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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 November 2011 (03.11.2011)

PCT

(10) International Publication Number
WO 2011/137146 A1

(51) International Patent Classification:
G06F 17/21 (2006.01)

(21) International Application Number:
PCT/US2011/034050

(22) International Filing Date:
27 April 2011 (27.04.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/769,839 29 April 2010 (29.04.2010) US

(71) Applicant *(for all designated States except US)*: **MONOTYPE IMAGING INC.** [US/US]; 500 Unicorn Park Drive, Woburn, Massachusetts 01801 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **LEE, Steve Hsing** [US/US]; 5307 Archstone Avenue, Tewksbury, Massachusetts 01876 (US).

(74) Agent: **BARCLAY, Jeffrey J.**; Fish & Richardson P.C., P.O. Box 1022, Minneapolis, Minnesota 55440-1022 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

— *with amended claims (Art. 19(1))*

(54) **Title**: INITIATING FONT SUBSETS

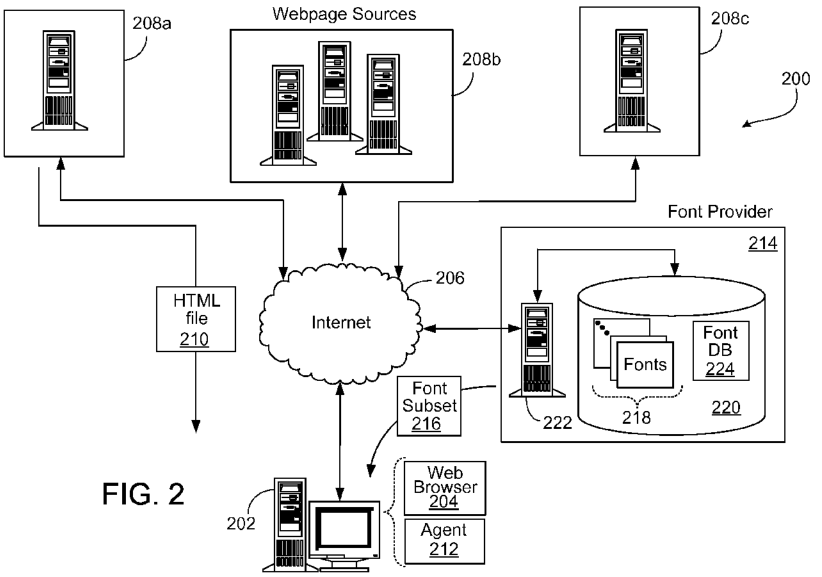


FIG. 2

(57) **Abstract**: A computer-implemented method includes analyzing content of a received electronic document in an autonomous manner to identify each unique character of one or more fonts included in the content of the electronic document. The method also includes initiating a request for a subset of one or more of the fonts included in the content of the electronic document, wherein the request includes each identified unique character of the respective one or more fonts.

Initiating Font Subsets

BACKGROUND

[001] This description relates to initiating the production and delivery of font subsets.

[002] In the ever-expanding connectivity provided by computer networks such as the Internet, various types of content such as text, graphics, audio, video, etc. may be exchanged between a few computing devices or broadcast to a multitude. Due to such connectivity, users located throughout various regions of the globe may access content providers who need the flexibility of providing content to users of different nationalities and cultures. For example, content provided to North American users may not be discernable to Asian users. As such, providers may tailor content for users in particular geographical regions or provide adjustable content to different users.

SUMMARY

[003] The systems and techniques described here relate to producing and delivering font subsets.

[004] In one aspect, a computer-implemented method includes analyzing content of a received electronic document in an autonomous manner to identify each unique character of one or more fonts included in the content of the electronic document. The method also includes initiating a request for a subset of one or more of the fonts included in the content of the electronic document, wherein the request includes each identified unique character of the respective one or more fonts.

[005] Implementations may include any or all of the following features. Analyzing the content of the electronic document may include identifying a source of a font or the absence of a source included of a font in the one or more fonts. Analyzing content of a received electronic document in an autonomous manner may be initiated by executing an agent. Analyzing content of the received electronic document may include parsing the content of the electronic document to identify the characters represented in the document. Analyzing content of the received electronic document may include filtering the identified characters represented in the electronic document to identify each unique character for each of the one or more fonts included in the content of the electronic document. Initiating a request for a subset for one or more of the fonts may include adding the identified characters to a request. A response to the request may include all characters of one or more of the fonts. Analyzing the content of the electronic document may be performed independent of platform.

[006] In another aspect a system includes a font server for providing to a computing device, upon request, an agent for analyzing content of an electronic document received by the computing device. The analysis performed by the agent identifies each unique character of one or more fonts included in the content of the electronic document. The font server is configured to produce, upon request from the computing device, a subset for one or more of the fonts in which each font subset includes each identified unique character of the respective font. The font server is further configured to initiate sending each font subset to the computing device.

[007] Implementations may include any or all of the following features. Analyzing the content of the received electronic document, the agent may identify a source of a font or the absence of a source of a font included in the one or more fonts. The computing device may execute the agent received from the font server to analyze the content of the received electronic document. The agent may parse the content of the electronic document to identify the characters represented in the document. The agent may filter the identified characters represented in the document to identify each unique character for each of the one or more fonts included in the content of the electronic document. The agent may initiate production of the request for the font server. The agent may be configured to operate independent of the computing device type. The font server may be configured to determine the portion of font characters to include in the font subset, and the portion of the font characters to include in the font subset may include all characters of the font. The font server may associate the produced font subset to the received electronic document.

[008] In another aspect a computing device includes a memory configured to store instructions and a processor configured to execute the instructions to perform a method. The method includes analyzing content of a received electronic document in an autonomous manner to identify each unique character of one or more fonts included in the content of the electronic document. The method also includes initiating a request for a subset of one or more of the fonts included in the content of the electronic document. The request includes each identified unique character of the respective one or more fonts.

[009] Implementations may include any or all of the following features. Analyzing the content of the electronic document may include identifying a source of a font or the absence of a source of a font included in the one or more fonts. Analyzing content of a received electronic document in an autonomous manner may be initiated by executing an agent. Analyzing content of the received electronic document may include parsing the content of the electronic document to identify the characters represented in the document. Analyzing content of the received electronic document may include filtering the identified characters represented in the electronic document to identify each unique character for each of the one or more fonts included in the content of the electronic document. Initiating a request for a subset of one or more of the fonts includes adding the identified characters to a request.

[010] In another aspect, one or more computer readable media storing instructions that are executable by a processing device, and upon such execution cause the processing device to perform operations that include analyzing content of a received electronic document in an autonomous manner to identify each unique character of one or more fonts included in the content of the electronic document. The operations also include initiating a request for a subset of one or more of the fonts included in the content of the electronic document. The request includes each identified unique character of the respective one or more fonts.

[011] Implementations may include any or all of the following features. Analyzing the content of the electronic document may include identifying a source of a font or the absence of a source of a font included in the one or more fonts. Analyzing content of a received electronic document in an autonomous manner may be initiated by executing an agent.

[012] These and other aspects and features and various combinations of them may be expressed as methods, apparatus, systems, means for performing functions, program products, and in other ways.

[013] Other features and advantages will be apparent from the description and the claims.

DESCRIPTION OF DRAWINGS

FIG. 1 illustrates exemplary content that may be transferred through a computer network.

FIG. 2 is a block diagram an Internet based computer network.

FIG. 3 illustrates requesting and delivering a software agent and a font subset from a front provider.

FIG. 4 is an exemplary file for producing a webpage.

FIGs. 5, 6 and 7 illustrate examples of code instructions.

FIG. 8 is an example flow chart of operations of a software agent.

FIG. 9 is a block diagram of computing devices and systems.

DETAILED DESCRIPTION

[014] Referring to FIG. 1, an exemplary webpage 100 is illustrated that may be accessed through the Internet (or other type of computer network) by a user of a computing device (e.g., computer system, personal digital assistant (PDA), cellular telephone etc.). For example, by providing an appropriate uniform resource locator (URL) to a web browser, the webpage may be accessed, packaged and sent from a corresponding content provider to the user's computing device for display. In the illustrated example, the content of the web page 100 is a birthday party invitation that includes text and graphics and may be accessed from a particular URL (i.e., www.invite.com). To provide an eye-catching invitation, multiple fonts are used that provide different typefaces, each of which can be considered as providing stylistic characters or glyphs. A "character" can be considered as a member of a set of shapes used for the organization, control and representation of information, and, a "glyph" can be considered as a specific instance of a character.

[015] Once accessed, a content provider may use one or more techniques to provide the content of the webpage 100 to the user's computing device. For example, the webpage 100 may be represented in one or more files (e.g., file 102) that use a markup language such as hypertext markup language (HTML) so that the webpage 100 conforms to the browser executed by the computing device. Standards such as the World Wide Web Consortium (W3C) standards for

HTML, CSS, XML and other standards may be implemented by the browser so the webpage 100 can be properly rendered on various types of platforms capable of displaying electronic content (e.g. personal computers, wireless telephones, personal digital assistants, handheld computers, set-top boxes and Internet appliances, etc.).

[016] Along with providing graphics (e.g., a graphic of a birthday cake in this instance), the one or more HTML files may reference one or more fonts used to render text on the webpage. For illustration, five different fonts are used in the webpage 100 to provide information associated with the invitation (e.g., time, date, location, event description, and menu). As such, the HTML file 102 calls out each font such that the recipient computing device is aware which font corresponds to each portion of textual information (e.g., present the date in a Times New Roman font). Upon receiving and executing the instructions included in the HTML file 102, the recipient computer device may locally retrieve the fonts needed to present the text. However, a considerable amount of computing devices are unable to locally store all the characters of each font that could be needed to render the webpage. Limited resources (e.g., memory) and the constant creation of new font types can limit each type of computing device from being ready to display text in any and all fonts. Fonts associated with different languages amplifies the issue that many computing devices are unable to provide all font types. Languages such as Chinese, Japanese, Korean, etc. use alphabets of characters that may number in the tens of thousands (e.g., over 10,000 characters) and call for 1 MB to 20 MB of memory to store the characters of a single language. Such memory needs are impractical, specially for computing devices with less robust onboard memory (e.g., cellular telephones). Further, storing complete character sets for even a few fonts may be inefficient if only a few characters are need from a set (to present a particular webpage). Referring to the illustrated example, a relatively small number of Times New Roman font characters are needed to present the event date (e.g., “A”, “p”, “r”, “i”, “l”, “2”, “0”, “t” and “h”) and storing the complete font character set may be considered an unwise use of device memory, certainly if a complete character set is stored for each possible user language (e.g., English, Chinese, Japanese, etc.).

[017] To conserve the use of local memory, some conventional techniques provide complete font sets with the received webpage content (e.g., an HTML file) or initiate the retrieval of the complete character sets of needed fonts upon receipt of the file associated with the webpage.

However, such techniques may still cause computing device memory to be filled with collections of font characters when only a few characters are actually being used by the webpage. For example, as illustrated in the figure, to present the webpage 100 on a computing device, complete font character sets 104, 106, 108, 110, 112 are provided along with the HTML file 102. As such, each possible character of the five fonts used by the webpage 100 is provided to the computing device, thereby consuming a considerable amount of memory of the computing device.

[018] The time needed for transferring fonts along with bandwidth consumption also provides a concern. For example, transferring complete character sets of fonts associated with languages such as Chinese, Japanese and Korean may need considerable bandwidth and transfer time.

[019] One or more techniques may be used to reduce the file transfer time, bandwidth consumption and needed memory space for preparing to present webpages such as webpage 100. For example, rather than providing complete character sets for each font, font subsets may be provided that include only that characters that appear in the webpage. As such, file transfer time and bandwidth needs are reduced and device memory is conserved while an appropriate set of font characters is provided for webpage presentation. Referring to the illustrated example, each of the font character sets 104-112 may be replaced with significantly smaller font subsets that can be provided with the HTML file 102 or relatively soon after receipt of the file. Similar to reducing transfer time and bandwidth needs based on font subsets for the stylistic appearance of characters, reductions may also be achieved for providing font subsets associated with different languages. For example, if a webpage contains only three hundred Chinese characters, it may be more efficient to provide a font subset that is restricted to only include the three hundred characters, which may reduce the size of the transferred font data from approximately 10 MB to 50 KB. In some situations while subsets may be produced for some font character sets (e.g., the Chinese language character set), other font character sets may be sent as complete sets. For example, due to the relatively small number of characters included in a complete font set (e.g., the Latin language character set), creating and transferring a subset may not significantly reduce the bandwidth or time needed to transfer the entire character set. Along with reducing the size of font character sets provided to a computing device to present a webpage or other types of electronic document, one or more techniques may be implemented to efficiently provide such font subsets to computing devices. For example, along with presenting the webpage, the user's computing device may be used for identifying the appropriate font subsets and subsequently request the identified subsets.

[020] Referring to FIG. 2, a computer network 200 includes a computer system 202 that a user may interact with (e.g., using a keyboard or pointing device such as a mouse) identify a target webpage to be presented with the computer system. For example, a web browser 204 or similar software application may be executed by the computer system 202 for the user to target one or more webpages. Upon being identified, operations of the web browser 204 may include requesting, via the Internet 206, content from one or more webpage sources 208 a,b,c for the target webpage(s). As illustrated, in this particular example a webpage page is requested from webpage source 208a and a corresponding HTML file 210 is sent from the source through the Internet 206 to the computer system 202.

[021] To identify the appropriate font subsets to present the webpage defined by the HTML file 210, the computer system 202 may use one or more techniques. For example, operations may be executed by the computer system 202 to scan the HTML file 210 to identify the individual font characters included in the webpage defined by the file. In one arrangement, the computer system 202 may execute a software agent 212 to identify the individual font characters and to send a request to initiate the production of font subsets (if appropriate) needed to present the webpage. Such agents can be considered as a software module that is executable in a substantially autonomous manner. For example, upon being provided to the computer system 202, a software agent may operate without considerable user interaction. By operating in a somewhat flexible manner, the software agent can adaptively identify font characters needed for webpage presentation. In this particular example, the software agent 212 scans the content of the HTML file 210 in a somewhat persistent manner for identifying the font characters. For example, the software agent may execute in a substantially continuous manner. In some arrangements the software agent is provided to the user computing device (e.g., computer system 202) very shortly after the delivery of the file or files (e.g., the HTML file 210). As such, electronic documents such as webpages, application pages, user interfaces, and the like may be perceived as being scanned nearly in real time as the documents are received.

[022] The computer network 200 also includes a font provider 214 that determines to produce font subsets (when appropriate) based upon information provided by software agents (e.g., software agent 212) executed by user computing devices (e.g., computer system 202). Once produced, the subset or subsets (e.g., illustrated as font subset 216) are packaged and sent by the font provider 214 to the requesting computing device. Along with determining to produce one or more font subsets, the font provider 214 is also capable of determining if complete font character sets should be provided to the requesting computing device. For example, predefined rules may be used by the font provider 214 in determining whether a font subset should be sent. One such rule may indicate that font character sets associated particular languages (e.g., Chinese) should have subsets created due to the large size of the complete character set. Subset determinations may also be provided in a dynamic manner. For example, based upon achievable file transfer rates, a file size threshold (e.g., 2 MB) may be determined such that subsets are produced for character sets larger than the threshold. If a font character set size falls below the threshold, the entire character set may be sent as the file transfer rate may be considered within an efficient range. Such thresholds may be dynamically adjusted, for example, by monitoring the achievable transfer rates, the threshold may be changed. For example, as the level of achievable transfer rates decreases, the threshold for creating subsets for character sets may correspondingly be decreased (e.g., lowered from 2 MB to 1 MB). One or more factors may account for determining the transfer rates as being within an efficient range. For example, geographic location of the user computing device and the font provider may factor into whether a subset should be produced and sent in place of a complete character set. If both the font provider and the user computing device are located relatively nearby (e.g., both in the eastern United States), relatively high transfer rates may be achievable and the entire character set may be sent. For a situation in which the user computing device is remote from the font provider (e.g., one in the United States and the other in India), the font provider may determine to subset the font character set to be sent. Similar to location based determinations, time of day, season of year and other temporal factors may be used by the font provider to determine if subsets need to be produced for one or more font character sets identified for transmission.

[023] In some arrangements, the font provider 214 may also provide the software agents to the computing devices for scanning received files (e.g., HTML file 210) for character identification. As such, the font provider 214 may operate independent of the webpage sources 208 a,b,c. Once a request is received from a user computing device, the font provider 214 may provide appropriate agent software to the requesting device. Once the electronic document is scanned by the software agent, appropriate fonts and font subsets may be provided by the font provided 214 based upon a request initiated by the executed agent.

[024] To provide the fonts and font subsets, the font provider 214 typically needs access to one or more libraries of fonts that may be stored local or remote to the font provider. As represented in the figure, a library of fonts 218 is shown being stored in a storage device 220 (e.g., one or more hard drives, CD-ROMs, etc.) on site. Being accessible by a server 222, the font library 218 may be used along with information provided from software agents to produce appropriate font subsets that can be provided along with complete font character sets (if determined appropriate by the font provider). Illustrated as being stored in a single storage device 220, the font provider 214 may use numerous storage techniques and devices to retain a collection of accessible sets of fonts (e.g., for different font styles, languages, etc.). The font provider 214 may also access fonts at separate locations for subset production. For example, upon identifying characters needed for a font subset, the server 222 may be used to collect needed characters from one or more sources external to the font provider 214 (e.g., via the Internet 206).

[025] Along with producing font subsets and providing them (along with complete font sets, if appropriate) to requesting computing devices, the font provider 214 may provide other functionality. For example, fonts and font subsets associated with particular webpages may be tracked for future requests. In one scenario, one or more font subsets (e.g., font subset 216) may be created for presenting a particular webpage (on a computing device). The associated between the font subsets and the webpage may be identified (e.g., by the server 222) and stored for later retrieval. As such, the subsets needed to present the webpage in the future (e.g., on another computing device) can be quickly identified and provided to the requesting computing device. In one arrangement, a font database 224 is stored at the font provider 214 (e.g., on the storage device 220) and includes records that represent the association between webpages and font subsets (and fonts). In some instances, the association is identified from information provided by font subset requests sent to the font provider 214 (e.g., from a software agent). The association between a webpage and appropriate font subsets may also be stored prior to the webpage being selected by a user (e.g., based on directions and information provided by a webpage source).

Other types of architectures and networking techniques may also be implemented for providing software agents and font subsets (and fonts) to user computing devices for presentation of electronic documents such as webpages.

[026] Referring to FIG. 3, a diagram 300 represents some operations for providing a software agent to a user's computing device along with appropriate font subsets (and fonts) needed to present a webpage of other type of electronic document. As illustrated in FIG. 2, upon a user identifying a webpage of interest with a web browser executed on a computing device (e.g., providing a URL to the web browser), one or more files (e.g., HTML files) may be provided to the user's computing device from a corresponding webpage source. As illustrated, such a file 302 may include content 304 (e.g., text, graphics, video, audio, etc.) for presenting to the user (via the web browser). The file 302 may also contain one or more instructions 306 for requesting that a software agent be provided to the user's computing device 202. Upon executing the instructions (labeled in the figure "Fetch agent instruction"), delivery of a request 308 may be initiated from the user computing device (e.g., the computer system 202) to the font provider 214 (e.g., server 222), as represented by graphical arrow 310. In response to the request 308, an agent (e.g., the agent 212) is sent from the server 222 of the font provider 214 to the user computing device (e.g., computer system 202), as represented by graphical arrow 312. In some instances the delivery of the agent may occur very soon after the file is received and any delay may go unnoticed by a user. In some arrangements, other information may be provided by the request 308. For example, the webpage of interest may be identified in the request (e.g., URL of the webpage provided) so that the font provider 214 may determine if a one or more font subsets (and possibly fonts) have been previously produced for the webpage.

[027] Received at the user computing device, the requested agent 212 is executed to scan the content 304 of the HTML file 302 (as represented by a graphical arrow 314) to identify characters for each font represented in the content. The agent 212 may also provide the functionality to identify each unique character of each font present. As such, multiple instances of the same font character may only be noted once by the agent, thereby consolidating the characters needed to be requested from the font provider 214 (for each received electronic document). In some arrangements, the agent 212 notifies the font provider 214 of each character identified for each font present in the webpage. Upon being provided this information, the font provider identifies each uniquely occurring character for each font for possible inclusion in a font subset. To provide such scanning operations, one or more techniques may be implemented, for example, the agent may parse the content 304 to identify each character present for each font. One or more filters may then be used (by the agent 212 or the font provider 214) to identify each unique character for each font. For example, if the characters “a”, “B”, and “c” for font A are detected in the content 304 and characters “x”, “Y”, and “Z” for font B are detected, the agent may identify a subset for font A as containing “a”, “B” and “c” while the subset for font B may contain “x”, “Y” and “Z”. Once scanned, identified font characters 316 are used by the agent 212 to produce a font subset request 318. In general, the request 318 includes each character identified by the agent 212, however, some characters included in the content of the page content 304 may not be included in the request 318. For example, characters identified as possibly being locally stored at the user computing device may not be included in the request 318. As such, the agent 212 may exclude some characters included in the page content 304 from the subset request 318.

[028] One or more techniques may be implemented to provide the font subset request 318 to the server 222 of the font provider 214, as represented by graphical arrow 320. For example, for an agent represented in JavaScript, a technique associated with a protocol such as the hypertext transfer protocol (HTTP) maybe used for transferring the request. By appending the identified unique characters to a query string of the URL of interest, a command (e.g., a GET command) can be used to provide the information to the server 222. Similarly, an agent that is provided as an application may provide the character information to the server 222 of the font provider 214 with a protocol such as HTTP. Once provided the request 318 for the font subset(s), the server 222 produces the one or more needed font subsets and complete font character sets (if appropriate) and replies to the user computing device. For example, as represented with graphical arrow 322, the font subset 216 (that may represent one or more font subsets and possibly one or more complete font character sets) is provided to the user computing device.

[029] Referring to FIG. 4, instructions of an exemplary HTML file 400 are illustrated that include requesting an agent (such as the agent 212 shown in FIG. 2) and assigning fonts to particular characters. In this particular example, upon instruction 402 being executed (e.g., by the computer system 202), an agent is requested from a font provider (e.g., the font provider

214). Once received by the computer system 202, the agent is executed to analyze the contents of the HTML file 400. For example, the agent may step through each of the remaining lines of the HTML file 400 and identify each character and font used to present the webpage associated with the contents of the file. For example, by analyzing instruction 404, the agent may identify that characters “A”, “B”, “C” and “D” in frutiger font are needed for webpage production. In this particular arrangement, the individual characters (e.g., “A B C D”) are provided by the instruction 404 along with a URL for accessing the font. Similarly, the executed agent also identifies, in instruction 406, that characters “Z” and “W” need to be represented in frutiger font to produce the webpage. As such, when producing a font subset request (e.g., request 318 shown in FIG. 3), the agent identifies each of the unique characters (i.e., “A”, “B”, “C”, “D”, “Z” and “W”) and the corresponding font (i.e., frutiger) needed to produce the webpage. In some arrangements, while scanning the content of the file, the agent may come across characters that are not to be included in a font subset request. For example, the HTML file may include an instance in which locally stored fonts on the computer system (executing the file) are to be used for representing particular characters. As such, fonts do not need to be attained from sources external to the computer system. Instruction 408 of the exemplary HTML file 400 illustrates such an occurrence. In this instance, the characters “M”, “P” and “Q” are called out by the instruction 408 without a URL for a particular font. As such, fonts local to the computer system executing the file 400 may be used to present the characters “M”, “P” and “Q”. Since a font or font subset is not needed for these particular characters, the agent does not include these characters in the request 318. However, while these characters are stored locally for this particular font, one or more of these characters may be included in the request for another font based upon another instruction (not shown) in the file 400 that calls out “M”, “P” and/or “Q” as being needed in a font attained from a source external to the computer system.

[030] Referring to FIG. 5, one or more techniques may be implemented to analyze the content of a file such as the HTML file 400 (illustrated in FIG. 4) to identify characters for font subsets. For a JavaScript based agent, a browser independent library (referred to as jQuery and which emphasizes interactions between JavaScript and HTML), may be used to analyze text content. To provide this functionality, a jQuery framework may be used to provide an agent for extracting unique characters from a string. The jQuery framework may also include an associative array (referred to as a JSON) for forming associations between the identified unique characters and a corresponding font. Stepping through the file in an iterative manner, the unique characters are identified and stored (e.g., cached) for further processing. The portion of code 500 presented in the figure can provide this functionality.

[031] Referring to FIG. 6, upon identifying the unique characters, one or more techniques may be implemented to group the identified characters accordingly based upon font. For example, each unique character (e.g., “A”, “B”, “C”, “D”, “Z” and “W”) identified for a particular font (e.g., frutiger) is a member of a group for that font. Additionally, for fonts that have relatively few members (e.g., a font associated with the Latin language), a group of unique characters may not be formed. For such fonts that include relatively fewer members, the entire font set may be sent without consuming considerable computational resources such as transfer time and bandwidth. As such, the complete font set be provided (e.g., from the font provider) for producing characters of that font. The portion of code 600 presented in the figure can provide this functionality.

[032] Referring to FIG. 7, once the identified unique characters for each font have been grouped (along with any identified fonts that have relatively small character sets), the agent provides this information to the font provider 214 (e.g., to the server 222 of the font provider). One or more techniques may be used to provide this information. For example, a command such as the HTTP GET command may be used to append identified characters and corresponding fonts to a URL query string. Upon receiving a request (provided by the HTTP GET command), the font provider 214 (e.g., the server 222 of the font provider) creates and sends one or more appropriate font subsets to the requesting computer system. Complete font sets may also be sent for identified fonts that include relatively few characters, based upon predefined rules associated with one or more factors (e.g., geographic location of user and/or font provider, temporal information, data transfer parameters such as achievable transfer rate, etc.). To provide the identified unique characters and corresponding fonts, a portion of code 700 is presented in the figure that can provide this functionality.

[033] Referring to FIG. 8, a flowchart 800 represents operations of a software agent (e.g., the software agent 212 shown in FIG. 2). Operations of the agent 212 are typically executed by a single computing device (e.g., computer system 202), however, operations of the agent may be executed by multiple computing devices. Along with being executed at a single site (e.g., at one computer system), operation execution may be distributed among two or more locations.

[034] Typically, the agent is requested by the user computing device (e.g., from code included in a file such as an HTML file) and provided from an external source such as the font provider 214. Upon being received and executed by the user computing device, operations may include analyzing 802 the content of an electronic document such as the HTML file that included the instructions to request the software agent. By using a software agent, the content of the electronic document is analyzed in a relatively autonomous and persistent manner. One or more techniques may be implemented to analyze the electronic documents, for example, the agent may step through the document and associate each included character with a corresponding font. Operations may also include identifying 804 each unique character that is associated with a font. For example, while a document may include many instances of the character “a” for a particular font (e.g., Times New Roman), the agent may filter the many instances down to a single instance such that the character “a” is only requested once for the Times New Roman font. Character identification may also include identifying fonts that have relatively small character sets. As such, entire character sets may be requested if one or more of the characters (included in these fonts) are used in the electronic document. Operations may also include requesting 806 appropriate subsets for each font represented in the electronic document. The requested subsets typically include the identified unique characters for each corresponding font. However, for instances in which the font includes relatively few characters, the requested subset may include the entire character set for such a font.

[035] FIG. 9 is a block diagram of computing systems 900 that may be used and implemented to perform operations associated with the agent 212. The computing systems may also be used by the font provider 214 to execute operations. Computing device 900 is intended to represent various forms of digital computers, such as laptops, desktops, workstations, personal digital assistants, servers, blade servers, mainframes, and other appropriate computers.

[036] Computing device 900 includes a processor 902, memory 904, a storage device 906, a high-speed interface 908 connecting to memory 904 and high-speed expansion ports 910, and a low speed interface 912 connecting to low speed bus 914 and storage device 906. Each of the components 902, 904, 906, 908, 910, and 912, are interconnected using various busses, and can be mounted on a common motherboard or in other manners as appropriate. The processor 902 can process instructions for execution within the computing device 900, including instructions stored in the memory 904 or on the storage device 906 to display graphical information for a GUI on an external input/output device, such as display 916 coupled to high speed interface 908. In other implementations, multiple processors and/or multiple buses can be used, as appropriate, along with multiple memories and types of memory. Also, multiple computing devices 900 can be connected, with each device providing portions of the necessary operations (e.g., as a server bank, a group of blade servers, or a multi-processor system).

[037] The memory 904 stores information within the computing device 900. In one implementation, the memory 904 is a computer-readable medium. In one implementation, the memory 904 is a volatile memory unit or units. In another implementation, the memory 904 is a non-volatile memory unit or units.

[038] The storage device 906 is capable of providing mass storage for the computing device 900. In one implementation, the storage device 906 is a computer-readable medium. In various different implementations, the storage device 906 can be a floppy disk device, a hard disk device, an optical disk device, or a tape device, a flash memory or other similar solid state memory device, or an array of devices, including devices in a storage area network or other configurations. In one implementation, a computer program product is tangibly embodied in an information carrier. The computer program product contains instructions that, when executed, perform one or more methods, such as those described above. The information carrier is a computer- or machine-readable medium, such as the memory 904, the storage device 906, memory on processor 902, or the like.

[039] The high speed controller 908 manages bandwidth-intensive operations for the computing device 900, while the low speed controller 912 manages lower bandwidth-intensive operations. Such allocation of duties is exemplary only. In one implementation, the high-speed controller 908 is coupled to memory 907, display 916 (e.g., through a graphics processor or accelerator), and to high-speed expansion ports 910, which can accept various expansion cards (not shown). In the implementation, low-speed controller 912 is coupled to storage device 906 and low-speed expansion port 914. The low-speed expansion port, which can include various communication ports (e.g., USB, Bluetooth, Ethernet, wireless Ethernet) can be coupled to one or more input/output devices, such as a keyboard, a pointing device, a scanner, or a networking device such as a switch or router, e.g., through a network adapter.

[040] The computing device 900 can be implemented in a number of different forms, as shown in the figure. For example, it can be implemented as a standard server 920, or multiple times in a group of such servers. It can also be implemented as part of a rack server system 924. In addition, it can be implemented in a personal computer such as a laptop computer 922. Alternatively, components from computing device 900 can be combined with other components in a mobile device (not shown).

[041] Embodiments of the subject matter and the functional operations described in this specification can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Embodiments of the subject matter described in this specification can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a computer-readable medium for execution by, or to control the operation of, data processing apparatus. The computer-readable medium can be a machine-readable storage device, a machine-readable storage substrate, a memory device, a composition of matter effecting a machine-readable propagated signal, or a combination of one or more of them. The term “data processing apparatus” encompasses all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them.

[042] A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub-programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[043] The processes and logic flows described in this specification can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit).

[044] Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read-only memory or a random access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto-optical disks, or optical disks. However, a computer need not have such devices. Moreover, a computer can be embedded in another device, e.g., a mobile telephone, a personal digital assistant (PDA), a mobile audio player, a Global Positioning System (GPS) receiver, to name just a few. Computer-readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

[045] Embodiments of the subject matter described in this specification can be implemented in a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the subject matter described is this specification, or any combination of one or more such back-end, middleware, or front-end components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network ("LAN") and a wide area network ("WAN"), e.g., the Internet.

[046] The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

[047] While this specification contains many specifics, these should not be construed as limitations on the scope of the invention or of what may be claimed, but rather as descriptions of features specific to particular embodiments of the invention. Certain features that are described in this specification in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

[048] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous. Moreover, the separation of various system components in the embodiments described above should not be understood as requiring such separation in all embodiments, and it should be understood that the described program components and systems can generally be integrated together in a single software product or packaged into multiple software products.

[049] Thus, particular embodiments of the invention have been described. Other embodiments are within the scope of the following claims. For example, the actions recited in the claims can be performed in a different order and still achieve desirable results.

WHAT IS CLAIMED IS:

1. A computer-implemented method comprising:

analyzing content of a received electronic document in an autonomous manner to identify each unique character of one or more fonts included in the content of the electronic document; and

initiating a request for a subset of one or more of the fonts included in the content of the electronic document, wherein the request includes each identified unique character of the respective one or more fonts.

2. The computer-implemented method of claim 1 in which analyzing the content of the electronic document includes identifying a source of a font included in the one or more fonts.

3. The computer-implemented method of claim 1 in which analyzing the content of the electronic document includes identifying the absence of a source of a font included in the one or more fonts.

4. The computer-implemented method of claim 1 in which analyzing content of a received electronic document in an autonomous manner is initiated by executing an agent.

5. The computer-implemented method of claim 1 in which analyzing content of the received electronic document includes parsing the content of the electronic document to identify the characters represented in the document.

6. The computer-implemented method of claim 5 in which analyzing content of the received electronic document includes filtering the identified characters represented in the electronic document to identify each unique character for each of the one or more fonts included in the content of the electronic document.

7. The computer-implemented method of claim 1 in which initiating a request for a subset of one or more of the fonts includes adding the identified characters to a request.

8. The computer-implemented method of claim 1 in which a response to the request includes all characters of one of the fonts.

9. The computer-implemented method of claim 1 in which analyzing the content of the electronic document is performed independent of platform.

10. A system comprising:

a font server for providing to a computing device, upon request, an agent for analyzing content of an electronic document received by the computing device, wherein the analysis performed by the agent identifies each unique character of one or more fonts included in the content of the electronic document;

wherein the font server is configured to produce, upon request from the computing device, a subset for one or more of the fonts in which each font subset includes each identified unique character of the respective font; and

wherein the font server is further configured to initiate sending each font subset to the computing device.

11. The system of claim 10 in which by analyzing the content of the received electronic document, the agent identifies a source of a font included in the one or more fonts.

12. The system of claim 10 in which by analyzing the content of the received electronic document, the agent identifies the absence of a source of a font included in the one or more fonts.

13. The system of claim 10 in which the computing device executes the agent received from the font server to analyze the content of the received electronic document.

14. The system of claim 10 in which the agent parses the content of the electronic document to identify the characters represented in the document.

15. The system of claim 14 in which the agent filters the identified characters represented in the document to identify each unique character for each of the one or more fonts included in the content of the electronic document.

16. The system of claim 10 in which the agent initiates production of the request for the font server.

17. The system of claim 10 in which the agent is configured to operate independent of the computing device type.

18. The system of claim 10 in which the font server is configured to determine the portion of font characters to include in the font subset.

19. The system of claim 18, in which the portion of the font characters to include in the font subset includes all characters of the font.

20. The system of claim 20, in which the font server associates the produced font subset to the received electronic document.

21. A computing device comprising:

a memory configured to store instructions; and

a processor configured to execute the instructions to perform a method comprising:

analyzing content of a received electronic document in an autonomous manner to identify each unique character of one or more fonts included in the content of the electronic document; and

initiating a request for a subset of one or more of the fonts included in the content of the electronic document, wherein the request includes each identified unique character of the respective one or more fonts.

22. The computing device of claim 21, in which analyzing the content of the electronic document includes identifying a source of a font included in the one or more fonts.

23. The computing device of claim 21, in which analyzing the content of the electronic document includes identifying the absence of a source of a font included in the one or more fonts.

24. The computing device of claim 21, in which analyzing content of a received electronic document in an autonomous manner is initiated by executing an agent.

25. The computing device of claim 21, in which analyzing content of the received electronic document includes parsing the content of the electronic document to identify the characters represented in the document.

26. The computing device of claim 25, in which analyzing content of the received electronic document includes filtering the identified characters represented in the electronic document to identify each unique character for each of the one or more fonts included in the content of the electronic document.

27. The computing device of claim 21, in which initiating a request for a subset of one or more of the fonts includes adding the identified characters to a request.

28. One or more computer readable media storing instructions that are executable by a processing device, and upon such execution cause the processing device to perform operations comprising:

analyzing content of a received electronic document in an autonomous manner to identify each unique character of one or more fonts included in the content of the electronic document; and

initiating a request for a subset of one or more of the fonts included in the content of the electronic document, wherein the request includes each identified unique character of the respective one or more fonts.

29. The computer readable media of claim 28, in which analyzing the content of the electronic document includes identifying a source of a font included in the one or more fonts.

30. The computer readable media of claim 28, in which analyzing the content of the electronic document includes identifying the absence of a source of a font included in the one or more fonts.

31. The computer readable media of claim 28, in which analyzing content of a received electronic document in an autonomous manner is initiated by executing an agent.

AMENDED CLAIMS

[received by the International Bureau on 13 September 2011 (13.09.2011)]

1. A computer-implemented method comprising:

analyzing content of a received electronic document in an autonomous manner to identify each unique character of one or more fonts included in the content of the electronic document; and

initiating a request for a subset of one or more of the fonts included in the content of the electronic document, wherein the request includes each identified unique character of the respective one or more fonts.

2. The computer-implemented method of claim 1 in which analyzing the content of the electronic document includes identifying a source of a font included in the one or more fonts.

3. The computer-implemented method of claim 1 in which analyzing the content of the electronic document includes identifying the absence of a source of a font included in the one or more fonts.

4. The computer-implemented method of claim 1 in which analyzing content of a received electronic document in an autonomous manner is initiated by executing an agent.

5. The computer-implemented method of claim 1 in which analyzing content of the received electronic document includes parsing the content of the electronic document to identify the characters represented in the document.

6. The computer-implemented method of claim 5 in which analyzing content of the received electronic document includes filtering the identified characters represented in the electronic document to identify each unique character for each of the one or more fonts included in the content of the electronic document.

7. The computer-implemented method of claim 1 in which initiating a request for a subset of one or more of the fonts includes adding the identified characters to a request.

8. The computer-implemented method of claim 1 in which a response to the request includes all characters of one of the fonts.

9. The computer-implemented method of claim 1 in which analyzing the content of the electronic document is performed independent of platform.

10. A system comprising:

a font server for providing to a computing device, upon request, an agent for analyzing content of an electronic document received by the computing device, wherein the analysis performed by the agent identifies each unique character of one or more fonts included in the content of the electronic document;

wherein the font server is configured to produce, upon request from the computing device, a subset for one or more of the fonts in which each font subset includes each identified unique character of the respective font; and

wherein the font server is further configured to initiate sending each font subset to the computing device.

11. The system of claim 10 in which by analyzing the content of the received electronic document, the agent identifies a source of a font included in the one or more fonts.

12. The system of claim 10 in which by analyzing the content of the received electronic document, the agent identifies the absence of a source of a font included in the one or more fonts.

13. The system of claim 10 in which the computing device executes the agent received from the font server to analyze the content of the received electronic document.

14. The system of claim 10 in which the agent parses the content of the electronic document to identify the characters represented in the document.

15. The system of claim 14 in which the agent filters the identified characters represented in the document to identify each unique character for each of the one or more fonts included in the content of the electronic document.

16. The system of claim 10 in which the agent initiates production of the request for the font server.

17. The system of claim 10 in which the agent is configured to operate independent of the computing device type.

18. The system of claim 10 in which the font server is configured to determine the portion of font characters to include in the font subset.

19. The system of claim 18, in which the portion of the font characters to include in the font subset includes all characters of the font.

20. The system of claim 10, in which the font server associates the produced font subset to the received electronic document.

21. A computing device comprising:

a memory configured to store instructions; and

a processor configured to execute the instructions to perform a method comprising:

analyzing content of a received electronic document in an autonomous manner to identify each unique character of one or more fonts included in the content of the electronic document; and

initiating a request for a subset of one or more of the fonts included in the content of the electronic document, wherein the request includes each identified unique character of the respective one or more fonts.

22. The computing device of claim 21, in which analyzing the content of the electronic document includes identifying a source of a font included in the one or more fonts.

23. The computing device of claim 21, in which analyzing the content of the electronic document includes identifying the absence of a source of a font included in the one or more fonts.

24. The computing device of claim 21, in which analyzing content of a received electronic document in an autonomous manner is initiated by executing an agent.

25. The computing device of claim 21, in which analyzing content of the received electronic document includes parsing the content of the electronic document to identify the characters represented in the document.

26. The computing device of claim 25, in which analyzing content of the received electronic document includes filtering the identified characters represented in the electronic document to identify each unique character for each of the one or more fonts included in the content of the electronic document.

27. The computing device of claim 21, in which initiating a request for a subset of one or more of the fonts includes adding the identified characters to a request.

28. One or more computer readable media storing instructions that are executable by a processing device, and upon such execution cause the processing device to perform operations comprising:

analyzing content of a received electronic document in an autonomous manner to identify each unique character of one or more fonts included in the content of the electronic document; and

initiating a request for a subset of one or more of the fonts included in the content of the electronic document, wherein the request includes each identified unique character of the respective one or more fonts.

29. The computer readable media of claim 28, in which analyzing the content of the electronic document includes identifying a source of a font included in the one or more fonts.

30. The computer readable media of claim 28, in which analyzing the content of the electronic document includes identifying the absence of a source of a font included in the one or more fonts.

31. The computer readable media of claim 28, in which analyzing content of a received electronic document in an autonomous manner is initiated by executing an agent.

32. A system comprising:

a font server for producing, upon a request from a computing device, a subset of one or more fonts, wherein each font subset includes each unique character of the respective font as identified from content of an electronic document by an agent included in the computing device; and

wherein the font server is configured to initiate sending each font subset to the computing device.

33. The system of claim 32 in which the font server is configured to provide the agent to the computing device.

34. The system of claim 32 in which the agent analyzes the content of the electronic document to identify a source of a font included in the one or more fonts.

35. The system of claim 32 in which the agent analyzes the content of the received electronic document to identify the absence of a source of a font included in the one or more fonts.

36. The system of claim 32 in which the agent parses the content of the electronic document to identify the characters represented in the document.

37. The system of claim 36 in which the agent filters the identified characters represented in the electronic document to identify each unique character for each of the one or more fonts included in the content of the electronic document.

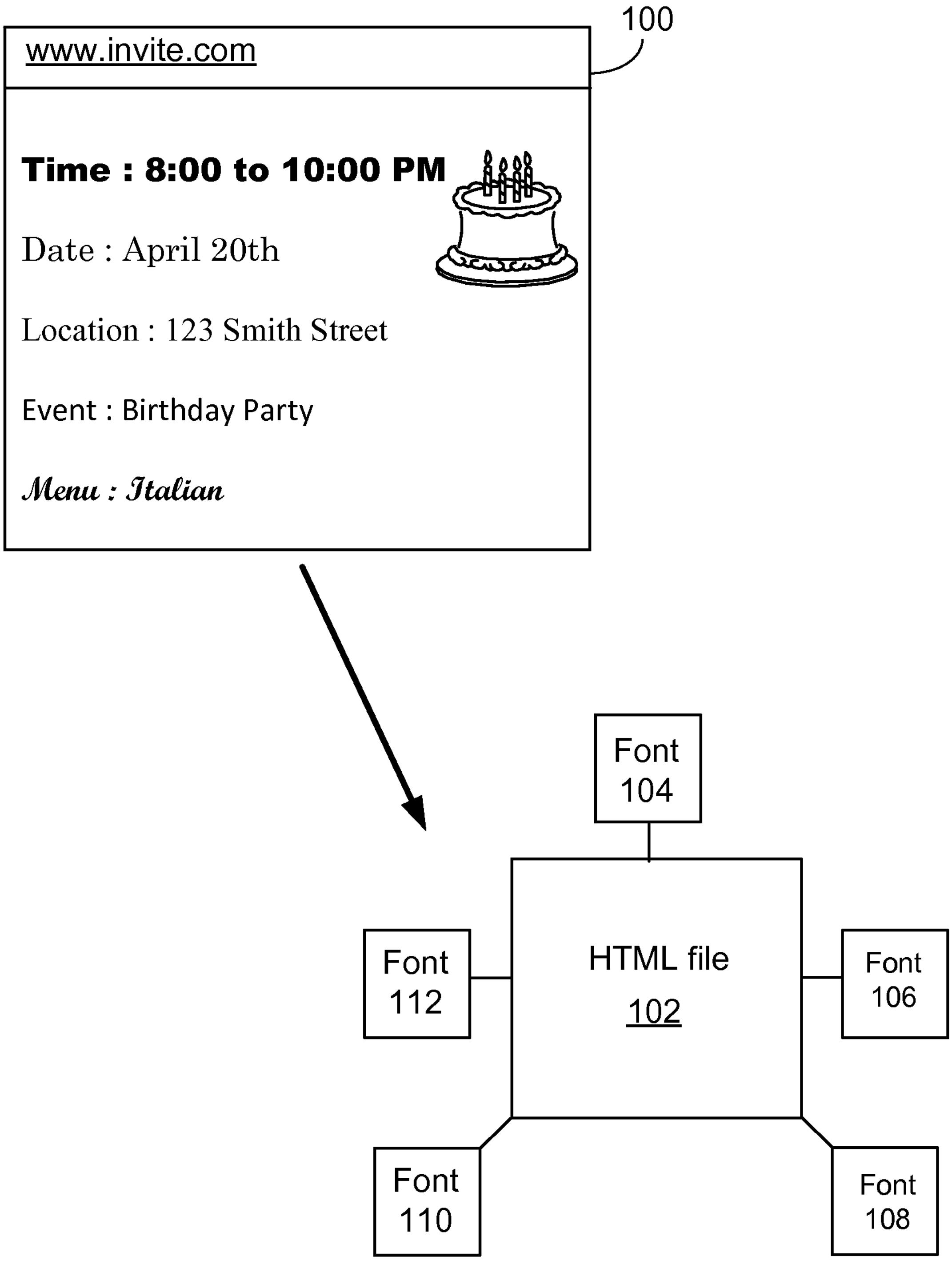
38. The system of claim 32 in which the agent initiates production of the request for the font server.

39. The system of claim 32 in which the agent is configured to operate independent of the computing device type.

40. The system of claim 32 in which the font server is configured to determine the portion of font characters to include in the font subset.

41. The system of claim 40, in which the portion of the font characters to include in the font subset includes all characters of the font.

42. The system of claim 32, in which the font server associates the produced font subset to the electronic document.



Time : 8:00 to 10:00 PM

Date : April 20th

Location : 123 Smith Street

Event : Birthday Party

Menu : Italian

FIG. 1

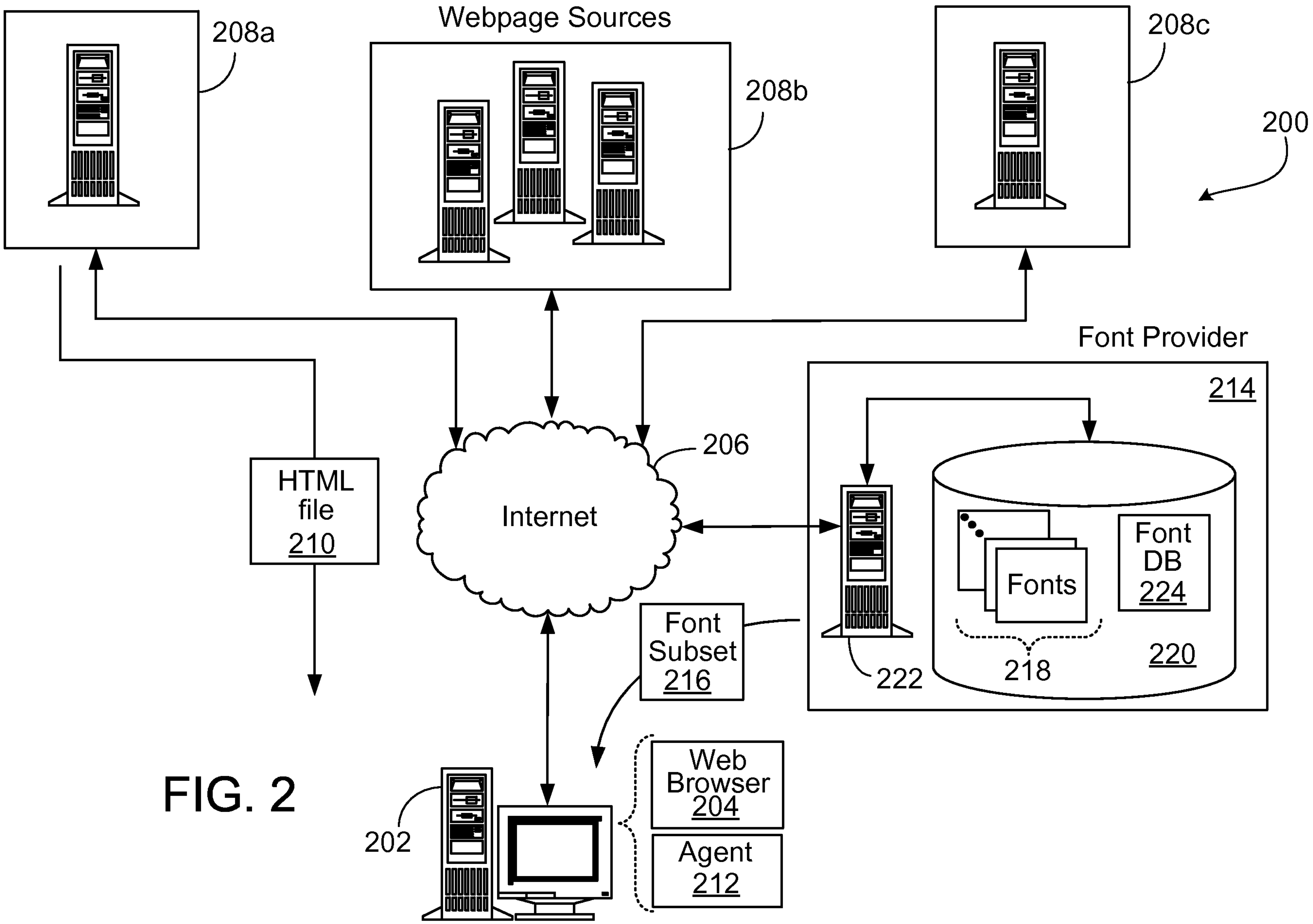

Webpage Sources
208a
208b
208c
200
Font Provider
214
206
Internet
HTML
file
210
Font
DB
224
Fonts
218
220
222
Font
Subset
216
Web
Browser
204
Agent
212
202

FIG. 2

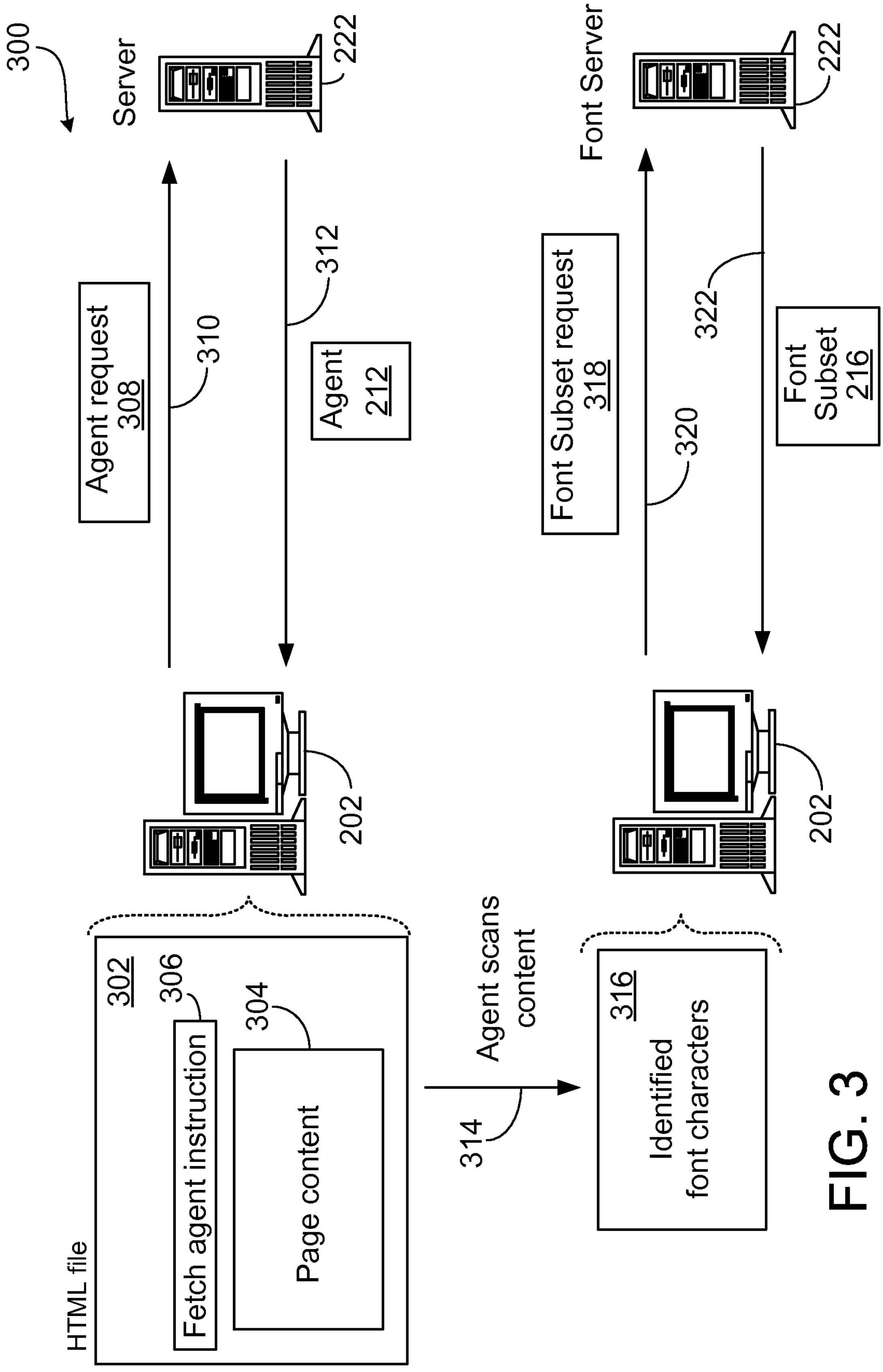
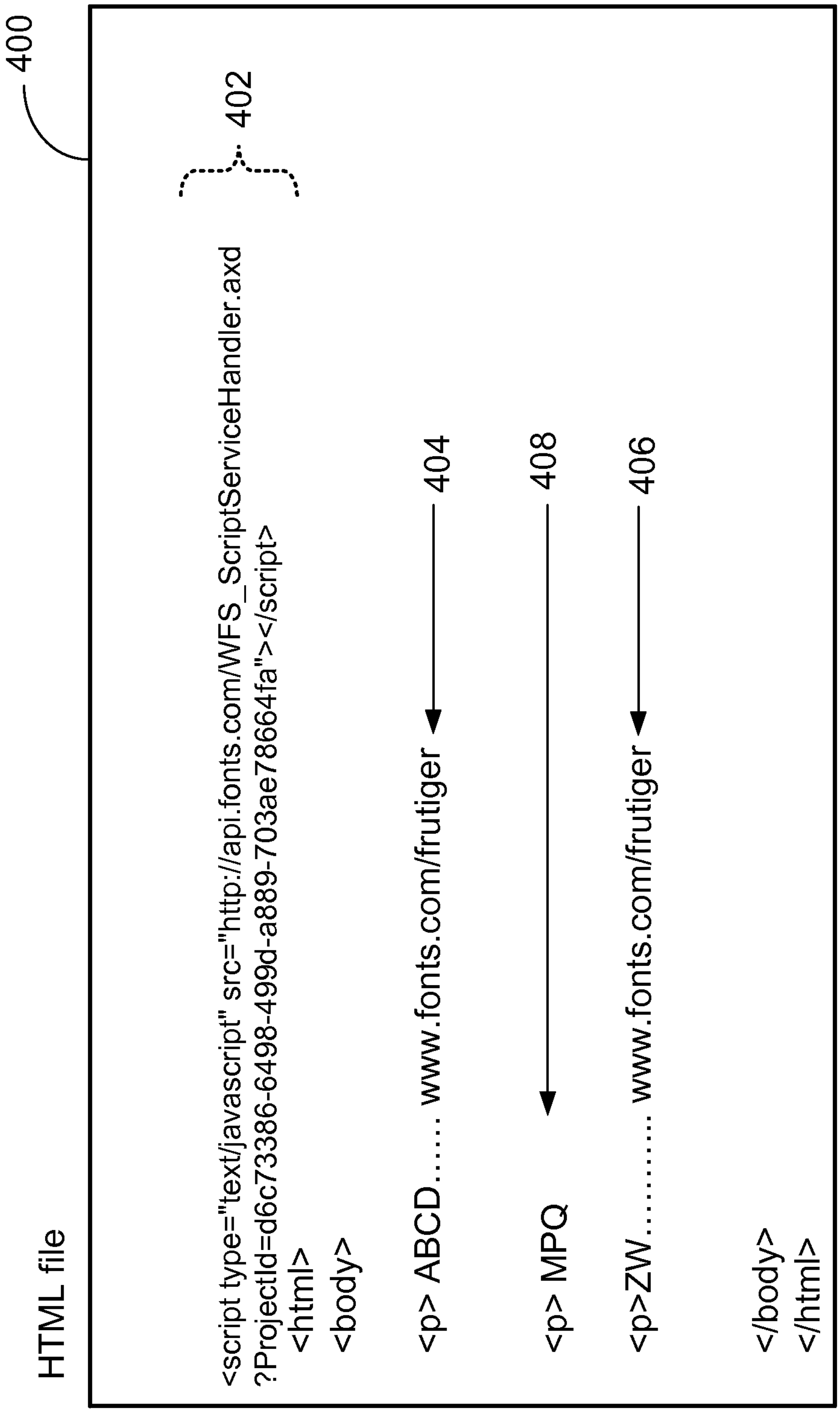
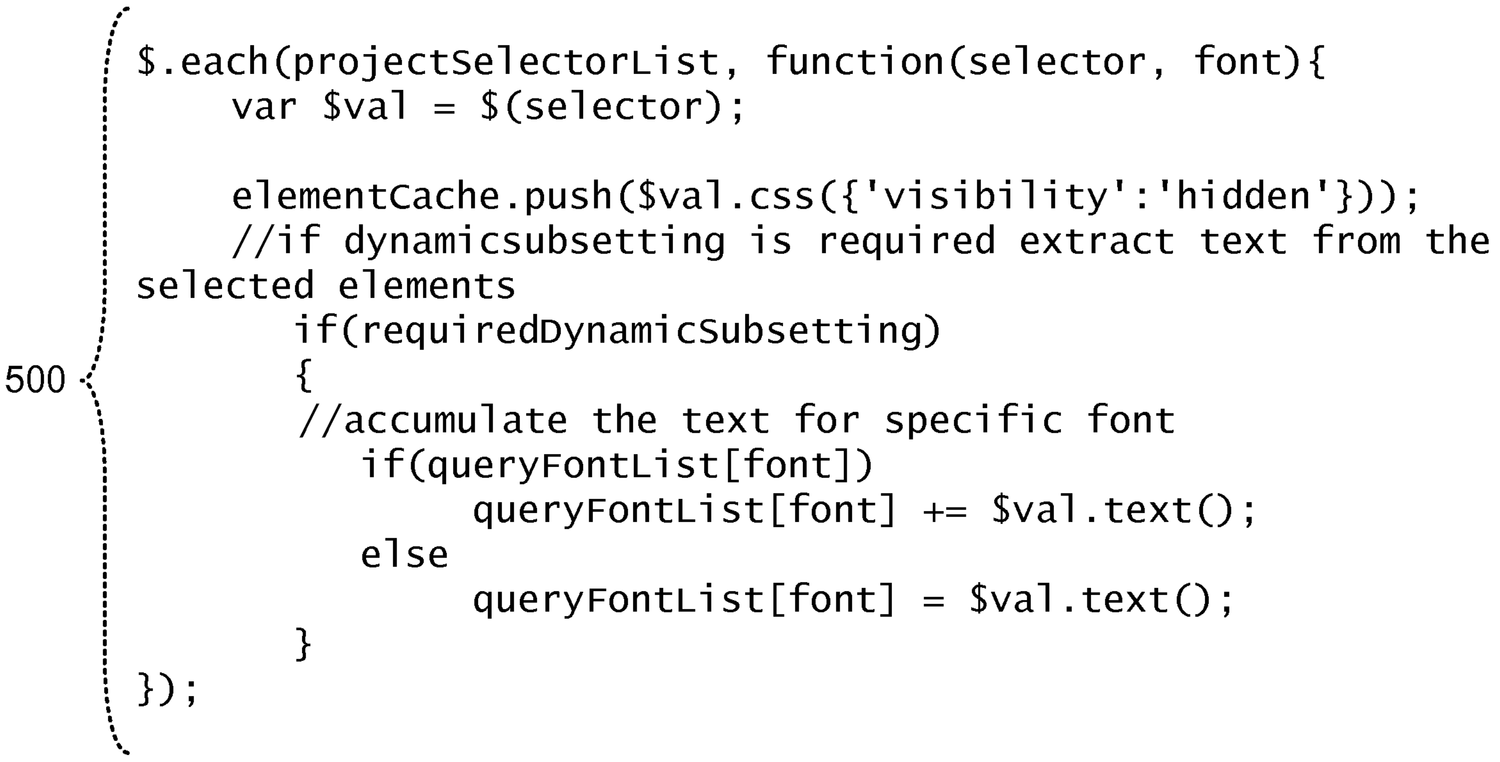

300
Server
222
Agent request
308
310
312
Agent
212
202
HTML file
302
306
Fetch agent instruction
304
Page content
Agent scans content
314
316
Identified font characters
Font Server
222
Font Subset request
318
320
322
Font Subset
216
202

FIG. 3



```
<script type="text/javascript" src="http://api.fonts.com/WFS_ScriptServiceHandler.axd
?ProjectId=d6c73386-6498-499d-a889-703ae78664fa"></script>
<html>
<body>

<p> ABCD...... www.fonts.com/frutiger

<p> MPQ

<p>ZW.......... www.fonts.com/frutiger


</body>
</html>
```

FIG. 4



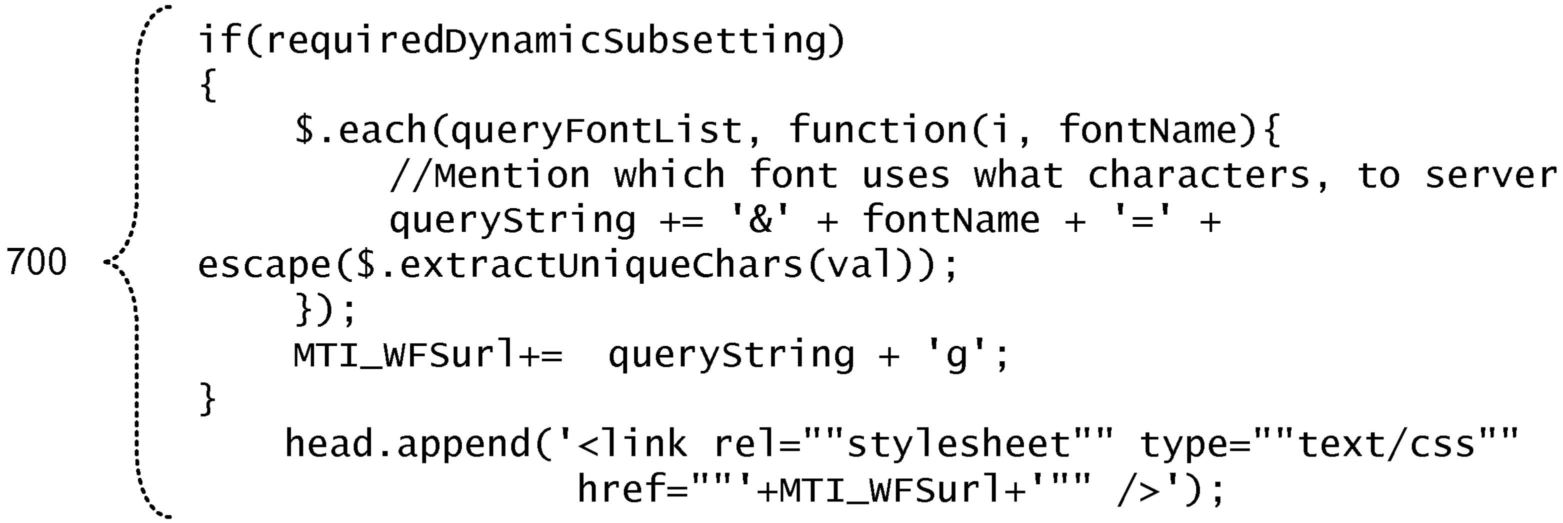
```
$.each(projectSelectorList, function(selector, font){
    var $val = $(selector);

    elementCache.push($val.css({'visibility':'hidden'}));
    //if dynamicsubsetting is required extract text from the
selected elements
       if(requiredDynamicSubsetting)
       {
       //accumulate the text for specific font
          if(queryFontList[font])
               queryFontList[font] += $val.text();
          else
               queryFontList[font] = $val.text();
       }
});
```

FIG. 5

600

```
$('*',$body).filter(function(){
    var $this = $(this);
    var hasProjectFont = false;
    var fontFamily = $this.css('font-family');

    $.each(projectFontList, function(i, fontName){
        if(fontFamily.replace(/^\s|\s$/g,'') == fontName.replace(/^\s|\s$/g,''))
        {
            // if the current element is not in element cache
            if($.inarray($this, elementCache) < 0)
            {
                elementCache.push($this.css({'visibility':'hidden'}));
                if(requiredDynamicSubsetting)
                {
                    if(queryFontList[fontFamily])
                    {
                        queryFontList[fontFamily]+= $this.text();
                    }
                    else
                    {
                        queryFontList[fontFamily]= $this.text();
                    }
                }
                hasProjectFont = true;
            }
        }
    });
     return hasProjectFont;
});

function extractUniqueChars(node, ignoreWhiteSpace) {
      var text = extractText(node, ignoreWhiteSpace);
      var uniq = "";
      var textLength = text.length;
      var curChar = null;
      var charAt = text.charAt;
      for (var i = 0; i < textLength; i++) {
         curChar = text.charAt(i);
         if (uniq.indexOf(curChar) == -1) {
            uniq += curChar;
         }
      }

      return uniq;
   }
```

FIG. 6



```
if(requiredDynamicSubsetting)
{
    $.each(queryFontList, function(i, fontName){
        //Mention which font uses what characters, to server
        queryString += '&' + fontName + '=' +
escape($.extractUniqueChars(val));
    });
    MTI_WFSurl+=  queryString + 'g';
}
    head.append('<link rel=""stylesheet"" type=""text/css""
                href=""'+MTI_WFSurl+'"" />');
```

FIG. 7

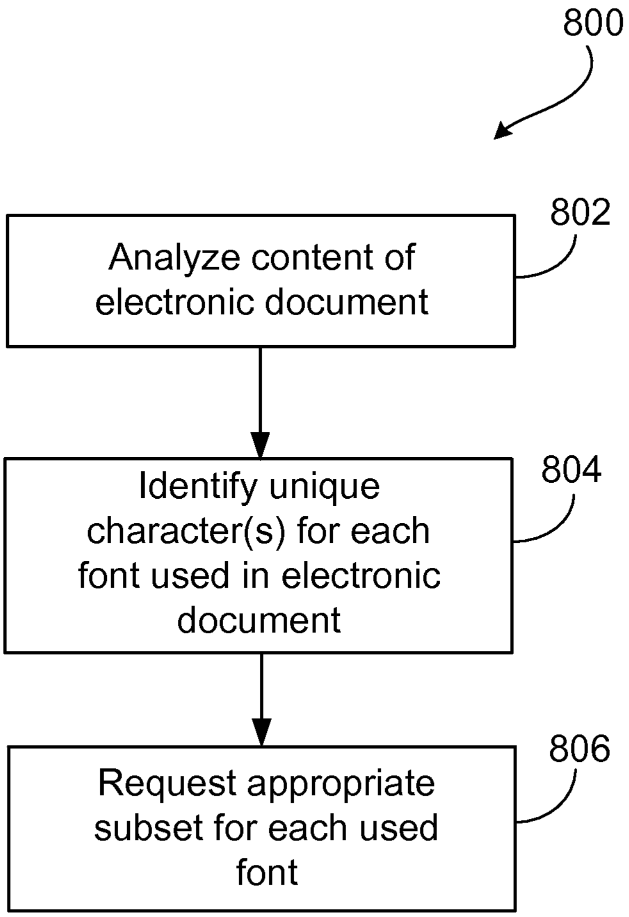

800
Analyze content of electronic document
802
Identify unique character(s) for each font used in electronic document
804
Request appropriate subset for each used font
806

FIG. 8

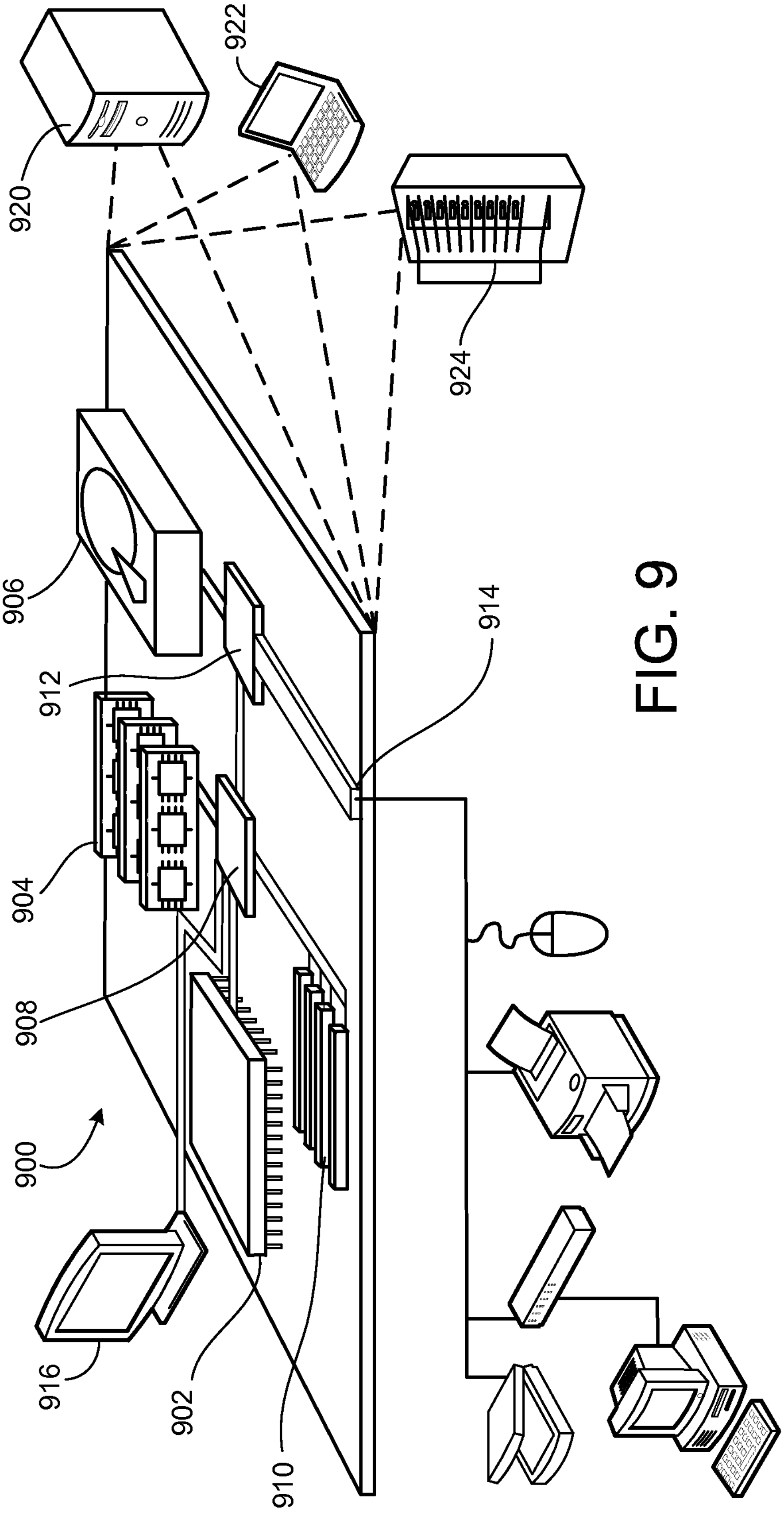


FIG. 9

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2011/034050

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/21
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

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

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

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 01/91088 A1 (OPENTV CORP [US]; ADLER DOUGLAS R [US]; NAWROCKI GREGORY J [US]; KORP) 29 November 2001 (2001-11-29) the whole document -----	1-31
X	US 2008/028304 A1 (LEVANTOVSKY VLADIMIR [US] ET AL) 31 January 2008 (2008-01-31) the whole document -----	1,4-21, 24-28,31
A	US 5 940 581 A (LIPTON DANIEL I [US]) 17 August 1999 (1999-08-17) cited in the application abstract column 1 - column 2, line 61 ----- -/--	8,19

☒ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search 29 June 2011	Date of mailing of the international search report 15/07/2011
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer Woods, Justin

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	Anonymous: "Announcing Speakeasy: A new open-source language tool from Typekit", , 28 October 2010 (2010-10-28), pages 1-3, XP55001644, http://blog.typekit.com/2010/10/28/announcing-speakeasy-a-new-open-source-language-tool-from-typekit/ [retrieved on 2011-06-29] the whole document -----	1,10,21, 28

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2011/034050

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 0191088 A1	29-11-2001	AU 6311501 A	03-12-2001
		CA 2409981 A1	29-11-2001
		EP 1301916 A1	16-04-2003
		JP 2004501442 A	15-01-2004
		US 7155672 B1	26-12-2006
US 2008028304 A1	31-01-2008	EP 2047381 A2	15-04-2009
		JP 2009545064 A	17-12-2009
		WO 2008013720 A2	31-01-2008
US 5940581 A	17-08-1999	NONE	