AUTOMATED MEDIA DELIVERY SYSTEM

OPTIMIZATION OF MEDIA CONTENT USING GENERATED INTERMEDIATE MEDIA CONTENT

CROSS REFERENCE TO RELATED APPLICATIONS

This application is a Continuation of U.S. Ser. No. 12/238,842, filed Sep. 26, 2008, which is a Divisional of U.S. Ser. No. 12/173,747, filed Jul. 15, 2008, which is a Divisional of U.S. Ser. No. 11/269,916, filed Nov. 7, 2005, which is a Continuation-in-Part of U.S. Ser. No. 09/929,904, filed Aug. 14, 2001, now U.S. Pat. No. 6,964,009 granted on Nov. 8, 2005, which is a Continuation-in-Part of U.S. Ser. No. 09/425,326, filed Oct. 21, 1999, now U.S. Pat. No. 6,792,575, granted on Sep. 14, 2004, each of which is hereby incorporated in its entirety by this reference thereto.

BACKGROUND OF THE INVENTION

[0001] 1. Technical Field

[0002] 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.

[0003] 2. Description of the Prior Art

[0004] 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.

[0005] 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:

[0006] Current Workflow for Creating Web Graphics

[0007] Original Artwork/Asset Creation

[0008] Use third-party point products

[0009] Asset Editing

[0010] Scale/reduce/slice

[0011] Asset Format Conversion

[0012] JPEG/GIF/PNG

[0013] Asset Staging

[0014] Place in Web file system

[0015] Edit HTML

[0016] Create/Modify HTML for particular page

[0017] Store HTML on Web server

[0018] View final pages

[0019] Repeat process for each version of each graphic on each page

[0020] Estimated timeTime

[0021] Two hours per page times the number of pages

[0022] 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.

[0023] 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. 17-20, 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."

[0024] 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."

[0025] John Mills Dudley, Network-Based Classified Information Systems, AU-A-53031/98 (Aug. 27, 1998/08/98) 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."

[0026] Cordell et al, Automatic Data Display Formatting with A Networking Application, U.S. Pat. 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."

[0027] 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. Pat. 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."

[0028] Wistendahl et al, System for Mapping Hot Spots in Media Content Interactive Digital Media Program, U.S. Pat. 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."

[0029] Rogers et al, Method for Fulfilling Requests of A Web Browser,. U.S. Pat. No. 5,701,451 (Dec. 23, 1997) and Lagarde etat al, Method for Distributed Task Fulfillment of Web Browser Requests, U.S. Pat. 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."

[0030] 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."

[0031] 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.

[0032] 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.

[0033] 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.

[0034] 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.

[0035] 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.

[0036] The key to IrisTransactive is the compositing subsystem. It requires three things to build a shopping solution using image composition.

[0037] 1) The original images must be created. It is suggested that the image be converted to Flashpix for better performance.

[0038] 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 5 k and 50 k.

[0039] 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.

[0040] 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 5 k-50 k 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.

[0041] 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.

[0042] 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.

[0043] MGI Software sells point solutions that require end-users to download a viewer to process a proprietary image format.

[0044] 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.

[0045] 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.

[0046] It would be advantageous to empower an end user with flexibility and control by providing interactive page capabilities.

[0047] 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.

[0048] 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.

[0049] 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.

[0050] 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.

[0051] 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.

[0052] 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.

[0053] It would be advantageous to provide a system that generates media, whereby the media is automatically uploaded.

[0054] It would be advantageous to provide a system that automatically caches generated media so identical requests can be handled without regeneration of images.

[0055] It would be advantageous to provide a system that resides behind the Web server, thereby eliminating security issues.

[0056] It would be advantageous to provide a system wherein the client browser does not require a plug-in.

[0057] It would be advantageous to provide a system wherein the system does not require any changes to a Web server.

[0058] It would be advantageous to provide a system wherein the system manages the Web server media cache.

[0059] It would be advantageous to provide a system wherein the Web media is generated only if requested by a client browser.

[0060] 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.

[0061] 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.

[0062] 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.

[0063] 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

[0064] 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.

[0065] 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.

[0066] 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

[0067] FIG. 1 is a schematic diagram showing the placement of the system within a current Web infrastructure according to the invention;

[0068] 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;

[0069] FIG. 3 is a schematic diagram showing delivery of an HTML document and media to a Web browser according to the invention;

[0070] FIG. 4 is a schematic diagram showing the components involved in Web site administration according to the prior art;

[0071] FIG. 5 is a schematic diagram showing the components of the system involved in Web site administration according to the invention;

[0072] FIG. 6 is a simple overview showing the components of the system according to the invention;

[0073] FIG. 7 is a schematic diagram showing the process flow of a proprietary enabled page delivered to a Web browser according to the invention;

[0074] FIG. 8 is a flow chart showing an authoring process according to the invention;

[0075] FIG. 9 is a flow chart showing an HTML parsing process according to the invention;

[0076] FIG. 10 is a flow chart showing a media creation process according to the invention;

[0077] FIG. 11 is a screen shot showing an administration tool according to the invention;

[0078] FIG. 12 displays a structure of a database record used for the system according to the invention;

[0079] FIG. 13 shows original media to be processed according to the invention;

[0080] FIG. 14 shows a portion on an HTML document with a proprietary tag according to the invention;

[0081] FIG. 15 shows an HTML document and an HTML document source according to the invention;

[0082] FIG. 16 shows a generated GIF image according to the invention;

[0083] FIG. 17 is a schematic diagram of an image system within a typical Web infrastructure according to the invention;

[0084] FIG. 18 is a schematic diagram showing delivery of an HTML document and original media according to the invention;

[0085] FIG. 19 is a schematic diagram showing components of Web site administration according to a preferred embodiment of the invention;

[0086] FIG. 20 is a simple overview showing components of the image system according to a preferred embodiment of the invention;

[0087] 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;

[0088] FIG. 22 shows a flowchart of a content generation procedure according to a preferred embodiment of the invention; and

[0089] FIG. 23 is a flow chart showing an authoring process according to a preferred embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0090] 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.

[0091] 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.

[0092] 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.

[0093] 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.

[0094] 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.

[0095] 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.

[0096] 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.

[0097] 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.

[0098] 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.

[0099] 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.

[0100] The System.

[0101] A preferred embodiment of the system 100 is provided.

[0102] 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.

[0103] 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.

[0104] 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.

[0105] 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.

[0106] The media processing engine 650 interprets the proprietary tag and generates the image. The media

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.