URL queue, and so on. Alternatively, the policy may start with a particular server URL queue but then limit the number of outstanding requests per server before requests for another server are used. In this example, the fairness policy would dictate that "never more than x outstanding requests per server" would be used. Thus, the first x URLs (which would include the home page and x-1 relative links) would be taken from the first server URL queue before using the URLs from a second server URL queue, and so on. Another fairness policy could be based on the total number of bytes received from a particular server, irrespective of the number of URL requests generated from a particular server URL queue. In this manner, the policy could restrict the total number of bytes cached from any particular server. Another approach to fairness would be limit the link depth of documents retrieved from a particular server. A link depth limitation would ensure that a page with a large number of nested or embedded links would not unnecessarily bias the download process to the exclusion of other servers. Still another approach would be to identify certain types of sites on the favorites list and serve such sites last because they would otherwise conserve too much of the download cycle. Thus, for example, a directory site (with thousands of links) might be identified and placed on the backend of the service list to ensure that the other sites get their chance to contribute to the download.

Of course, one of ordinary skill will appreciate that the above examples are not mutually exclusive and are not limiting, so that any fairness policy that has a goal of ensuring that each server on the favorites list has at least an opportunity to contribute to the download may be employed (irrespective of whether content is actually received by the client during the session). Thus, the above examples are merely representative of the types of criteria that may be used to implement a fairness policy. One or more of these examples may be combined to provide even more fine tuning of the precise amount of content cached from each of the favorite sites during the download session. The result of this load balancing approach is that each server at least has a theoretical and practical opportunity to contribute content; the user thus obtains a significant percentage of the favorite sites requested, even though some of the particular content may not be downloaded during the actual session. This is typically not a problem, however, since the cache will usually retain a copy (albeit possibly outdated) of the particular content that does not make it down to the cache during a particular download session. Thus, the viewer will still have a "complete" version of the favorite site, and the load balancing process ensures that he or she will have a broader scope of the overall list.

Referring now to FIGURE 6, a flowchart is shown of the process response routine that is run for each GET request submitted to the network by the cache manager at step 264 of FIGURE 5. It begins at step 270 by decrementing the count of the number of outstanding requests, since the particular request has been delivered to the network for service. At step 272, a test is run to determine whether a timer has expired for a particular request without the document being received. If the response to the test at step 272 is negative, the document has been received and the routine continues at step 274 to parse the document links. Although not meant to be limiting, during this step duplicate URLs or links that have already been visited are stripped from the document and thus are not returned back to the server URL queue. In addition, non-local links (except possibly images, e.g., .gif or .jpeg files) are also stripped from the document. Removal of such links ensures that more efficient processing of the document may be achieved during the next download session involving the document. Although the above processing is desired, other types of intelligent processing may be applied to the returned document. Thus, for example, the step may include a command to follow a certain type of string, or to limit the link depth of the document, or to apply a byte-count restriction, and so on. The particular type of intelligent processing is not limited in any way, and any technique that may be useful in facilitating the subsequent download process may be employed. After step 274, the routine is done.

If the outcome of the test at step 272 is positive, which indicates that the document has not yet been received in response to the original request, the routine continues at step 276 to test whether a retry count has run a predetermined number of times. If so, the routine terminates. If not, the routine continues at step 278 to re-queue the request to the head of its particular server URL queue and to increment the retry counter for this particular request. By placing the URL request at the head of the server URL queue, it has a better chance of being re-serviced quickly (depending, of course, on the particular fairness policy being implemented). After step 278, the process terminates.

Turning now to FIGURE 7, a block diagram is shown of a representative Web server platform which supports content to be downloaded to the Web client. This platform is representative of a Web site. The platform 312 comprises an IBM RS/6000 computer 318 (a reduced instruction set of workstation) running the AIX Operating System 320 (Version 4.1 or above) and a Web server program 322, such as Netscape Enterprise Server Version 2.0, that supports interface extensions. The platform 312 also includes a graphical user interface (GUI) 324 for management and administration. The various models of the RISC-based computers are described in many publications of the IBM Corporation, for example, RISC System/6000, 7013 and 7016 POWERstation and POWERserver Hardware Technical Reference, Order No. SA23-2644-00. The AIX operating system is described in AIX Operating System Technical Reference, published by IBM Corporation, First Edition (November 1985), and other publications. While the above platform is useful, any other suitable hardware/operating system/ web server combinations may be used.

The Web Server accepts a client request and returns a response. The operation of the server 322 is governed by a number of server application functions (SAFs), each of which is configured to execute in a certain step of a sequence. This sequence, illustrated in FIGURE 8, begins with authorization translation (AuthTrans) 330, during which the server translates any authorization information sent by the client into a user and a group. If necessary, the AuthTrans step may decode a message to get the actual client request. At step 332, called name translation (NameTrans), the URL associated with the request may be kept intact or it can be translated into a system-dependent file name, a redirection URL or a mirror site URL. At step 334, called path checks (PathCheck), the server performs various tests on the resulting path to ensure that the given client may retrieve the document. At step 336, sometimes referred to as object types (ObjectType), MIME (Multipurpose Internet Mail Extension) type information (e.g., text/html, image/gif, etc.) for the given document is identified. At step 338, called Service (Service), the Web server routine selects an internal server function to send the result back to the client. This function can run the normal server service routine (to return a file), some other server function (such as a program to return a custom document) or a CGI program. Our prime concern here is the situation where the server function runs a CGI program resident on the Web server. At step 340, called Add Log (AddLog), information about the transaction is recorded. At step 342, called Error, the server responds to the client when it encounters an error. Further details of these operations may be found in the Netscape Web Server Programmer's Guide, Chapter 5, which is incorporated herein by reference.

Thus, the Web server includes a known set of server application functions (SAFs). These functions take the client's request and other configuration data of the server as input and return a response to the server as output. Referring back to FIGURE 7, the Web server also includes an Application Programming Interface (API) 326 that provides extensions to enable application developers to extend and/or customize the core functionality thereof (namely, the SAFs) through software programs commonly referred to as "plug-ins".

In the preferred embodiment, the optimization routine comprises a set of instructions (program code) in a code module resident in the memory of the processor of the data processing system. Alternatively, the routine may be program code resident in a random access memory of a computer. Until required by the computer, the set of instructions may be stored in another computer memory, for example, in a hard disk drive, or in a removable memory such as an optical disk (for eventual use in a CD ROM) or floppy disk (for eventual use in a floppy disk drive), or downloaded via the Internet or other computer network. When the code is electronically delivered, a computer program product is said to comprise the program data (electronically delivered) stored in the substrate of a computer-readable storage medium such as the hard drive, floppy disk or other conventional storage media. In addition, although the various methods described are conveniently implemented in a general purpose computer selectively activated or reconfigured by software, one of ordinary skill in the art would also recognize that such methods may be carried out in hardware, in firmware, or in more specialized apparatus constructed to perform the required method steps.

It will be appreciated that the client described herein may be any suitable computer or other device directly or indirectly connected or connectable in any known or later-developed manner to a computer network, such as the Internet, intranet, extranet, etc., and may represent the data processing system described above, taken alone or in combination with a television set or other device.

Further, although the invention has been described in terms of a preferred embodiment in a specific environment, those skilled in the art will recognize that many modifications, variations in hardware and operating system architectures, and so on, are possible. Thus, for example, the mechanism described herein may be implemented conveniently as a plug-in to a conventional browser program operating in a personal computer. Moreover, the preferred processing flow and steps may be altered. Thus, for example, it is not required that the process operate during a time-bounded downloading or caching session. The optimization technique may be conveniently applied in any Internet client that desires to cache content from a plurality of servers according to a fairness policy wherein each of the servers has an opportunity to contribute to the download.

## Claims

1. A method of retrieving content for delivery to a client (100) connectable to a network via a communication link (227), the client including a cache, said method comprising the steps of:

   (a) defining a set of servers from which content is desired to be retrieved and stored in the cache;
   (b) determining (254) an activity level for the communication link as content is being downloaded to the cache from the servers; and
   (c) if the activity level for the communication link is less than a threshold level, issuing (260) to the cache additional requests for content such that each of the servers of the set has an opportunity to deliver content to the client.

2. The method of Claim 1 wherein the additional requests for content are issued to the set of servers as one request per server in an ordered sequence.

3. The method of Claim 1 wherein the additional requests for content are issued to the set of servers up to a predetermined number of requests per server in an ordered sequence.

4. The method of any preceding Claim wherein the additional requests for content are issued to the set of servers based on predetermined parameters.

5. The method of Claim 4 wherein the predetermined parameters include a link depth of a document located on a particular server.

6. The method of Claim 4 wherein the predetermined parameters include the number of bytes received in the cache from a particular server.

7. The method of any preceding Claim further including the step of processing the content received from a server to remove duplicative links.

8. The method of any preceding Claim further including the step of processing the content received from a server to remove non-local links.

9. The method of any preceding Claim wherein the threshold level of the communication link is a given number of outstanding requests for content.

10. The method of any of Claims 1-8 wherein the threshold level of the communication link is related to an average link utilization rate for a given monitored interval.

11. The method of any preceding Claim wherein the content is downloaded to the cache during a predetermined time period.

12. The method of Claim 11, futher comprising the additional step of determining (250) whether a given download period has terminated, wherein said step of determining an activity level for the communication link is performed if the given download period has not terminated; and wherein the additional requests for content are issued to the cache such that each of the servers at least has an opportunity to deliver content to the client during the given download period.

13. The method as described in Claim 12 wherein the given download period is selectable.

14. The method as described in Claim 12 wherein the given download period is approximately one hour.

15. The method of any preceding Claim, wherein said network is the World Wide Web, and said client and servers are a Web client and Web servers respectively.

16. A computer program product for retrieving content for delivery to a client (100) connectable to a network via a communication link (227), the client including a cache, the computer program product comprising: a computer-readable storage medium having a substrate; and a program encoded in the substrate of the computer-readable storage medium, wherein the program comprises means for performing the method of any preceding claim.

17. A computer comprising a modem (222) connected to a communication link (227); a processor (204); a memory (206) including a cache; a browser program running on the processor for providing World Wide Web information retrieval and including means for generating a list of Web sites to be downloaded; a cache control program (225) running on the processor and including means for initiating download requests to the communication link based on the list of Web sites; and a policy control program running on the processor and including

means for issuing multiple HTTP GET requests to the cache control program to ensure that each of the Web sites on the list has an opportunity to contribute content during a given download session.

**18.** The computer of claim 17, wherein the Web browser program includes menu means for generating the list of Web sites to be downloaded during an off-peak download session; the cache control program includes means for receiving downloads that are stored in the cache for off-line browsing; and the cache control program ensures that each of the Web sites on the list has an opportunity to contribute content during a given download session.

**19.** A data processing system (100) comprising a remote control unit (106) and a base unit (102) connectable to a monitor (104) for providing Internet access under the control of the remote control unit, the base unit comprising the computer of claim 17 or 18.

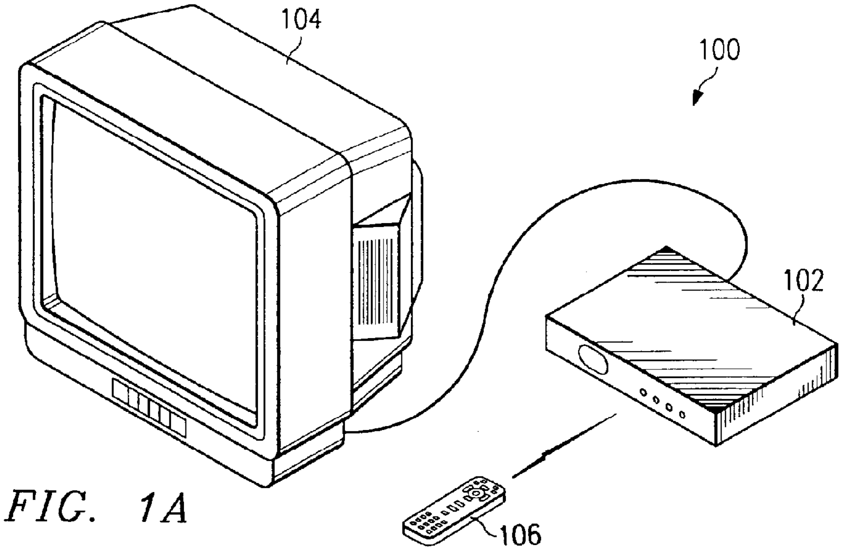

104
100
102
106

FIG. 1A

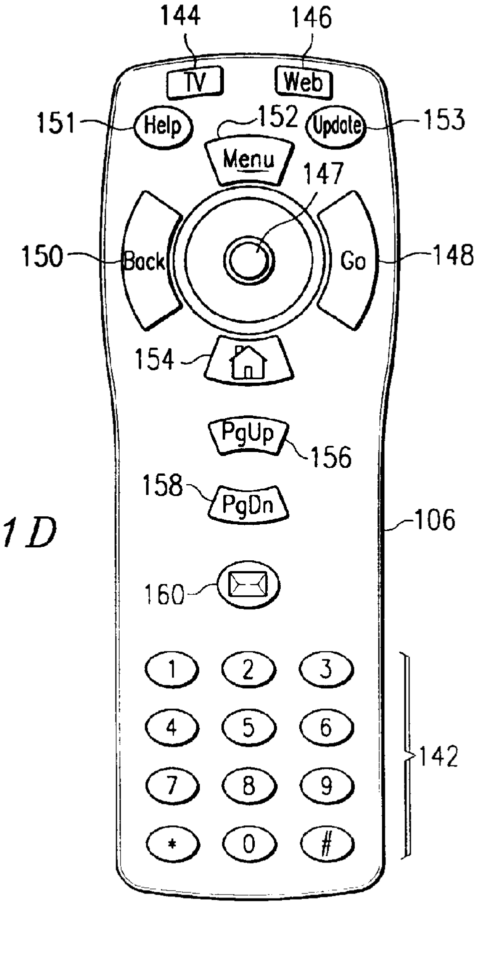

144
146
TV
Web
151
Help
152
Update
153
Menu
147
150
Back
Go
148
154
PgUp
156
158
PgDn
106
160
1
2
3
4
5
6
7
8
9
*
0
#
142

FIG. 1D

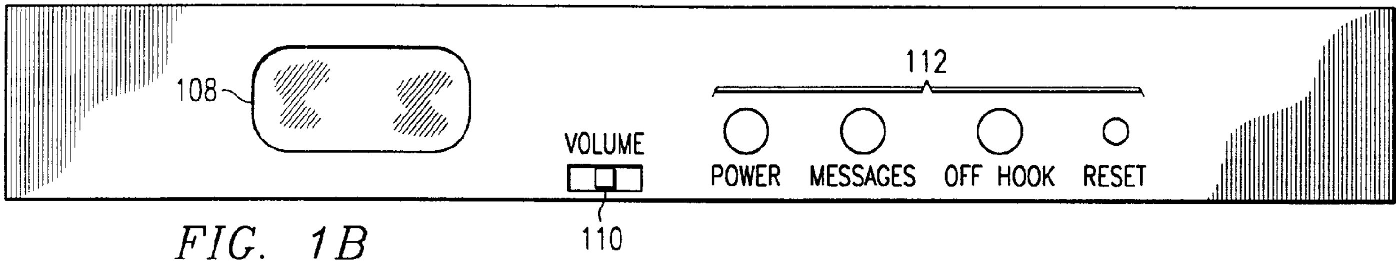

108
VOLUME
110
112
POWER
MESSAGES
OFF HOOK
RESET

FIG. 1B

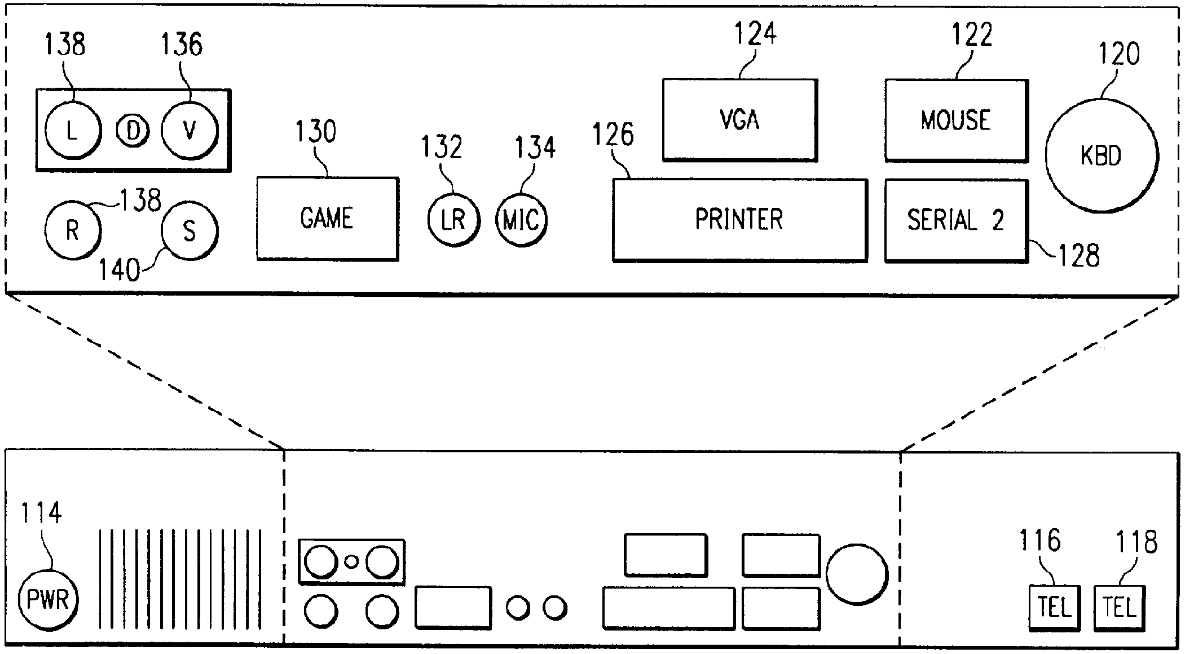

138
136
L
D
V
R
138
S
140
130
GAME
132
LR
134
MIC
126
PRINTER
124
VGA
122
MOUSE
SERIAL 2
128
120
KBD
114
PWR
116
TEL
118
TEL

FIG. 1C

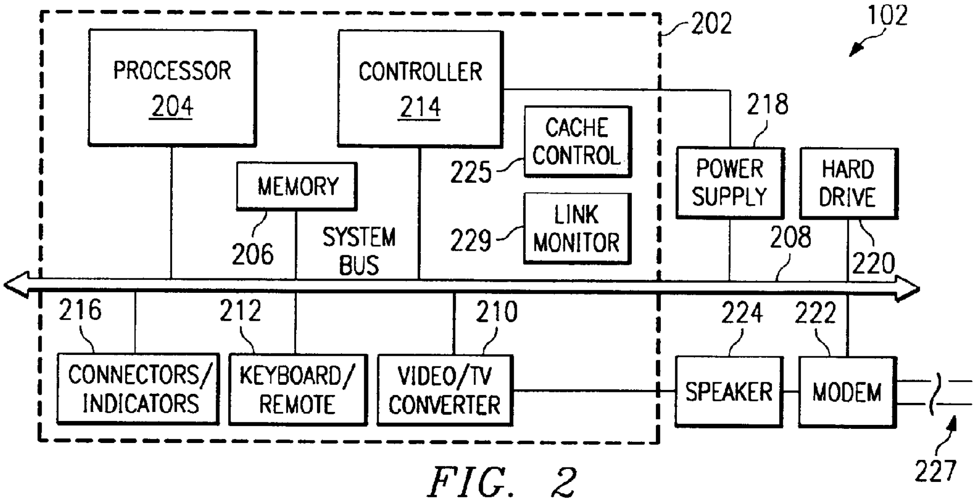
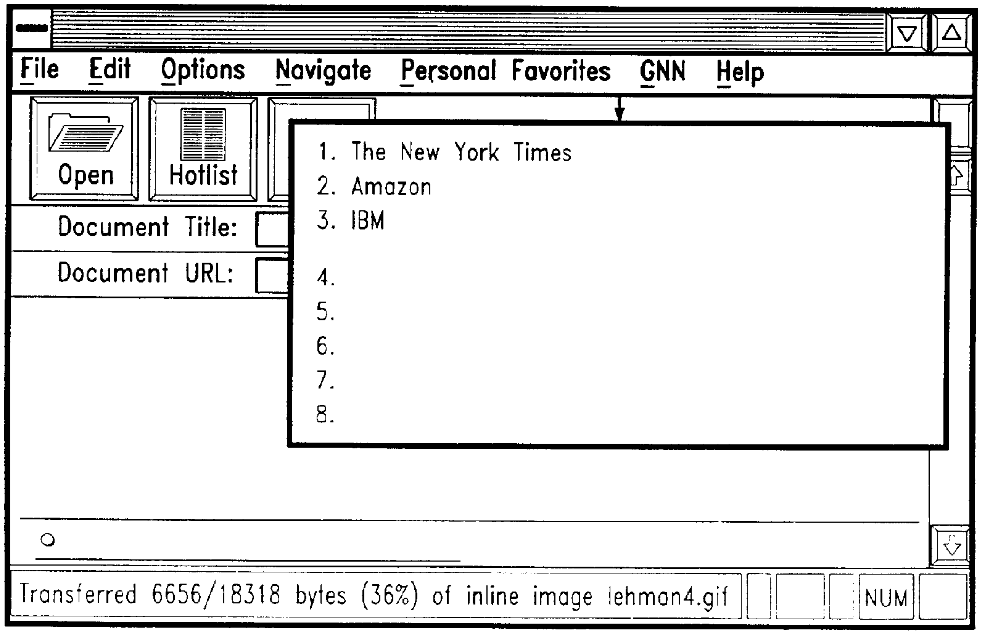


*FIG. 2*



1. The New York Times
2. Amazon
3. IBM
4.
5.
6.
7.
8.

Transferred 6656/18318 bytes (36%) of inline image lehman4.gif

*FIG. 3*

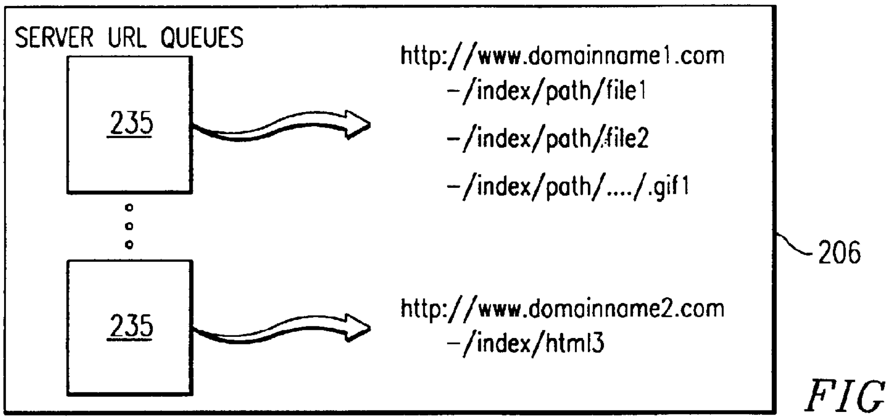

SERVER URL QUEUES
235
235
http://www.domoinname1.com
-/index/path/file1
-/index/path/file2
-/index/path/..../.gif1
http://www.domoinname2.com
-/index/html3
206

*FIG. 4*

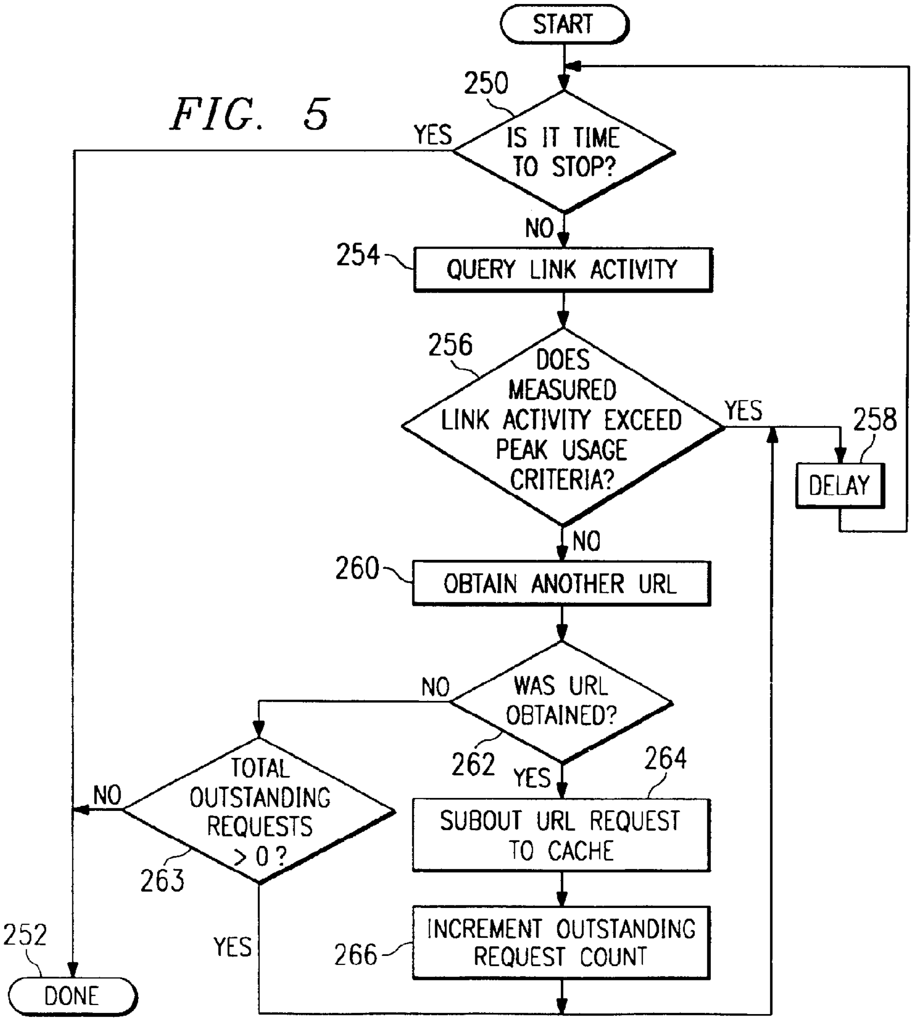

START
250
IS IT TIME TO STOP?
YES
NO
254
QUERY LINK ACTIVITY
256
DOES MEASURED LINK ACTIVITY EXCEED PEAK USAGE CRITERIA?
YES
258
DELAY
NO
260
OBTAIN ANOTHER URL
NO
WAS URL OBTAINED?
262
YES
264
SUBOUT URL REQUEST TO CACHE
TOTAL OUTSTANDING REQUESTS > 0 ?
NO
263
YES
266
INCREMENT OUTSTANDING REQUEST COUNT
252
DONE

*FIG. 5*

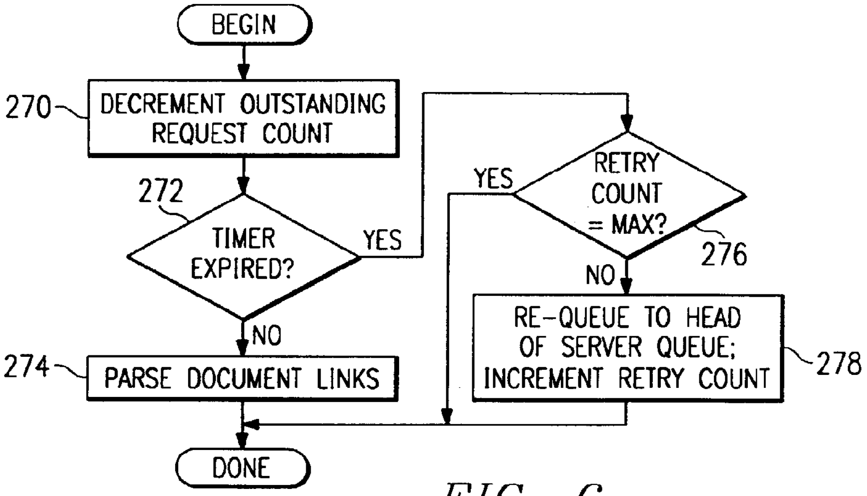

BEGIN
270
DECREMENT OUTSTANDING REQUEST COUNT
272
TIMER EXPIRED?
YES
NO
274
PARSE DOCUMENT LINKS
RETRY COUNT = MAX?
YES
276
NO
RE-QUEUE TO HEAD OF SERVER QUEUE; INCREMENT RETRY COUNT
278
DONE

FIG. 6

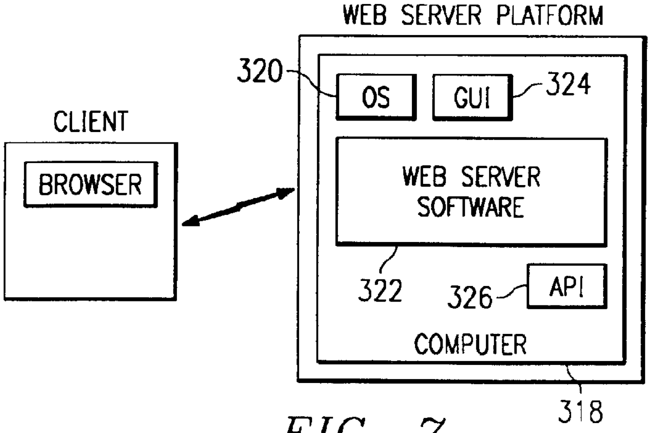


FIG. 7

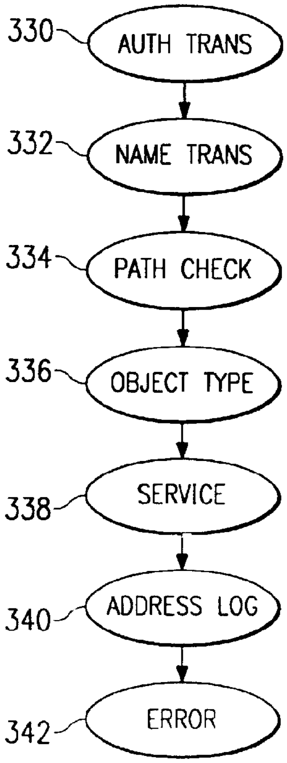


FIG. 8

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 883 068 A2

# (12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
**09.12.1998 Bulletin 1998/50**

(51) Int Cl.$^{6}$: **G06F 17/30**

(21) Application number: **98201847.5**

(22) Date of filing: **28.05.1998**

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE**
Designated Extension States:
**AL LT LV MK RO SI**

(30) Priority: **28.05.1997 US 47998 P**
**21.08.1997 US 915662**

(71) Applicant: **Home Information Services, Inc.**
**New York, NY 10019 (US)**

(72) Inventor: **Ranger, Dennis**
**New York, NY 10019 (US)**

(74) Representative: **Quintelier, Claude et al**
**Gevers & Vander Haeghen,**
**Patent Attorneys,**
**Rue de Livourne 7**
**1060 Brussels (BE)**

## (54) Entity retrieving system and method for retrieving entities

(57) The present invention relates to an entity retrieving system and a method for retrieving entities. The system and method according to the invention comprises mutation means provided for cooperating with a memory, a processor and an interface and for establishing, for each particular entity to be retrieved upon request of a user, if said particular entity pertains to a dependent classes of the class identified by the user, and retrieving, upon establishing that said particular entity pertains to one of said dependent classes of said selected particular class, said additional properties of said dependent class.

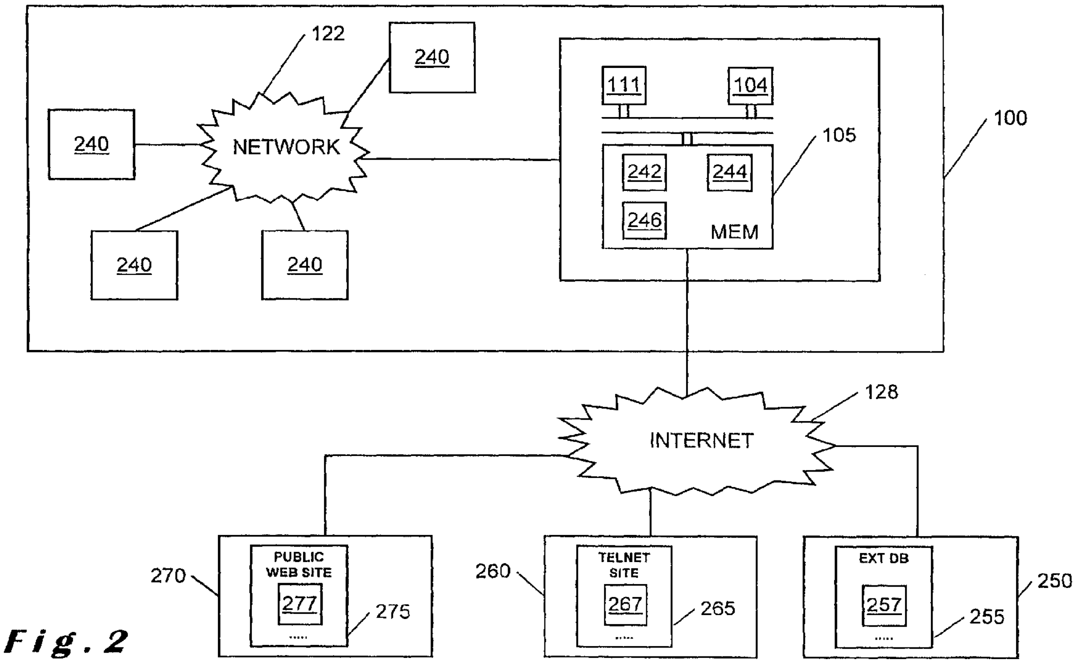


*Fig. 2*

Printed by Jouve, 75001 PARIS (FR)

(Cont. next page)

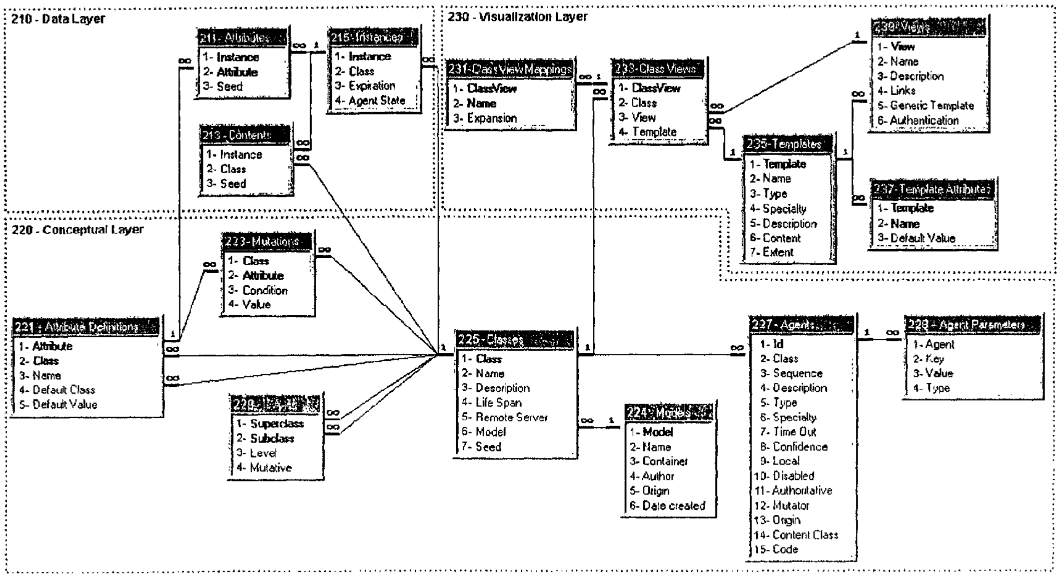

210 - Data Layer
211 - Attributes
1- Instance
2- Attribute
3- Seed
215 - Instances
1- Instance
2- Class
3- Expiration
4- Agent State
213 - Contents
1- Instance
2- Class
3- Seed
230 - Visualization Layer
231-ClassView Mappings
1- ClassView
2- Name
3- Expansion
233-Class Views
1- ClassView
2- Class
3- View
4- Template
239 Views
1- View
2- Name
3- Description
4- Links
5- Generic Template
6- Authentication
235-Templates
1- Template
2- Name
3- Type
4- Specialty
5- Description
6- Content
7- Extent
237-Template Attributes
1- Template
2- Name
3- Default Value
220 - Conceptual Layer
223-Mutations
1- Class
2- Attribute
3- Condition
4- Value
221 - Attribute Definitions
1- Attribute
2- Class
3- Name
4- Default Class
5- Default Value
229
1- Superclass
2- Subclass
3- Level
4- Mutative
225-Classes
1- Class
2- Name
3- Description
4- Life Span
5- Remote Server
6- Model
7- Seed
224 - Models
1- Model
2- Name
3- Container
4- Author
5- Origin
6- Date created
227 - Agents
1- Id
2- Class
3- Sequence
4- Description
5- Type
6- Specialty
7- Time Out
8- Confidence
9- Local
10- Disabled
11- Authoritative
12- Mutator
13- Origin
14- Content Class
15- Code
228 - Agent Parameters
1- Agent
2- Key
3- Value
4- Type

*Fig. 3*

## Description

The present invention relates to an entity retrieving system connectable to at least one data source, said system having a memory into which a plurality of classes are stored, each class defining the structure of said entities, said structure comprising a plurality of property definitions, each property definition identifying property values to be retrieved dedicated to said property definition, said property values being stored in said data sources, said system further comprising an interface and a processor connected to each other, wherein :

(a) said classes comprise at least one dependent class hierarchically linked to at least one other class, said dependent class comprising additional property definitions specifying additional property values, in addition to the property values of the classes from which it depends,
(b) said interface is provided for receiving a query and supplying said query to said processor, said query comprising an identifier for identifying a particular class and at least one of said property values,
(c) said processor being provided, under control of said query, for selecting, among said classes, said particular class dedicated to said identifier, and
(d) said interface is further provided : i) for accessing said data sources, under control of said processor; ii) for retrieving property values pertaining to at least one particular entity that comprise said at least one of said property values; and iii) for outputting the retrieved entities.

Such a system is known from US patent 5,560,005, which enables to transform entities stored in a plurality of remote, heterogeneous structured relational databases into a homogeneous data model. In this patent, several classes are defined, e.g. "SERVICE PROVIDER" forming an abstract class and "HOSPITAL", "PHYSICIAN" and "PRACTISE" forming derived base classes or dependent classes hierarchically linked to the "SERVICE PROVIDER" class. Classes dependent from the dependent classes may also occur. This forms an hierarchic modelled structure of the classes into which property values or data may be incorporated. A query supplied by a user to the known system is formed by a request comprising the class name and search arguments forming thus the identifier to be supplied to the system. The system searches in the remote databases the table entries matching the provided search arguments and returns the property values of the table entries to the requester.

A problem with the known system is that the system will return a number of data, from the found particular entities, i.e. only the properties to be incorporated in that particular class. This means that if the user selects a particular class and the entity to be returned belongs in fact to a dependent class hierarchically dependent from the particular class, the number of properties returned to the user will be limited to the properties to be incorporated in that particular class. If the user wants to check if a particular entity pertains to a dependent class, he has to supply an additional query to the system, wherein the particular dependent class is mentioned. The more dependent classes there are and/or the more entities are found, the more time consuming this checking operation will be.

This problem is solved in the system according to the invention, which system further comprises mutation means provided for cooperating with said memory, said processor and said interface and for establishing, for each particular entity to be retrieved, if said particular entity pertains to one of said dependent classes of said selected particular class, and retrieving, upon establishing that said particular entity pertains to one of said dependent classes of said selected particular class, said additional properties of said dependent class. In case a user requests now for entities belonging to a particular class, the mutation means of the system will check if the found entity also pertains to one of the dependent classes of the particular class and, if this is the case, return the additional properties to be incorporated to the dependent class. The answer to the request of the user will therefore be more complete than expected, without requiring the user to manually check if the entity belongs to a particular class.

Preferably, agents are stored in the memory of the system, each agent being dedicated to at least one of said classes, each agent being provided for cooperating with said interface for accessing at least one particular property value pertaining to said particular entity of said class, each agent comprising : (a) an address for addressing one of said data sources from which said particular property values are to be retrieved; (b) a series of instructions indicating which data from the addressed data source are to be retrieved by said agent; and (c) at least one agent parameter for assigning, for each property value to be retrieved, a portion of said data to one of said property definitions. In particular, a reliability parameter is assigned to said agent.

Such a reliability parameter enables to indicate how trustworthy the addressed data source is. In case for example a first agent retrieves a first entity and a second agent retrieves a second entity, wherein the first entity is identical to the second entity except for one or more property values that are not equal to each other, the system may then, based on the reliability parameter of the agents, decide that one data source has provided a wrong property value or that there is one entity with an uncertainty concerning the property values that do not correspond or that those two entities in fact refer to two different entities.

According to a preferred embodiment, said system further comprises means for displaying the retrieved entities to a user and means for generating, upon receipt of said retrieved entities, at least one list, each list comprising a plurality of property value ranges for subdividing said retrieved entities, and said interface is further provided for enabling said user to select one property value range within one of said lists.

Such lists are very convenient when a large number of entities have been retrieved by the system. With known systems, the user would have to scroll through the results, without having an idea how the entities are structured. In this case, a logical structure of the retrieved entities is presented to the user, wherein the user can select a range. The entities belonging to that range will be shown to the user. This system enables thus to the user to view the results in a more structured way.

The present invention further relates to a method for retrieving entities by using a system according to anyone of the preceding claims comprising the steps of : a) receiving said query and supplying said query to a processor; b) selecting among said classes, said particular class dedicated to said identifier, c) accessing said data sources, under control of said processor, d) retrieving property values pertaining to at least one particular entity that comprise said at least one of said property values and e) outputting the retrieved entities. According to the invention, said method further comprises the step of establishing, for each particular entity to be retrieved, if said particular entity pertains to one of said dependent classes of said selected particular class, and for retrieving, upon establishing that said particular entity pertains to one of said dependent classes of said selected particular class, said additional properties of said dependent class.

The invention will now be described with reference to the annexed drawings, wherein :

Figure 1 illustrates schematically a preferred embodiment of the entity retrieving system according to the present invention.

Figure 2 illustrates schematically from which data sources the entities may be retrieved by the entity retrieving system according to an alternative embodiment of the present invention.

Figure 3 illustrates a database structure according to a preferred embodiment which is used for implementing the system according to the present invention.

Figure 4 illustrates an example of a number of classes with their hierarchy.

Figure 5 illustrates an example of the attributes and contents of a class

Figure 6 illustrates an example of an entity of the class according to Figure 5.

Figure 7 illustrates an example of agents returning inconsistent and ambiguous property values.

Figure 8 illustrates a display supplied to the user when a relatively large number of entities are retrieved by the system according to the present invention.

Figures 9 to 11 illustrate examples of agents.

As illustrated in Figure 1, the entity retrieving system 100 according to the present invention comprises a bus 102 to which a memory 105, an interface 111 and a processor 104 are connected. In particular, the memory is formed by a main memory 106, such as a random access memory (RAM), a read only memory (ROM) 108 and a storage device 110, such as a magnetic disk or optical disk. The interface comprises a display 112 for showing the retrieved entities to a user, an input device 114 such as a keyboard enabling a user to input a query, a communication interface 118 and optionally a cursor control 116 such as a mouse.

The system according to the invention is connectable to data sources, which may be internal to the system, for example data sources stored in the memory 105, and/or external to the system. For external data sources, use is made of the communication interface 118 for connecting the external data source to the system. Data sources may be located in a host device 124, connected to the system through the intermediary of the local network 122 and a network link 120. Data sources may further be stored in a server 130 connected to the network 122 through the intermediary of a communication network such as the Internet 128 which is accessible via an internet service provider (ISP) 126 or the like.

Figure 2 shows an entity retrieving system 100 according to an alternative embodiment, wherein data sources are stored. The retrieved entities may be represented to a plurality of user using a personal computer 240 with a user interface, in particular a web browser. The personal computers are connected, through the intermediary of network 122 to a network computer 126, in particular a web server, enabling a connection to the internet 128. The network computer comprises a memory 105 into which several databases are stored, for example spreadsheets 242, internal web sites 244 or other databases 246. External data source formats may be formed by external databases 257, telnet sites 267 or public web sites 277, stored in respective memories 255, 265 and 275 in respective computer systems 250, 260, 270. It is also conceivable, according to another embodiment of the present invention, to have for example video or sound data sources. The entity retrieving system of the present invention may be applied for retrieving entities stored in data sources having different formats, such as will be described further.

Referring to Figure 3, there is shown a possible database structure used when implementing the entity retrieving system according to the present invention. It forms a structured generic model or a so called "meta model" of a specific modelled structure, an example of which will be described when referring to Figures 4 to 6. It should be clear that this relational database structure is only one possible way to implement the present invention. In Figure 3, there are shown a number of tables, each table having a number of fields, each field defining a certain function. Many to one relationships between the fields are indicated by interconnecting lines with an indication "∞" on the many side and "1" on the one side. For example, instances 215 may have many attributes 211. Accordingly, there is an indication "∞" on the many side of the Instance field 211-1 and an indication "1" on the one side of the Instance field 215-1.

In Figure 3, distinction is made between a conceptual layer 220 forming an intermediary between a data layer 210 and a visualisation layer 230.

CONCEPTUAL LAYER

The conceptual layer 220 comprises fields describing how data is organised within a defined model. The main part of the conceptual layer 220 is formed by the classes 225. A class 225 pertains to a model 224. The model table enables to support multiple models and dependent models. A specific model might be composed of a plurality of dependent models. For example, a banking model might have an accounting and a lending dependent model. The model table 224 comprises fields with the following meaning :

224-1 Model is a unique identifier, in particular a number for identifying the model. For example, the banking model has number "1001", the accounting "1101" and the lending model "1102"

224-2 Name indicates the name of the model in a human readable manner. In the example given above, this field would for example be "banking", "accounting" and "lending".

224-3 Container : If the model is a dependent model, this field contains the unique identifier of the model from which it depends. For example the "accounting" model would have in this field the unique identifier of the banking model, i. e. "1001". It should be clear that a dependent model may be dependent of a dependent model. For example, the accounting model may comprise several dependent models, which would have in this field the number "1101".

224-4 Author indicates the name or a user ID of the person that created this model.

224-5 Origin : If this model has been imported from a data source, the address of the data source is indicated here. In case the data source has been accessed through the Internet, it would be an URL.

224-6 Date Created indicates the creation date of the model.

Each model 224 has at least one class 225. Each class 225 is provided for defining the structure of entities to be retrieved. The class table 225 comprises the fields :

225-1 Class is a unique identifier for identifying the class, in particular a number.

225-2 Name is another unique identifier for the class but in a format convenient for human use. In particular it is formed by a string of characters, e.g. "book" or "product".

225-3 Description is provided for enabling the operator maintaining the system to add annotation and comments for this class.

225-4 Life Span indicates how long, for example in seconds, the entities belonging to that class should be kept in the memory of the system. For example a class in which the price of the entities is retrieved should have a relatively short life span, whereas a class from which the data is not quickly outdated may have a longer life span.

225-5 Remote server : In case a class is defined in another model and/or in a remote data source, the address of this model / data source is mentioned in this field.

225-6 Model contains the model unique identifier 224-1 to which the class is dedicated.

225-7 Seed indicates an Attribute Definition unique identifier, which is the attribute field 221-1 as will be described further, indicating which value is unique for each instance in the class. For example, a book may have as unique value its ISBN or ID number.

Each class can have a plurality of dependent classes or can be a dependent class from a plurality of classes. For this purpose, an Is A table 229 is provided for defining the hierarchy. It comprises the fields :

229-1 Superclass comprising the unique identifier of a class,

229-2 Subclass comprising the unique identifier of a dependent class of the superclass identified in field

229-3 Level showing the number of intermediate classes between the superclass 229-1 and the possibly indirect subclass 229-2, wherein level 0 indicates that the superclass and subclass are equal, level 1 indicates that subclass is a direct dependent class of the superclass, level 2 indicates that there is one intermediate dependent class between the subclass 229-2 and the superclass 229-1, etc.; this multiple level architecture improves the performance of the system and

229-4 Mutative indicating whether or not one or more mutation patterns or agents are dedicated to the class.

Each class has a plurality of attribute definitions. The attribute definitions table 221 comprises property definitions, with the following fields :

221-1 Attribute is a unique identifier for an attribute definition, eg. a number

221-2 Class identifies the class 225-1 that contains this attribute as part of its structure

221-3 Name is an identifier of the attribute in text format

221-4 Default Class is an identifier of another class if the property contains a reference to this other class. For example a supplier attribute in a product class could refer to a supplier class.

221-5 Default Value comprises a default value in case the property value for this attribute is not found.

One or a plurality of mutation patterns can be dedicated to each class. The mutations table 223 comprises mutation pattern portion with one condition, for example book's pages is greater than or equal to 50. A mutation pattern may be formed by a plurality of conditions each condition being defined in a mutation pattern portion, for example book's pages is greater than or equal to 50 and price is less than $10. This table comprises the fields :

223-1 Class is the identifier of the class 225-1 to which the mutation pattern element is dedicated

223-2 Attribute is the identifier of the attribute 221-1 on which a condition applies, for example the attribute pages

223-3 Condition is an operator for example "greater than or equal to", "equal to", "not equal to", ...

223-4 Value is the value to which the property value should be compared, for example "50"

One or a plurality of agents can be dedicated to each class. The agents table 227 comprises the following fields :

227-1 ID is a unique machine readable identifier for the agent

227-2 Class identifies the class to which the agent is dedicated

227-3 Sequence is a number defining sequential order of invocation of the agents for a class (optional) It defines the agent's priority. If two agents are ready to be run, the one with the greater priority has precedence. The lower the sequence, the greater the priority.

227-4 Description is an annotation for providing a human readable description of the agent

227-5 Type specifies whether the agent is an attribute agent or a content agent. An attribute agent is provided for retrieving attributes while a content agent is provided for retrieving contents. The difference between contents and attributes is explained further when referring to the attributes and contents tables 211 and 213.

227-6 Specialty specifies the nature of the data source the agent queries, e.g. ODBC, Web, Corba, Telnet

227-7 Time Out indicates how long an agent should wait when the data source is not responding

227-8 Confidence indicates how trustworthy the property values retrieved from the data source (see origin 227-13) is

227-9 Local indicates whether or not the agent is local; if an agent is local then the agent is only used for the class to which it is dedicated, not in its dependent classes

227-10 Disabled indicates whether the agent is not to be used. This field is used for debugging and diagnostic purposes

227-11 Authoritative : if this field is yes and if this agent receives an empty response to its request, then the entity does not exist

227-12 Mutator indicates whether this agent is a mutation agent

227-13 Origin indicates the data source identity, in particular the path name of the data source from which the property values are to be retrieved

227-14 Content Class identifies the class 225-1 of the references returned by the agent, if the agent is a content agent.

227-15 Code comprises the instructions, forming a parameterised query, to be executed when running the agent

When an agent applies the code 227-15, data is returned comprising the requested property values. Each property value has to be extracted as a portion of the returned data. For this purpose, agent parameters are dedicated to the agent. The agent parameters table 228 comprises the fields :

228-1 Agent identifies the agent 227-1 to which the agent parameter is dedicated.

228-2 Key is a field dependent of the specialty 227-6 of the agent;
for ODBC agents, key is an index (e.g. 1,2,3,...) assigned to each portion of data returned by a query;
for Web agents, key is the identifier of the property definition to which the portion of data will be assigned

228-3 Value is a field dependent of the specialty 227-6 of the agent;
for ODBC agents, value is the identifier of the property definition to which the portion of data identified by the key field will be assigned
for Web agents, value is the pattern used for identifying the portion of data to be extracted and assigned to the property definition indicated by the key field 228-2

228-4 Type is a field dependent of the specialty 227-6 of the agent;
for ODBC agents, this field is not used for Web agents, type is a code indicating whether to

perform pattern matching on the HTML or on the text without the HTML tags.

For the purpose of clarity, an example of agents with parameters is given in Figures 9 to 11, wherein the reference numbers indicate in which field from Figure 3 the values are stored. Figure 9 gives an example of an attribute agent with specialty "ODBC", figure 10 a content agent with specialty "ODBC" and figure 11 an attribute agent with specialty "WEB". The fields provided below specialty are shown in function of the selected type and specialty.

VISUALISATION LAYER

The visualisation layer 230 is provided for comprising data from which a predetermined presentation of an entity is selected and produced. A view is here defined as what a group of users is allowed to see; it is represented as a set of templates attached to classes. It should be noted that some classes can have no template for a given view, meaning that the user has no access to the data requested or that there is a view to be inherited from one classes from which the dependent class depends, or that a default view has been assigned.

A class view table 233 is provided for determining a single template given a view and a class, or a single view given a template and a class or a list of classes given a template and a view. This table comprises the fields :

233-1 Class View is a unique identifier, e.g. a serial number, for a class view
233-2 Class is the unique identifier of the Class 225-1 to which the class view is dedicated
233-3 View is the unique identifier of the View 239-1 to which the class view is dedicated
233-4 Template is the unique identifier of the Template 235-1 to which the class view is dedicated

To each class view, one ore more class view mappings can be dedicated. The class view mappings table 231 holds variable substitution data. When a template is processed, for example as HTML or VRML generation, "value holders" such as "%supplier" are substituted by their values. A value holder can refer by name either to a class defined attribute, a class view mapping variable or a template variable. An attribute has precedence over a class view mapping variable which has precedence over a template variable. In other words, the value of a value holder in a template will default to the value of a template variable only as a last resort. The class view mappings table comprises the fields :

231-1 Class view is a unique identifier of the Class View 233-1 to which the class view mapping is dedicated.
231-2 Name is the name of the variable, for example supplier.
231-3 Expansion is a value of a variable, in particular a template.

Class views are dedicated to views. The view table 239 comprises the definition of a view. A view represents what a group of users is allowed to see. A view is a set of templates assigned to classes. Each template is retrievable from the class view table, given the view and a class. The view table comprises the fields :

239-1 View is a unique identifier for the view.
239-2 Name is the name of the view, for example "Inventory managers"
239-3 Description is provided for holding annotations
239-4 Links is the text of a default link template for the view, which is used when no template of the type link (see 235-3) has been dedicated to a view.
239-5 Generic template is a number identifying a default template 237-1 used when more than one entity is found; although the user has requested for one entity, for example one book; this can occur when there is a "conflict of opinion" as will be explained further.
239-6 Authentication indicates the name of a user's group if the present view is restricted to particular users. A password could be requested for some particular views. This password protection of views is performed with techniques known as such.

Each class view is dedicated to a template. The templates table 235 comprises data related for producing a presentation of an instance of a class, for example HTML, XML or VRML presentations. This table comprises the fields :

235-1 Template is a unique identifier for a template
235-2 Name is a text indicating the name of the template
235-3 Type indicates the type of the template, for example an object template (in particular an item), a space template (in particular a page), a link template (for representing a value of an attribute, in particular a hyperlink), ...
235-4 Specialty determines the presentation medium, for example HTML, VRML, XML, ...
235-5 Description enables the manager of the system to add comments and annotations
235-6 Content gives the actual text of the template, with embedded value holders
235-7 Extent indicates spatial dimensions for three dimensional objects or spaces for VRML presentations

Each template may comprise a number of template attributes. The template attributes table 237 comprises template variables used in value substitution, as explained with reference to the class view mappings table 231. The template attributes table comprises the fields :

237-1 Template is the unique identifier of the template 235-1 to which the template attribute is dedicated
237-2 Name identifies the name of the variable
237-3 Default value comprises, if applicable a default value of that variable

DATA LAYER

In order to improve response time, instances or entities retrieved by the system according to the present invention are preferably cached. The content and attributes of instances are stored separately. This enables to cache for example the attributes of an instance and not its contents or vice versa. An instance is cached no longer than is permitted by its class's life span 235-4.

The instance table 215 is provided for holding the instances or entities cached by the system according to the present invention. This table comprises the fields :

215-1 Instance is a unique identifier of an instance
215-2 Class is the identifier of the class 225-1 to which the entity or instance pertains
215-3 Expiration indicates the moment when a cached instance expires. This moment is calculated on the moment the data is retrieved plus the life span indicated in the life span field 225-4.
215-4 Agent state is a list of agent identifiers 227-1 that were used for retrieving the cached contents or attributes of an instance.

The contents table 213 comprises the content, retrieved by means of a content agent, of cached instances. A content is a list of references to instances of a given class. The content table comprises the fields :

213-1 Instance is the identifier 215-1 of the cached instance to which the content pertains
213-2 Class is the identifier of the class 225-1 of the cached content
213-3 Seed is a value from a list of values that makes up the content of an instance, that, together with the content class, specifies an instance of that class. The content seeds form a portion of property values to be retrieved.

The attributes table 211 comprises the attributes, retrieved by means of a content agent, of cached instances. This table comprises the fields :

211-1 Instance is the identifier 215-1 of the cached instance to which the content pertains
213-2 Attribute is the identifier of the attribute definition 221-1 to which the cached attribute pertains to
213-3 Seed is the cached value or property value of the attribute.

Figure 4 illustrates an example of a store model having a plurality of classes : "PRODUCT", "BOOK", "AUDIO TAPE" and "BEST-SELLER". Classes "BOOK" and "AUDIO TAPE" are dependent classes from the "PRODUCT" class. A best-seller is a particular book. Therefore, the "BEST-SELLER" class depends from the class "BOOK". In case a class "VIDEO TAPE" has to be added to the store model, it can easily be added as a dependent class of the "PRODUCT" class.

Each class has a number of property definitions, in particular attribute and content definitions. Referring to Figure 5, the "PRODUCT" class has for example the attribute definitions 221 with name 221-3 "ID", "SUPPLIER" and "TYPE", wherein ID is a unique identifier for a particular product, the supplier attribute is a reference to a supplier from the "SUPPLIER" class, and type describes the type of the product. ID and Type identify property values for simple data, e.g. numbers or strings, whereas Supplier identifies a property value referring to another entity, e.g. a Supplier entity pertaining to the Supplier class. The default class 221-4 field for this property definition will therefore be "SUPPLIER". The "BOOK" class is a dependent class from the "PRODUCT" class and inherits therefore all the property definitions from the classes from which it depends. In addition, it comprises additional property definitions, such as illustrated in Figure 5, i.e. the attributes "Title", "Author" and the contents "EDITION", "REVIEW", "CHAPTER 1", "CHAPTER 2".

Attributes are "single-valued" in the sense that each attribute has only one value. The title of a book is a single piece of data. Content properties, on the other hand, refer to open-ended lists of or references to other entities. For example, Chapters and Reviews are content properties of a book; they list the book's chapters and reviews. Content properties can also be inherited. For example, the Sales content property of a Book could be inherited from Product.

Figure 6 illustrates an example of a book entity having the following property values : "93-21123" as ID, "Doubleday" as Supplier, "Book" as Type, "War & Peace" as Title and "Tolstoy, Leo" as Author. The content properties refer to other entities. This entity is found by the system according to the invention after the user has input a request, for example :

http://www.server.com/query.pl?Product=93-21123&View=Customer For such a query, it is assumed that a product can only be retrieved based on its ID number. In this case, the seed field 225-7 would indicate the unique identifier of the attribute definition for ID number. The search possibilities could be enhanced by permitting to indicate in a query any property value pertaining to a class. The request would have then for example the following format :

http://www.server.com/query.pl?Product. ID=93-21 123&View=Customer

The interface 111 receives this query and supplies it to the processor 104 of the system. The processor will select from the memory the "Product" class, since the user has keyed in Product as particular class. An agent dedicated to the Product class, is provided for retrieving the Supplier and Type property values based on the ID number. These property values are for example stored in an internal data source, for example a relational database 246 (see Figure 2). The agent comprises an address in field Origin 227-13 indicating the path name of the database 246 data source. In order to enable to retrieve data from different types of data sources, there are provided different types of agents. For a relational database such as Oracle ®, the agent is an ODBC agent type. The agent further comprises a series of instructions indicating which data from the addressed data source are to be retrieved by the agent, for example :

"Select Key, Type, Supplier FROM Products"

The agent further comprises in its agent parameters 228 for assigning, for each property value to be retrieved, a portion of the data to one of the property definitions. In this case, "Key" is assigned to "ID" property definition, "Type" to "Type" property definition and "Supplier" to "Supplier" property definition.

This agent co-operates with the interface 111 for accessing the data source, under control of the processor 104 and for retrieving the requested data. In the example mentioned hereinabove, the following data will be returned : "93-21123" forming the ID, "Doubleday" forming the Supplier and "Book" forming the type. This data is stored in the memory of the system and will be provided to the user such as in known entity retrieving systems.

According to the invention, the system further comprises mutation means. The system will in particular check if mutation patterns or mutation agents are dedicated to one of the dependent classes. This can in particular be performed by verifying, for each dependent class, if the mutative field 229-4 is yes. In this case, there is checked if there are mutation patterns or mutation agents dedicated to the classes "Book" and "Audio Tape". A mutation pattern dedicated to the classes book comprises for example the condition : "If the Product Type = "book" then mutate the Product into a Book". The system thus verifies if the found property value for the product type falls within the condition. For this purpose, the processor compares the property values stored in the memory with the condition of the mutation pattern. In this case the retrieved property for the product type is a "Book". Consequently, a mutation occurs, wherein the class of the entity becomes "Book" and wherein the system will retrieve the additional property values pertaining to the class "Book".

In particular, retrieving the additional property values is again performed by one or several agents dedicated to this class. For example, an agent will be provided for retrieving a book's author and title from a web site given the book's ID. Since the data will be retrieved from a web site, the agent type is in this case "Web". The address in the Origin field is in this case an URL of the web site where the data should be retrieved, the instructions form in this case the steps required for accessing a web page where the requested data is shown. This is performed by providing the ID number "93-21123" and assigning this number to a corresponding parameter on the web site, for example LCCN Number. The agent further comprises agent parameters, by means of which pattern matching is performed, wherein the property values are extracted from web content by applying regular expressions, a known technique i.a. from "Mastering Regular Expressions" by Jeffrey E. F. Friedl, January 1997. For example the title property value could be found by searching on the page the expression "TITLE :" and looking for a series of words after the expression and located between spaces. It will be clear that this technique is applicable to all web pages having a predetermined structured presentation of the data. The ID is for example mentioned after the expression "LCCN NUMBER :" and the author after the expression "Author/Other Name :". In the example given, the following additional property values are retrieved by the system : "War & Peace" as title and "Tolstoy, Leo" as author.

Another agent could for example be provided for finding books chapters in a Telnet site given the book's ID. A further agent could be provided for accessing a relational database where book reviews are gathered from multiple sources, given a book's title and author. It should also be clear that the agents dedicated to a class to which the dependent class is hierarchically linked are also used. In this case, agents dedicated to the class products have already been processed, since book is a particular product and the user input the class "product" in his query. Now, the system has retrieved additional data, i.e. additional property values such as the title of the book, the author, reviews and chapters. This additional data is supplied to the user, although he had requested data relating to a product. The answer to the request of the user is automatically completed by the system according to the present invention with additional data which was unexpected by the user.

When several agents are provided for retrieving, from different data sources, property values that should correspond, it may occur that some property values retrieve are not equal to each other. For example, a customer's telephone number may be recorded differently in two data sources, or there might be three different authors for the same book title. In the first case, it is probable that the same customer has two phone numbers (an inconsistency), in the second case, we may be dealing with three altogether different books (an ambiguity).

Inconsistencies and ambiguities are unavoidable when integrating multiple data sources that were not conceived together and that may not even be managed by the same organisation. It is an object to deal in an appropriate way with ambiguities and inconsistencies within data. The manner in which the system according to the invention deals with those problems will be explained by means of an example.

Assume that agents are looking for a Person named Bob Smith. An agent A looks for a person's address given the person's name. Two agents B and C look for a person' age given the person's name, each agent targeting a separate data source. This example is illustrated in Figure 7.

The agent A returns with not one but two "Bob Smith", one living in New York and the other in Newark. The question is: Are we looking at two Bob Smith or only one but with a conflicting address? The answer depends on how much we trust agent A to be accurate or, in other words, how much we trust its data source to return correct addresses. For this purpose, a reliability or confidence parameter 227-8 is assigned to the agent. Let's say we trust agent A for 100%. Then we have no choice but to see two Bob Smith and two entities are thus shown to the user. In case agent A has a confidence parameter of only 10%, then the system will show only one entity showing two possibilities for a property value, e.g. "New York OR Newark".

Assume now agent A has a 100% reliability parameter. Agent B and C for the Bob Smith in New York obtain his age. Both agree that it is 35. However agents B and C for the Bob Smith in Newark disagree about his age. Agent B indicates 24 and agent C 27. Are we looking at two Bob Smith living in Newark? In this case, Agents B and C have been declared as fallible. That they disagree is not sufficient grounds to see two separate Bob Smith living in Newark; so we simply record the fact that there is a "conflict of opinion" between data sources about the age the Bob Smith living in Newark. Because of ambiguities and inconsistencies, a request to the system according to the invention to find an entity may end up returning more than one entity, with some "conflicts of opinion" about some of them. When this occurs, the user is presented with a display using the generic template 239-5 for the requested view, e.g. a Web page, that gives a choice between these entities and highlights conflicts.

In case agents B and C have substantially a same reliability parameter, which is relatively low, for example 10%, then one entity will be presented to the user with an indication of two property values for the age : "24 OR 27", such as illustrated in Figure 7. In case agents B and C have substantially the same reliability parameter, which is relatively high, for example 90%, then the systems interprets that there are two distinct entities as being two separate entities which will be presented to the user, each with its own age. In case agent B is substantially more reliable than agent C, for example agent B is at least 25 % more reliable than agent C, then the system will rely on the property value retrieved by agent B, i.e. 24, and only the entity retrieved having this value will be presented to the user.

Consequently, by providing a reliability parameter to the agents, the system interprets inconsistencies and ambiguities in property values, filters out unreliable property values and/or presents it in an appropriate fashion to the user. It should be noted that this provision of a reliability parameter could also be applied in other systems, in particular in systems without mutation means, and in general to any retrieving system provided for retrieving data from one or a plurality of data sources.

Referring to Figure 4, there is shown a dependent class "Best-seller" dependent from the class "Book". Now that the product requested by the user has been determined by the system as being a book, the system will further verify if the found entity should not further be mutated to the class "Best-seller". A mutation agent, i. e. an agent the mutator field 227-12 of which indicating that the agent is a mutation agent, which agent is dedicated to the class "Best-seller" is for example provided for accessing an external database or web site comprising a best-seller list. Based for example on the book's title and the author, the agent will search, in the same manner as explained hereinabove, in the addressed data source for the requested data. If the agent finds the requested book in the database, this means that the entity is a best-seller and the found entity is mutated to the class "Best-seller" as explained hereinabove. On the other hand, if the requested entity is not present in the best-seller data source, for example the agent receives a message such as "Could not be located", this signifies that the book is not a best-seller and therefore a mutation will not be performed.

The example of the mutation pattern described hereinabove had a single value condition with an operator "equal to". In general, it should be noted that all types of conditions are conceivable, with all types of operators such as "larger than", "between ... and ...". A list of single values is also conceivable. A mutation pattern could also have as condition "If all the other mutation patterns from the same level fail, then mutate to this dependent class". Such a mutation pattern signifies that the class to which the dependent classes of that level are hierarchically linked, is an abstract class, i.e. a class for which comprise no entities or instances. In the example given, if the store only sells books and audio tapes, the product class could then be an abstract class, since every entity is either a book or an audio tape. The mutation pattern for audio tape would be "If product type is not a book then mutate to audio tape class".

A further aspect of the present invention is the visualisation of the retrieved entities, in particular which property values are presented to the user based on its request. In the given example, the user has input in its query : "View=Customer", wherein all the found properties, except the supplier would be shown to the user. Another view could for example be a view for the staff, which view would require a password and provide in addition to the property values mentioned (including the supplier property value) an indication of the stock of the product in the store. An additional agent or above mentioned agent provided for retrieving the supplier and the type would be provided for retrieving in the internal database of the store, the numbers in stock of the requested product.

Based on the requested view, corresponding to views name 239-2, and the requested class corresponding to class name 225-2, a dedicated class view 233 is determined by the system, having a dedicated template 233-4. The content field 235-6 of the template table 235 will supply the instructions for producing the requested view to a user.

In general, once an entity has been recovered from the data sources, it can be shown to the user who requested it. How the entity is shown and how much of it is shown depends on the template that's used to generate the presentation. Each template is dedicated to a class. A dependent class can inherit presentations from its classes from which it depends or can define its own templates.

There are different kinds of templates, including :

a page template to display an entity as a full page, for example, a Book page will show the book's title, author, price, availability etc.;

an object template to display a summary of an entity in the page of another related entity, for example, a Book page would also show a list of its Reviews, each one summarized as a few lines and displayed using their object template; and

a link template to display an attribute value, in particular a hyperlink to an entity, for example, the summary of a Review listed in a Book page would include a hyperlink to the Review page where the full review can be read.

A user display, in particular a web page, is constructed on the fly from the templates. Each template is in fact a parameterised web page or VRML scene or other presentation to a user. The blanks are filled by the values of the entity's attributes and content properties before display. When the value filling a "blank" is a reference to another entity, then a hyperlink to that entity is automatically inserted. Hyperlinks are thus always correct and current, reflecting the data in the data sources at the time of the request.

If no template for a view on a class is defined (or inherited) for a given group of users, then no user belonging to that group can see entities of that class. For example, there would be no views defined on the class Inventory for customers, meaning that customers could not peruse the Inventory. If an attribute of a class is not shown in a template for a given group, then no member of that group can see the value of that attribute. For example, the views for class Employee would include the Salary attribute only for the Manager group (meaning that only managers can see employee salaries).

Between the moment an user requests to view an entity and the moment that entity is viewed for example as a Web page or a VRML scene or other presentation, a lot can happen as the relevant agents access data sources, triggering further agents until eventually all activated agents are done. Instead of waiting for all agents to have completed before displaying a Web page, a more dynamic approach may preferably be applied, a technique which is known as "server push", showing a Web page as soon as some data is available about the requested entity and then refreshing the page automatically when new data is retrieved. In particular, when mutation is applied, the user will first see data relating to the requested class and upon refreshing, the user will also see the additional property values. This way, the user does not have to wait too long to get a feedback and may elect to follow a hyperlink before the entity is completely shown and while the incomplete page is waiting to be refreshed with additional property values.

In a preferred embodiment, there is first checked which view is requested by the user and determined which properties should be supplied to the user. Based on this determination, only the agents required for the requested view are triggered, in order to supply more quickly the requested data to the user.

A further aspect of the present invention relates to supplying the results of a query input by the user in a more structured manner. Queries supplied to the system may lead in a large number of entities to be retrieved and presented to the user. With known systems, the user would be confronted with long lists of results, in particular hyperlinks, to scroll through.

The system according to the present invention takes advantage of the fact that the retrieved entities are dedicated to a structured model and thus that it has some understanding of entities. It is thus possible to organise long lists of entities. All entities belong to a class with defined properties. Using that knowledge, the system takes a long list and splits into smaller lists. Each smaller list represents entities falling within some range for a property value. For example, for a big list of Employees, the system could break the list down according to employee ID. A first sub-list would contain references to employees with IDs less than 236 and the other smaller list would contain references to employees with IDs greater than 342. This example is illustrated in Figure 8.

Since there is more than one property definition that can be used to create smaller lists, the system offers alternate subdivisions of the oversized content. This is in particular performed for each of the property values that may be viewed by the user. As illustrated in Figure 8, smaller lists are created for the first name, last name, city and state.

If the smaller sub-list is still too large for comfort, the system applies the same operation again on the sub-list, until the user reaches a list small enough to be laid out in full. This is performed automatically for example by assigning to the system a predetermined maximum number of entities that the sub-list may not be exceeded.

Sometimes the amount of data that would be returned by an agent is so large that the system can only accept some of it from the data source and must discard the rest. Imagine what could happen if an agent returned a million references to Customers. The manageable portion that is accepted by the system is displayed using the same technique. As the user accesses restricted subsets of the original list, a more specific query is sent, yielding a smaller number of references. Again that smaller number, for example 100,000, may still be too large to be all taken in by the system. Eventually though, when the user has navigated to a narrow enough subset, the highly constrained query will return a complete yet manageable set of answers which can all be accepted and displayed by the system according to the present invention.

Consequently, such a subdivision of the results or automatic indexing provides to the user a logical structure of the retrieved entities, wherein the user can select a range. It should be noted that this provision could also be applied in other systems, in particular in systems without mutation means, and in general to any displaying system for displaying large sets of data.

In summary, when a user supplies a query to the user, the following steps are performed, according to a preferred embodiment of the present invention:

1. The query is received by the interface and supplied to the processor
2. If required, the user is asked to enter a user ID and a password for authentication, in particular when the requested view is a password protected view
3. The query is processed by the system
4. A plurality of data sources, addressed by the agents, are accessed to retrieve data pertaining to the requested entities, which data is mapped into property values as defined in the agent parameters of the agent.
5. The found entities are presented to the user according using templates dedicated to the requested view. In particular, this step may be performed before the previous step has been finished, in such a manner that the page presented to the user is dynamically updated when more property values are retrieved. In case the number of results is too large, then the list of results is subdivided in smaller list indexed according to several property values and presented to the user. The user may select one of these smaller lists.
6. Meanwhile, there is checked if mutation of each found entity should occur, using mutation patterns and mutation agents. If mutation occurs, additional property values are retrieved and presented to the user, by dynamically updating the users screen.
7. A new query may be input by the user to the system, in particular by selecting a link in the presentation supplied to the user.

The system according to the present invention improves the manner in which data is extracted from data sources, integrated into a model and presented to the user, in particular when the data has to be retrieved from a plurality of data sources which may have different formats.

## Claims

**1.** An entity retrieving system connectable to at least one data source, said system having a memory into which a plurality of classes are stored, each class defining the structure of said entities, said structure comprising a plurality of property definitions, each property definition identifying property values to be retrieved dedicated to said property definition, said property values being stored in said data sources, said system further comprising an interface and a processor connected to each other, wherein

(a) said classes comprise at least one dependent class hierarchically linked to at least one other class, said dependent class comprising additional property definitions specifying additional property values, in addition to the property values of the classes from which it depends,
(b) said interface is provided for receiving a query and supplying said query to said processor, said query comprising an identifier for identifying a particular class and at least one of said property values,
(c) said processor being provided, under control of said query, for selecting, among said classes, said particular class dedicated to said identifier, and
(d) said interface is further provided

i) for accessing said data sources, under control of said processor,
ii) for retrieving property values pertaining to at least one particular entity that comprise said at least one of said property values and
iii) for outputting the retrieved entities,

characterised in that
said system further comprises mutation means provided for cooperating with said memory, said processor and said interface and for establishing, for each particular entity to be retrieved, if said partic-

ular entity pertains to one of said dependent classes of said selected particular class, and retrieving, upon establishing that said particular entity pertains to one of said dependent classes of said selected particular class, said additional properties of said dependent class.

**2.** A system according to claim 1, wherein

(a) said mutation means comprise a mutation pattern dedicated to said dependent class, said mutation pattern comprising at least one condition, each condition assigning at least one predetermined property value range to one of said property definitions of said class to which said dependent class is hierarchically linked, and
(b) said mutation means are further provided for establishing if said particular entity pertains to said dependent class by verifying if the property value dedicated to said property definition of said particular entity falls within said predetermined property value ranges.

**3.** A system according to claim 1 or 2, into which memory agents are stored, each agent being dedicated to at least one of said classes, each agent being provided for cooperating with said interface for accessing at least one particular property value pertaining to said particular entity of said class, each agent comprising :

(a) an address for addressing one of said data sources from which said particular property values are to be retrieved,
(b) a series of instructions indicating which data from the addressed data source are to be retrieved by said agent,
(c) at least one agent parameter for assigning, for each property value to be retrieved, a portion of said data to one of said property definitions.

**4.** A system according to claim 3, wherein said mutation means are further provided

(a) for establishing if said agent comprises a mutation indicator indicating that said agent is a mutation agent, and
(b) for establishing if said particular entity pertains to said dependent class by verifying if said agent has established that said at least one particular property value pertaining to said particular entity is present, upon establishing that said agent comprises said mutation indicator.

**5.** A system according to claim 3 or 4, wherein a reliability parameter is assigned to said agent.

**6.** A system according to anyone of the claims 3 to 5, wherein said agent comprises a speciality indicator for indicating the type of the addressed data source.

**7.** A system according to anyone of the claims 3 to 6, wherein said agent comprises an authoritative indicator for indicating that if the property values retrieved by said agent are empty, then an indication that the entity does not exist is shown to a user of the system.

**8.** A system according to anyone of the preceding claims, said system further comprising means for displaying the retrieved entities to a user, said system further comprises means for generating, upon receipt of said retrieved entities, at least one list, each list comprising a plurality of property value ranges for subdividing said retrieved entities, and said interface is further provided for enabling said user to select one property value range within one of said lists.

**9.** A system according to anyone of the preceding claims, into which memory a plurality of models are stored, wherein each of said classes are assigned to one of said models.

**10.** A method for retrieving entities by using a system according to anyone of the preceding claims comprising the steps of:

(a) receiving said query and supplying said query to a processor;
(b) selecting among said classes, said particular class dedicated to said identifier,
(c) accessing said data sources, under control of said processor,
(d) retrieving property values pertaining to at least one particular entity that comprise said at least one of said property values and
(e) outputting the retrieved entities,

characterised in that said method further comprises the step of

(f) establishing, for each particular entity to be retrieved, if said particular entity pertains to one of said dependent classes of said selected particular class, and
(g) retrieving, upon establishing that said particular entity pertains to one of said dependent classes of said selected particular class, said additional properties of said dependent class.

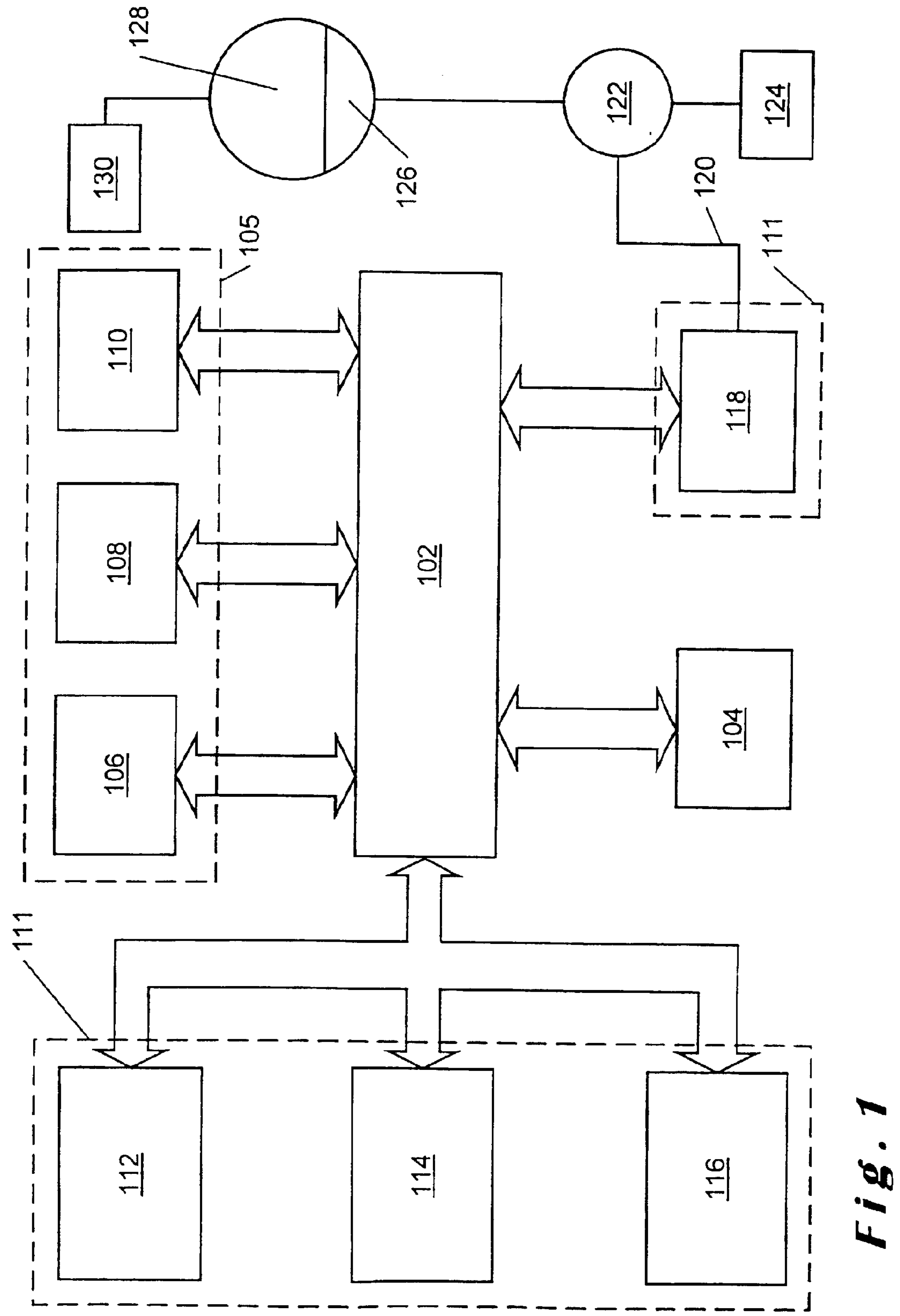

128
130
126
122
124
120
105
111
110
108
106
102
118
104
111
112
114
116

Fig. 1

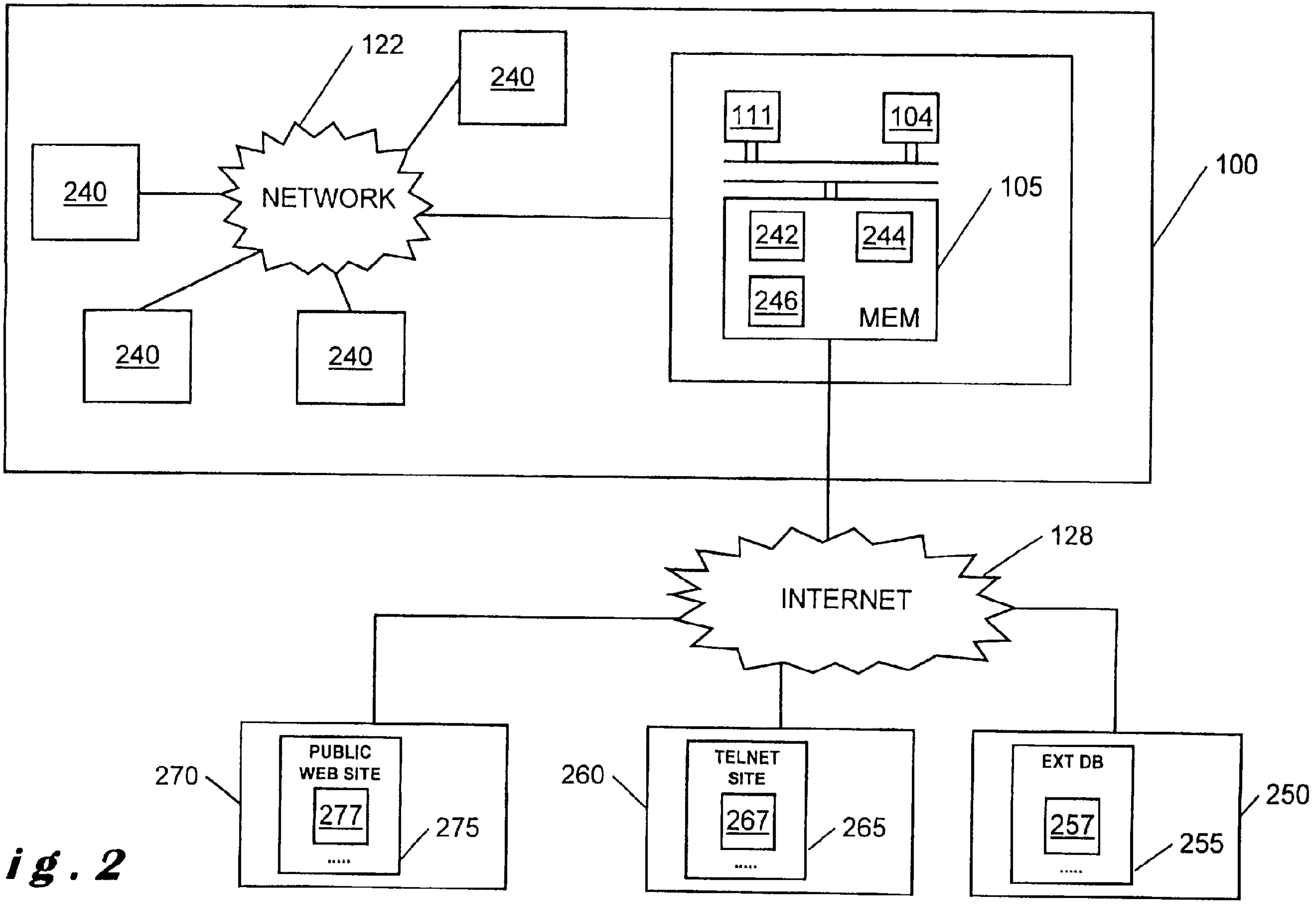

100
122
240
240
240
240
NETWORK
111
104
105
242
244
246
MEM
128
INTERNET
270
PUBLIC
WEB SITE
277
.....
275
260
TELNET
SITE
267
.....
265
250
EXT DB
257
.....
255

Fig. 2

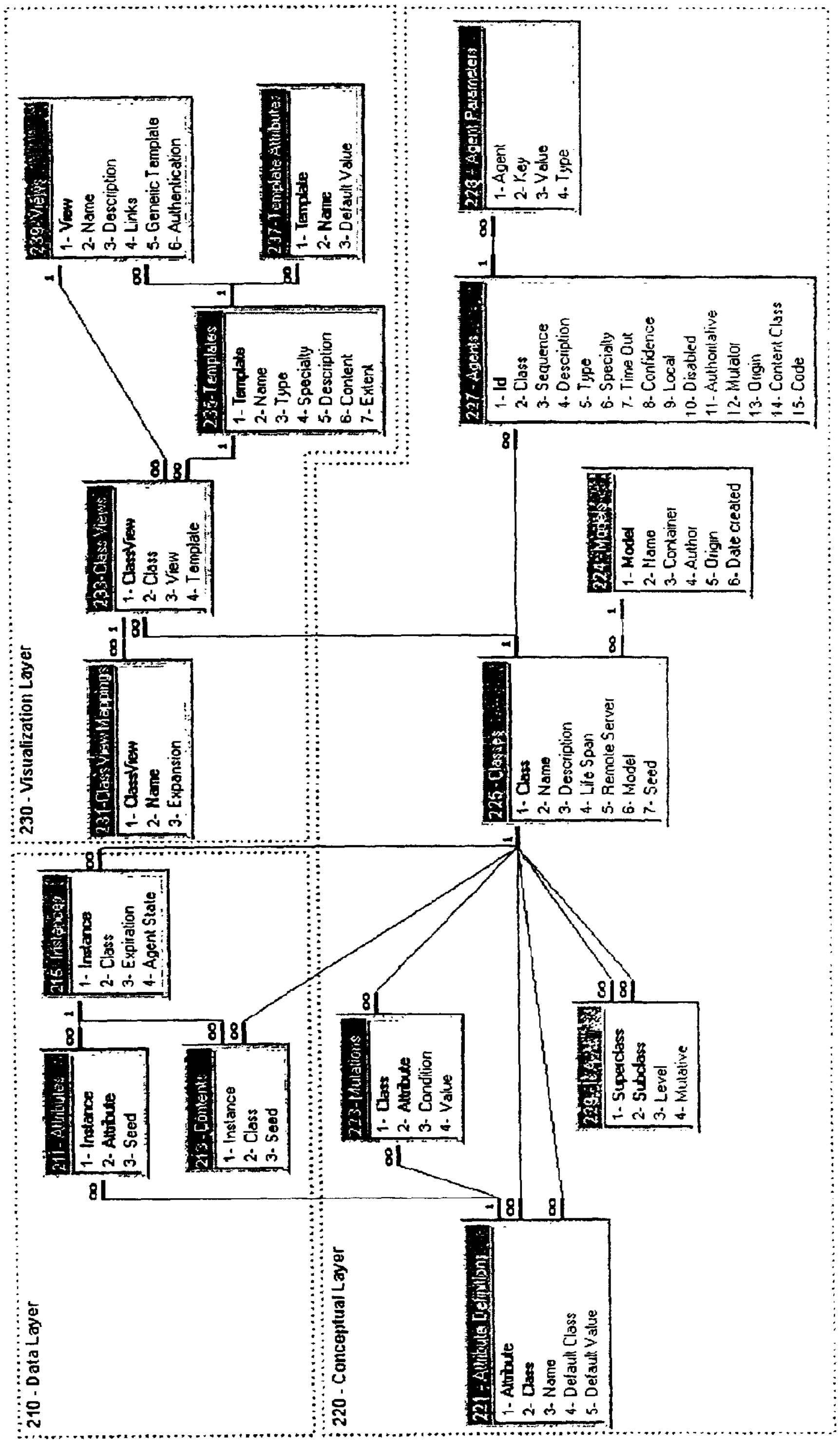

210 - Data Layer
220 - Conceptual Layer
230 - Visualization Layer
211 - Attributes
1- Instance
2- Attribute
3- Seed
213 - Contents
1- Instance
2- Class
3- Seed
215 - Instances
1- Instance
2- Class
3- Expiration
4- Agent State
221 - Attribute Definitions
1- Attribute
2- Class
3- Name
4- Default Class
5- Default Value
223 - Mutations
1- Class
2- Attribute
3- Condition
4- Value
229 -
1- Superclass
2- Subclass
3- Level
4- Mutative
225 - Classes
1- Class
2- Name
3- Description
4- Life Span
5- Remote Server
6- Model
7- Seed
224 - Models
1- Model
2- Name
3- Container
4- Author
5- Origin
6- Date created
227 - Agents
1- Id
2- Class
3- Sequence
4- Description
5- Type
6- Specialty
7- Time Out
8- Confidence
9- Local
10- Disabled
11- Authoritative
12- Mutator
13- Origin
14- Content Class
15- Code
228 - Agent Parameters
1- Agent
2- Key
3- Value
4- Type
231 - Class View Mappings
1- ClassView
2- Name
3- Expansion
233 - Class Views
1- ClassView
2- Class
3- View
4- Template
235 - Templates
1- Template
2- Name
3- Type
4- Specialty
5- Description
6- Content
7- Extent
237 - Template Attributes
1- Template
2- Name
3- Default Value
239 - Views
1- View
2- Name
3- Description
4- Links
5- Generic Template
6- Authentication
1
∞

*Fig. 3*

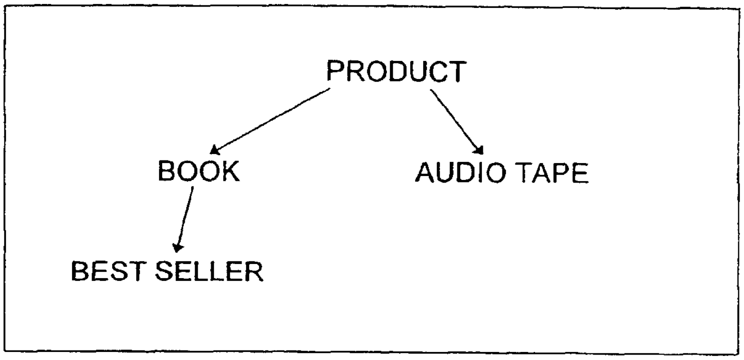


Fig. 4

BOOK

| ATTRIBUTES | DEFAULT CLASS |
|---|---|
| (product) ID | - |
| (product) Supplier | Supplier |
| (product) Type | - |
| Title | - |
| Author | - |

| CONTENTS | DEFAULT CLASS |
|---|---|
| Edition | Edition |
| Review | Review |
| Chapter 1 | Chapter |
| Chapter 2 | Chapter |

Fig. 5

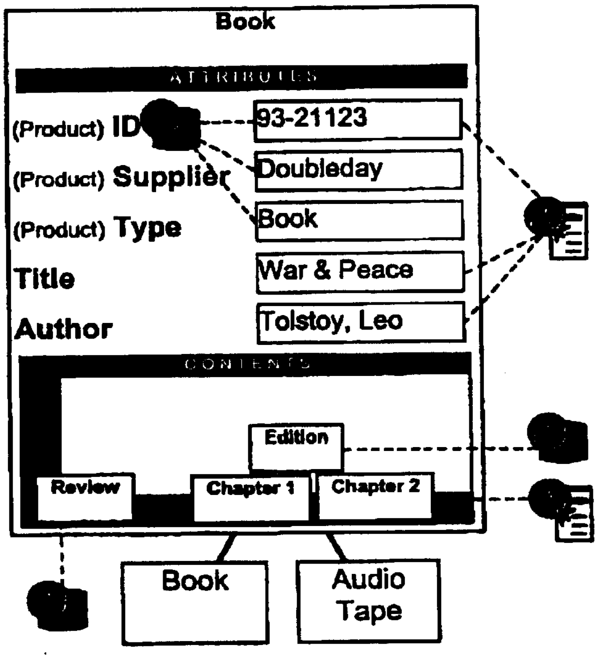

Book
ATTRIBUTES
(Product) ID
(Product) Supplier
(Product) Type
Title
Author
93-21123
Doubleday
Book
War & Peace
Tolstoy, Leo
CONTENTS
Edition
Review
Chapter 1
Chapter 2
Book
Audio Tape

*Fig. 6*

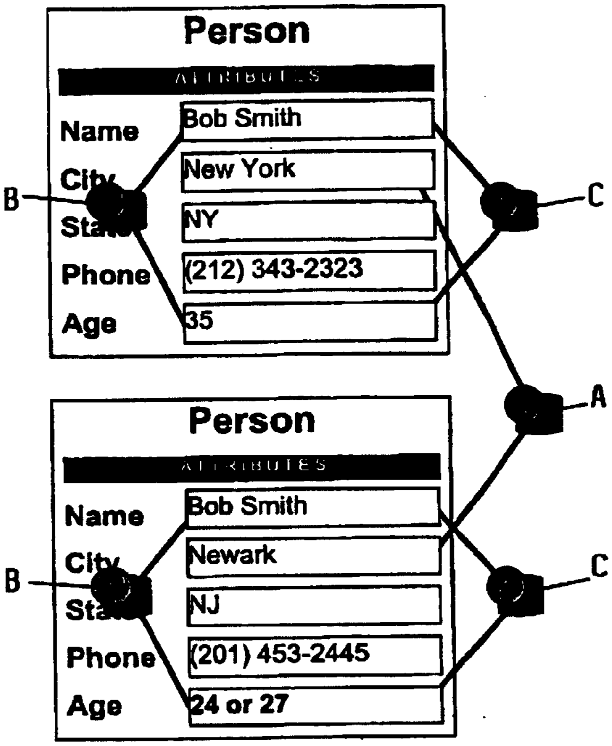
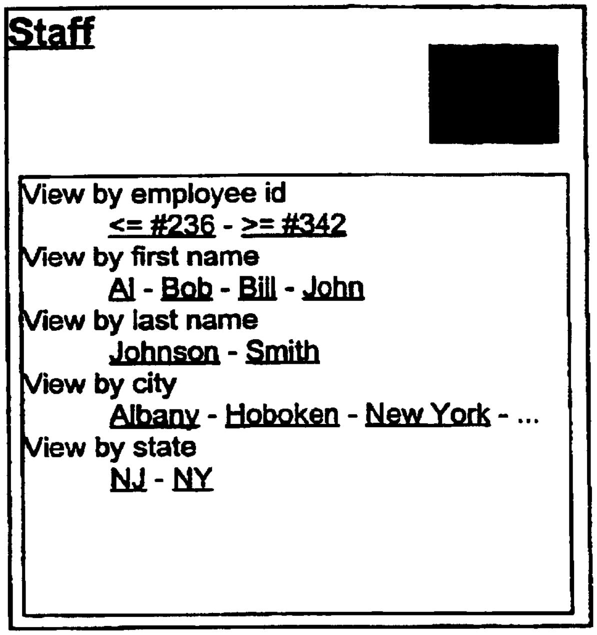

Person
ATTRIBUTES
Name
Bob Smith
City
New York
State
NY
Phone
(212) 343-2323
Age
35
Person
ATTRIBUTES
Name
Bob Smith
City
Newark
State
NJ
Phone
(201) 453-2445
Age
24 or 27
A
B
B
C
C

*Fig. 7*



Staff

View by employee id

<= #236 - >= #342

View by first name

Al - Bob - Bill - John

View by last name

Johnson - Smith

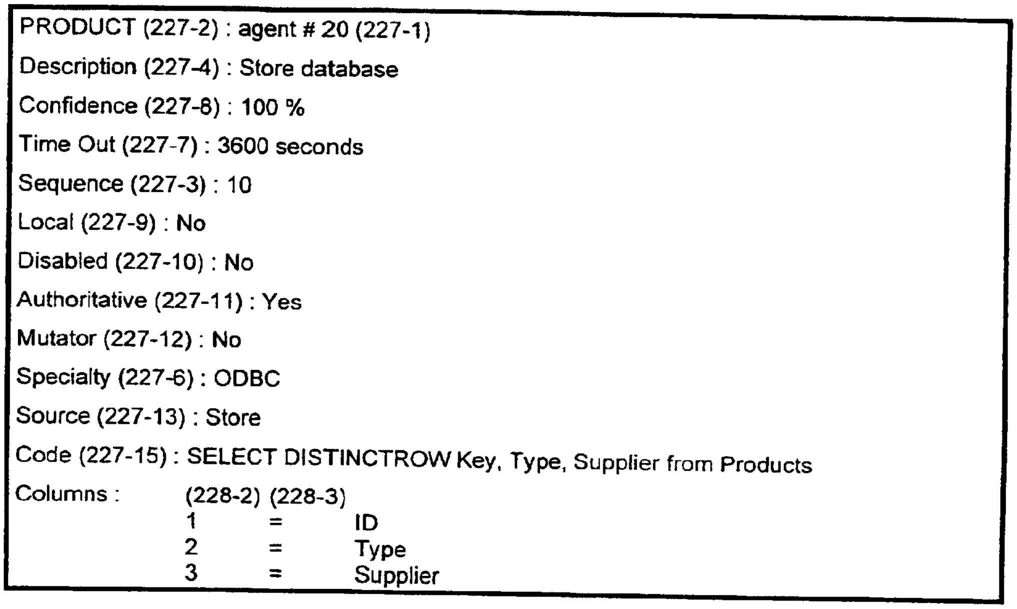
View by city

Albany - Hoboken - New York - ...

View by state

NJ - NY

*Fig. 8*



PRODUCT (227-2) : agent # 20 (227-1)

Description (227-4) : Store database

Confidence (227-8) : 100 %

Time Out (227-7) : 3600 seconds

Sequence (227-3) : 10

Local (227-9) : No

Disabled (227-10) : No

Authoritative (227-11) : Yes

Mutator (227-12) : No

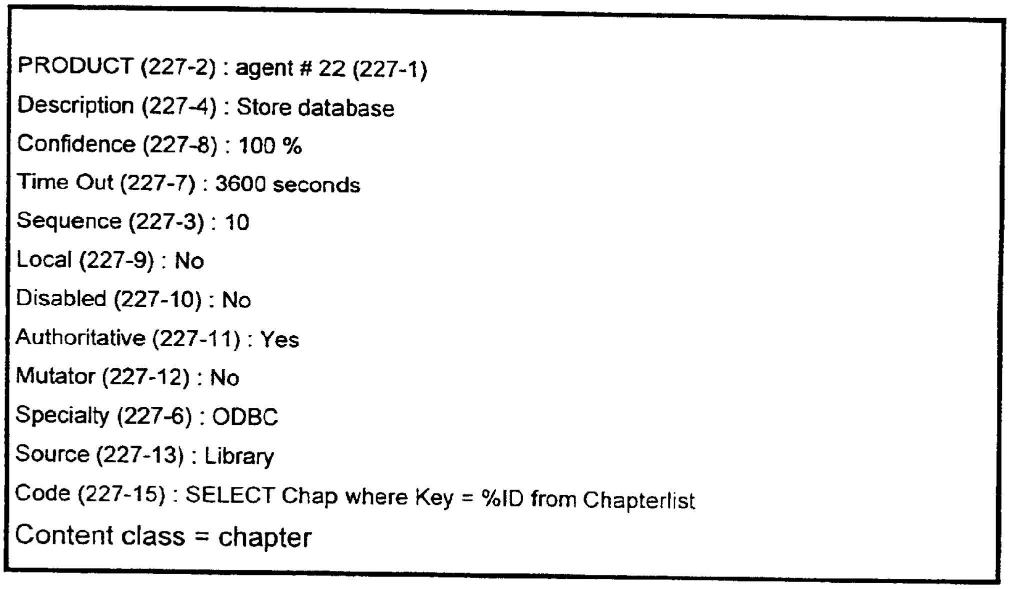
Specialty (227-6) : ODBC

Source (227-13) : Store

Code (227-15) : SELECT DISTINCTROW Key, Type, Supplier from Products

```
Columns :     (228-2) (228-3)
              1     =     ID
              2     =     Type
              3     =     Supplier
```

*Fig. 9*



PRODUCT (227-2) : agent # 22 (227-1)

Description (227-4) : Store database

Confidence (227-8) : 100 %

Time Out (227-7) : 3600 seconds

Sequence (227-3) : 10

Local (227-9) : No

Disabled (227-10) : No

Authoritative (227-11) : Yes

Mutator (227-12) : No

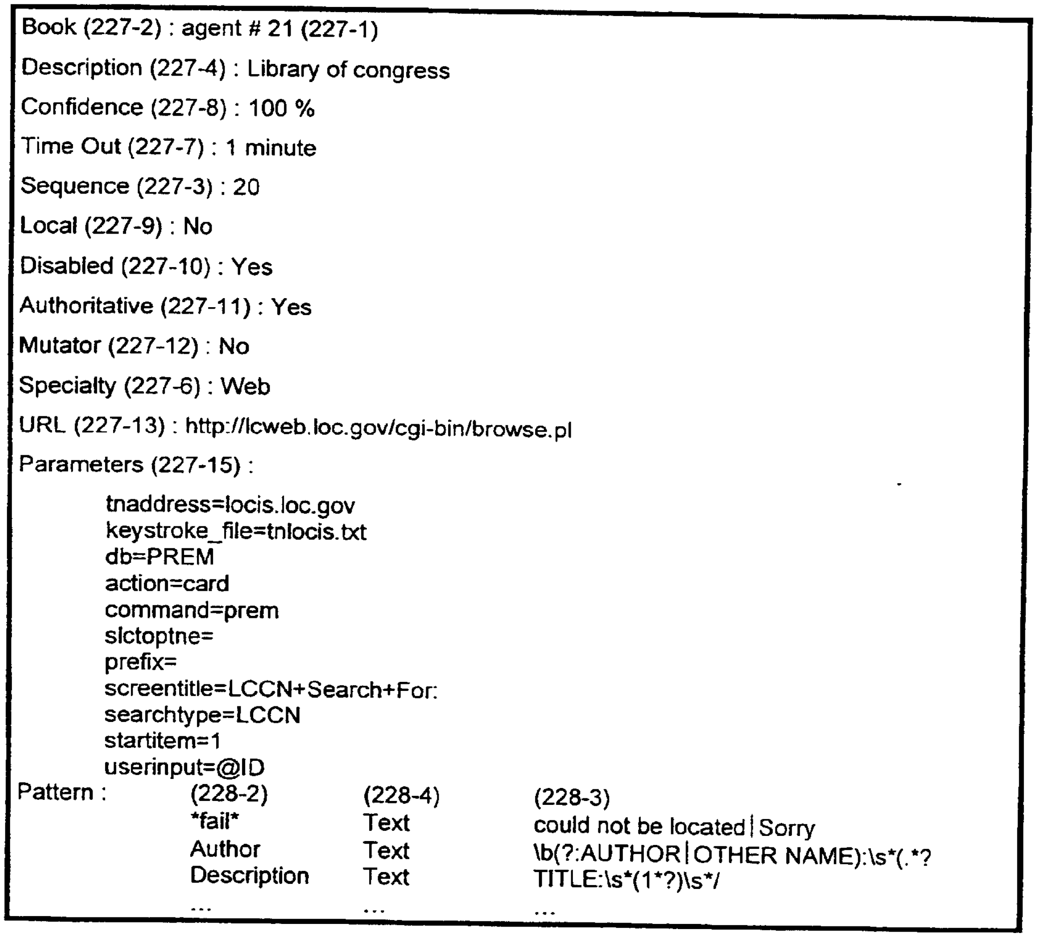
Specialty (227-6) : ODBC

Source (227-13) : Library

Code (227-15) : SELECT Chap where Key = %ID from Chapterlist

Content class = chapter

*Fig. 10*



```
Book (227-2) : agent # 21 (227-1)
Description (227-4) : Library of congress
Confidence (227-8) : 100 %
Time Out (227-7) : 1 minute
Sequence (227-3) : 20
Local (227-9) : No
Disabled (227-10) : Yes
Authoritative (227-11) : Yes
Mutator (227-12) : No
Specialty (227-6) : Web
URL (227-13) : http://lcweb.loc.gov/cgi-bin/browse.pl
Parameters (227-15) :
        tnaddress=locis.loc.gov
        keystroke_file=tnlocis.txt
        db=PREM
        action=card
        command=prem
        slctoptne=
        prefix=
        screentitle=LCCN+Search+For:
        searchtype=LCCN
        startitem=1
        userinput=@ID
Pattern :       (228-2)        (228-4)        (228-3)
                *fail*         Text           could not be located|Sorry
                Author         Text           \b(?:AUTHOR|OTHER NAME):\s*(.*?
                Description    Text           TITLE:\s*(1*?)\s*/
                ...            ...            ...
```

*Fig. 11*

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 886 409 A2

# (12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
23.12.1998 Bulletin 1998/52

(51) Int. Cl.$^6$: **H04L 29/06**, G06F 1/00

(21) Application number: **98107899.1**

(22) Date of filing: **30.04.1998**

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE**
Designated Extension States:
**AL LT LV MK RO SI**

(30) Priority: **01.05.1997 JP 113939/97**

(71) Applicant:
**Digital Vision Laboratories Corporation**
**Minato-ku, Tokyo (JP)**

(72) Inventor: **Muratani, Hirofumi**
**Takatsu-ku, Kawasaki-shi (JP)**

(74) Representative:
**Lins, Edgar, Dipl.-Phys. Dr.jur.**
**Gramm, Lins & Partner GbR,**
**Theodor-Heuss-Strasse 1**
**38122 Braunschweig (DE)**

## (54) Information providing system

(57) An information providing system comprises an encryption unit for encrypting content data using a first key. The first key is included in message data which is associated with the content data and is separately transmitted to a user site. The message data is also encrypted using a second key within a security module. The second key is further encrypted using a third key within the security module. The third key is never read out to the outside of the security module.

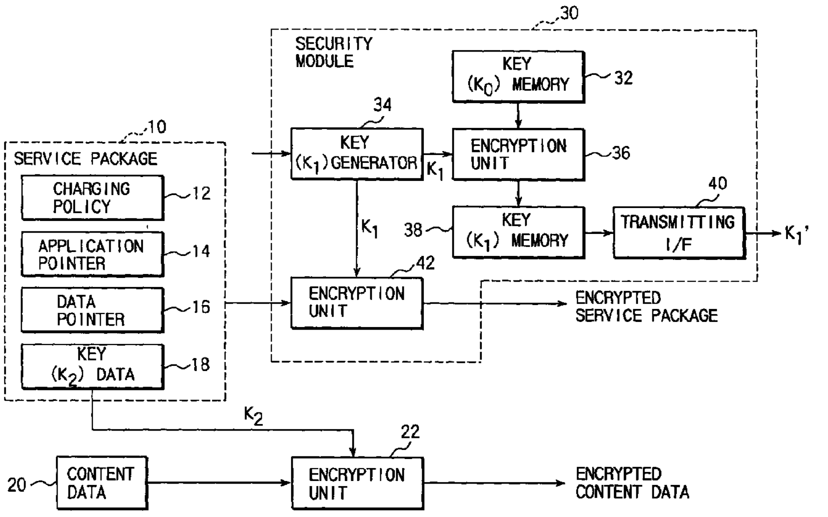


FIG.2

EP 0 886 409 A2

Printed by Xerox (UK) Business Services
2.16.6/3.4

## Description

The present invention relates to an information providing system and more particularly to an information providing system which allows information to be provided readily while ensuring information protection.

The present application is based on Japanese Patent Application No. 9-113939, filed May 1, 1997, the content data of which is incorporated herein by reference.

With the advance of the Internet and the development of large-volume storage media such as DVDs and the like, various information providing services have been implemented regardless of online or offline. The information providers, which are companies that provide information as business, charge users for usage of information.

Forms of charging include charging on the basis of the amount of time that information is used, charging on the basis of the amount of information (the number of bytes) that is used, charging on the basis of a unit of information (for example, a movie), and so on. Under the present circumstances, only the forms of charging determined by the information providers are implemented. To be specific, in a closed system in which dedicated hardware is used, as in a cable television broadcasting service, a charging program is described in an application program installed in a data processing terminal (computer) on the user side or a server on the information provider side. In order to change the form of charging, therefore, it is required to rewrite the application program itself. It is thus not so easy to change the charging form. To provide a variety of forms of charging, it is desirable to add a new form or forms of charging and allow users to make a choice from the set forms of charging. However, this needs to modify the program considerably.

In addition, with the recent evolution of multimedia techniques, a case is also increasing in which one user subscribes to a number of information providers and receives a number of information providing services. In this case, an application program will be needed for each individual information providing service. The conventional charging facility, which is contained in an application program, cannot be commonly used in different application programs. For this reason, when an information service provider creates a new application program, it is also required to create a new charging program. However, the charging function is inherently independent of application programs and should be able to be used in common to different application programs. Preparing a charging program for each application program offers drawbacks that the program developing time is useless and each program increases in size and complexity.

The inventor of the present application proposed previously a system in which the usage of information (data processing function) and the charging process are separated from each other, the former being implemented by an application program, and the latter being implemented by a platform that differs from the application program (Japanese Patent Application No. 8-259,433). Here, the information provider separates information to be provided (hereinafter referred to as content data) or information, such as addresses, that identify content data and control information (referred to as a service description) required to utilize the content data or the information providing service. The service description includes information for identifying an application program utilizing content data to be provided, information for identifying a charging policy associated with the utilization of the content data, and information indicating a key needed to decrypt the content data in encrypted form. An example of the service description is such that a video playback application "A" is needed to utilize a video data "B", the charge for that video data "B" is 1000yen and the user must pay a fee to Mr. "C" in accordance with a settlement method "D". In this manner, users are allowed to utilize content data on the basis of the service description.

If the service description remained unprotected on the information transmitting path from an information provider to a user or at the user site, the service description might be altered. In such case, the information provider would be unable to collect a charge and have its digital rights infringed. The digital rights include a copy right or a counterpart right for service creation or service provision which should be belonged to the provider.

The provider's digital rights include the right of the service description as well as the copyright of the content data. For example, the information provider has rights to claim that "content data should be used in this manner", "content data should not be used in this manner", etc. For example, one who wrote a computer program can claim that the program may be run but no copying is allowed, or the program may be copied but no modifications are allowed and can define a charging policy such that the charge is 10yen per minute as the service description. The utilization that does not observe the service description constitutes an infringement of the digital rights. If the service description was not protected, then malicious users could rewrite the charging policy to thereby make the charges for information free. In such case, the information provider would suffer a great loss because the charging processor fails to work.

In order to protect the provider's digital rights, therefore, it is required to protect the service description as well as the content data. Like the content data, the service description is digital data and hence may be protected by encryption. That is, the content data and the service description are encrypted so that they cannot be interpreted at the time of utilization in the absence of a key, such as a token or ticket, issued by the information provider. The key is transmitted from the information provider to the user via a protected secure path independently of the service description.

FIG. 1 is a block diagram of such a conventional system. A server 1 located on the provider side converts content data 3 into an encrypted form in an encryption unit 4 and then sends it to a terminal 2 located on the user side. An encryption key is generated by a key generator 6 and then transmitted by a key management unit 5 to the user device 2 over a secure path which is different from that for the data 3. On the user side, the key is stored in a key management unit 8 and the encrypted content data is stored in a decryption unit 7. Using the key in the key management unit 8, the content data is decrypted in the decryption unit 7, whereby content data 9 is made available.

However, even if the key is transmitted to the user site over a secure path, once the key is passed to the user or the user's application program, there arises the possibility that the service description after decryption may be altered at the user site. Thus, there is an essential drawback that the provider's digital rights may not be protected.

Even if the service description is passed to a user in an encrypted form that is not dependent on various content data transmitting forms such as broadcasting, on-demand, DVD, etc., a key is passed to the user on demand. It is not known when the user will make a service request. For this reason, the information provider is required to run the key issuing server all the time. This will cost the information provider and is not suitable for information providing service by individuals.

Accordingly, it is an object of the present invention to provide an information providing system which permits information to be provided readily while ensuring information protection.

According to the present invention, there is provided an information providing system comprising a provider device for providing information to users; a user device for utilizing information; and an information storage card adapted to be connected to the provider device and the user device and comprising means for storing a second key, in which the provider device comprises means for sending to the user device, a service package that describes information necessary for utilization of the provided information, the service package being encrypted in accordance with a first encryption system, and means for sending to the user device, a first key used in the first encryption system, the first key being encrypted using the second key which is stored in the information storage card; and the user device comprises means for decrypting the encrypted first key within the information storage card.

The service package after decryption is disabled from being retained within the user device or being output from the user device to outside.

The encrypted service package is decrypted within the information storage card and the decrypted service package is disabled from being output to outside of the information storage card.

The user device comprises service package decryption means for decrypting the encrypted service package and means for disabling the service package decryption means from decrypting the encrypted service package when it is not guaranteed that the decrypted service package should not be retained within the user device nor be output to the outside of the user device.

The service package comprises information for identifying information to be provided, information for identifying an application program that utilizes the information to be provided, and information indicating a charging policy relating to the utilization of the information to be provided, and the user device comprises an application program execution unit that operates in response to the decrypted application program identifying information, a charging unit that operates in response to the decrypted charging policy identifying information.

The application program execution unit is implemented by an application program, and the charging unit is implemented by a platform that is different from the application program.

The provider device comprises means for sending to the user device, a second charging policy identical to the charging policy contained in the encrypted service package without encryption.

The first key used in the first encryption system is generated in the information storage card.

The first key used in the first encryption system is generated by an authorized agent and is written into the information storage card.

The provider device comprises means for sending a ticket to the user device, the ticket associating information identifying the service package with information identifying a key used to encrypt that service package and the user device comprises means for identifying a key associated with the service package to be utilized on the basis of the ticket.

The information providing system further comprises a repeater unit for receiving the message data from the provider device and transmitting the received message data to the user device.

According to the present invention, there is provided another information providing system comprising a provider device for providing information to users; a user device for utilizing information; and a security module adapted to be connected to the provider device and the user device and comprising means for storing a second key in such a way that it cannot be read out to outside, in which the provider device comprises means for sending to the user device, a service package that describes information necessary for utilization of information, the service package being encrypted in accordance with a first encryption system, a first key used in the first encryption system being encrypted using the second key stored in the security module; and the user device comprises means for decrypting the encrypted first key within the security module.

According to the present invention, there is provided an information providing device for providing information to users with an information storage card for storing a second key, the device comprising means for transmitting, a service package that describes information necessary for utilization of the provided information, the service package being encrypted in accordance with a first encryption system; and means for transmitting a first key used in the first encryption system, the first key being encrypted using the second key which is stored in the information storage card.

According to the present invention, there is provided still another information providing system for providing content data and message data in association with the content data, comprises means for calculating a first value of a unidirectional function or unidirectional hash function to which the whole message data or parts of the message data are input, a second value of a key generation function to which the first value is input, or a third value of a key generation function to which the first value and data included in the associated message or stored in the device are input, as a key.

The information providing system further comprises means for attaching the message data with a digital signature.

The message data contains charging information concerning a charge for usage of the content data.

The message data contains data described in a format including SGML, HTML, MHEG, or XML, and their extended or limited format.

According to the present invention, there is still another information utilization device for use with an information providing system in which content data and its associated message data are provided and the content data is encrypted, the device comprising means for calculating a first value of a unidirectional function or unidirectional hash function to which the whole message data or parts of the message data are input, a second value of a key generation function to which the first value is input, or a third value of a key generation function to which the first value and data included in the associated message or stored in the device are input, as a key.

The message data is attached with a digital signature.

The message data contains charging information concerning a charge for usage of the content data.

The message data contains data described in a format including SGML, HTML, MHEG, or XML, and their extended or limited format.

According to the present invention, there is provided still another information providing system comprising an information providing device which, in encrypting content data using an encryption key, uses a value of a unidirectional function or unidirectional hash function to which at least two parts of message data associated with the content data as the encryption key and transmitting the encrypted content data; a repeater unit for receiving the message data from the information providing device and transmitting the received message data; and an information utilization device which, in decrypting the encrypted content data transmitted from the information providing device using an decryption key, uses a value of a unidirectional function or unidirectional hash function to which the whole message data or parts of the message data associated with the content data and transmitted from the information providing device or the repeater unit as the decryption key.

The information providing device encrypts the message data, and the repeater unit decrypts the received encrypted message data, encrypts the message data again and transmits the encrypted message data.

The information providing device sends the message data with a provider's signature attached, and the repeater unit verifies the signature on the received message data and transmits the message data with a message data receiver's signature attached.

The repeater unit is in the form of the information utilization device.

According to the present invention, there is provided an encryption device for encrypting content data and its associated message data to be separately transmitted, the device comprising means for

calculating an value of a unidirectional function or unidirectional hash function to which the whole message data or parts of the message data are input; and
means for encrypting the content data using the value of the unidirectional function or unidirectional hash function as a key.

The invention can be more fully understood from the following detailed description when taken in conjunction with the accompanying drawings, in which;

FIG. 1 is a block diagram of a conventional information providing system;
FIG. 2 is a block diagram illustrating a system configuration on the provider side in accordance with a first embodiment of an information providing system of the present invention;
FIG. 3 is a block diagram illustrating a system configuration on the user side in accordance with the first embodiment of the present invention;
FIG. 4 is a detailed block diagram of the service instance of FIG. 3;
FIG. 5 is a block diagram of a security module in accordance with a second embodiment of the information providing system of the present invention;
FIG. 6 shows a security module for a user having a repeater function according to a second embodiment of the present invention;
FIG. 7 shows a security module for a repeater

according to the second embodiment of the present invention;

FIG. 8 shows a security module for an information provider having the repeater function according to the second embodiment of the present invention;

FIG. 9 shows a communication protocol between two security modules for two terminals;

FIG. 10 shows a system for an information provider according to a third embodiment of the present invention;

FIG. 11 shows a system for an agent or repeater according to the third embodiment of the present invention;

FIG. 12 shows a system for a user according to the third embodiment of the present invention;

FIG. 13 shows a modified system for the information provider according to the fourth embodiment of the present invention;

FIG. 14 shows a modified system for the agent according to the third embodiment of the present invention; and

FIG. 15 shows a modified system for the user according to the third embodiment of the present invention.

A preferred embodiment of an information providing system according to the present invention will now be described with reference to the accompanying drawings.

(First Embodiment)

Referring now to FIG. 2, there is illustrated an arrangement of a system on the information provider side in accordance with a first embodiment of the invention. In the present invention, as in the conventional system described previously, in order to allow the charging function to serve as a platform, a data processing unit, such as a server, on the information provider side creates a service package 10 that contains a pair of content data (name of the content data) the information provider provides and information (referred to as service description) required for control of the information or information representing the correspondence relationship between the content data and the service description. The service package 10 may include data described in accordance with a format such as SGML(Standard Generalized Markup Language), HTML(HyperText Markup Language), MHEG(Multimedia and Hypermedis Experts Group), XLM(eXtensible Markup Language), and their expanded or limited formats. The service package is such that MPEG data "$D_1$" (the name or address of content data) is encrypted using a key "$K_2$", processed by an application program "$A_1$", and subjected to a charging process "$C_1$". The user can actually make use of the content data on the basis of the service description in the service package 10. Thus, the service package 10 comprises a charging policy 12 indicating the form of charging, an application pointer 14 indicating an application program that utilizes content data, a data pointer 16 indicating the name or address of content data provided, and key data 18 indicating the key $K_2$ required to encrypt data. The charging policy 12 includes a usage fee, a usage condition, a paying method, or a payee.

Content data 20 provided is encrypted in an encryption unit 22 using the key $K_2$ and then transmitted to a user site over a transmitting interface (not shown). The encryption key $K_2$, which may be chosen freely by the provider, should preferably be one peculiar to the content data. Encrypted content data may be distributed online via the Internet or offline using DVD.

The encryption system may be either a common key system or a public key system. In the common key system, the key used to encrypt data and the key used to decrypt encrypted data are the same. In the public key system, on the other hand, the encryption key and the decryption key are different, and one of the keys is made public with the other kept secret. The provider encrypts data using the user's public key, and the user decrypts the encrypted data using his or her secret key. Thus, the public key encryption can be adopted only when each user is identified. According to the public key system, the content data is encrypted by using a public key $K_2$ so that a secret key corresponding to the public key must be transmitted to the user. Even with the colon key system, the key $K_2$ need not necessarily be contained in the service package 10 that is protected in accordance with the present invention. If there is a separate secure path, the key may be transmitted to the user over that path.

The service package 10 itself is encrypted for transmitting to the user. This transmitting may be made either online or offline. It should be noted however that the service package 10 is encrypted using a key $K_1$ which is different from the content encryption key $K_2$, and the key $K_1$ itself is also encrypted using still another key $K_0$ for transmitting to the user site. For encryption of the service package 10 and encryption of the encryption keys $K_1$ and $K_2$, in addition to the common key system the public key encryption system may also be used provided that each user is identified.

The key $K_1$ and the key $K_0$ for encrypting the key $K_1$ are kept absolutely unknown to the user. In the present invention, therefore, a security module 30 is used which is physically disabled from readout of data to outside. The encryption is performed inside the module 30. In order to increase terminal versatility, the module 30 should preferably be made of a semiconductor information storage card (a smart card, PCMCIA card, or the like) which provides the physical protection of data. However, if the terminal is implemented as a dedicated one, the module does not necessarily be removably mounted like a card, but can be fixedly mounted to part of the terminal.

The security module 30 comprises a key ($K_0$) memory 32, a key ($K_1$) generator 34, encryption units 36 and 42, an encrypted key ($K_1$') memory 38, and an encrypted key ($K_1$') transmitting interface 40. The key generator 34 generates an encryption key $K_1$ peculiar to the service package 10 in accordance with information that identifies the externally supplied service package 10. However, the key $K_1$ need not necessarily be generated within the module 30; it may be generated by a reliable organization and written into the security module 30.

The service package 10 is encrypted in the encryption unit 42 using the key $K_1$ and then transmitted to the user site via a transmitting interface not shown. The key $K_1$ used to encrypt the service package 10 is further encrypted in the encryption unit 36 using the key $K_0$ stored in the security module 30. The key ($K_0$) memory 32 consists of a nonvolatile memory. The key $K_0$ is stored at the time of creating the module 30 in such a way that it can never be accessed from outside at a later time. The key $K_0$ may be a key specific to the information provider irrespective of data/service package.

The encrypted key $K_1$' is stored in the memory 38. This is intended to omit work of encrypting the key $K_1$ each time the same service package 10 is encrypted. Thus, the key $K_1$' memory 38 can store keys $K_1$' obtained by encrypting encryption keys $K_1$, one for each of different service packages, using the key $K_0$. If, when a certain service package is specified, the corresponding key exists among the keys $K_1$' already stored, it is only required to read that key from the memory 38.

The key $K_1$' stored in the memory 38 is transmitted to the security module at the user site without being accessed from outside. When a semiconductor information storage card is used as the security module, the key $K_1$' is transmitted to the user's card in accordance with a card-to-card communications protocol. Thus, the key $K_1$' obtained by encrypting the service package encryption key $K_1$ is not output to the outside of the security module 30 nor does it become known to the user, preventing the service package 10 from being altered by a third party.

It is not required that the encrypted service package encrypted key K1' be transmitted simultaneously with the encrypted service package. As described previously, a key $K_1$ is peculiar to a service package. If, therefore, there exist multiple encrypted service packages and multiple encrypted service package encryption keys $K_1$' at the user site and the correspondence relationship between the service packages and the encryption keys is unknown, the encrypted service packages cannot be decrypted. It is therefore desirable to send from the provider to the user information that is used to establish a correspondence between information for identifying service packages and information for identifying keys $K_1$ used to encrypt those service packages. The information used to establish a correspondence is referred to as a ticket. By so doing, the user will be able to know from the ticket a key $K_1$ associated with a service package he or she wants to utilize.

FIG. 3 shows an arrangement of the terminal at the user site. At the user site as well, a security module 50 is used which preferably is made of a semiconductor information storage card. The security module 50 comprises a receiving interface 52, decryption units 54 and 60, a key ($K_0$) memory 56, a key ($K_1$) memory 58, and a service execution unit 62.

The user accepts the encrypted key K1' from the provider by secure module-to-module communications. The key $K_1$' will therefore not be transmitted to any user who does not subscript to the provider. The encrypted key K1' is supplied through the receiving interface 52 to the decryption unit 54. Like the provider's security module 30, the user's security module 50 has the memory 56 for storing the encryption key $K_0$. The memory 56, which is also a nonvolatile memory, is stored with the key $K_0$ at the time of creation of a card and is later made inaccessible from outside. Thus, the service package encrypted key K1' which was encrypted on the provider side using the key $K_0$ can be decrypted on the user side. Note that the public key encryption system may also be used here. The decrypted key $K_1$ is temporarily stored in the memory 58. This is also intended to omit work of decrypting the same encrypted key $K_1$' each time the same service package is decrypted. The memory 58 has an enough capacity to store keys $K_1$ corresponding to a plurality of service packages.

The encrypted service package is received by a receiving interface 64 and then temporarily stored in a service package memory 66. The encrypted content data are received by a receiving interface 68 and then stored in a content data memory 70. The encrypted service package is decrypted in the decryption unit 60 in the security module 50 using the key $K_1$ stored in the memory 58 and then entered into the service execution unit 62. The key $K_2$ contained in the decrypted service package is supplied from the security module 50 to a decryption unit 72. This decryption unit 72 decrypts the encrypted content data stored in the memory 70 using the key $K_2$. The decrypted content data is are delivered to a content reproducing unit (for example, a display unit) 74.

The user's terminal further comprises a user interface 76 and a service control unit 78. The service control unit 78 controls the key $K_1$ memory 58, the service package memory 66, and the service execution unit 62.

Referring now to FIG. 4, the service execution unit 62 comprises a charging module 82 which performs a charging process on the basis of the charging policy 12, an application program 84 which is run on the basis of the application pointer 14 and the data pointer 16, and a data transfer processing module 86 which allows content data 88 to be received in cooperation with the charging module 82 and the application program 84. The service execution unit 62 thus comprises hardware and software which are required to implement the information providing service on the basis of the service description, and parameters that allow the hardware and software to work properly, i.e., the unit is a collection of facilities required to implement the information providing service.

The operation of the first embodiment will be described next. The first embodiment makes it a condition that the user uses the provider-distributed security module 50 having the key $K_0$ memory 56. An encrypted service package and encrypted content data are transmitted to the user site online via the internet or offline using a large-volume storage medium such as a DVD, i.e., in an arbitrary mode. When utilizing an information providing service, the user receives a key $K_1$' peculiar to the service package from the provider. The user plugs the security module 50 that has received the key $K_1$' into the terminal. In the module 50, the encrypted service package is decrypted by the decryption unit 60 and a service instance is produced from the service package. At the same time, the encrypted content data is decrypted by the decryption unit 72 using the key $K_2$ contained in the service package.

In the decrypted service package, the application program 14 and the data pointer 16 activate a predetermined application program 84. As the application program runs, the data transfer processing module 86 reads content data 88 from a server or storage medium to initiate the usage of the information providing service and the charging module 82 charges usage of the information providing service according to the charging policy 12.

As described above, according to the first embodiment, the content data 20 and the service package 10 are encrypted using separate encryption keys and then transmitted to the user. The encryption key $K_1$ used to encrypt the service package 10 is further encrypted using an additional encryption key $K_0$ and then transmitted to the user. This additional key $K_0$ used to encrypt the service package encryption key $K_1$ is held in the memory 32 in the security module 30 that cannot be accessed from outside and the encrypted key $K_1$' itself is directly transmitted to the user site on a module-to-module communications basis, in other words, in an externally inaccessible state. Therefore, the user and the application program cannot rewrite the service package, which prevents the charging policy from being altered for illegal utilization of services.

A modification of the first embodiment will be described. In FIGS. 2 and 3, the security module has only circuits that meet minimum requirements built in. If it has room, however, the encryption unit 22 (FIG. 2) and the decryption unit 72 (FIG. 3) may be built into the security modules 30 and 50, respectively.

In addition, the provider's security module 30 and the user's security module 50 may be arranged identically. An example therefor is illustrated in FIG. 5. A key ($K_0$) memory 100 is connected to an encryption/decryption unit 102 to which a key ($K_1$) memory 106 and a key ($K_1$') memory 108 are connected. A transmitting/receiving interface 110 is connected to the key ($K_1$') memory 108. Information identifying a service package is given to a key ($K_1$) generator 104, which generates a key $K_1$ used to encrypt that service package. The key $K_1$ is stored in the key memory 106 and supplied to a service package encryption/decryption unit 112 to which a transmitting/receiving interface 114 and a service execution unit 116 are connected.

In the module used by the provider, a key $K_1$ peculiar to a service package is generated or received from an external organization to encrypt the service package in the service package encryption/decryption unit 112. The resulting encrypted service package is transmitted over the transmitting/receiving interface 114 to a user site. At the same time, the key $K_1$ is encrypted by the encryption/decryption unit 102 using a key $K_0$ and the resulting encrypted key $K_1$' is transmitted over the transmitting/receiving interface 110.

In the module at the user site, the encrypted key $K_1$' received from the provider over the transmitting/receiving interface 110 is decrypted in the encryption/decryption unit 102 using a key $K_0$ and the decrypted key $K_1$ is then stored in the key ($K_1$) memory 106. The encrypted service package received over the transmitting/receiving interface 114 is decrypted in the service package encryption/decryption unit 112 for application to the service execution unit 116.

According to such an arrangement, the provider and the user are allowed to use the security modules of the same arrangement, providing an advantage of reduced cost. In this case as well, if there is room to accommodate more hardware in the security module, the data encryption/decryption unit may also be built into the module. In addition, if the user keeps a security module of the same arrangement as the provider's security module, there is no need for the provider to directly send the service package encrypted key K1' to the user. In such a case, the user will be allowed to send the key $K_1$' to other users. Further, the user can also send the key $K_1$' via security modules of a plurality of users, allowing the key to be communicated from individual to individual like word-of-mouth communication. This will eliminate the need of operating the key publishing server all the time and is therefore suitable for an information providing service by individuals. In this case, the users who merely repeats the key will not need all the hardware of FIG. 5, but requires only the key ($K_1$') memory 108 and the transmitting/receiving interface 110. However, when the public key system is used, the encrypted key $K_1$' must be decrypted once at each repeater terminal; therefore, in the arrangement of FIG. 5 it is only the service package encryption/decryption unit 112, the transmitting/receiving interface 114 and the service execution unit 116 that can be omitted.

(Second Embodiment)

Referring to FIGS. 6 to 8, there is illustrated an arrangement of a second embodiment in which the service package and key can be repeated via a repeater. The key must be repeated using the security module. The service package is not necessary to be repeated using the security module since it is encrypted. The service package can be stored in a personal computer as a file and read out to be transmitted to the other repeater or user.

FIG. 6 shows a security module for the user having a repeater function. An input/output interface 202 receives the encrypted service package and the encrypted key $K_1$' from the information provider or repeater. The encrypted key $K_1$' is supplied to a key receiver 208 of a key management section 204. The key management section 204 comprises a key controller 206, key memory 210, and key transmitter 212 in addition to the key receiver 208. The encrypted key $K_1$' is written into the key memory 210 by the key receiver 208. The key $k_1$' read out from the key memory 210 is supplied to the key transmitter 212. The key transmitter 212 sends out the key $k_1$' via the I/O interface 202. Thus, the encrypted key $K_1$' is repeated by the user's security module.

The user's security module further comprises a service package decryption unit 214, a service execution unit 216, and a service control unit 218. The I/O interface 202 supplies the input service package (encrypted service package) to the package decryption unit 214 in which the encrypted service package is decrypted by using the key $K_1$' supplied from the key memory 210. The encrypted service package is supplied to the service execution unit 216 which causes the information providing service to be started. In the same manner as the first embodiment, the service execution unit 216 is controlled by a service control unit 218.

FIG. 7 shows a security module for the repeater. An input/output interface 222 receives the encrypted key $K_1$' from the information provider or repeater. The encrypted key $K_1$' is supplied to a key receiver 228 of a key management section 224. The key management section 224 comprises a key controller 226, key memory 230, and key transmitter 232 in addition to the key receiver 228. The encrypted key $K_1$' is written into the key memory 230 by the key receiver 228. The key $k_1$' read out from the key memory 230 is supplied to the key transmitter 232. The key transmitter 232 sends out the key $k_1$' via the I/O interface 222. Thus, the key $K_1$' is repeated by the repeater's security module.

FIG. 8 shows a security module for the information provider having a repeater function. An input/output interface 242 receives data necessary for synthesize the service package. The security module comprises a key management section 244 which is formed of a key generator 248, key controller 246, key memory 250, and key transmitter 252. The key generator 248 generates an encrypted key $K_1$' which is an encrypted form of the encryption key $K_1$ of the service package 10. The key $K_1$' is stored in the key memory 250. The key $k_1$' read out from the key memory 250 is supplied to the key transmitter 252. The key transmitter 252 sends out the key $k_1$' via the I/O interface 242.

The provider's security module further comprises a service package synthesis unit 254, a service package encryption unit 256, and a service package generation controller 258. The I/O interface 202 supplies the input data to the service package synthesis unit 254 in which the service package is synthesized based on the input data. The service package output from the service package synthesis unit 254 is encrypted by the service package encryption unit 256. The encrypted service package is externally output from the I/O interface 242.

FIG. 9 shows a key transmission protocol between two terminals each having a security module. When the user of a terminal "A" wishes to receive the key from a terminal "B", the terminal "A" sends a transmission request to the terminal "B". The terminal "B" communicates with its security module a transmission command and then sends a reception request to the terminal "A". The terminal "A" communicates with its security module a reception command and then sends a reception agreement to the terminal "B". When the terminal "B" sends a start command to its security module, the security modules of the terminal "B" and terminal "A" start verification process and then the session is setup between the security modules of the terminal "B" and terminal "A". The key is transmitted from the security module of the terminal "B" to the security module of the terminal "A". After the session is terminated, the security module of the terminal "B" reports the terminal "B" of the complete of transmission and the security module of the terminal "A" reports the terminal "A" of the complete of reception.

According to the second embodiment, there can be provided a security module in which the service package and the key can be repeated.

According to the first and second embodiments, as in the conventional system described previously, in order to allow the charging function to serve as a platform, a data processing unit, such as a server, on the information provider side creates a service package 10 that contains a pair of content data (name of the content data) the information provider provides and control information (referred to as service description) required to utilize the content data.

An information providing system can be provided which provides service package security protection at the user site, including the transmitting path from the information provider to the user.

(Third Embodiment)

In the third embodiment, the provider encrypts content data and the encrypted content data are transmitted to the user directly or via an agent for repeating data. The service description (hereinafter referred to as a message) associated with the content data is transmitted to the user directly or via the agent. The examples of the service description is the same as that of the first embodiment. The number of agents through which the data is transmitted is not limited to one. The agent is not limited to a person who only repeats the data. Other users can be agents. The third embodiment is characterized in that the content data and/or message are transmitted to an end user via another user or agent. FIG. 10 is a schematic representation of a server on the provider side, FIG. 11 is a schematic representation of a system on the agent side, and FIG. 12 is a schematic representation of a terminal on the user side.

Content data CN, such as video, music, images, etc., which are information to be delivered, are encrypted by an encryption unit 312 and then published as encrypted content data CN' (= T(CN)) to the agent or user. A message M associated with the content data contains multiple (at least two) submessages $M_1$ and $M_2$. The encryption unit 312 uses, as an encryption key K, an output of a correlation unit 314 in which the submessages $M_1$ and $M_2$ are input to a unidirectional function or unidirectional hash function $f(M_1, M_2)$, a value of a predetermined key generation function to which the output of the correlation unit 314 is input, or a value of the predetermined key generation function to which the output of the correlation unit 314 and data included in the associated message or stored in the device are input, therefore, the content data encryption key K are correlated with the submessages $M_1$, $M_2$ associated with the content data.

Unlike the content data CN, the message M is appended with a digital signature of the provider "A" in a signature unit 316 and a signed message $M_{sign}$ (= $S_A(E_B(M))$) is transmitted to the agent or user. Here, $E_B(M)$ is an encrypted message obtained by encrypting the message M using a public key of the agent "B" (or a common key that the provider "A" and the agent "B" share) in order to transmit the message M to the agent B. $S_A(E_B(M))$ represents the encrypted message $E_B(M)$ appended with the digital signature of the provider "A". Hereinafter, E, D, S and V represent operations of encryption, decryption, signature, and verification, respectively. The purpose of transmitting of the message M with the digital signature is to guarantee that the message is a true message transmitted from the provider.

As shown in FIG. 11, in the message agent's system, the message $M_{sign}$ with the digital signature is verified by a signature verification unit 322 and the original message M (= $D_B(V_A(M_{sign}))$) is reproduced. Here, $V_A(M_{sign})$ is the encrypted message for which verification has been made that it was signed by the provider "A", and $D_B(V_A(M_{sign}))$ is a message in plaintext obtained by decrypting the encrypted message $V_A(M_{sign})$ using the secret key of the agent "B" (or the common key that the provider "A" and the agent "B" share).

Suppose here that the agent is malicious and alters part of submessages $M_1$ and $M_2$ to yield $M_1'$ and $M_2$ by using a forgery unit 324. It is assumed that at least one of $M_1' \neq M_2$ and $M_1' \neq M_2$ is satisfied. The agent "B" attaches the own digital signature to the forged message M' by using a signature unit 326. The forged message with the signature $M'_{sign}$ (= $S_B(E_C(M'_{sign}))$) is transmit to the user. Alternatively, the signed message $M'_{sign}$ may be transmitted to the user via still another agent.

As shown in FIG. 11, the agent's system can be implemented by a conventional data processing apparatus, such as a personal computer. For a honest agent, the message is received and merely retransmit to the agent or user. Therefore, the forgery unit 324 is replaced with a mere buffer memory.

As shown in FIG. 12, in the user's system, the message $M'_{sign}$ with the digital signature transmitted from the provider or agent (in this case, the message is forged by the agent) is verified in a signature verification unit 332. The original forged message M' (= $D_C(V_B(M'_{sign}))$ is obtained. Here, $V_B(M'_{sign})$ is the message M' forged and signed by the agent "B", and $D_C(V_B(M'_{sign})$ is a message in plaintext obtained by decrypting the encrypted message M', for which verification has been made that it was signed by the agent "B", using the secret key belonging to the user "C" (or the common key that the agent "B" and the user "C" share). The encrypted content data CN' is decrypted by a decryption unit 336. The decryption unit 336 uses, as a decryption key K', an output of a correlation unit 334 in which the submessages $M_1'$ and $M_2'$ contained in the message M' are input to a unidirectional function or unidirectional hash function $f(M_1', M_2')$, a value of a predetermined key generation function to which the output of the correlation unit 314 is input, or a value of the predetermined key generation function to which the output of the correlation unit 314 and data included in the associated message or stored in the device are input. If the message M is not forged by the agent, then the key K' supplied to the decryption unit 336 will be equal to the key K supplied to the encryption unit 312 on the provider side. In this case, however, the message is forged as described above, the key K' supplied to the decryption unit 336 is $f(M_1', M_2')$.

If the message M is not forged by the agent, then the key K' equals to the key K and the decrypted content data CN" (= R(CN')) matches the content data CN on the provider side.

However, when, as shown in FIG. 11, the agent forges the message, the key K' to the decryption unit 336 is $K' = f(M_1', M_2') \neq f(M_1, M_2)$, resulting in a mismatch between the decrypted content data CN" and the original content data CN. In other words, the original content data are not available on the user side. In such case, the user will suppose that the message given by the agent is not correct, that is, the agent would have committed some injustice, and then notifies the provider of that. As a result, the provider sends the correct message directly to the user, thus allowing the user to decrypt and utilize the content data.

In the event that the agent has altered the submessage M1 to M1', in order to cause the decryption key K' (= f($M_1$', $M_2$')) to match the encryption key K (= f($M_1$, $M_2$)), it is required to change the other submessage M2 as well to M2' that satisfies f($M_1$', $M_2$') = K. Since the function f is a unidirectional one, however, it is almost impossible for the malicious agent to find $M_2$' that satisfies f($M_1$', $M_2$') = K, i.e., $M_2$ that allows K' to be changed to K.

As described so far, according to the third embodiment, by making a content data encryption key a unidirectional function or unidirectional hash function of the whole message or parts of the message (in this embodiment, two parts of the message) associated with the content data, that is, by correlating the encryption key with the message, the decryption of the content data becomes disabled when the message is forged. It therefore becomes possible to prevent unauthorized usage of content data and protect the digital rights of information providers. In addition, since the encryption key is never known to the users and agents, encrypted content data can never be decrypted illegally. Moreover, since the message containing at least two submessages used to generate an encryption key can be transmitted offline to the final user via the agent or other agents, there is no need of installing a key issuing server and hence the cost involved in providing information is reduced. This is suitable for an information providing service by individuals.

(Detailed Example)

Detailed example of a specific application of the third embodiment will be described next. As an example, it is assumed that the content data CN are MPEG-compressed video and the message M contains charging information for billing a user for video. The provider "A" describes in the submessage M1 charging information such that user should pay the provider "A" 1000yen for this video. The submessage M2 may contain the name of video information (content data) or a data pointer indicating the address of the video information, an application pointer indicating which of application programs is to be used for the content data, etc, as shown in FIG. 2. It should be noted that the number of submessages used to generate an encryption key is not limited to two, but may be three or more.

The provider "A" passes the message M to an agent "B" with its signature attached.

The agent verifies the signature of the message M and then sends it to a user directly or via another agent with its signature attached.

The user can decrypt the content data using the key generated from the submessages M1 and M2 of the message M thus received.

Suppose here that a malicious agent "B" alters the submessage M1 to a submessage M1' describing that user should pay the agent "B" 1,000yen and sends it to the user with its signature attached. In such a case, the user will know from this message M1' that it is to the agent "B" that he or she should pay 1,000yen for usage of content data. However, since the message has been forged, the correct encryption key cannot be obtained, so that the user fails to decrypt the encrypted content data. It turns out, on contact with the provider "A", that the agent "B" forged the message. Thus, it is possible to cause a malicious agent's plot to forge a message and take a charge to be paid to the provider "A" to end in failure.

(Fourth Embodiment)

In the third embodiment, it is not required that a digital signature is attached to the message. The fourth embodiment is a modification of the third embodiment in which the message is directly transmitted without a digital signature. FIG. 13 is a schematic representation of a server on the provider side, FIG. 14 is a schematic representation of a system on the agent side, and FIG. 15 is a schematic representation of a terminal on the user side. The provider does not attach the digital signature to the message. The agent does not perform a signature verification and merely relay the received message to the other agent or user. The user correlates the submessages included in the received message.

Additional advantages and modifications will readily occur to those skilled in the art. Therefore, the present invention in its broader aspects is not limited to the specific details, representative devices, and illustrated examples shown and described herein. Accordingly, various modifications may be made without departing from the spirit or scope of the general inventive concept as defined by the appended claims and their equivalents.

For example, in the first and second embodiments, the user cannot know the charging policy data without decryption and a security module is need for decryption. The user has no security module before service utilization. This is inconvenient for users. For this reason, it is desirable that a second charging policy identical to a charging policy to be transmitted in encrypted form be prepared separately and transmitted to the user site without encryption. The user can know the second charging policy data to decide whether to utilize the service or not. In this case, it is the charging policy contained in the decrypted service package that is transmitted to the service execution unit 62. However, a third party might alter the second charging policy in plaintext to render a charge for service free. This affords little user protection. It is therefore desirable that the user device be equipped with means for making a comparison between the decrypted charging policy and the second charging policy and disabling service utilization when the comparison indicates inequality.

In the first and second embodiments, in order to keep the decrypted service package unknown to the user, the decryption unit 60 and the service execution unit 62 are installed in the security module 50. That is, hardware is used to prevent alteration of information. Of course, the service package may be protected by software. A certificate that guarantees that the service package and the key $K_1$ are not output to outside nor retained may be attached to a service instance itself which is software for implementing the service package. In the absence of this certificate, the service package is disabled from being decrypted. In this case, the decryption unit 60 and the service execution unit 62 need not be installed in the security module 50. Further, if, when hardware is used to prevent alteration of information, the user's terminal is reliable, the decryption unit 60 and the service implementing unit 62 may not necessarily be provided in the security module 50. Although being implemented by a platform, the service package decryption unit and/or charging system may be implemented as an application program as with normal data processing.

It is not necessarily required that an agent or agents intervene between a provider and a user according to the above-mentioned embodiments. With no agent, a message may be transmitted from the provider to the user together with encrypted content data. In this case, the need of signature transfer processing through message is saved.

The transfers of content data and messages may be made online via the internet or offline through DVDs.

In the third embodiment, the signature encryption may be either public key-based or common key-based. The message may include not only charging information but also data described in accordance with a format such as SGML (Standard Generalized Markup Language), HTML (HyperText Markup Language), MHEG(Multimedia and Hypermedia Experts Group), XML(eXtensible Markup Language), and their extended or limited format or the like.

When the correlation unit 341 on the user side which calculates a unidirectional function or unidirectional hash function value is built into the security module, such as a semiconductor chip, a smart card, or the like, which is physically disabled against readout, as well as a decrypting unit for the content data, the security can be further increased because the key K is never read out to outside. The reason is that, if a user intervenes between the module in which the unidirectional function or unidirectional hash function is used to calculate the key K and the module in which the content data are decrypted, the user will be able to know the value for key K to thereby decrypt the content data, associate an entirely different message with the decrypted content data, and encrypt the content data using a different key. It must be avoided to make it possible to decrypt encrypted content data readily with no need of a message.

As described above, according to the present invention, there is provided an information providing system which has a facility of protecting content data and provider's rights at the user site including a transmitting path from an information provider to a user and permits information to be delivered readily.

## Claims

1. An information providing system comprising:

   a provider device for providing information to users;
   a user device for utilizing information; and
   an information storage card adapted to be connected to the provider device and the user device and comprising means for storing a second key,
   characterized in that
   the provider device comprises means (42) for sending to the user device, a service package that describes information necessary for utilization of the provided information, the service package being encrypted in accordance with a first encryption system, and means (40) for sending to the user device, a first key used in the first encryption system, the first key being encrypted using the second key which is stored in the information storage card; and
   the user device comprises means (54) for decrypting the encrypted first key within the information storage card.

2. The information providing system according to claim 1, characterized in that the service package after decryption is disabled from being retained within the user device or being output from the user device to outside.

3. The information providing system according to claim 2, characterized in that the encrypted service package is decrypted within the information storage card and the decrypted service package is disabled from being output to outside of the information storage card.

4. The information providing system according to claim 2, characterized in that the user device comprises service package decryption means (60) for decrypting the encrypted service package and means for disabling the service package decryption means from decrypting the encrypted service package when it is not guaranteed that the decrypted service package should not be retained within the user device nor be output to the outside of the user device.

5. The information providing system according to

claim 1, characterized in that the service package comprises information (16) for identifying information to be provided, information (14) for identifying an application program that utilizes the information to be provided, and information (12) indicating a charging policy relating to the utilization of the information to be provided, and the user device comprises an application program execution unit (62) that operates in response to the decrypted application program identifying information, a charging unit that operates in response to the decrypted charging policy identifying information.

**6.** The information providing system according to claim 5, characterized in that the application program execution unit is implemented by an application program, and the charging unit is implemented by a platform that is different from the application program.

**7.** The information providing system according to claim 5, characterized in that said provider device comprises means for sending to the user device, a second charging policy identical to the charging policy contained in the encrypted service package without encryption.

**8.** The information providing system according to claim 1, characterized in that the first key used in the first encryption system is generated in the information storage card.

**9.** The information providing system according to claim 1, characterized in that the first key used in the first encryption system is generated by an authorized agent and is written into the information storage card.

**10.** The information providing system according to claim 1, characterized in that the provider device comprises means for sending a ticket to the user device, the ticket associating information identifying the service package with information identifying a key used to encrypt that service package and the user device comprises means for identifying a key associated with the service package to be utilized on the basis of the ticket.

**11.** The information providing system according to claim 1, which further comprises a repeater unit for receiving the message data from the provider device and transmitting the received message data to the user device.

**12.** An information providing system comprising:

a provider device for providing information to users;

a user device for utilizing information; and

a security module adapted to be connected to the provider device and the user device and comprising means for storing a second key in such a way that it cannot be read out to outside, characterized in that

the provider device comprises means (42) for sending to the user device, a service package that describes information necessary for utilization of information, the service package being encrypted in accordance with a first encryption system, a first key used in the first encryption system being encrypted using the second key stored in the security module; and

the user device comprises means (54) for decrypting the encrypted first key within the security module.

**13.** An information providing device for providing information to users with an information storage card for storing a second key, the device comprising:

means (42) for transmitting, a service package that describes information necessary for utilization of the provided information, the service package being encrypted in accordance with a first encryption system; and

means (40) for transmitting a first key used in the first encryption system, the first key being encrypted using the second key which is stored in the information storage card.

**14.** An information providing system for providing content data and message data in association with the content data, characterized by comprises:

means (314) for calculating an value of a unidirectional function or unidirectional hash function to which the whole message data or parts of the message data are input; and

means for encrypting the content data to be provided using the value of the unidirectional function or unidirectional hash function as a key.

**15.** The information providing system according to claim 14, characterized by further comprising means for attaching the message data with a digital signature.

**16.** The information providing system according to claim 14, characterized in that the message data contains charging information concerning a charge for usage of the content data.

**17.** The information providing system according to claim 14, characterized in that the message data contains data described in a format including

SGML, HTML, MHEG, or XML, and their extended or limited format.

**18.** An information utilization device for use with an information providing system in which content data and its associated message data are provided and the content data is encrypted, the device comprising:

means for calculating a first value of a unidirectional function or unidirectional hash function to which the whole message data or parts of the message data are input, a second value of a key generation function to which the first value is input, or a third value of a key generation function to which the first value and data included in the associated message or stored in the device are input; and
means for decrypting the encrypted content data using the value of the unidirectional function or unidirectional hash function as a key.

**19.** The information utilization device according to claim 18, characterized in that the message data is attached with a digital signature.

**20.** The information utilization device according to claim 18, characterized in that the message data contains charging information concerning a charge for usage of the content data.

**21.** The information utilization device according to claim 18, characterized in that the message data contains data described in a format including SGML, HTML, MHEG, or XML, and their extended or limited format.

**22.** An information providing system characterized by comprising:

an information providing device which, in encrypting content data using an encryption key, uses a first value of a unidirectional function or unidirectional hash function to which the whole message data or parts of the message data are input, a second value of a key generation function to which the first value is input, or a third value of a key generation function to which the first value and data included in the associated message or stored in the device are input, as the encryption key and transmitting the encrypted content data;
a repeater unit for receiving the message data from the information providing device and transmitting the received message data; and
an information utilization device which, in decrypting the encrypted content data transmitted from the information providing device using an decryption key, uses a value of a unidirectional function or unidirectional hash function to which the whole message data or parts of the message data associated with the content data and transmitted from the information providing device or the repeater unit as the decryption key.

**23.** The information providing system according to claim 22, characterized in that the information providing device encrypts the message data, and the repeater unit decrypts the received encrypted message data, encrypts the message data again and transmits the encrypted message data.

**24.** The information providing system according to claim 22, characterized in that the information providing device sends the message data with a provider's signature attached, and the repeater unit verifies the signature on the received message data and transmits the message data with a message data receiver's signature attached.

**25.** The information providing system according to claim 22, characterized in that the repeater unit is in the form of the information utilization device.

**26.** An encryption device for encrypting content data and its associated message data to be separately transmitted, characterized by comprising:

means for calculating an value of a unidirectional function or unidirectional hash function to which the whole message data or parts of the message data are input; and
means for encrypting the content data using the value of the unidirectional function or unidirectional hash function as a key.

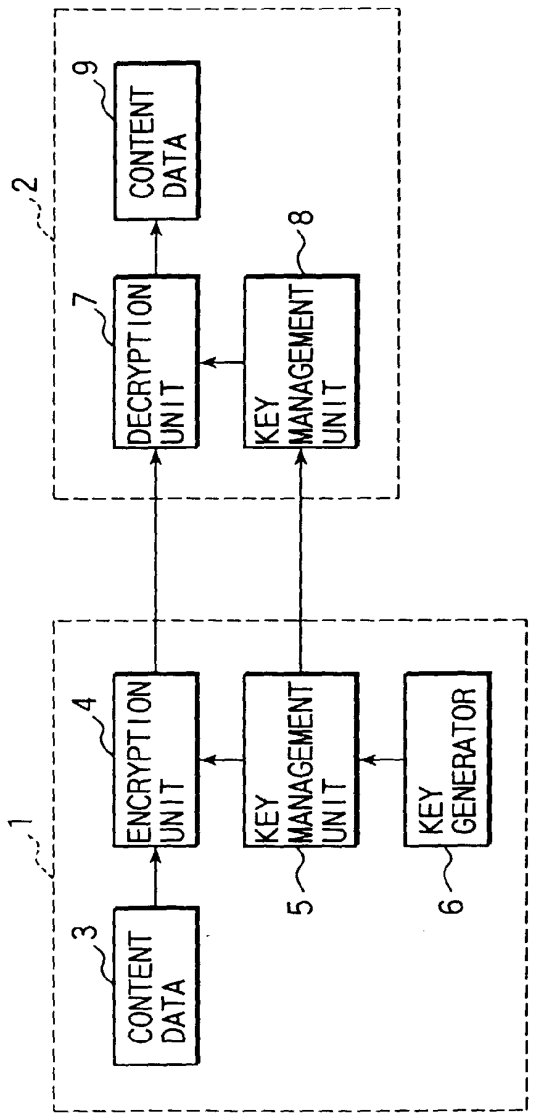

1
3
CONTENT DATA
4
ENCRYPTION UNIT
5
KEY MANAGEMENT UNIT
6
KEY GENERATOR
2
7
DECRYPTION UNIT
8
KEY MANAGEMENT UNIT
9
CONTENT DATA

FIG.1

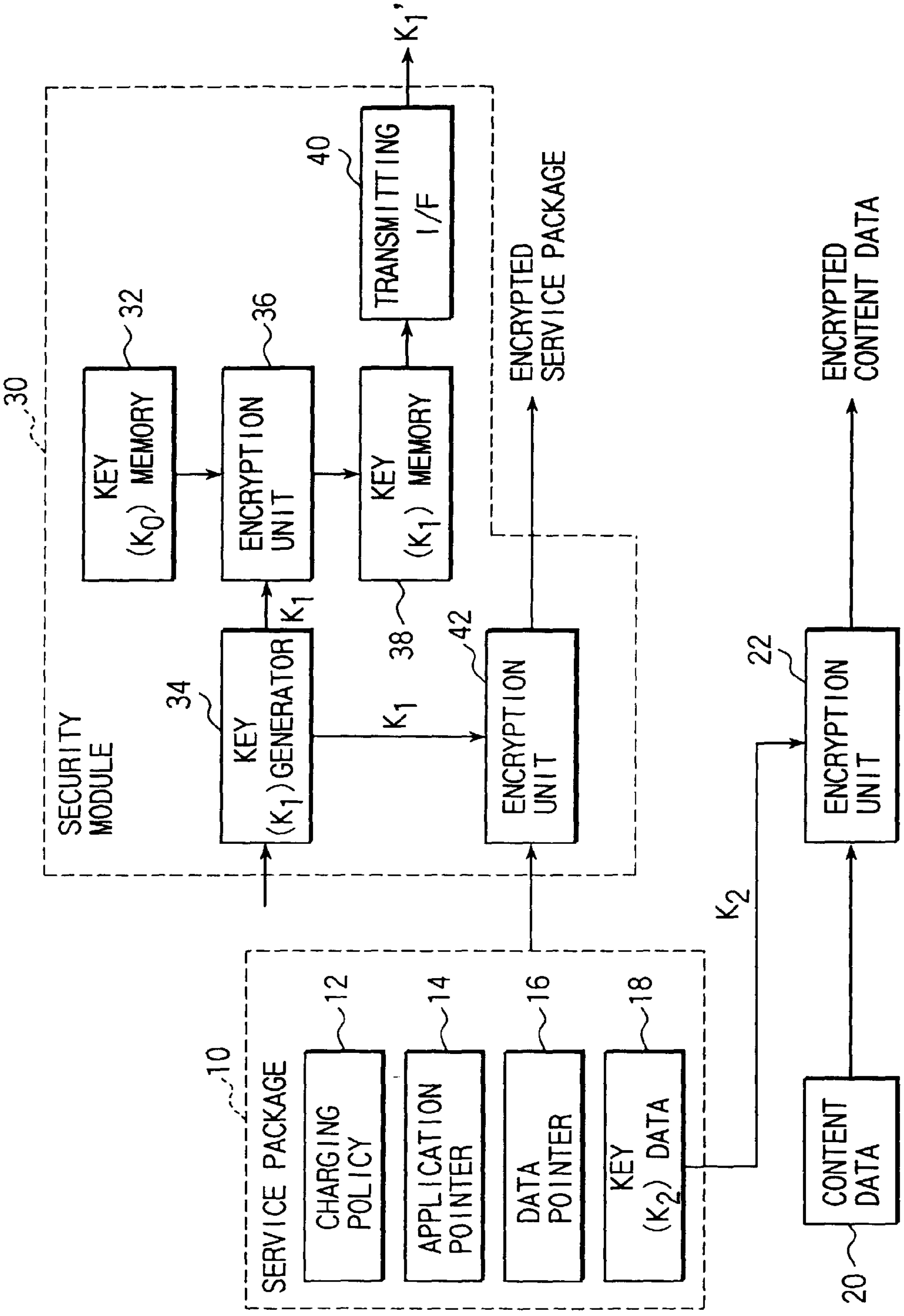

30
SECURITY MODULE
KEY ($K_0$) MEMORY
32
ENCRYPTION UNIT
36
KEY ($K_1$) MEMORY
38
TRANSMITTING I/F
40
$K_1$'
KEY ($K_1$) GENERATOR
34
$K_1$
$K_1$
ENCRYPTION UNIT
42
ENCRYPTED SERVICE PACKAGE
10
SERVICE PACKAGE
CHARGING POLICY
12
APPLICATION POINTER
14
DATA POINTER
16
KEY ($K_2$) DATA
18
$K_2$
CONTENT DATA
20
ENCRYPTION UNIT
22
ENCRYPTED CONTENT DATA

FIG. 2

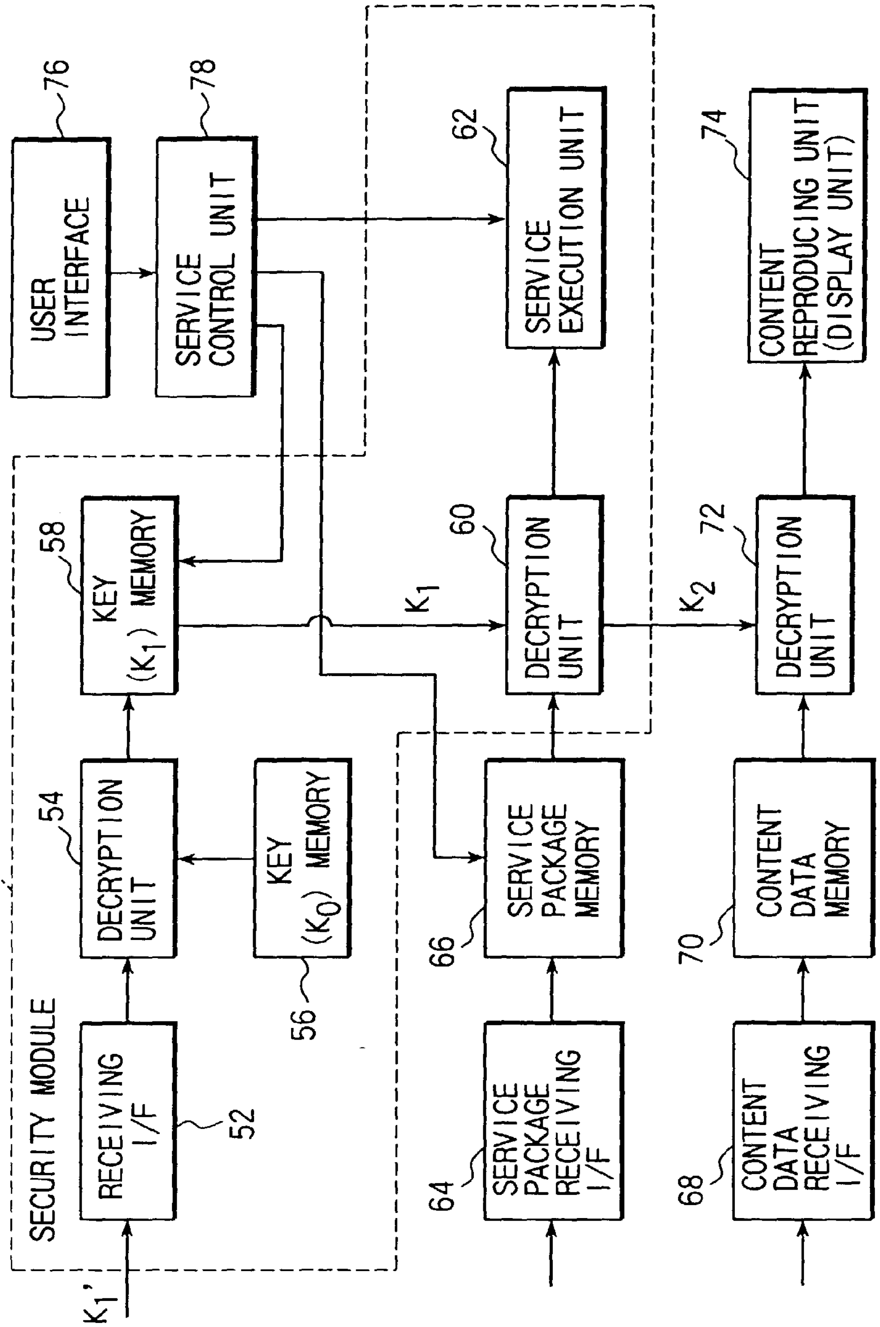

SECURITY MODULE
50
K1'
RECEIVING I/F
52
DECRYPTION UNIT
54
KEY (K0) MEMORY
56
KEY (K1) MEMORY
58
USER INTERFACE
76
SERVICE CONTROL UNIT
78
K1
DECRYPTION UNIT
60
SERVICE EXECUTION UNIT
62
SERVICE PACKAGE RECEIVING I/F
64
SERVICE PACKAGE MEMORY
66
K2
CONTENT DATA RECEIVING I/F
68
CONTENT DATA MEMORY
70
DECRYPTION UNIT
72
CONTENT REPRODUCING UNIT (DISPLAY UNIT)
74

FIG. 3

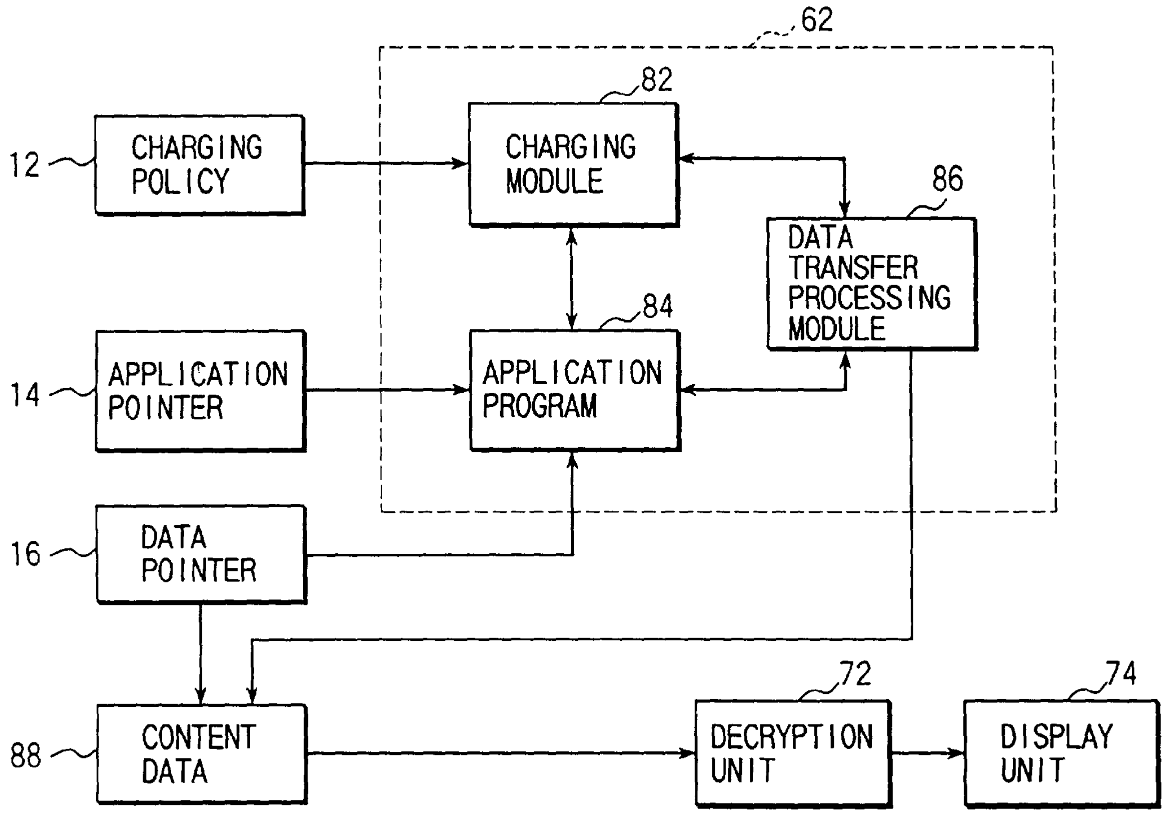

62
12
CHARGING POLICY
82
CHARGING MODULE
86
DATA TRANSFER PROCESSING MODULE
84
14
APPLICATION POINTER
APPLICATION PROGRAM
16
DATA POINTER
88
CONTENT DATA
72
DECRYPTION UNIT
74
DISPLAY UNIT

FIG. 4

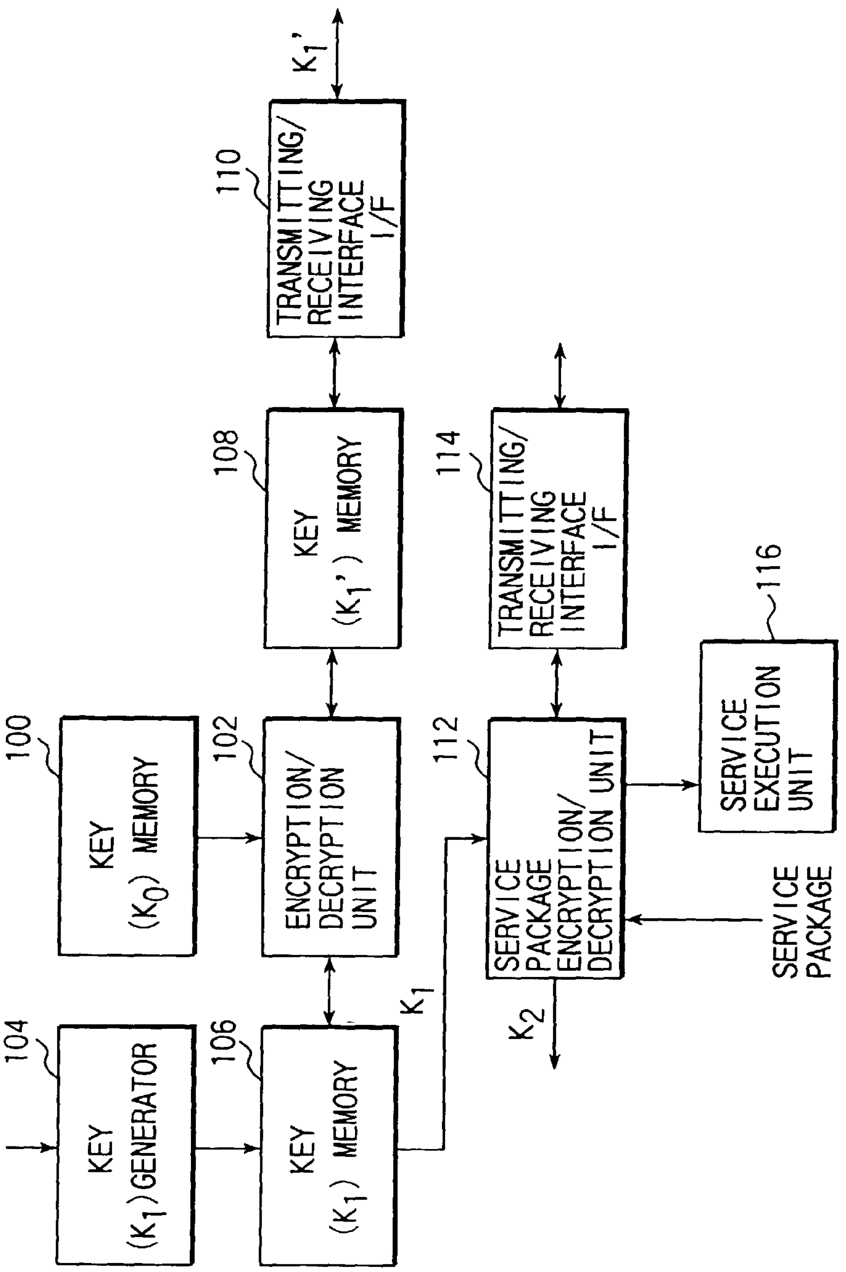

104
KEY
$(K_1)$GENERATOR
106
KEY
$(K_1)$ MEMORY
100
KEY
$(K_0)$ MEMORY
102
ENCRYPTION/
DECRYPTION
UNIT
108
KEY
$(K_1')$ MEMORY
110
TRANSMITTING/
RECEIVING
INTERFACE
I/F
$K_1'$
$K_1$
112
SERVICE
PACKAGE
ENCRYPTION/
DECRYPTION UNIT
114
TRANSMITTING/
RECEIVING
INTERFACE
I/F
$K_2$
SERVICE
PACKAGE
116
SERVICE
EXECUTION
UNIT

FIG. 5

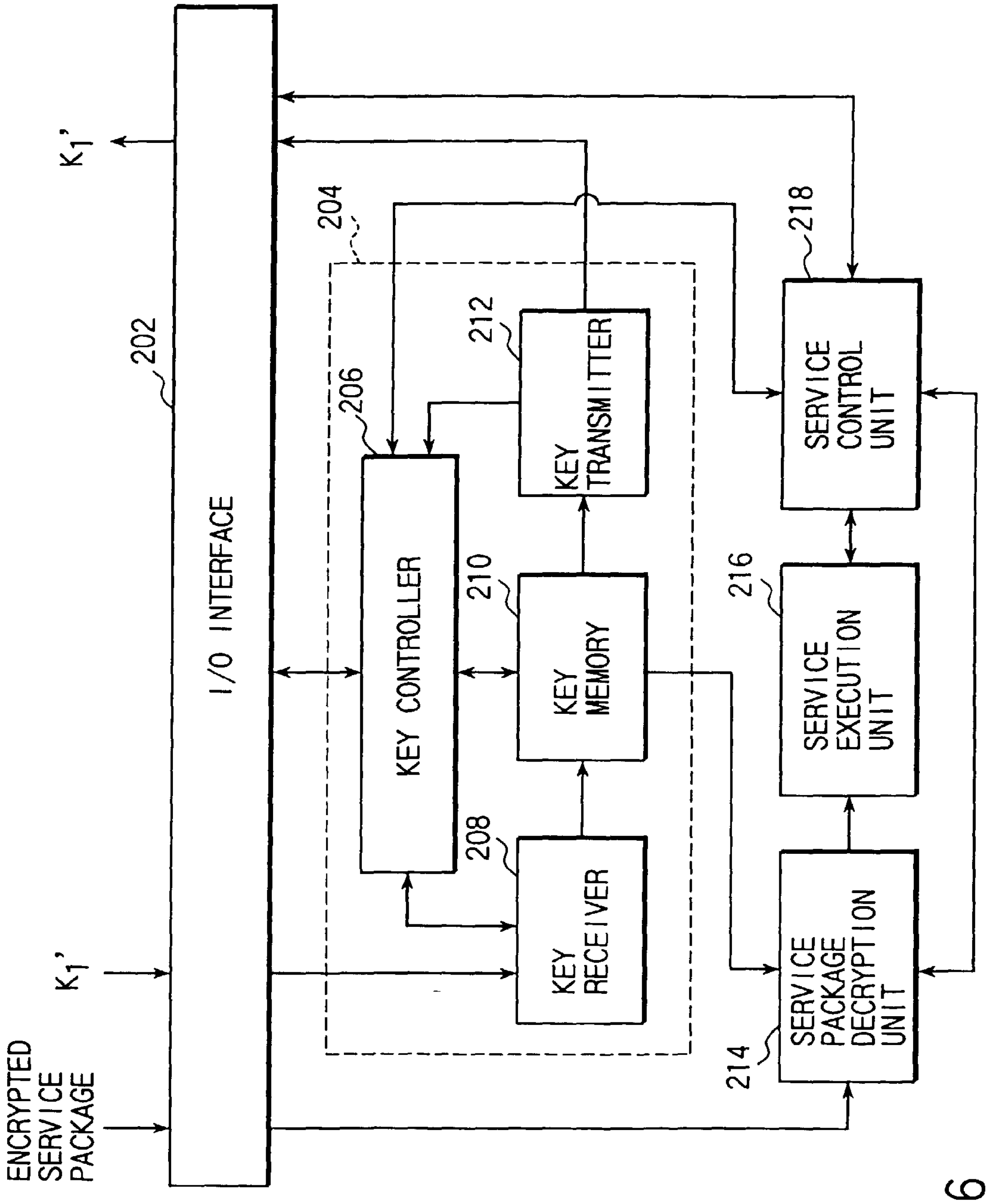

ENCRYPTED SERVICE PACKAGE
K1'
K1'
I/O INTERFACE
202
204
KEY CONTROLLER
206
KEY RECEIVER
208
KEY MEMORY
210
KEY TRANSMITTER
212
SERVICE PACKAGE DECRYPTION UNIT
214
SERVICE EXECUTION UNIT
216
SERVICE CONTROL UNIT
218

FIG.6

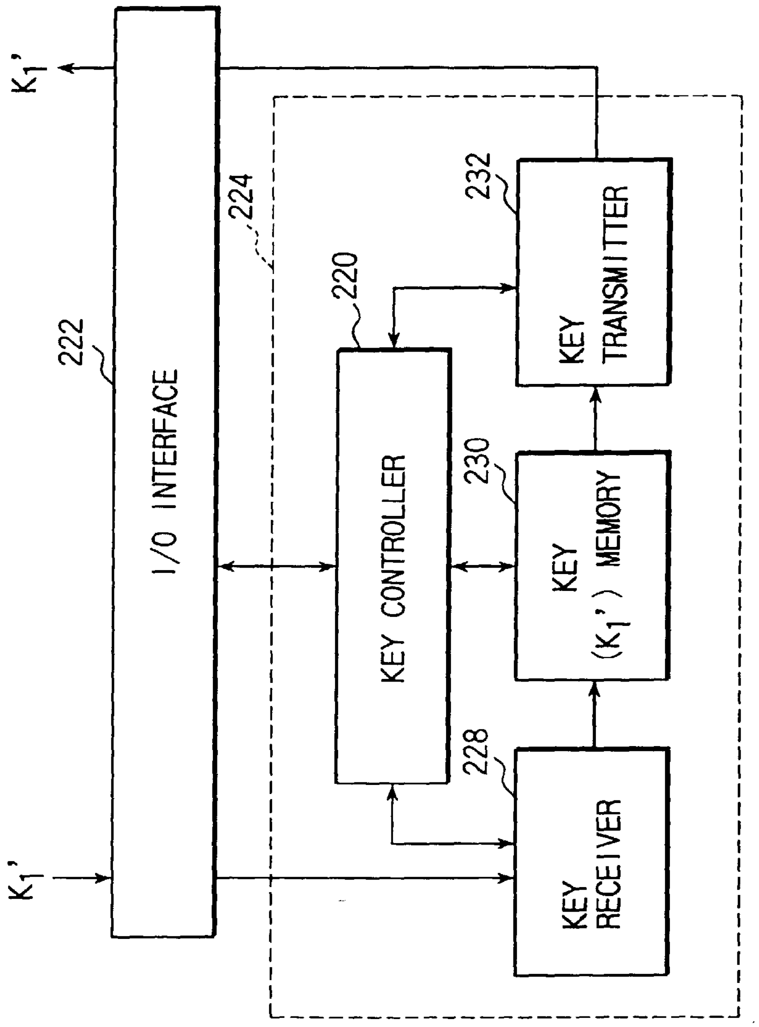

$K_1'$
$K_1'$
222
I/O INTERFACE
224
220
KEY CONTROLLER
232
KEY TRANSMITTER
230
KEY ($K_1'$) MEMORY
228
KEY RECEIVER

FIG. 7

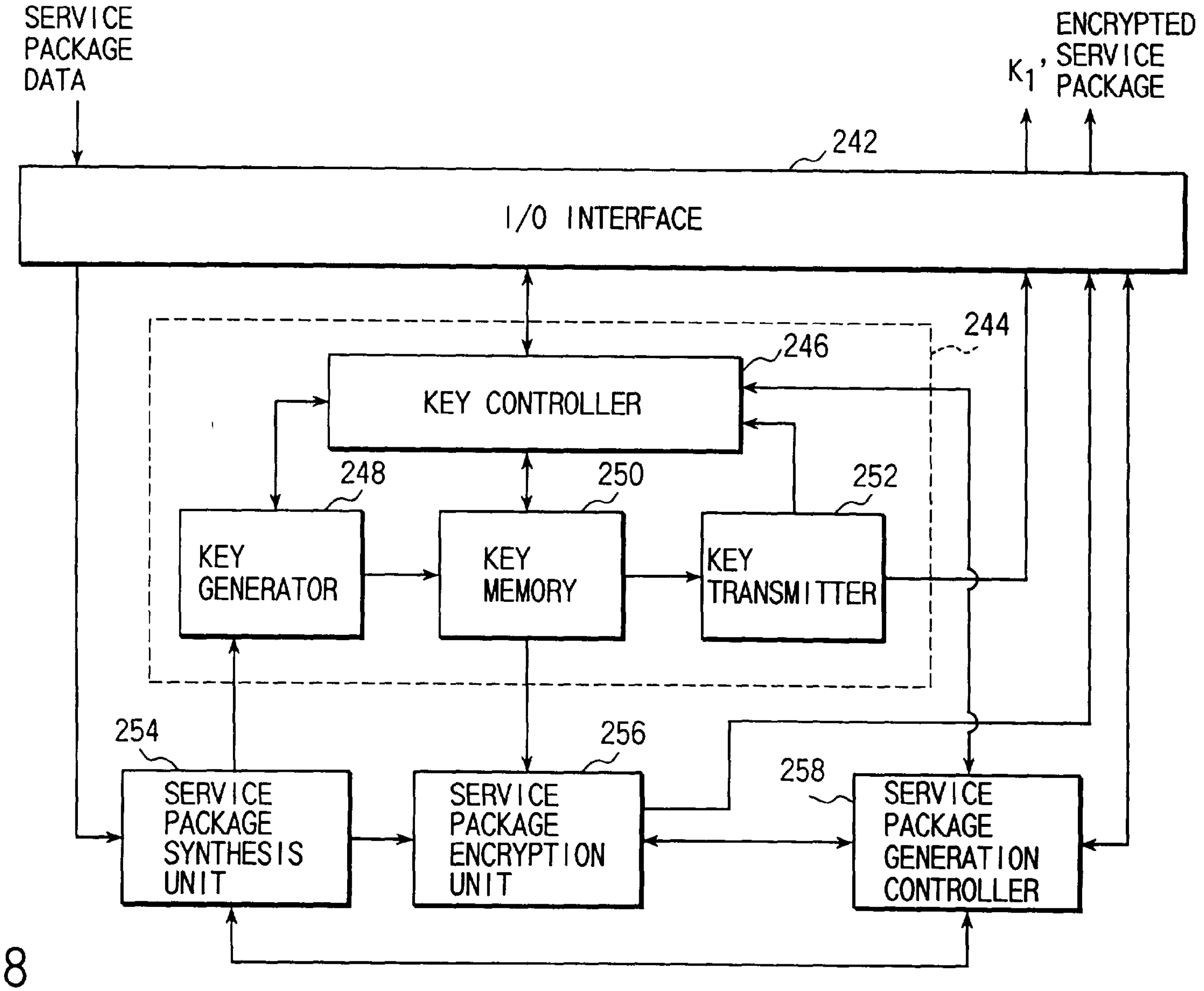

SERVICE PACKAGE DATA
K1, ENCRYPTED SERVICE PACKAGE
242
I/O INTERFACE
244
246
KEY CONTROLLER
248
250
252
KEY GENERATOR
KEY MEMORY
KEY TRANSMITTER
254
256
258
SERVICE PACKAGE SYNTHESIS UNIT
SERVICE PACKAGE ENCRYPTION UNIT
SERVICE PACKAGE GENERATION CONTROLLER

FIG. 8

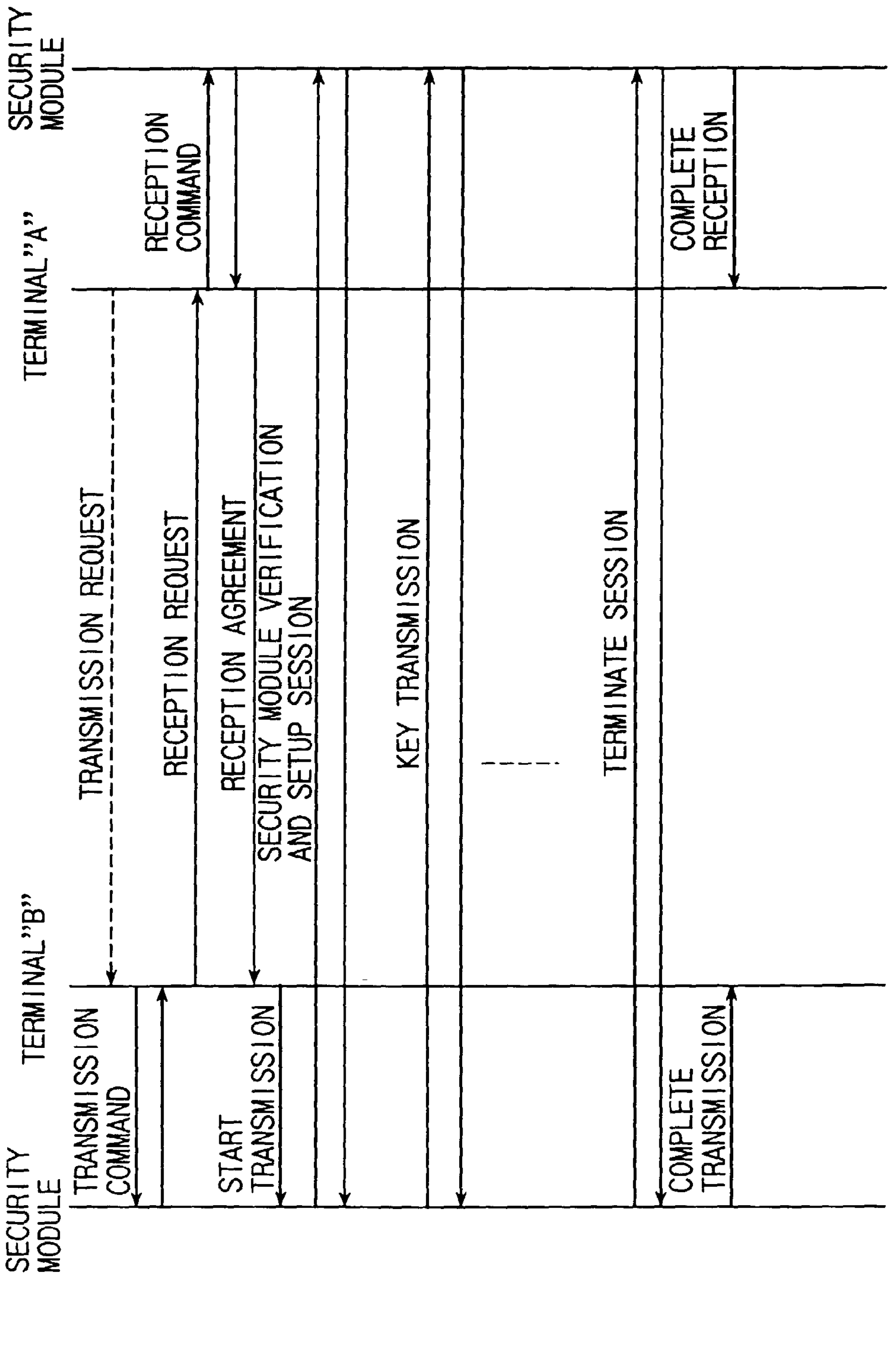

SECURITY MODULE
TERMINAL "B"
TERMINAL "A"
SECURITY MODULE
TRANSMISSION COMMAND
TRANSMISSION REQUEST
START TRANSMISSION
RECEPTION REQUEST
RECEPTION COMMAND
RECEPTION AGREEMENT
SECURITY MODULE VERIFICATION AND SETUP SESSION
KEY TRANSMISSION
TERMINATE SESSION
COMPLETE TRANSMISSION
COMPLETE RECEPTION

FIG. 9

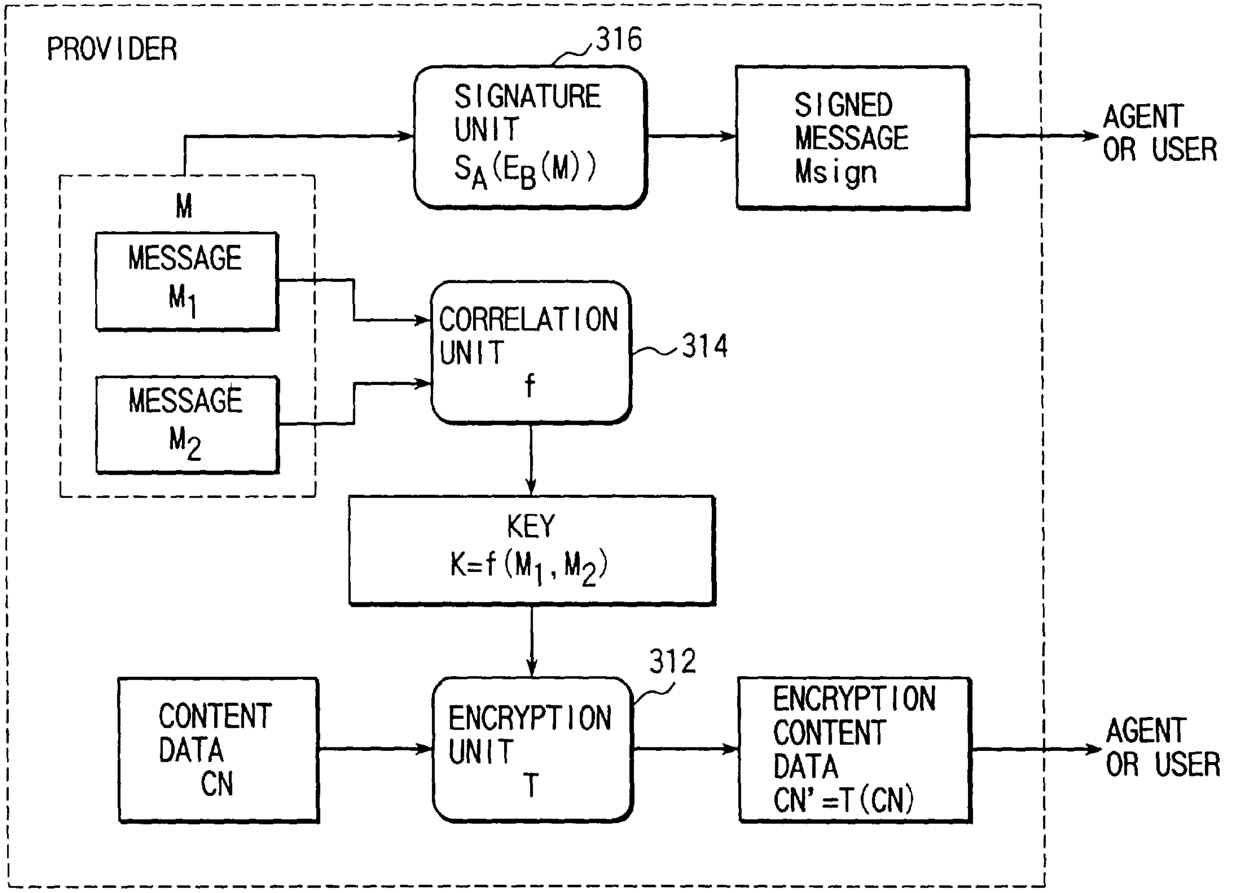

PROVIDER
316
SIGNATURE
UNIT
$S_A(E_B(M))$
SIGNED
MESSAGE
Msign
AGENT
OR USER
M
MESSAGE
$M_1$
MESSAGE
$M_2$
CORRELATION
UNIT
f
314
KEY
$K=f(M_1,M_2)$
312
CONTENT
DATA
CN
ENCRYPTION
UNIT
T
ENCRYPTION
CONTENT
DATA
CN'=T(CN)
AGENT
OR USER

FIG. 10

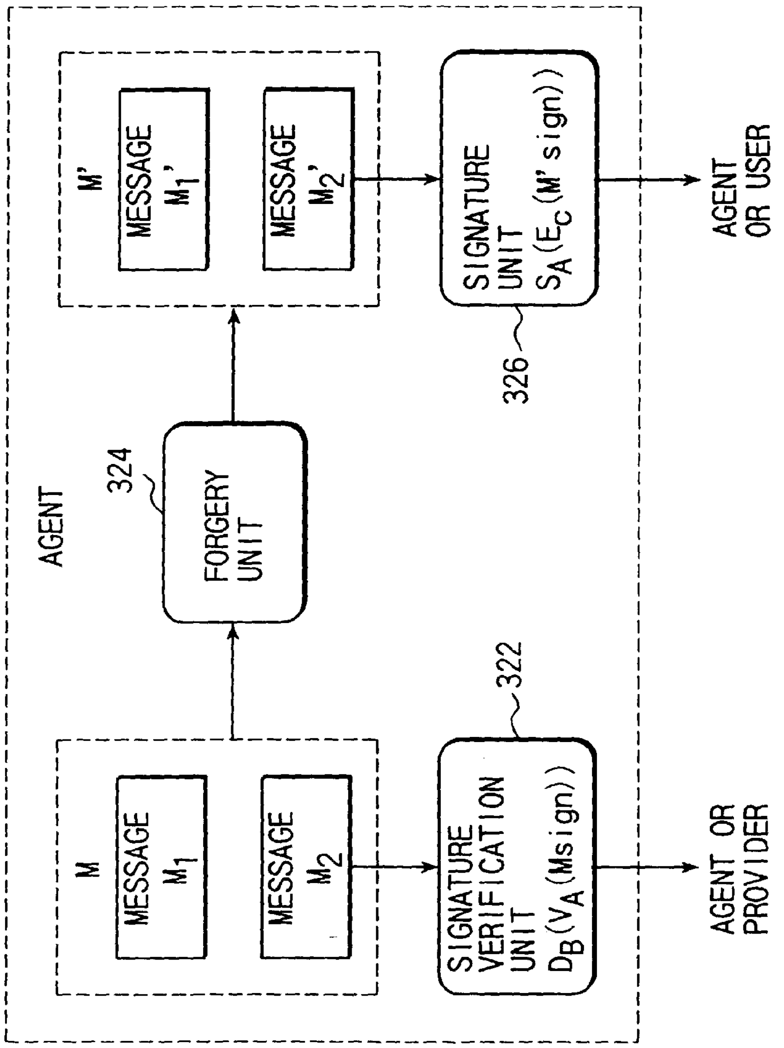

AGENT
M
MESSAGE M1
MESSAGE M2
SIGNATURE VERIFICATION UNIT DB(VA(Msign))
322
AGENT OR PROVIDER
324
FORGERY UNIT
M'
MESSAGE M1'
MESSAGE M2'
SIGNATURE UNIT SA(EC(M'sign))
326
AGENT OR USER

FIG.11

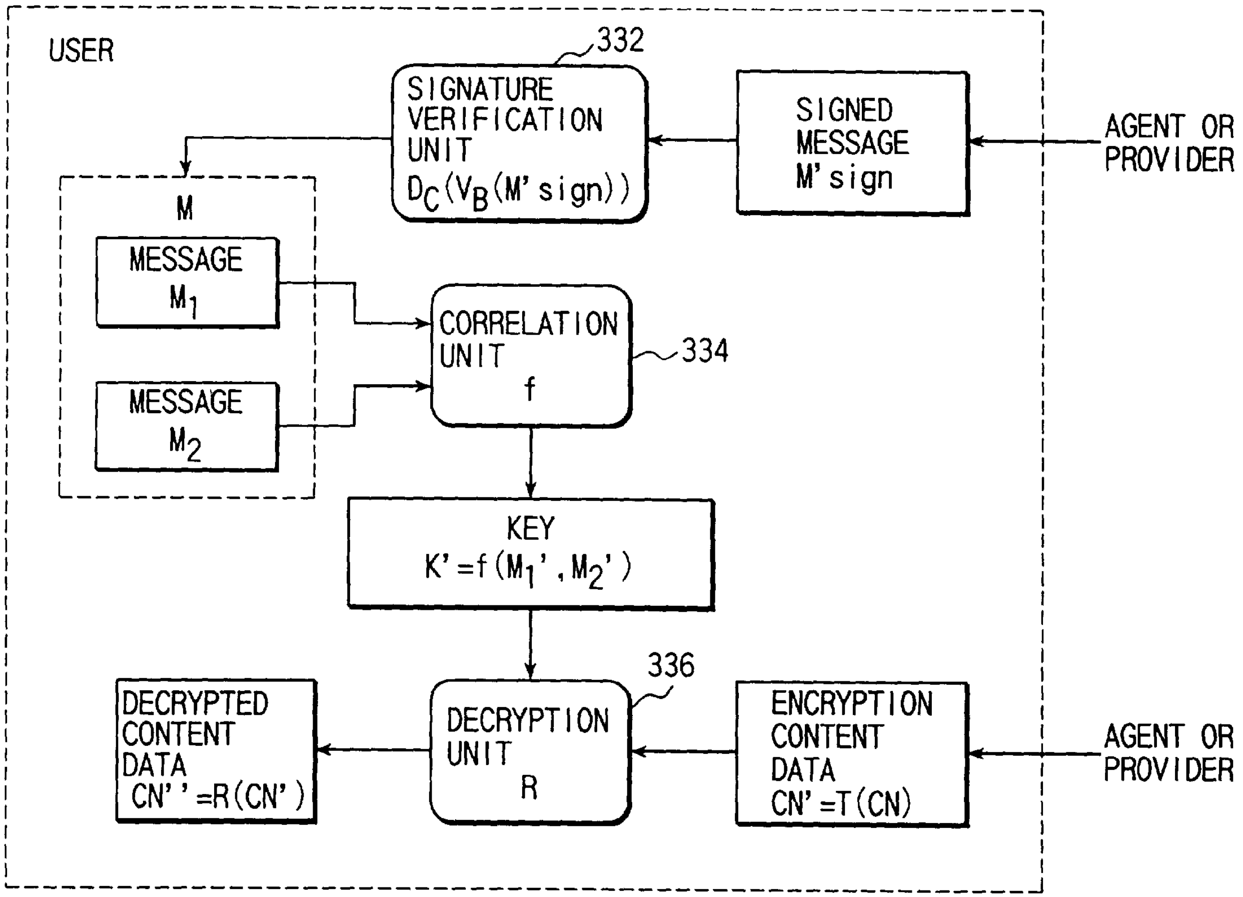

USER
332
SIGNATURE VERIFICATION UNIT $D_C(V_B(M'sign))$
SIGNED MESSAGE M'sign
AGENT OR PROVIDER
M
MESSAGE $M_1$
MESSAGE $M_2$
CORRELATION UNIT f
334
KEY $K'=f(M_1',M_2')$
336
DECRYPTED CONTENT DATA $CN''=R(CN')$
DECRYPTION UNIT R
ENCRYPTION CONTENT DATA $CN'=T(CN)$
AGENT OR PROVIDER

FIG. 12

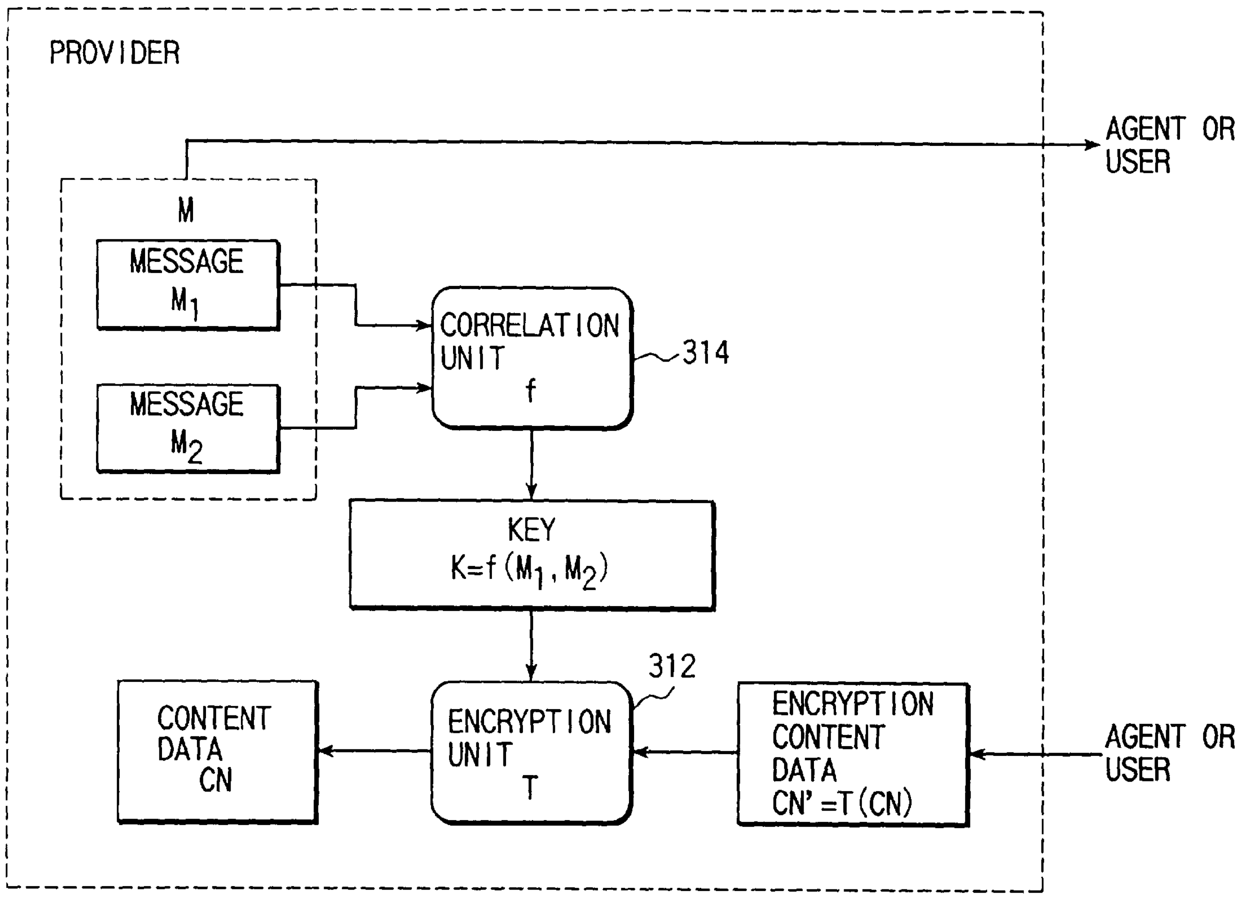

PROVIDER
AGENT OR USER
M
MESSAGE M1
MESSAGE M2
CORRELATION UNIT f
314
KEY K=f(M1, M2)
312
CONTENT DATA CN
ENCRYPTION UNIT T
ENCRYPTION CONTENT DATA CN'=T(CN)
AGENT OR USER

FIG. 13

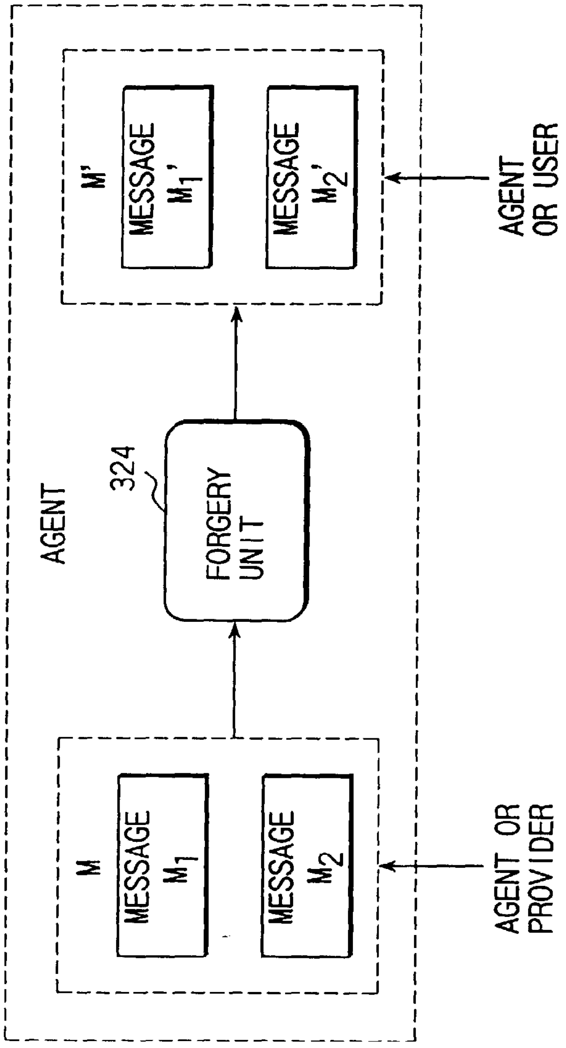

AGENT
324
FORGERY UNIT
M
MESSAGE $M_1$
MESSAGE $M_2$
AGENT OR PROVIDER
M'
MESSAGE $M_1$'
MESSAGE $M_2$'
AGENT OR USER

FIG. 14

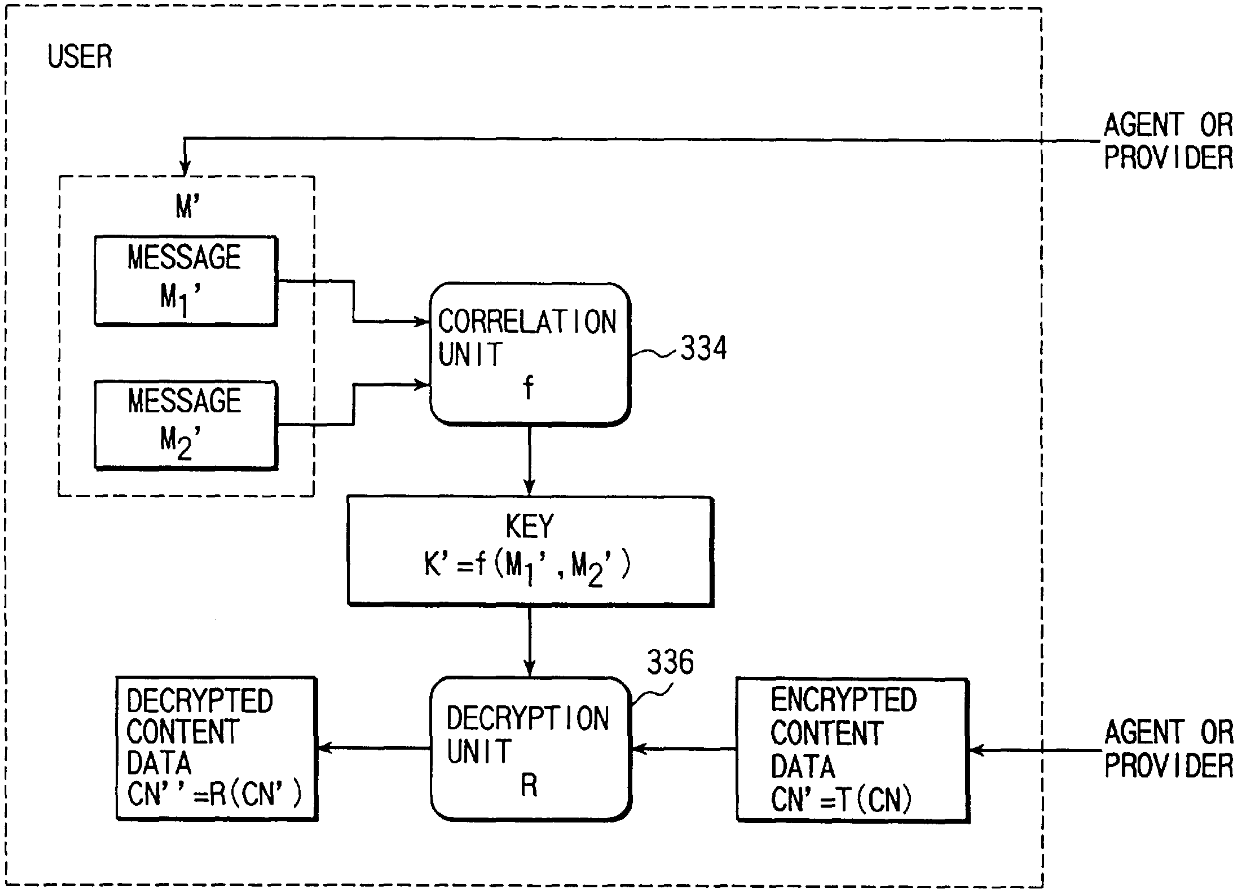

USER
AGENT OR PROVIDER
M'
MESSAGE M1'
MESSAGE M2'
CORRELATION UNIT f
334
KEY K'=f(M1',M2')
336
DECRYPTED CONTENT DATA CN''=R(CN')
DECRYPTION UNIT R
ENCRYPTED CONTENT DATA CN'=T(CN)
AGENT OR PROVIDER

FIG. 15

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 895 171 A2

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
**03.02.1999 Bulletin 1999/05**

(51) Int. Cl.$^6$: **G06F 17/60**

(21) Application number: **98114099.9**

(22) Date of filing: **28.07.1998**

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE**
Designated Extension States:
**AL LT LV MK RO SI**

(30) Priority: **28.07.1997 US 901573**

(71) Applicant: **Neoforma, Inc.**
**Mountain View, CA 94040 (US)**

(72) Inventor: **Mcvicker, Wayne D.**
**Mountain View, CA 94941 (US)**

(74) Representative:
**Schoppe, Fritz, Dipl.-Ing.**
**Schoppe & Zimmermann**
**Patentanwälte**
**Postfach 71 08 67**
**81458 München (DE)**

# (54) System for planning projects

(57) A computer aided planning tool provides a platform having a process-based structure including various planning templates for a project linked to information stored in a relational database related to the task performed utilizing that particular template. The planning tool enables users to navigate from information available in various information galleries to process tools utilized to design the components of the project. The database comprises product catalog information and a library of additional information for targeted access. The information stored in the database is unmodifiable or customizable by designation of the entity which submits that information by including the log-in name and unique password of that entity when the information is submitted. A particular page at a web site may be accessed based on information selected during utilization of a particular planning template to supplement database information, rather than requiring users to browse the web site for the relevant page. Additionally, automated two-way e-mail over the Internet is provided to facilitate the acquisition of additional information which is then accessible by users and can be automatically imported for use in the planning templates, including use in applications programs, such a spreadsheet for calculating costs for the project.

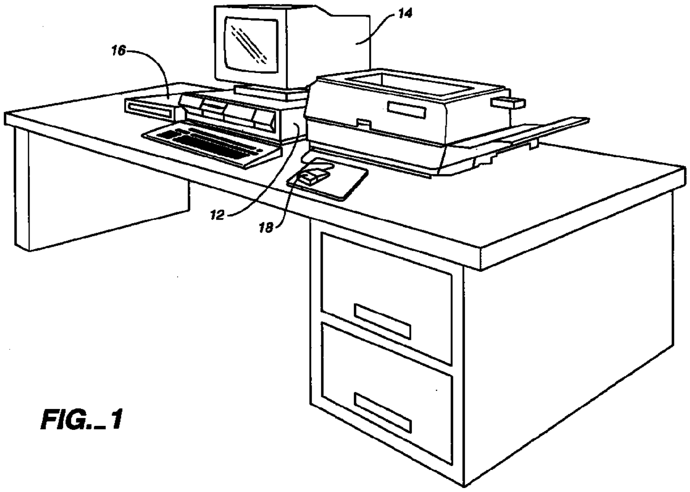


FIG._1

Printed by Xerox (UK) Business Services
2.16.7/3.6

## Description

### Field of the Invention

**[0001]** The present invention relates to productivity tools and, more particularly, to computer aided design (CAD) productivity tools. Specifically, one embodiment of the invention provides a computer aided design tool for persons who plan and implement projects.

### Background of the Invention

**[0002]** Persons who design projects are required to know or learn a great deal of information about a project for which they are assigned the responsibility to design. After that information is possessed, various productivity tools are known for designing projects. These productivity tools include various CAD tools which are commercially available for increasing design productivity. These productivity tools are generally provided for professional designers, such as architects and engineers.

**[0003]** For example, some projects require an in-depth current knowledge and expertise about the project, such as the design of a radiotherapy department for a healthcare facility, which necessitates knowledge of applicable regulations regarding acceptable dosage levels to radiotherapy technicians, materials and equipment utilized in the construction of a radiotherapy department, available commercial products, costs, etc. The design can require a number of complicated calculations, such as the thickness of the shielding that is needed to meet the regulations for radiation dosage levels received by technicians, and the thickness of the shielding depends upon the shielding material which in turn can depend upon the commercial availability of the material and trade-offs between the effectiveness of the material and the cost. This complicates the design process. Therefore, a need has existed for productivity tools for facilitating the design process and reducing the time to design a project, such as the radiotherapy department at a healthcare facility, as well as to enable a person to implement a project design through the purchase of products.

**[0004]** A useful productivity tool for designing a radiotherapy department is known. This productivity tool is commercially available as the Radiotherapy Department Toolkit from Neoforma, Inc. located in Mountain View, California. This toolkit includes a software-based planning tool and product catalog for radiotherapy departments resident on a database to enable facility planning to be efficient and fast by streamlining the complicated tasks of designing a radiotherapy department and purchasing needed equipment. Advantageously, in addition to providing a tool for configuring a radiotherapy department, the Radiotherapy Department Toolkit enables persons designing a radiotherapy department to access product information for selecting equipment for the department from the product catalog database. Additionally, this toolkit incorporates a web browser to enable persons to visit the web sites of various manufacturers whose equipment is included in the product catalog database. The Radiotherapy Department Toolkit enables persons to select shielding material manufacturers, to review current regulations, to generate standardized physics reports, and to simplify the design of complex shield barriers by automating tedious calculations. Therefore, this toolkit has proved to be a beneficial productivity tool for equipment planners, architects, physicists, physicians, administrators, and regulators.

**[0005]** Although the Radiotherapy Department Toolkit has various advantages, there are various limitations. Like CAD tools in the architectural and engineering fields that are limited to one aspect of a multi-faceted project, the Radiotherapy Department Toolkit is a specialized tool for the design of a single department. For example, a radiotherapy department may be only one of various departments at a healthcare facility, such as a hospital that additionally has emergency and operating rooms, a maternity ward, an intensive care unit, patient care and physical therapy departments, etc. Furthermore, the product catalog stored in the database of the Radiotherapy Department Toolkit is specialized, as are the other features, including calculation of shielding needs for radiotherapy equipment. Furthermore, the web browser of the Radiotherapy Department Toolkit simply enables persons to connect to the web sites of equipment manufacturers. Thereafter, persons must browse until the product information being sought is located at the web site.

**[0006]** Therefore, a project planting tool platform is needed that provides an expandable infrastructure that can be adapted as needed to far-ranging project planning applications. Such an adaptable productivity tool for planning a project must be flexible enough to accommodate expansion, such as the addition of planning modules to design various needed aspects of an overall project, and yet be efficiently configured to enhance productivity. Known productivity tools for planning the design of a project do not provide the infrastructure for substantially open-ended expansion and yet possess a level of integration and associational linkages, such as linkages and cross-linkages to and within a database and/or applications programs, such as spreadsheets, to serve as a platform for an efficient planning tool. Therefore, known productivity tools are inherently limited and constitute no more than special-purpose application programs that are not structured to be configured for other applications. At the same time, however, the project planning tool must also enable a person to implement the design for a project at the equipment level.

### Summary of the Invention

**[0007]** One embodiment of the present invention provides an infrastructure for an efficient project planning tool as a platform for a planning tool for planning and implementing a project. The platform provided by the project planning tool can be tailored to any project planning application, from designing, outfitting, and determining the cost of a building project to planning an entire healthcare facility instead of a single department.

**[0008]** The computer aided planning tool in accordance with the invention provides a platform having a process-based structure. The process-based structure comprises various planning templates linked to information stored in a relational database related to the task performed utilizing that particular template in connection with the project being planned. The computer aided planning tool enables users to navigate from information available in various information galleries to process tools utilized to design the components of the project. The database comprises product catalog information and a library of additional information for targeted access. The information stored in the database is unmodifiable or customizable by designation of the entity which submits that information by including the log-in name and unique password of that entity when the information is submitted. For example, product vendors can designate their information unmodifiable and distribute their own files of information to users directly. Database information can be supplemented by enabling users to connect to web sites over the Internet, and in accordance with the invention a particular page at a web site may be accessed based on information selected during utilization of a particular planning template, rather than requiring users to browse the web site for the relevant page. Additionally, automated two-way e-mail over the Internet is provided to facilitate the acquisition of supplemental information which is then accessible by users and can be automatically imported for use in the planning templates, including use in applications programs, such as spreadsheets for calculating costs for the project.

**[0009]** As a result, the computer aided planning tool in accordance with the invention enables users to readily select processes applicable to planning projects for which they have been assigned responsibility and to efficiently navigate within planning templates and access product and library information that enables projects to be implemented. This significantly enhances user productivity.

## Brief Description of the Drawings

**[0010]** The above and other objectives and features and the concomitant advantages of the present invention will be better understood and appreciated by those skilled in the art in view of the description of the preferred embodiments given below in conjunction with the accompanying drawings. In the drawings:

Fig. 1 illustrates one embodiment of the computer aided planning tool in accordance with the invention; and

Figs. 2-21 illustrate various screens that are displayed by the computer aided planning tool shown in Fig. 1 as a project is being planned.

## Detailed Description of the Preferred Embodiments

**[0011]** The computer aided planning tool in accordance with the various embodiments of the invention is executed on a computer 12, as shown in Fig. 1. The computer aided planning tool in accordance with one embodiment of the invention is preferably a 32-bit CAD application compatible with a Microsoft Windows 95 or Microsoft NT 3.51 or later operating system available from Microsoft, Inc. located in Redmond, Washington. The computer 12 comprises a minimum of 16 MB of random access memory (RAM) and preferably includes 24 MB of RAM. The computer 12 also comprises a hard disk drive having 40 MB of free storage space available. The computer is also provided with an Internet connection, such as a modem, for connection to web sites of other entities and exchange of e-mail.

**[0012]** In an alternative embodiment, the computer aided planning tool can be available by connecting to the web site of the supplier of the computer aided planning tool. The computer aided planning tool can be ported to the web and executed on a web server. Therefore, the requirements for the computer 12 would be reduced.

**[0013]** Means for displaying information preferably in the form of a monitor 14 connected to computer 12 is also provided. The monitor 14 can be a 640 x 480, 8-bit (256 colors) VGA monitor and is preferably an 800 x 600, 24-bit (16 million colors) SVGA monitor. The computer 12 is also preferably connected to a CD-ROM drive 16. As shown in Fig. 1, a mouse 18 is provided for mouse-driven navigation between design processes comprising the computer aided planning tool, such as navigation from room to room within one of a plurality of department planning templates, for example, a radiotherapy department for a healthcare facility. The mouse 18 also enables persons utilizing the computer aided planning tool (referred to hereafter as "users") to review and select products and services (collectively referred to hereafter as "products") available from various vendors from a plurality of different categories of products for implementing a project, such as radiotherapy equipment for the radiotherapy department of a healthcare facility that is being planned.

**[0014]** The computer aided planning tool preferably provides three-dimensional visualization of images on monitor 14, such as an entire radiotherapy department for a healthcare facility and the individual rooms in the radiotherapy department being planned. The computer aided planning tool also comprises a relational database having an easy-to-use graphical interface, so that stored product information can be accessed. Three-dimensional visualizations are also preferably provided for specific implementing products. The three-dimensional visualizations of implementing products are linked to screens displayed by the computer aided planning tool on monitor 14 featuring product specifications, drawings, and manufacturer details. All product screens are interrelated for easy navigation between views so users can learn more about the product, vendor, and how to contact the vendor, which are available to users at the click of mouse 18.

**[0015]** The computer aided planning tool also comprises functional modules linked to the information displayable in a planning template and other screens. For example, a spreadsheet application executed in the background can process information related to the project design and use product information accessed from the product catalog stored in the relational database to generate items in a table for determining the cost of various implementing components of a design, as well as the overall cost of the project design. For example, a shielding functional feature for planning a healthcare facility can automate the complicated calculation of shielding barriers for a radiotherapy department to facilitate the design process for the radiotherapy department for a healthcare facility.

**[0016]** The computer aided planning tool in accordance with the various embodiments of the invention comprises a blend of software tools consisting of code executed by a computer, such as computer 12 or a web server connected with computer 12, and a relational database accessible by the computer. The software tools can be encoded on a CD-ROM and downloaded into the computer to execute the computer aided planning routine in accordance with the invention. Data can also be initially provided on a CD-ROM and updated by providing revised data on another CD-ROM or by allowing data to be downloaded over the Internet as the need for updating the data arises. In one exemplary implementation to be described in more detail below, the data stored on the CD-ROM can provide a library or catalog of information related to healthcare for use in planning a healthcare facility, such as products available from various vendors of products to the healthcare industry. Additionally, the computer aided planning tool in accordance with the invention provides access to the Internet so that a web site can be browsed for information related to the project that the computer aided planning tool in accordance with the invention is being utilized to plan. The browser software is integrated into the software planning tool in accordance with the invention in a seamless fashion so that access to web sites of interest can be effected at various times while a project is being planned. In one exemplary implementation to be described in more detail below, web sites of various vendors of products to the healthcare industry are accessible by the computer aided planning tool in accordance with the invention by clicking mouse 18 on an entity name or logo for obtaining additional information relating to planning a healthcare facility, such as supplemental information about products available from various vendors of products to the healthcare industry. The integration of various sources of data for access is a salient feature of the computer aided planning tool in accordance with the invention.

**[0017]** Furthermore, the integration of a local database of product information with Internet access to the web sites of product vendors provided by the computer aided planning tool in accordance with the invention affords a targeted, reliable, easy, and timesaving approach to selection of products needed for the project being planned. This is based on a two-tiered approach. The first tier is that the database provided by the computer aided planning tool provides a general product catalog, but access to that general product catalog is based on the particular selection from among the set of targeted processes that has been selected and using one of various planning templates comprising the computer aided planning tool. Based on the particular planning template that is selected, knowledge-based access is provided to limit the portions of the database, that is, to limit the portions of the general product catalog, that are accessible for product information. The particular portions of the database that are accessible are limited to the particular targeted planning process that has been selected, which means that unlike the prior art, the entire general product catalog in the local database does not have to be browsed for applicable product information. Moreover, the information stored in the particular portions of the relational database becomes immediately accessible in the context of the process within the planning template, which obviates the need for users to browse for the information.

**[0018]** The general product catalog can be updated by providing revised data on another CD-ROM or preferably by allowing data to be downloaded over the Internet as the need for updating the data arises. For example, an updated CD-ROM can be created by downloading general product information from product vendors and providing the updated product information to users of the computer aided planning tool in accordance with the invention. Preferably, however, the general product catalog is updated by users directly downloading general product catalog information over the Internet. In any event, the general product catalog stored in the database accessible by the computer aided planning tool in accordance with the invention is preferably maintained by the product vendors. Therefore, the responsibility for providing the data to update the general product catalog resides with the vendors whose products appear in the database.

**[0019]** In this regard, each vendor is assigned a log-in name and a unique password by the supplier of the computer aided planning tool in accordance with the invention. The unique password enables the product vendor to access the portion of the archival database maintained by the supplier of the computer aided planning tool in accordance with the invention dedicated to the particular products of that product vendor included in the general product catalog. Access is preferably provided over the Internet. This enables product vendors to update information regarding their products that appear in the general product catalog as the need arises for updating that information, such as discontinuing a product, adding a new product to a product line, etc. One advantage of the computer aided planning tool in accordance with the invention is that the responsibility for updating general product information preferably resides with the set of product vendors, so that the best interests of the product vendors are served by their diligently updating their general product information as business circumstances dictate. This can be easily and quickly accomplished by downloading data over the Internet with the assurance that the integrity of the general product catalog can be maintained by the assignment of unique passwords. This also reduces the burden of database maintenance on the supplier of the computer aided planning tool in accordance with the invention.

**[0020]** One embodiment of the computer aided planning tool in accordance with the invention provides a set of targeted processes. In one exemplary implementation to be described in more detail below, the target processes relate to planning a healthcare facility, such as selecting the departments that will comprise the healthcare facility, architecting the physical layout of the facility including the various rooms from operating rooms to patient rooms to waiting rooms, selecting actual equipment to deploy in the architected rooms, such as medical equipment, furniture, and even decorations, and other products available from various vendors of products to the healthcare industry. The targeted processes are frilly enabled by the computer aided planning tool in accordance with the invention. A relational database is provided for providing information needed in the planning process. Access to applicable information stored in the database is determined by the particular targeted process that is selected. A spreadsheet is executed in the background for performing calculations needed in the planning process. In one exemplary implementation to be described in more detail below, the spreadsheet application software can compute costs for the healthcare facility being planned, such as cost information depending upon the materials, equipment, accessories, and other costs involved in constructing and furnishing a healthcare facility. A suite of conventional software application tools can be provided to enable planning, such as architectural application software.

**[0021]** In order to implement the targeted processes provided by the computer aided planning tool in accordance with the invention, a process-based graphic user interface (GUI) is provided. The process-based interface comprises a menu of available selections corresponding to the targeted processes that are provided by the computer aided planning tool in accordance with the invention. Additionally, the GUI of the computer aided planning tool in accordance with the invention is intuitive, and a sufficient number of help screens are provided that the need for hardcopy documentation in the form of a user manual is obviated.

**[0022]** Preferably, each of the targeted planning processes is connected to a portion of the general product catalog stored in the database related to that planning process, so that the general product catalog provides the backbone of the computer aided planning tool in accordance with the invention. This enables users of the computer aided planning tool in accordance with the invention to migrate easily from a targeted planning process directly to specific product information, which substantially reduces the time needed to fully plan a project and obtain concrete information, such as cost information, for project implementation. Tables generated by users including the results of any calculations performed by the spreadsheet can be utilized to generate an order report utilizing the reporting features of the computer aided planning tool.

**[0023]** The computer aided planning tool in accordance with the invention provides a specialized design tool integrated with the product database backbone. The computer aided planning tool is also integrated with community provided information in a library database for enabling entities provided with the computer aided planning tool to design and implement a project including complicated technical aspects of the project. In one embodiment of the invention, the library information stored in the database comprises information which can only be modified by the entity which has submitted the information and, alternatively, the information can be customizable. Typically, product information submitted by vendors is submitted with the log-in name and unique password of the vendor so that users cannot modify the information. An entity which submits library information can also include its log-in name and unique password with library information to render that information unmodifiable. Otherwise, information in the database can be imported by users and modified. Enabling product and library information to be designated as unmodifiable can better assure the integrity of the database.

**[0024]** The computer aided planning tool preferably includes integrated spreadsheet applications software integrated with the computer aided planning tool, which is linked to predetermined selections from the planning templates and executes in the background for calculating the quantity and cost of components needed to implement a project For example, the spreadsheet applications software can be configured to calculate the amount of shielding material based on a selected type of material using conventional equations for determining the amount of a material based on the shielding properties of the particular material and then calculates the associated cost of utilizing that material in a radiology department of a healthcare facility. The spreadsheet is integrated with the database and linked by selection of the planning template to product information and library information to import the information needed to perform the needed calculations to determine quantity and cost which are then displayed. Preferably, the computer aided planning tool displays the particular product, together with the quantity and cost, in tabular form. Therefore, complicated aspects of planning a project are processed in the background by the computer aided planning tool, so that complicated procedures connected with a selected planning template and various procedures that require expert knowledge are performed in the background simply through the selection of a template and straightforward selection of available components, such as products available to implement those components. This facilitates planning a project, particularly a complicated project that may require expert knowledge in a particular field. Advantageously, even an expert in the particular field is able to save a great deal of time, since tedious calculations are performed automatically by the computer aided planning tool.

**[0025]** The library of information stored in the database is also preferably updatable through connection over the Internet to the web site of the entity which published the information. The library of information comprises submittals of information by the universe, or community, of entities which are provided with the computer aided planning tool. Any person who has authorized access to the computer aided planning tool at an entity provided with the computer aided planning tool can author and submit information for access by the entire community of entities which also are provided with the computer aided planning tool and users at those entities. Any such entity or person who desires to author information for inclusion in the library can add the information to the web site of the entity where the information is authored and/or can submit the information to the supplier of the computer aided planning tool. The submitted information is published through the library available to entities provided with the computer aided planning tool. The information published by the supplier of the computer aided planning tool and/or not submitted but simply accessible through browsing the web site of the entity at which the information was authored enables direct access by users of the computer aided planning tool. Preferably, the computer aided planning tool includes linkages to the addresses of the web sites of entities which publish in the subject areas related to the selected planning template as part of the relational database, so that users can easily determine the web site addresses which they can browse if they desire to seek additional information. The interests of all are served by the submission of information for publication in the library.

**[0026]** For example, one group of entities provided with the computer aided planning tool is, of course, persons who desire to plan a project, such as a healthcare facility. It is important that those persons are informed regarding any applicable regulatory requirements for the project, such as the maximum level of exposure to radiation that is permitted to technicians who work in a radiation therapy or radiology department at a healthcare facility. Moreover, these persons are also well-served to learn from the experience of other entities which have implemented a similar project and to also determine whether or not an industry or trade association or other professional group has published guidelines or standards that facilitate the implementation of a project, such as guidelines by the American Medical Association for equipping a hospital emergency room or intensive care unit. Moreover, some entities may voluntarily publish information about a project that may be invaluable to another entity planning a project at a later time and the prestige that can attach to the publishing entity when other entities re-use part or all of an implementation developed by the publishing entity, such as one healthcare planner implementing an intensive care unit based on the implementation of such a department submitted by Stanford University Hospital.

**[0027]** Another group of entities provided with the computer aided planning tool and well-served by submission and publication of information is any governmental agencies which oversee regulation of the implemented project, such as a healthcare facility. The fact that there are regulations regarding an implementation that incorporates regulated equipment, such as radiation equipment in a healthcare facility, has been mentioned earlier. Other regulations may also apply to a project being planned, such as building codes that apply to a healthcare facility. Government agencies and other regulatory or oversight agencies can easily and effectively disseminate needed information by submitting the information to the provider of the computer aided planning tool for publication or having that information available at their web site for access by users of the computer aided planning tool.

**[0028]** An additional group of entities provided with the computer aided planning tool and well-served by submission and publication of information is professional consultants who specialize in the planning and/or implementation of various aspects of the given type of project and who are available to be retained by the entity accessing the library for information, for example, a hospital architectural firm which could be retained to render actual plans for construction of a hospital. The library accessible by the computer aided planning tool provides a vehicle for a professional consultant to publicize expertise and prior experience by having that information available at the web site of the consultant for access by users of the computer aided planning tool.

**[0029]** As a further and quite obvious example, a group of entities provided with the computer aided planning tool and motivated to submit information for publication is vendors of products for actual implementation of the various components of the project that is the subject of the planning. The library accessible by the computer aided planning tool provides a mechanism for a product vendor to publicize important product development and product announcements, such as the development of new oncology equipment for the radiotherapy department at a healthcare facility, by having that information available at its web site for access by users of the computer aided planning tool.

**[0030]** The groups of entities that would publish in the library accessible through the computer aided planning tool in accordance with the invention vary from one type of project being planned to another. Therefore, there is no limit to the community of entities which are provided with the computer aided planning tool and persons at those entities who would author and submit information.

**[0031]** Considered in more detail, the product database stores product information. The product information comprises graphic and/or text information relating to products. The product graphics and/or text for the various product information stored in the database is preferably provided and updated as needed (i.e., maintained) by the respective vendors of those products. Similarly, the library information comprises graphics and/or text information relating to products or other aspects of a project. The graphics and/or text for the various library information stored in the database is also preferably provided and updated as needed (i.e., maintained) by the entities of the community which submit that information for publication. The computer aided planning tool displays this graphic and/or text when a planning template is active and the product is selected for implementing a component of the project being planned.

**[0032]** When a particular component is selected within a planning template, one embodiment of the computer aided planning tool in accordance with the invention enables additional information relating to that component, such as a product, to be obtained, that is, the computer aided planning tool provides a mechanism to obtain information to supplement product information stored in the database. The additional information is obtained by selecting a specific product, for example, by clicking mouse 18 on a product listed on a screen displaying one or more products of a vendor. In response to selection of the specific product, the computer aided planning tool connects over the Internet to the vendor and directly accesses the page at the web site of the vendor where the additional information about that particular product appears. This obviates the need for users of the computer aided planning tool to browse the web sites of vendors for additional product information when specific products have been selected.

**[0033]** In another embodiment, the additional information is obtained by a two-way e-mail exchange between the computer aided planning tool and the vendor of the selected implementing product or entity which authored the library information. To facilitate the exchange of e-mail, the computer aided planning tool preferably provides check boxes within the planning template, which can be selected to request a quote or supplemental information or references relating to a product. If a check box is not selected, the request is preferably defaulted to a request for supplemental information relating to the product. In response to selection of a check box, the computer aided planning tool preferably automatically assembles a pre-formatted e-mail message by the computer aided planning tool to the e-mail address of the vendor or publishing entity stored in the database together with the product or library information. The assembled request message contains: an identification of the product or publication about which supplemental information is requested, the e-mail address of the vendor or product entity to which the query is to be sent as well as the e-mail address of the computer aided planning tool which sends the request, and a pre-formatted text message dependent upon the selected check box, such as "Please reply with pricing information for the product identified in the subject of this message" or "Please reply with all available product information respecting the product identified in the subject of this message." Preferably, the computer aided planning tool enables customized notes to be appended to the pre-formatted text, for example "Please also specify earliest delivery date and method of shipment." If the vendor or publishing entity does not have a current e-mail address, the assembled e-mail request preferably additionally incorporates the facsimile transmission number of the requesting entity, and the e-mail address is preferably defaulted to the e-mail address of the supplier of the computer aided planning tool. Upon receipt of the e-mail message with the default e-mail address by the supplier of the computer aided planning tool, the supplier can convert the e-mail message to a facsimile transmission and send the facsimile transmission to the vendor or publishing entity with a request to reply by facsimile transmission to the telephone number of the computer aided planning tool which sent the request. For example, if additional information is desired about a large number of products needed for implementation of a project being planned, an e-mail message is assembled for each product as to which additional information is sought and sent to each of the product vendors, which may result in a substantial volume of e-mail messages being sent to product vendors by the computer aided planning tool.

**[0034]** If the vendor or publishing entity to which the request is addressed also possesses e-mail capability, a pre-formatted reply is assembled. The assembled reply message contains: an identification of the product or publication about which supplemental information was requested, the e-mail address of the computer aided planning tool which sent the request as well as the e-mail address of the vendor or publishing entity which sends the reply, and a pre-formatted text message dependent upon the selected check box, such as

"The price of the product identified in the subject of this message is $...." or "Supplemental product information respecting the product identified in the subject of this message is...." Preferably, the reply can also include a separate section for reply to any customized note appended to the pre-formatted text, for example, "The earliest delivery date is ... and method of shipment is by overland freight."

**[0035]** Two-way e-mail over the Internet provides significant advantages. One advantage is that e-mail affords an effective mechanism for communicating product and published information. Another advantage is that e-mail obviates the need to interrupt the planning process to telephone or draft a letter request for supplemental information. E-mail is particularly convenient when the vendor or publishing entity is located In another time zone or another country. Consequently, international products and published library information from abroad can more logically be included in the database. Therefore, the computer aided planning tool has global geographical application.

**[0036]** Also, a preferred embodiment of the computer aided planning tool in accordance with the invention automatically processes replies by vendors and publishing entities to e-mail requests. Users route e-mail replies to an e-mail queue when the e-mail application of the computer on which the computer aided planning tool executes is active. When the computer aided planning tool is thereafter executed, the e-mail replies in the e-mail queue are automatically parsed, and the pre-formatted replies are disassembled and the supplemental information is de-embedded and entered in the tables of the planning templates that existed at the time when the check boxes of those templates were checked to generate the corresponding requests. The format of the de-embedded information is compatible with the other information contained in the tables that have been generated within the planning template. This provides a very powerful and automated mechanism for inclusion of supplemental information for planning a project and implementing components of that project. This supplemental information is also processed by any spreadsheet application program executing in the background to perform calculations, such as the total cost of implementing a project.

**[0037]** One implementation of the computer aided planning tool in accordance with the invention will now be described. It is to be understood that the implementation to be described is by way of example and not by way of limitation. The features of the computer aided planning tool in accordance with the invention apply generally to planning a project and are not restricted in any manner to the specific implementation to be described below. Other exemplary potential implementations will be mentioned following a detailed description of the one exemplary implementation.

**[0038]** One exemplary implementation provides a computer aided planning tool for healthcare. In one implementation, for example, the computer aided planning tool can be a healthcare facility planning tool.

**[0039]** The computer aided planning tool comprises a relational database comprising product information from vendors. The database also comprises a library comprising worldwide regulatory agency regulations and other information, such as files generated by record-and-verify systems like VARiS, which can be accessed from the library stored in the database. Also, the library can store information relating to architectural solutions that present a sampling of architectural drawings and graphics of architectural design implementations from around the world for evaluation by an architect involved in a planning project for a healthcare facility. The computer aided planning tool in accordance with one implementation of the invention can include a spreadsheet application, such as Lotus 1-2-3 available from Lotus Development Corporation, which is pre-configured to perform various calculations within a selected planning template such as the cost of a project being planned. A spreadsheet can be also be configured to calculate shielding in connection with a process which comprises designing a radiotherapy department. Automated shielding calculations are provided by the known Radiotherapy Toolkit and therefore do not in and of themselves constitute the present invention. Products available from vendors, such as commercially available doors ad wall assemblies, can be selected, and parameters included in the information relating to those implementing components stored in the product database are called by the spreadsheet and used in the calculations performed by the spreadsheet. Regulatory agency defaults can also be used in the spreadsheet calculations.

**[0040]** The computer aided planning tool, which comprises a program or set of coded instructions which is executed by computer 12 or a web server to which computer 12 is connected, will now be described in detail with reference to Figs. 2-21. Fig. 2 illustrates an opening screen, which can be referred to as the process screen, that is displayed by monitor 14 when the computer aided planning tool is opened or accessed. In the event that the computer aided planning tool is executed by a web server connected to computer 12, the process screen can be denoted as the "Home Page," as indicated at the lower right hand corner of Fig. 2.

**[0041]** The process screen shown in Fig. 2 is the screen in which the interface to the various processes throughout the computer aided planning tool is defined. Generally, the computer aided planning tool comprises many different screens associated, or linked, in many ways. Each screen is treated as a unique object, which enables screens to be bundled through linkages in different orders to create different modules. The process-based platform provided by the computer aided planning tool provides navigation through a variety of tools and a variety of screens of information accessed from the associated relational database, and the process-