MOBILE-COMMERCE STORE GENERATOR THAT AUTOMATICALLY EXTRACTS AND CONVERTS DATA

Abstract: (00102) Electronic documents, such as e-commerce stores and electronic forms, can be displayed on a range of devices, which have different screen properties. In order to present e-commerce stores in a visually consistent and readable way, they need to be converted. The present disclosure is applied to e-commerce stores that can be converted and provides a mobile engine with data from one or more e-commerce stores in an internal standardized format so that the mobile engine can adjust and optimize the data so that it can be optimally rendered into one or more mobile commerce stores for targeted end user machines. The automatic content identifier can be used to extract the relevant data from one or more e-commerce stores.

Published:
— with international search report (Art. 21(3))
— before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))
MOBILE-COMMERCE STORE GENERATOR THAT AUTOMATICALLY EXTRACTS AND CONVERTS DATA

CROSS-REFERENCE TO RELATED APPLICATION(S)

[0001] This application relates to and claims priority from U.S. Patent, Application No. 61/728,528, filed on November 20, 2012 and U.S. Patent Application No. 14/050,623, filed on October 10, 2013 the entire disclosures of which is incorporated herein by reference.

FIELD OF THE DISCLOSURE

[0002] The present disclosure relates generally to electronic document analysis and electronic-commerce (“e-commerce”), and more specifically to exemplary embodiments of system, method, software arrangement and computer-accessible medium which can automatically access an electronic document, extract data from the electronic document, process the data and adjust and optimize the data such that it can be rendered into a mobile commerce store.

BACKGROUND INFORMATION

[0003] The Internet provides users access to a multitude of websites. These websites can be made up of web pages, which can be linked together using a hypertext markup language (“HTML”) code. Websites can also include links or access to other electronic documents, including but not limited to, spreadsheets, PDFs, word processing documents and post-script documents. These websites can further include e-commerce stores for persons to shop electronically (e.g., online). Persons can purchase products in the e-commerce stores. Products can include, but are not limited to, physical goods (e.g., cell phones), digital goods (e.g., e-books), services (e.g., technical support) and virtual goods (e.g., extensions of warranty). While many of these electronic documents, including web pages, are optimally viewed on large-screen devices such as desktops, televisions and laptops, they may not be optimally viewed on small-screen devices, such as personal digital assistants (“PDAs”) and
mobile phones, due to their different display capabilities, such as different total pixel count, pixels per inch and the graphics capabilities of the device itself or its screen. This can make reading or navigating between the electronic documents more complex and more demanding than on a large-screen device. These challenges can be particularly apparent in the context of websites, more particularly, e-commerce stores. Navigating through e-commerce stores specifically made for large-screen devices can be especially daunting. Entering the necessary transaction information during a web-based payment flow can be difficult on small screens. This can lead to consumers refraining from making purchases through the store. Since mobile website traffic has been growing rapidly (e.g., comprising approximately 10 percent of total Internet traffic), it can be advantageous to easily and efficiently adapt e-commerce stores for small-screen devices.

Various technological advancements have been described to optimize small-screen viewing of websites. However, these approaches to create small-screen versions of websites have several disadvantages.

Some of the existing approaches for providing or facilitating small-screen versions of websites and examples of their disadvantages are described below:

One such approach for creating small-screen versions of websites can be automatic transcoding. Automatic transcoding creates small-screen versions of web pages through an automated proxy. While this process can streamline the creation of small-screen versions of web pages, the results can exhibit one or many of the following shortfalls: the whole content of the page can be transcoded, so the page becomes very long, and unnecessary content can be shown; the menu items may not be optimized; the reference to which elements should be displayed and which should not can be lost; the positioning relationship of elements in 2 dimensional (x, y) or 3 dimensional space (x, y and in front of each other) can be lost; the loss of positioning can also lead to an order that does not correspond to the reading order of
the original document; and styling can be partially or fully lost. Additionally, since automatic transcoding generally removes client-side scripts, the resulting web page may not contain the same functionality as the original web page. Moreover, it can be very difficult or even impossible to optimize the small-screen pages to work well with individual small-screen devices, such as personal digital assistants (“PDAs”) and mobile phones. Since automatic transcoding does not facilitate users to customize the web pages, there can often be major flaws in the small-screen versions of the web pages.

[0007] In order to improve on the lack of customization that can result from using automatic transcoding, another system and method has been suggested to provide content authors with a means to control how the resulting content will look. This "selective transcoding" can provide content authors the ability to more easily control the style and content of the small-screen web pages. It has been suggested that this method and system facilitates the creation of small-screen web pages that more closely match the original web pages.

[0008] However, the above-described method and system have certain disadvantages. For example, by giving too many of the decision-making responsibilities to a content author, and taking them away from an automatic system, the transcoding process can become more time-consuming, can require repetitive user input, and can lead to sub-optimal viewing experiences in case of user error. Furthermore, the method only removes the document length shortfall and some of the positioning and display shortfalls.

[0009] Another approach for improving automatic transcoding involves extensible transcoder annotations ("XTAs") that provide rules to the transcoder to improve how the web pages can be transcoded. The transcoder executes the XTA instructions. A person can remotely edit the XTAs. In this case, the approach above, however, this approach also potentially improves the transcoding results, but at the expense of additional manual work. Furthermore,
XTAs can only address the document length, and display shortfalls and some of the positioning shortfalls.

[0010] The above conversion processes can also cause e-commerce sites to lose their functionality. Since so many websites have e-commerce stores, and the number of people who shop online or electronically has been rising, this can be highly problematic. Moreover, the process of manually creating a mobile store from scratch using the information from the original website can be very time-consuming and cumbersome. Therefore, an automated approach to generate a store suitable for a mobile device can be preferable.

[0011] Thus, there is a need to provide exemplary embodiments of automatic system, method, software arrangement and computer-accessible medium for, e.g., analyzing and processing electronic document code to optimize it for transcoding, selecting one or more regions of interest using the results, and thereafter automatically transcoding websites, which can improve access to the specific content and web pages within these sites for specific small-screen devices. There is also a need to recreate and/or adapt e-commerce stores on small-screen devices (hereinafter "m-commerce stores"). Furthermore, as indicated herein, there is a need to retain all or some of the styling information, including visibility settings and positioning, and to minimize or eliminate some of the shortfalls of transcoding. It is noted that the above described problem areas are merely representative. Other areas can exist where the exemplary embodiments of automatic system, method, software arrangement and computer-accessible medium for analyzing and transcoding electronic documents can be advantageous.

SUMMARY OF EXEMPLARY EMBODIMENTS

(00121) Exemplary embodiments of the present disclosure provide certain exemplary solutions to the problems of the techniques described above.
For example, **exemplary embodiments** of **automatic** system and method for analyzing electronic documents to optimize such **documents** for **transcoding** can therefore **be** preferable to **make** the transcoder **function** and perform in an improved manner, with no or minimal manual effort. It can be **further** advantageous to provide systems and methods for allowing **e-commerce** stores to present and **sell** **products** on small-screen devices by accessing **e-commerce** stores, extracting data from the **e-commerce** stores, processing the data from the e-commerce store, and rendering the data into a **mobile-friendly** format, with few or no manual steps.

According to certain **exemplary embodiments** of the present disclosure, **exemplary embodiments** of the system, method, software arrangement and **computer-accessible medium** can be provided for analyzing electronic **documents** by identifying a variety of the electronic **documents'** properties, selecting the **document** properties, developing one or more machine-readable **descriptions** of how the properties of the **documents** can be located in the electronic documents, referred herein as **regions of interest**, providing the **regions of interest** to a transcoder, and modifying the electronic **document** via the **transcoder using** the **regions** of interest.

According to certain **exemplary embodiments** of the present disclosure, **exemplary embodiments** of the system, method, **software arrangement and computer-accessible medium** can be provided for analyzing and extracting **data** from one or more **e-commerce** stores, converting/transcoding the data into an **internal** standardized format, and adjusting and optimizing the data such that it can be optimally rendered into an **m-commerce** store.

Representative and/or exemplary electronic documents can include but are not **limited to**, **e.g.**, any type of **document** that can have a clear definition on how to store and how to display data, and hence, can be transcoded into, including but not **limited to**, **e-commerce** documents.
stores. In one exemplary embodiment, the analysis of electronic document can include a logical block identifier known as an identifying strategy that can generate a representation of the electronic document in logical blocks, and a logical block ranker known as a ranking strategy, which can go through the blocks identified, and rank them in relation to one another. In another exemplary embodiment the blocks can be scored via the ranking strategy. The numbers can represent the probability that the electronic document blocks can contain certain functional element types, including, but not limited to, content and menu. In typical embodiments, the blocks picked as representing the menu and the blocks picked as representing the content can each be defined in machine-readable descriptions, known as regions of interest that can specify the locations of the portions of the electronic document code that can be transcoded. These regions of interest can be used to identify relevant data from one or more e-commerce stores which can be extracted and converted. However, any conceivable functional type can each have its own regions of interest. In another exemplary embodiment of the present disclosure, a styling recipe can complement the regions of interest.

To address such exemplary need, exemplary embodiments of methods, systems, software arrangements, and computer-accessible media can be provided for automatically analyzing electronic documents in general, and e-commerce stores specifically, to optimize or prepare them for converting and/or transcoding, and thereafter subsequently convening/transcoding them.

These and other objects, features and advantages of the present disclosure will become apparent upon reading the following detailed description of embodiments of the present disclosure, when taken in conjunction with the appended claims.
BRIEF DESCRIPTION OF THE DRAWINGS

[0019] Further objects, features and advantages of the present disclosure will become apparent from the following detailed description taken in conjunction with the accompanying Figures showing illustrative embodiments of the present disclosure, in which:

[0020] FIG. 1 is a block diagram of an exemplary embodiment of a system according to an exemplary embodiment of the present disclosure;

[0021] FIG. 2 is a schematic representation of an exemplary electronic document analysis engine illustrated in FIG. 1 and its exemplary components including the identifying strategy, the ranking strategy and the styling engine according to an exemplary embodiment of the present disclosure;

[0022] FIG. 3 is a flow diagram of an exemplary embodiment of an automatic-rendering and analysis method according to the present disclosure;

[0023] FIG. 4 is a flow diagram of an exemplary embodiment of a conversion and rendering method according to the present disclosure;

[0024] FIG. 5 is a block diagram of an exemplary comparing device and its internal hardware on which the exemplary system can be implemented according to an exemplary embodiment of the present disclosure;

[0025] FIGS. 6A and 6B are exemplary web pages rendered on a small-screen device that use the exemplary embodiment of an automatic rendering and analysis method, and which are generated by the exemplary embodiment of the system, software arrangement and computer-accessible medium according to the present disclosure;

[0026] FIG. 7 is a block diagram of an exemplary embodiment of a system for generating an m-commerce store according to an exemplary embodiment of the present disclosure;
FIG. 8 is a block diagram of the exemplary mobile engine illustrated in FIG. 7 according to an exemplary embodiment of the present disclosure;

FIG. 9 is a flow diagram of an exemplary embodiment of an exemplary m-commerce store creation according to an exemplary embodiment of the present disclosure; and

FIG. 10 is a block diagram of the exemplary end-user machine, secure validation server and mobile engine of FIG. 7 according to an exemplary embodiment of the present disclosure.

Throughout the drawings, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components, or portions of the illustrated embodiments. Moreover, while the present disclosure will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments and is not limited by the particular embodiments illustrated in the figures and the accompanying claims.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

The exemplary embodiments of the system, method, software arrangement and computer-accessible medium according to the present disclosure, referred to herein as “bMobilized” without any limiting effect, can be implemented using the following exemplary techniques, procedures and arrangements.

In the following description and/or claims, the terms "coupled" and/or "connected" can be used, but are certainly not limited to, to indicate not only that two or more elements can be "directly or physically connected" to each other, but also that two or more elements can be "electrically connected" to each other where intervening elements can be present. Further, in the following description and claims, the terms "includes", "has" "with"
or "comprises" are intended not to exclude other components but to further include other components unless otherwise indicated.

[0033] According to certain exemplary embodiments of the present disclosure, system, method, software arrangement and computer-accessible medium can be provided for analyzing electronic documents by identifying a variety of the electronic documents' properties, selecting the document properties, developing one or more machine readable descriptions of how the documents' properties can be located in the electronic documents, herein referred to as regions of interest, providing the regions of interest to a transcoder, and modifying the electronic document via the transcoder using the regions of interest.

An exemplary region of interest can include, but is not limited to, a pattern matching rule that can identify one or more elements of an electronic document. In other words, it can describe the regions of the electronic document from which code, including, but not limited to HTML, can be fetched for conversion. These patterns, similar to regular expressions or Cascading Style Sheets ("CSS") selectors, can range from simple element names to rich contextual patterns. If all conditions in a certain region of interest can be true for a certain element, then the region of interest can match the element. The syntax of a region of interest expression can vary depending on the electronic document native code (e.g. HTML, PDF, etc.). Examples of regions of interest for HTML can be: 1) +div [ ]; 2) *div [ ]; and 3) -div | j. The regions of interest can be matched with attributes and values to supply extra information about the element's content. For example, +div [id-content] can include the first div with content on the page, and -table [ipos=0] can remove the first table on the page.

[0034] FIG. 1 shows a block diagram of one exemplary embodiment of a system 100 that can be configured to access network content. According to this exemplary embodiment of FIG. 1, the system 100 can comprise a plurality of end-user computing devices in the form of end-user machines 11SA and 118B (collectively 118), a web site server 102 can host a
website 122 which can serve electronic documents, and an optional client machine 120 that can be adapted or configured to convert electronic documents to make them more easily viewable and readable for end-user machines 118. For clarity purposes, and not to provide any limitations, according to one exemplary embodiment, only a single website 122 can be shown. However, in another exemplary embodiment, a plurality of websites can be hosted on the web server 102. Moreover, electronic documents can be herein, but not in any way that limits their description or definition, referred to as web pages that can be served by website 122.

[0035] In one exemplary embodiment, for a first user test request, client machine 120 can select an electronic document served by website 122, to be converted, and communicate with electronic document conversion engine 110 via network 120. In a typical embodiment, the network 120 can include the Internet. However various networks can be used including a local area network, a wide area network, a point-to-point dial-up connection, a cell phone network, etc.

[0036] In another exemplary embodiment, the end-user machine 118 can access the electronic document by placing a call to the electronic document conversion engine 110, where the electronic document conversion engine 110 can act as a way to access the document or a transcoded version of it. Examples of such a setup can include network-proxies, a web page or other electronic document with a field to enter the location of the document or an application on the end-user machine 118 that can modify the electronic document request accordingly. The electronic document conversion engine 110 can then identify the electronic document and find the corresponding parent structure for conversion, such as the corresponding website or domain in the example of web pages. This electronic document, or electronic document parent structure, can then be selected for conversion.
In another exemplary embodiment, information can be provided to the electronic document conversion engine 110, via the network 120, when an electronic document can be requested by the end-user machine 118. Examples of this can include applications that can display electronic documents and can also send information on the documents to the electronic document conversion engine 110. Another example can be a script in the website 122 sending information that a page has been requested to the document conversion engine. The document conversion engine 110 can then locate the corresponding parent structure for conversion, such as the corresponding website 122, or domain in the example of web pages. This electronic document or electronic document parent structure can then be chosen for conversion.

In another exemplary embodiment of the present disclosure, a list of electronic documents, for which the electronic document parent structures can be identified, if required, can be analyzed as described herein below, and the results can be stored in one or more data store 106.

In certain exemplary embodiments of the present disclosure, the exemplary electronic document conversion engine 110 can be connected to a data store 106, which can have automatic parameters stored therein. The electronic document conversion engine 110 can communicate with an electronic document rendering engine 104, regarding the webpage of the website 122 to analyze, and the automatic parameters stored in data store 106. In a typical embodiment of the present disclosure, the electronic document rendering engine 104 can receive the Uniform Resource Locator ("URL") of the page to analyze. In other exemplary embodiments, the electronic document rendering engine 104 can also receive the Hypertext Markup Language ("HTML") of the page to be analyzed, the logo and icons to identify, and any CSS. It can also receive parameters regarding the maximum number of candidates for each functional block type that the electronic document analysis engine 108,
defined below, will be looking for. In certain exemplary embodiments, the maximum number of candidates for each functional block type can be automatically set to one. In another exemplary embodiment, a screenshot can also be taken. The client machine 120 can alter the automatic parameters via customization tools 112.

The exemplary electronic document rendering engine 104 can retrieve the layout attributes of web pages served by website 122 from website server 102 via network 120. Other electronic documents, including electronic forms, can also be rendered by the electronic document rendering engine 164. The exemplary code that the electronic document rendering engine 104 retrieves can be HTML and/or CSS, but it can be other codes as well. The electronic document rendering engine 104 can be any tool that can calculate display and layout attributes, including, but not limited to, web browsers, electronic document interpreters, or HTML+CSS parsers.

The exemplary electronic document rendering engine 104 can be coupled to the electronic document analysis engine 108 (together known as an automatic content identifier 124), which can identify logical block elements in the web pages with the help of the electronic document rendering engine 104, and can rank the block elements and sort the elements according to their rank. In some embodiments, the automatic content identifier 124 can be a web service that can run on a web server, but it can be any type of server. In a typical embodiment; the electronic document analysis engine 108 can score the block elements, and rank them according to the score. The blocks with the highest score can be saved in the data store 106 in a format that can facilitate the ability to retrieve respective blocks in the web page HTML. The data store 106 can include a cloud-based storage service, a database, a local disk drive or any other suitable method of storing data for later access in response to end-user machine 118 requests. The highest scored blocks saved in the data store 106 can be specified in the regions of interest. In another exemplary embodiment of the
present disclosure, the electronic document analysis engine 108 can also create a style
definition for each block and elements in each block. The resulting styling recipe cars
complement the regions of interest. The electronic document analysis engine 108 can also be
coupled with a logging and quality assurance ("QA") info storage 116. The logging and QA
info storage can serve as a repository for the analysis of the results produced by the automatic
content identifier 124. The storage can facilitate the retrieval of records of all the automatic
content identifier 124 analyses in order to evaluate their accuracy. The evaluation process can
be either manual (e.g., human operator), assisted (e.g., human operator with the support of
evaluation tools), or fully automated. The outcome of such evaluations achieves a double
purpose;

1) to pro-actively identify and correct potential analysis issues; and
2) to provide valuable data to further improve the accuracy of the identification
strategy, for instance, by producing additional data to be used to train the process used by the
electronic document analysis engine 108.

[0042] The exemplary electronic document analysis engine 108 can be coupled with
the electronic document conversion engine 110, which can capture and extract the content
from the electronic document analysis engine 108, and parses it into objects. In a typical
embodiment, the electronic document conversion engine 110 can rearrange HTML or other
code for end-user machines 118. The other code can include scripts and styling tags. In one
exemplary embodiment of the present disclosure, the objects can be returned by the electronic
document conversion engine in the order in which they were recognized by the automatic
content identifier 124. In another exemplary embodiment, electronic document conversion
engine 110 can identify and modify the links that make up the menu block, and put them in
association with each other. This can be done through the creation of a hierarchy, as
described in U.S. Patent Application 2007/0130125, which is incorporated herein by reference
in its entirety. In another exemplary embodiment, the links can be put into association with each other via changing link suffixes to incorporate server side logic that can associate it with corresponding menu objects and components of the next electronic document. The electronic document conversion engine 110 can be coupled with customization tools 112, that the client machine 120 can access to customize the converted web pages. The electronic document conversion engine can also be coupled with the device adapter 1.1.4, which can prepare and adapt the objects for targeted end-user machines US. The device adapter 114 can adjust the electronic documents based on end-user machine 118 capabilities such as its overall screen resolution, screen pixels per inch, style-interpretation, graphics and scripting capabilities. In typical embodiments, the device adapter can be any framework, including, but not limited to, Sencha Touch. Adjustment examples can include:

a) increase the font size for end user machines 118 with high overall screen resolution and screen pixels per inch greater than a regular computer screen, so that screen pixels per inch in relation to viewing distance of two end-user machines 11.8 can approximately match;

b) use advanced style definitions, including but not limited to CSS3, in the context of HTML, if end-user machine 118 supports it. Otherwise, provide alternative style definition;

c) if end-user machines 118 support it, include particular graphics or functional elements with graphics, such as high color depth graphics or interactive visual content, such as animations, animated interaction, video players, maps and games. If the end-user machines 118 do not support them, exclude the elements; and

d) include scripting if the end-user machine’s 118 capabilities support it. Scripts, which can include JavaScript and <script> tags, can refer to scripts present in the original electronic document or inserted by the electronic document conversion engine 110 to achieve aforementioned or further usability improvements. If scripting is not supported by the end-user machine 118, it may not be included. The transcoded version of the electronic document...
without scripts can be used. In an alternative embodiment of the present disclosure, the electronic document conversion engine can execute the script and transcode the resulting version of the document.

End-user machines can be coupled to the electronic document conversion engine via the network. The End-user machines can be large-screen devices (such as, e.g., desktops and laptops) or small-screen devices (such as, e.g., personal digital assistants - "PDAs", and mobile phones). In one exemplary embodiment, the large-screen devices, and small-screen devices, can also access the customization tools.

Upon a request from an end-user machine for a given web page served by website via the network, the electronic document conversion engine can retrieve information stored in data store about the blocks-identified and scored. The electronic document conversion engine can use the blocks to transcode (e.g., HTML) for display on end-user machines. In typical embodiments, this can be accomplished by converting the information stored in data store into objects using PHP, although other programming languages can be used, including C++. The objects can be JavaScript Object Notation ("JSON") objects, or any objects that can be interpreted by the device adapter. For the small-screen devices, the electronic document conversion engine can also apply general styling for optimal viewing on the small-screen devices. In one exemplary embodiment, a general styling sheet (e.g., one that conforms to certain preferred practices) can be used by default. However, in another exemplary embodiment, client machine can append the general styling sheet with additional styling. In certain exemplary embodiments, the electronic document conversion engine can also change all links and URLs in the document to either refer to the original files or web pages, or to represent the objects/regions of interest that have been retrieved from the electronic document analysis engine.
device adapter 118 can downsize certain images and transcoded documents, and further adjust the objects so that they can be optimally rendered and displayed on the target end-user machine 118,

[0045] In some exemplary embodiments, on first request for web page by end-user machine 118, the device adapter 114 can cause menu and component objects one level deep, including but not limited to widgets, content and map, to be cached by a browser on end-user machine 118. In part, this can decrease latency between the end-user machine 118 and the server 102. The objects can be stored in document object model ("DOM") and/or WebKit storage. Once in DOM, they can be retrieved on subsequent end-user machine 118 requests. Therefore, on subsequent requests from the end-user machine 118, for a given web page served by the website 122 via the network 120, some of the information one level deep can be fetched from end-user machine 118, instead of from the server 102.

[0046] in another exemplary embodiment, the exemplary electronic document conversion engine 110 can request the electronic document rendering engine 104 and electronic document analysis engine 108 every time without the use of a data store.

[0047] In certain exemplary embodiments of the present disclosure, the given web page that the end-user machine 118 requests does not have stored information in data store 106. If the stored information is not in data store 106, then the electronic document conversion engine 110 can tell the electronic document rendering engine 104 to retrieve the web page served by website 122 on server 102 to analyze. The electronic document rendering engine 104, the electronic document analysis engine 108, the electronic document conversion engine 110 and the device adapter 114 then can carry out the procedure described above.

[0048] In other exemplary embodiments of the present disclosure, the electronic document rendering engine 104 and the electronic document/analysis engine 108 can be used standalone without the electronic document conversion engine 110 and the device adapter
For example, the regions of interest can be used in other contexts such as but not limited to: document search; selective rendering of a document preview image; or comparative statistics over a range of websites.

In typical embodiments, the automatic content identifier 124, the electronic document conversion engine 104, the device adapter 114, the customization tools 112, the data store 106, and the logging and QA into storage 116 can be, e.g., together or in part, implemented on a server and/or computing device 126. The server can comprise memory, input/output ports, external devices, a central processing unit ("CPU"), external devices/resources and one or more buses. The memory can comprise any known type of transmission media and/or data storage, including but not limited to, random access memory ("RAM"), a data object, a data cache, read-only memory ("ROM") and the like. External devices can include, but are not limited to, speakers, a screen, a keyboard, a monitor and a mouse.

Referring to FIG 2, a schematic block diagram can show an exemplary electronic document analysis engine 108 in greater detail. The exemplary structure shown in FIG. 2 is exemplary, and contemplates an engine that can be used for the system 100. In certain exemplary embodiments, one exemplary purpose of the electronic document analysis engine 108 can be to, using the layout attributes, automatically find and identify block elements. The block elements can be categorized per their function. For example, the electronic document analysis engine 108 can categorize blocks as content, menu, logo and/or color scheme blocks. The regions of interest, which can assist in the transcoding and optimization of the web page, can be generated from these block elements. Separate regions of interest can be generated for each functional block type. In some exemplary embodiments, the electronic document analysis engine 108 can generate one or more potential regions of interest for each block type.
The electronic document analysis engine t08 can include an identifying strategy block 200, a ranking strategy block 202 and a styling engine 204. After the electronic document rendering engine 104 calculates the layout attributes of a web page, the identifying strategy block 200 can identify logical block elements using the layout attributes calculated by the electronic document rendering engine 104. In certain exemplary embodiments, the identifying strategy block 200 can divide the electronic document web page into individual blocks, in certain exemplary embodiments of the present disclosure, this can be accomplished by using HTML tags such as "<div> </div>" or any other text based mark-up structure wrapping elements inside an open and a closing information, such as opening and closing tags.

In another exemplary embodiment of the present disclosure, blocks can refer to elements in any binary file structure that can identify the beginning and end of data elements. The identified blocks' coordinates and size can be calculated following the layout and styling rules of the electronic document as defined in the corresponding electronic document format. The coordinates and size can be defined in the context of the display area of the entire electronic document. For the avoidance of doubt, the coordinates and size can refer to any shape, including, but not limited to, rectangles, or circles, or can also refer to a composite shape, such as two rectangles or a rectangle and a circle.

In another exemplary embodiment, the identification strategy block 200 can further identify blocks by utilizing one or more search directions, which can be determined by the cultural, language and layout specific attributes of the analyzed document. For example, search directions can be based on the reading direction of the language, including, but not limited to, left to right for English, right to left for Hebrew, and up-down for Chinese. The use of this additional identification step can be appropriate when: (i) only a few blocks can be identified using the text-based mark-up structures or binary file structures (e.g. below the
lower quartile of the average block count on the corresponding electronic document type); (ii) and/or the identified block’s size is too small (below the size expected for functional blocks); (iii) and/or the blocks can be positioned outside of the regular document display area; (iv) and/or the blocks can be positioned in a way that their order in the electronic document code does not correspond to their visual and reading order.

[0053] In typical embodiments of the present disclosure, where the identification strategy block 200 can utilize one or more search directions to identify blocks, the native blocks can be first identified using the initial identification logic for text-based or binary files. Native blocks can be blocks identified in the native format of the electronic document. For example, they can be the blocks identified by <div> tags. These native exemplary blocks can be logically combined into new blocks that may not originally be present in the native format of the electronic document. For example, in test-based electronic documents, such as HTML or XML based files, the new blocks can be identified by combining additional tags around certain text elements such as individual characters, words, sentences or word combinations separated by a linefeed intended for display.

[0054] Next, the coordinates and sizes for all of the identified blocks can be computed following the layout and styling rules of the electronic document as defined in the corresponding electronic document format.

[0055] In some exemplary embodiments of the present disclosure, in order to identify additional blocks using the search directions, for each block identified, the next adjacent block can be identified using coordinates and size along a search direction, if such a block can be found within a set geometric distance between the first block’s end coordinates in the specified search direction and for second block’s beginning coordinates. For example, when searching top-down, the first block’s bottom coordinates can be compared to the second block’s top coordinates. For example, two blocks can be considered adjacent if they can be
aligned along the reading direction, even if not perfectly aligned, within a certain tolerance. For example, the alignment of two blocks on Cartesian planes can be evaluated considering the overlapping of the y coordinates, for horizontal alignment, or the x coordinates for vertical alignment, of the two blocks. Blocks that can be considered adjacent using this process can be aggregated into one logical block. In addition to the overlapping, other rules can be applied to determine if the two blocks can be adjacent, such as proportions of the geometric dimensions of the two blocks. These steps, which can be used to find eligible adjacent blocks to aggregate, can be repeated until there can be no further element within the distance.

[0056] A combined into logical blocks. The properties of the logical blocks can be determined based on the properties of the two or more blocks that can be part of it. For example, in order to determine the total inner text length of the logical block, the lengths of the inner text of each block can be computed together. In certain exemplary embodiments, the coordinates and size of the logical block can be computed by taking the minimum shape that can contain the shapes of all the combined blocks. The minimum shape, can be, but is not limited to, a rectangle. In other exemplary embodiments, shapes may not be aggregated into one resulting shape, but each logical block can be the union of the shapes of each element.

[0057] In another exemplary embodiment of the present disclosure, a logical block can be the combination of native blocks only, while in other exemplary embodiments, it can be the combination of native blocks and other logical blocks.

[0058] In another exemplary embodiment of the present disclosure, the aforementioned strategy for finding logical blocks can be executed multiple times. After each step, the newly found logical blocks can be added to the native blocks.

[0059] In one exemplary embodiment of the present disclosure, the aggregation into logical blocks can be executed considering multiple directions. For example, both left-to-
right and top-to-bottom reading directions can be used if the electronic document does not aggregate any lines of text: left-to-right direction can be used to aggregate complete lines of text; top-down direction can be used to find small columns of text-flow; and then again left-to-right direction can be used to aggregate columns that belong together or as individual table rows, such as navigation menu bars; and, finally, another top-to-bottom process can be used to aggregate all the rows together with the other identified blocks into major sections of the electronic document.

[0060] The identifying strategy block 200 can also exclude content that may not be relevant. The blocks, identified utilizing the aforementioned methods described above, can serve as candidates for the location of functional areas of the web page. For example, the identifying strategy can categorize the blocks as candidates for being menu blocks (e.g., menu items and structure) and/or content blocks. In some exemplary embodiments, it can also identify logo blocks, color scheme blocks or other blocks. In one exemplary embodiment, the identifying strategy block 200 can search for the logo among the <img> in the page. If it finds a logo image, which can be ranked based on the image URL attributes, position, and/or size, and the image may not be a part of a map, it can download the image from the web page as it is. In another exemplary embodiment, if the logo is only part of an image map, then the identifying strategy block 200 may not download the image. Instead, it can cut the area of the logo from the screenshot and save it in data store 106. One or more fallback measures can also be used if the logo is not found among the <img> tags. In one exemplary embodiment, the identifying strategy block 200 can search among the background images.

[0061] The identifying strategy block 200 can be coupled to a ranking strategy block 202, which can rank the candidate blocks based on their likelihood of being for certain functions. For example, it can rank the candidate blocks according to their likelihood of being menu blocks and/or content blocks, in some exemplary embodiments, ranking strategy block
in order to rank the candidate blocks, can filter the block elements by size, by location and display properties, and/or by their content. For example, it can block elements by function based on whether the block elements contain text, images, or other content important for the display of the resulting web page or other electronic document. Any of the aforementioned filters can be optional, although, it can be preferable to utilize at least one filter. In certain exemplary embodiments, in order to rank the candidate blocks, the ranking strategy block 202 can compute a score for each of the block elements, and assign each block a number. For example, the ranking strategy block 202 can score the links on a page according to the probability that they can be menu items. However, in other exemplary embodiments, the ranking strategy block 202 can rank the block elements by function, without assigning them scores. For example, the ranking strategy block 202 can remove block elements not necessary through filtering. The exemplary blocks can also be ranked based on their natural order.

In certain exemplary embodiments whereby the ranking strategy 202 can compute scores, block elements can be subject to one or more scoring processes. During each single scoring process, elements can be evaluated based on scoring rules with the purpose of establishing whether they can be good candidates to perform a specific function inside the document (e.g., such as being relevant content rather than a navigational element or rather than a company logo) or among a group of documents. The score for each element can be based on the evaluation of at least one of these three exemplary factors:

1) the native code properties of the element. In the context of HTML pages, the properties can be the HTML tags and attributes, the proportion of link text, the proportion of inner text of and the role of the element in the HTML hierarchy, and the like;

2) the visual layout of the element, including, but not limited to, geometrical properties such as shape, dimensions, area, and coordinates; and
3) the combination of native code properties and visual properties. For example, the link density in an HTML element can be evaluated as the number of links that the element contains—native code property—divided by the area that such element occupies—geometric property.

The artificial intelligence procedures of the ranking strategy block 202 responsible for scoring can be trained to recognize the relevant correlations among all the aforementioned factors, and use such findings to properly weigh the probability that a block element serves a particular function. In some exemplary embodiments, the ranking strategy block 202 can also adjust the scoring based on linear comparison, specifically by using a heatmap. For example, the ranking strategy block 202 can use a heatmap with regards to the block elements’ size, location and/or inner text length. The ranking strategy block 202 can also categorize the block elements by adjusting the scoring based on input made during the exemplary AI Training Process. In some exemplary embodiments, the AI Training Process can comprise assembling a list of websites, manually picking the locations of the menu and content on the websites, and compiling data regarding the probability that specific portions of the web page contain the content or menu. The exemplary ranking strategy can automatically compare the results of this process with the preliminary results of the scoring process.

In a certain exemplary embodiment of the present disclosure, menu blocks can be ranked by the following criteria. For the avoidance of doubt this shall be seen as an exemplary list to illustrate the scoring and ranking mechanism and not as a complete list:

1) score geometric properties using the block's x and y axis position compared to the heatmap, the aspect ratio (e.g., width divided by height) and the size of the block. In a certain exemplary embodiment, a block with a high aspect ratio, e.g. 2 or above, at the top of an electronic document, can be a good indication of a menu block and can hence be assigned a high score;
2) score links based on all the parent blocks that lead from the main body of the electronic document to the individual link. In a certain exemplary embodiment, the properties of the parent blocks that can be evaluated can include the type of the block, such as a <div> in HTML, and any reference to style and/or functionality, such as "class" or "Id" attributes and their values in HTML. In a certain exemplary embodiment, the weight of scoring these parent block properties can be determined by evaluating a large number of electronic documents of the relevant type, and finding statistically significant identifiers, such as the "class" attribute containing the word "menu" in the context of HTML;

3) score the proportion of links in comparison to the block's size and text length in characters. The higher the count of links within the block, in comparison to the size of the block (e.g., width x height) or the text contained within the block, the higher the probability that the block can be a menu block. In another exemplary embodiment, the link text length in characters can be compared to the block's text length;

4) score internal links, links that stay within the electronic document's parent structure, by number and in proportion to the total links found within the block;

5) score the individual length in characters of the text within the links and the average of all these link text lengths within a block. If the individual length of a link text and/or the average link text length can be short, but greater than two, then the probability that the block can be the menu block can be high;

6) score the text displayed within the links in the block if it contains specific words or groups of words that can indicate that this block can be a menu, such as "home" or "contact us." The more links within the block that contain such words or groups of words, the higher the probability that the block can be the menu block;

7) the more parent elements a block has until the main body of the electronic document can be reached, the higher the probability that the block can be a menu block;
8) if the block's definition, styling or identification properties could indicate that the role of the block can be the navigation menu, the score can be increased accordingly. Examples include: pre-defined block definitions or tag names, such as the "nav" tag, attributes or style references such as "Id" or "class" or "datarole" with their values containing the word "menu", “nav” or related expressions. The definition of the storage format of the electronic document can, in some cases, include references to aforementioned tags, block names, identifiers or style references, e.g., the HTML 5 definition for HTML documents, h1 another exemplary embodiment, the score of a block can also be decreased if a block's role can be indicated to be another function than the menu; and

9) if the electronic document can consist of several sub-documents, such as "femes" in the context of HTML webpages, the sub-document can have an identification that can indicate the role in the page. The identifiers can contain text such as "nav" or "menu" and therefore the probability that the block can be the menu block can be higher.

In a certain exemplary embodiment, content blocks can be ranked by the following criteria. For the avoidance of doubt, this can be seen as an exemplary list to illustrate the scoring and ranking mechanism and not as a complete list:

1) score geometric properties using the blocks x and y axis position compared to the heatmap, the aspect ratio (e.g., width divided by height) and the size of the block. In a certain exemplary embodiment, a block with a low aspect ratio, e.g., or below, in the center of the document, can be a good indication of a content block and can hence be assigned a high score;

2) score the links per block size (e.g., width x height). The greater the block size in comparison to the number of links within the block, the higher the probability that the block can be the content block;
3) score the block by the proportion of link text as a share of the total text within the block. The lower the proportion of link text, the higher the probability for the block to be the consent block;

4) score the block’s scoring-size in comparison to the maximum scoring-size found among all the individual blocks within the electronic document. In a certain exemplary embodiment, the scoring-size can be the block’s height times two plus the block’s width. In another exemplary embodiment, the scoring-size can be the regular size (e.g., width x height);

5) in XML, XML-like or HTML based electronic document formats, score the block by the proportion of the displayed text within a block as share of the total text that defines the block within the stored electronic document. The higher the proportion of the displayed text, the higher the probability that the block can be the content block;

6) score the proportion of text within the block as a share of the longest text length in characters found among all the individual blocks in the electronic document. The higher the proportion, the higher the probability of the block being the content block, in another exemplary embodiment the amount of text belonging to links can be subtracted before calculating the score. In another exemplary embodiment, blocks with a very short text length (e.g. less than 150 characters) can be given a very low score;

7) score the text within the block as a proportion of the total text displayed by the electronic document. If the text in the block significantly exceeds the majority of the text in the document, e.g. 80% or above, the block can be scored low as the probability that the block contains only the content block of the electronic document can also be low;

8) if the block's definition, styling or identification properties can indicate that the role of the block can be the content, the score can be increased accordingly. Examples can include pre-defined block definitions or tag names, such as the "section" tag, attributes or style references such as "id" or "class" or "datarole" and their value containing the word
"content", "main" or related expressions. The definition of the storage format of the

electronic document can, in some cases, include references to aforementioned tags, block

names, identifiers or style references, e.g., the HTML 5 definition for HTML documents. In

another exemplary embodiment, the score of a block can also be decreased if a block's role

can be indicated to be another function than the content;

9) if the electronic document consists of several sub-documents, such as "frames" in the context of HTML web pages, the sub-document can have an identification that can indicate the role in the page. The identifiers can contain text such as "main" or "content" and, therefore, the probability that the block can be the content block can be higher;

10) if the block contains predominantly external links and can be in a publicly accessible electronic document, the probability of this block containing advertisement can be increased, and hence the probability of this block being the content block can be lower; and

11) if the block contains the menu block, the block can have a high probability of being too big to contain only the content. The probability of the block being the content block can be thus reduced.

For the avoidance of doubt, the exemplary sum of all individual probabilities of a block being the menu or the content block can be reflected in the score or the rank of a block in the list of possible content or possible menu blocks.

Once the block elements can be categorized according to their type and then scored, the ranking strategy block 202 can choose one or more candidate block elements with the highest probability of being one or more certain functional types, including menu and content, mid assign them as those functional types. The number of candidates that the electronic document analysis engine 108 looks for and saves can be saved in data store 106 as an automatic parameter. In an embodiment, the number of candidates can be automatically set to one per functional type, but the number can be changed via customization tools 112.
While assigning the candidate block elements as certain functional types, the ranking strategy block 202 can compare a block’s computed score or rank for the respective function to the other blocks’ score or rank. For each functional block category, the block ranking or scoring highest can be chosen as the functional block. In one exemplary embodiment, each block can only take one function. When this can lead to a block being considered for two functions, the decision can be made on score, rank, a set order of functions, such as menu before content blocks, or a combination of the above. In a different exemplary embodiment, a more elaborate scoring strategy, such as a strategy identifying blocks as menu or content, which is described below, can be used to define the function of a block considered for multiple functions. In another exemplary embodiment of the present disclosure blocks can be classified for multiple functions. In another exemplary embodiment of the present disclosure, multiple blocks for each function can be facilitated.

For clarity purposes, the following example only assumes that two functional block types were scored, but there could be more. In a certain exemplary embodiment involving only the scoring of menu and content blocks, the ranking strategy 202 can choose only the block element with the highest probability of being the real menu, and can assign it as the menu. Then, the ranking strategy 202 can choose the block element with the highest probability of being the real content, and assign it as the content. If the block element with the highest probability of being the real menu also has the highest probability of being the real content, then the ranking strategy block 202 can choose the block element with the second highest probability of being real content to be the content.

Data store 106 can save the information on the highest scored menu (e.g., one or more) and the highest scored content (e.g., one or more), as well as all other automatically identified blocks in a format that facilitates the retrieval of the respective blocks in the website HTML, even if the respective HTML has changed like the regions of interest. The blocks
identified as each functional type based on their score, can be picked as the corresponding regions of interest, including, but not limited to, the regions of interest for content and menu; In certain exemplary embodiments, these regions can be machine readable instructions that can specify the locations of portions of the electronic document to be converted. The regions of interest can tell the electronic document conversion engine 116 what areas of the electronic document should be converted and rendered.

{0071} In another exemplary embodiment of the present disclosure, regions of interests corresponding to blocks can be identified for a binary file format. One or more exemplary regions of interest to locate the blocks within the binary file can be identified by finding a unique series of two or more bytes identifying the blocks, such as a block id stored as a series of bytes within in the file. In another exemplary embodiment of the present disclosure, the starting point of a region of interest within the file can be defined by the number of bytes the first byte of the Mock can be offset against the first byte in the file. For the avoidance of doubt, exemplary bytes can be an exemplary unit for reading the binary file.

{0072} The exemplary ranking strategy procedure and/or block can also be coupled to a styling engine 204 that can create a style definition for each block and each element in the block. The style can be defined in the electronic document itself in accordance with the rules for styling in the respective document format, such as inline styling in an HTML document. Alternatively, external styling definitions referring to the document, if permitted by the document's styling logic, such as Cascading Style Sheets, can be used. Both of the styling definitions can be combined in accordance with the rules of the electronic document. The overall style can be for an end-user machine 118. In one exemplary embodiment, the styling engine 204 can ignore the old style of the original document. In another exemplary embodiment, the styling engine 204 can create a new style that can utilize references within the electronic document (e.g., headline or link identifiers). In another exemplary embodiment.
the styling engine 204 can recreate the visual style of the original electronic document. The styling engine 204 can determine background colors, text and anchor-text styling (e.g., color, font, weight, alignment), image styling, and margin and padding, all based on the original electronic document. For example, the styling engine 204 can determine background colors of the original website by sampling pixels. The styling engine 204 can also analyze the colors of the original website to determine if they can be readable on end-user machines 118. For example, the styling engine 204 can measure color distance and luminosity, and create a greater contrast, if needed.

In one exemplary embodiment of the present disclosure, configuration data can be used to determine which blocks chosen by electronic document analysis engine 108 and stored in data store 106 can be shown to the end-user machine 118. This can be accomplished by either, automatically selecting the blocks using a filtering algorithm, manually selecting which blocks should be selected in a text or database editor or by manually selecting the blocks in a graphical user interface via customization tools 112.

Referring now to FIG. 3, a flow diagram of an exemplary embodiment of the present disclosure is shown for a rendering and analysis procedure 300. For clarity purposes, the exemplary embodiment of the procedure shown in FIG. 3 can focus on block elements categorized as menu and content. However, in certain exemplary embodiments, the exemplary embodiment of the procedure can be used to identify and rank block elements for other functional types. The exemplary procedure begins at procedure 302, where a web page can be inputted. For example, a web page can be inputted after a first user test request by the client machine 120. Next, in procedure 303, the web page’s layout attributes can be analyzed by the electronic document rendering engine 104.

At procedure 304, blocks in the web page can be identified by the identifying strategy 200 of the electronic document analysis engine 108. For example, the identifying
strategy 200 can go through the web page and divide the web page into native blocks using HTML tags or other text based mark-up structure wrapping elements inside an open and closing information. These native blocks can be logically combined into new blocks that may not originally be present in the native format of the electronic document. Additional blocks can be identified using one or more search directions, which can be determined by the cultural, language and layout specific attributes of the electronic document. Blocks that can be considered adjacent using this process can be aggregated into one or more logical blocks. The identifying strategy 200 can also exclude information that may not be relevant, and can categorize blocks based on their functional type, including, but not limited to, menu blocks and/or content blocks.

Continuing in procedure 306, the logical blocks identified and aggregated can be tanked and filtered. For example, the ranking strategy 202 can go through the individual blocks and assign them numbers. The numbers can represent the probability that the blocks can actually be a certain functional type, for example, menu or content. Next, in procedure 308, the blocks with the highest scores for menu and content can be determined. For example, the blocks that cannot be categorized as menu can be extracted, and one or more can be chosen to be converted. Further, blocks that can be categorized as content can be extracted, and one or more can be chosen to be convened. Alternatively, the ranking and scoring process described above can be used. The ranking strategy 202 can be responsible for choosing and extracting the blocks to be converted.

Continuing in procedure 310, the highest scored blocks cannot be stored for retrieval later using the regions of interest. For example, date store 112 cannot store the information. In one exemplary embodiment, only the blocks chosen as menu and content cannot be stored in data store 112. In another exemplary embodiment, more than one of the highest scoring blocks categorized as menu or content can be stored.
Next, in procedure 312, style definitions can be created for the blocks stored. For example, the styling engine 204 can create CSS definitions, and determine background colors, text and anchor-text styling, image styling and margin and padding, all based on the original web page. As another example, the styling engine can infer the correct color of the original web page by sampling pixels.

Referring now to FIG. 4, a flow diagram of an exemplary embodiment of the present disclosure is shown for a conversion and rendering procedure 408. The exemplary flow diagram begins at procedure 401, where, upon end-user machine's 118 request, a web page can be transcoded. At procedure 402, it can be determined whether there can be stored information for blocks for the web page requested, if there is no information stored, then the rendering and analysis exemplary procedure can be carried out in procedure 464. If there is information stored, the stored information about the web page blocks can be retrieved. For example, the electronic document conversion engine 110 can retrieve the stored information about the web page blocks from the data store 106 using, for example, the regions of Merest. Next, at procedure 408, the blocks retrieved can be transcoded for display on end-user machine 118. For example, the electronic-document conversion engine 110 can convert the information stored in data store 106 into objects using PHP. The electronic document conversion engine 110 can also identify and modify the links that make up the menu block, and put them into relationship with each other. This can be done through the creation of a hierarchy. Continuing at procedure 410, the resulting objects can be adjusted to be optimally rendered and displayed on target end-user machine 118. Next, at procedure 412, the web page can be rendered and displayed on target end-user machine 118.

In other exemplary embodiments of the present disclosure, the information used for transcoding can be stored for later use. In another exemplary embodiment of the
present disclosure, the information used for transcoding can be retrieved from website of server, or any other location, instead of from data store.

[0081] FIG. 5 shows one example of a computing device on which the system can be implemented. Such exemplary computing device can have a CPU, such as processor, a memory or storage arrangement, at least one data input port, at least one data output port, and a user interface, all interconnected by one or more buses. However, in other exemplary embodiments, where the computing device can be a server, the data input port(s), the data output port and the user interface do not have to be present. The memory storage can store the operating system software, and other software programs including a program for implementing the automatic content identifier, the electronic document conversion engine, the device adapter, the mobile analytics module, the mobile engine, the catalog feed adapter, and/or the mobile ads module. The memory/storage arrangement can further include data storage for storing collected data sets through one of the input port and/or storing results generated during execution of the program. The data storage can comprise data store and logging and QA info storage.

[0082] The program can be organized into modules which can include coded instructions that when executed by the processor, cause the computing device to carry out different aspects, modules, or steps of the procedure for rendering, analyzing and/or converting an electronic document according to an exemplary embodiment, or the program can carry out the ra-commerce store conversion process disclosed below. A part of memory/storage arrangement, such as data storage, can reside in a different
geographical location from that of processor 502 and be coupled to processor 502 through one or more computer networks.

The program 510 can also include a module including coded instructions, which, when executed by the processor 502, can cause the computing device 500 to provide graphics! user interfaces ("GUP") for the user to interact with the computing device 500, and direct the flow of the program 510.

Referring now to Fig. 6A, a representative web page 600 is shown rendered on the small-screen device MSB using the automatic analysis method, system, software arrangement and computer-accessible medium according to the present disclosure. The exemplary web page 600 of Fig. 6A illustrates a header 604, and a question form 606. Fig. 6B illustrates another exemplary web page 602, also rendered on the small-screen device 118B using the automatic analysis procedure. The exemplary web page 602 shows a header 604, and a menu 60S, which can be selected from the menu block candidates during the analysis process.

Referring now to Fig. 7, an exemplary embodiment of a system 700 for generating a commerce store is shown that can be configured to access network content. System 700 can comprise a plurality of end-user computing devices in the 6 rm of the end-user machines 118, a website server 702 hosting one or more e-commerce stores 714, a catalog feed adapter 708 that can convert e-commerce store data into a standardized format a mobile engine 710, and a configuration storage 712. System 700 can also include the automatic content identifier 124, and an HTML feed adapter 704.

In an exemplary embodiment of the present disclosure, for e-commerce stores 714 that can expose catalogs and product information through one or more data feeds, catalog feed adapter 708, via network 120, can convert/transcode one or more data feeds into a standardized format capable of being easily viewed by the mobile engine 710 to represent all
the information in the e-commerce store 714. Catalog feed adapter 708 can convert/transcode the data feeds to ensure that system 700 processes every catalog in exactly the same manner. The electronic document conversion engine 110 can be used as the catalog feed adapter 708, Data feeds can include, but are not limited to, XML and JSON and E-commerce stores that have Application Programming Interface ("API") functionality and can be Magenfo stores, Google stores and/or Amazon stores. The information from the data feeds that can be converted by the catalog feed adapter 708 can include, but is not limited to, categories, offers, pictures, item attributes and description. The data feeds can be converted/transcoded into objects using PHP, although other programming languages can be used, including C++. The network 120 can include the Internet; however various networks can be used including a local area network, a wide area network, a point-to-point dial-up connection and/or a cell phone or mobile network, etc.

Once the catalog feed adapter 708 converts the data feed into an internal standardized format, the mobile engine 710 can adjust and optimize the data feed such that it can be optimally rendered into a mobile commerce store ("m-commerce store"), either as a mobile website or mobile application. The mobile engine 710, which can prepare and adapt the data feeds for targeted end-user machines 118, can be device adaptor 114, or one or more other mobile engines. The m-commerce store can be stored on a server and/or computing device 716, or any other server including, but not limited to, website server 702. The configuration storage 712 can store configuration parameters for each commerce store. The parameters can include, but are not limited to, connection information, rendering preferences, currencies and payment methods accepted.

If e-commerce stores 714 do not expose data feeds, the automatic content identifier 124, via network 120, can analyze the e-commerce stores 714, and create one or more regions of interest to identify the relevant information in the HTML of e-commerce.
store 714 using the exemplary methods, systems, software arrangements and computer accessible mediums described above. The HTML feed adapter 704 can then extract information from the relevant HTML identified in the one or more regions of interest, and convert/translate the region of interest into an internal standardized format. For example, the information can be converted into objects using PHP, although other programming languages can be used, including C++. The mobile engine 710 can prepare and adapt the data feeds to be rendered on targeted end-user machines 118.

[0089] One or more signatures can be assigned to each product in the e-commerce store 714. This can be done to ensure that the product information and/or the process may not be maliciously altered before the cart information can be submitted to a secure validation server 718 to execute the transaction. After the information can be transformed into a standardized format, each product can be signed by applying an MD5 procedure to the basic properties of the product (e.g., including price, id, description, etc.) together with a secret code. Examples of product signature function in PHP can include:

```php
private function calculateOfferSignature($itemId, $idType, $title, $itemDescription, $price, $offerQuantity, $currencyCode, $secretCode) {
    return md5($secretCode . "|" . $itemId . "|" . $idType . "|" . $title . "|" . $price . "|" . $offerQuantity . "|" . $currencyCode . "|" . $itemDescription);
}
```

[0090] The catalog feed adapter 708, and/or the HTML Feed Adapter 704, can assign the signature, and all the product information associated with the products can be passed to the m-commerce store. The mobile engine 710 can verify the signature during a checkout request, which is described below.

[0091] End-user machines 118 can access the optimized m-commerce stores via network 120, and make catalog browsing requests and/or purchases. The mobile engine 710, once the request can be made, can receive the request, which can include product searching.
browsing and/or sorting requests. The mobile engine 710 can satisfy the request using the most recent data fetched from the original e-commerce website 714. M-commerce stores rendered and generated via mobile engine 710 can mirror the catalog information present in the original e-commerce website 714, but can also provide its own cart functionality and offer different checkout options based on the parameters in configuration storage 712.

[0092] End-user machines 118 can make purchases during a cart checkout. During a cart checkout, mobile engine 710 can receive the order from the m-commerce store via network 120, and validate every product to ensure that prices and other basic product information have not been altered from their original values. This can be carried out by the mobile engine 710 using the following exemplary routine:

1) recalculate the signatures on each product of the cart and verify that they match the product signature assigned by the catalog feed adapter 708 and/or HTML feed adapter 704;

2) recalculate the shopping cart total and verify that it matches the submitted cart total;

3) validate that the ordered quantity of each product can be equal to or lower than the available quantity in the inventory; and

4) upon the success of the above validation, encrypt the entire cart according to the selected payment system (e.g., PayPal, Amazon, etc.) and generate secure payment form that can be used to redirect the end-user machine 118 to the secure validation server 718.

[0093] If the validation succeeds, the mobile engine 712 can redirect end-user machine 118, and the validated cart information, to the secure validation server 718, where the payment, billing and shipping information can be provided and the actual transaction can take place. In certain embodiments of the present disclosure, payments can be processed at delivery or at a later step. In certain embodiments of the present disclosure, the secure validation server 718 can exist on the server and/or computing device 71.6. At the end of the
transaction, end-user machines 118 can have the option to return to the m-commerce store to continue shopping.

[0094] The mobile engine 710 can also be connected to a mobile analytics system 720, which can receive event tracking notifications related to end-user machine's activities related to the one or more m-commerce stores visited. The event-tracking information that the mobile analytics system can collect can include browsing of products, product details, searches, and/or products added to or removed from the cart. The mobile analytics system 720 can provide a series of statistics to the mobile engine 710 that can be used for advertising selection, product recommendations and/or promotions of different kinds. The m-commerce store can push tracking information from one or more end-user machines 118 to the mobile analytics system 720 based on actions and interactions, including, but not limited to, adding an item to the cart or a wish list, writing a product review, browsing a certain category, searching for specific keywords and/or purchase history. The mobile analytics system 720 can be Google analytics, an internal developed analytics service, or any third-party analytics service.

[0095] The mobile engine 710 can also be connected to a mobile ads module 722. The mobile ads module 722 can store advertising information and policies related to each mobile store. Upon request by the mobile engine 710, the mobile ads module can provide the dataset containing the advertising information to be optimized and rendered on specific pages and locations in one or more m-commerce stores.

[0096] The automatic content identifier 124, HTML feed adapter 704, the catalog feed adapter 708, the mobile engine 710, the configuration storage 712 and the mobile analytics module 720 can be, e.g., together or in part, implemented on a server and/or computing device 716. The server can comprise memory, input/output ports, external devices, a CPU, external devices/resources and one or more buses. The memory can comprise any known type of
transmission media and/or data storage, including but not limited to, RAM, a data object, a data cache, ROM and the like. External devices can include, but are not limited to, speakers, a screen, a keyboard, a monitor and a mouse.

Fig. 8 shows a block diagram illustrating an exemplary mobile engine 710. Upon a request from an end-user machine 118 for an m-commerce store served by the website server 702 via the network 120, or sewer 716, the device detection proxy 860 can recognize the end-user machine 118 type and viewport. A rendering engine 802 can then optimize the data feed, and/or information extracted from the e-commerce store 714, into a visual format to be rendered and displayed on the type of end-user machine 118 recognized by the device detection proxy 800. An advertisement ("ads") builder 804 can be included to generate advertising information to be included in the m-commerce store rendered on the end-user machine 118. The ads can be generated based on the configuration and/or the statistic information provided by the mobile analytics system 720 for the particular m-commerce store stored in configuration storage 712.

Referring to Fig. 9, a flow diagram of an exemplary embodiment of the present disclosure is shown for an exemplary m-commerce store creation procedure 900. The exemplary procedure begins at procedure 902 where it can be determined, either manually or automatically, whether the e-commerce store 714 exposes a data feed. If the e-commerce store 714 exposes a data feed, then the catalog feed adaptor 708 can convert the data feed into an object standardized format at procedure 906. The data feeds can be converted/transcoded into objects using PHP, although other programming languages can be used, including C++. If the e-commerce store 714 does not expose a data field, then the automatic content identifier 124 can analyze the e-commerce store 714 to determine the relevant information to be converted using the identifying and ranking procedure described above at procedure 904. Then at procedure 906, the HTML feed adapter can extract and convert the relevant
information identified from the automatic content identifier 124 into an object standardized format. At procedure 910, the objects can be optimally adjusted for optimal display, and rendering on targeted end-user machine 118. Lastly, at procedure 912, the m-commerce store 714 can be rendered and displayed on target end user machine 118, upon the end-user machine's 118 request for a given e-commerce store 714.

[0099] FIG. 10 illustrates a data flow among end-user machine 118, secure validation server 718, and mobile engine 710. End-user machine 118 can make a checkout request with the cart information for validation (1002) from m-commerce store. Mobile engine 710, upon the checkout request, can validate the cart information and provide a list of checkout methods (1004) that can redirect end-user machine 118 to secure validation server 718. The validation can be carried out by the mobile engine 710 using the following exemplary routine:

1) recalculate the signatures on each product of the cart, and verify that it matches the product signature assigned by the catalog feed adapter 708 and/or HTML feed adapter 704;

2) recalculate the shopping cart total, and verify that it matches the submitted cart total;

3) validate that the ordered quantity of each product can be equal to or lower than the available quantity in the inventory; and

4) upon the success of the above validation, encrypt the entire cart according to the selected payment system (e.g., Paypal, Amazon, etc.) and generate secure payment form that can be used to redirect the end-user machine 3.18 to the secure validation server 718.

[0100] The cart and payment info from m-commerce store can then be redirected (1006) to secure validation server 718. The secure validation server 718 can handle and protect the necessary payment information and carry out the actual transaction. After the transaction can be completed or cancelled, payment notification (1008) can be sent to mobile engine 710, and end-user machine 118 can be redirected back to m-commerce store.
The foregoing merely illustrates the principles of the disclosure. Various modifications and alterations to the described embodiments will be apparent to those skilled in the art in view of the teachings herein. It will thus be appreciated that those skilled in the art will be able to devise numerous systems, arrangements, and procedures which, although not explicitly shown or described herein, embody the principles of the disclosure and can be thus within the spirit and scope of the disclosure. Various different exemplary embodiments can be used together with one another, as well as interchangeably therewith, as should be understood by those having ordinary skill in the art. In addition, certain terms used in the present disclosure, including the specification, drawings and claims thereof can be used synonymously in certain instances, including, but not limited to, for example, data and information. It should be understood that, while these words, and/or other words that can be synonymous to one another, can be used synonymously herein, that there can be instances when such words can be intended to not be used synonymously. Further, to the extent that the prior art knowledge has not been explicitly incorporated by reference herein above, it is explicitly incorporated herein in its entirety. All publications referenced are incorporated herein by reference in their entireties.
WHAT IS CLAIMED IS:

1. A computer system for analyzing at least one electronic document, comprising:

   at least one hardware processing arrangement which, when executing a set of instructions, performs procedures comprising:

   receiving data associated with content of at least one electronic document;

   automatically converting the data into a standardized format; and

   automatically adjusting the data so that it can be optimally rendered into an m-commerce store;

   wherein if the at least one electronic document does not expose one or more data feeds, the at least one hardware processing arrangement automatically analyzes the data to determine a relevant content, and converts the determined relevant content into the standardized format.

2. The computer system according to claim 1, wherein the relevant content includes information that relates to one or more products.

3. The computer system according to claim 2, wherein the at least one hardware processing arrangement automatically analyzes the data using an automatic content identifier by creating one or more regions of interest to identify the relevant content.

4. The computer system according to claim 3, wherein if the at least one electronic document exposes the one or more data feeds, the automatic conversion of the data is carried out by a catalog feed adapter.
5. The computer system according to claim 4, wherein the catalog feed adapter converts the data into objects using PHP.

6. The computer system according to claim 1, wherein the at least one electronic document is hosted on a server.

7. The computer system according to claim 1, wherein the at least one electronic document is formatted using HTML.

8. The computer system according to claim 1, wherein the computer system is provided in a server.

9. The computer system according to claim 1, further comprising at least one end-user machine communicating with the at least one hardware processing arrangement.

10. The computer system according to claim 1, further comprising a hardware storage arrangement which stores configuration parameters for each m-commerce store.

11. The computer system according to claim 1, wherein the at least one hardware processing arrangement is further configured to execute:

   a mobile ads module which configures the at least one hardware processing arrangement to store advertising information and policies related to each m-commerce store, and

   a mobile analytics module which configures the at least one hardware processing arrangement to receive event tracking notifications related to end-riser machine activities.
related to one or more m-commercee stores visited; and further comprising secure validation server providing payment, billing and shipping information and effectuate a transaction to take place based thereon,

12. The computer system according to claim 1, wherein the at least one hardware processing arrangement renders the data for a target end-user machine.

13. The computer system according to claim 12, wherein the target end-user machine is a small-screen device.

14. A computer-implemented method for analyzing at least one electronic document, comprising;

- receiving data associated with content of at least one electronic document;
- using a computing device, automatically converting the data into a standardized format, and automatically adjusting and optimizing the data so that it can be optimally rendered into an m-commercee store;
- using a hardware processing arrangement, determining if the at least one electronic document does not expose one or more data feeds, and automatically analyzing the data to determine a relevant content, and
- converting the relevant content into the standardized format.

15. The computer-implemented method according to claim 14, wherein the relevant content includes information that relates to one or more products.
16. The computer-implemented method according to claim 14, wherein the hardware processing arrangement automatically analyzes the data using an automatic content identifier by creating one or more regions of interest to identify the relevant content.

17. The computer-implemented method according to claim 14, wherein if the at least one electronic document exposes the one or more data feeds, the automatic conversion of the data is carried out by a catalog feed adapter.

18. The computer-implemented method according to claim 17, wherein the catalog feed adapter converts the data into objects using PHP.

19. The computer-implemented method according to claim 14, wherein the data is optimally rendered for a target end-user machine.

20. The computer-implemented method according to claim 14, further comprising:
   - storing configuration parameters for each commerce store in a storage arrangement;
   - storing advertising Information and policies related to each m-commerce store in the storage arrangement;
   - receiving event tracking notifications related to end-user machine activities related to one or more m-commerce stores visited; and
   - storing a payment, billing and shipping information in, and carrying out an actual transaction using, a secure validation server.

21. A software arrangement provided for automatically analyzing at least one electronic document comprising:
a first module, which when executed by a processing arrangement, causes the processing arrangement to receive data associated with content of at least one electronic document;

if the at least one electronic document does not expose one or more data feeds, a second module, which when executed by a processing arrangement, causes the processing arrangement to automatically analyze the data to determine relevant content;

a third module, which when executed by a processing arrangement, causes the processing arrangement to automatically convert the relevant content into a standardized format; and

a fourth module, which when executed by a processing arrangement, causes the processing arrangement to automatically prepare and adapt the data to be rendered on to a targeted end-user machines.

22. The software arrangement of claim 21, wherein the second module is an automatic content identifier.

23. The software arrangement of claim 21, wherein the fourth module comprises:

a device detection proxy;

a rendering engine; and

an ads builder.
Start

Input Web page and Analysis Parameters

Render Web page

Identify Blocks in the Web page and Aggregate them into one or more Logical Blocks

Filter Said Blocks and Rank Them

Extract Blocks Categorized as Content. Choose One or More to be Converted
Extract Blocks Categorized as Menu. Choose One or More to be Converted.

Store Regions of Interest for Retrieving Block(s) Again, Even if the HTML Has changed

Create Style Definition for Block(s) Stored

End

Fig. 3
Start

On End-User Machine Request Transcode Web Page

Retrieve Layout Attributes of Web Page and Analyze Web Page

No

Stored Information about Web Page Blocks?

Yes

Retrieve Stored Information about the Web Page Blocks

Transcode Blocks for Display on End-User Machines

Adjust the Objects for Optimal Display on Targeted End-User Machine

Render and Display Web Page on Targeted End-User Machine

End

Fig. 4
Fig. 5
Fig. 6A
Fig. 8
Fig. 9
INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 2013/070897

A. CLASSIFICATION OF SUBJECT MATTER

G06Q 30/00 (8/2012.01)
H04N 21/2343 (2011.01)

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 3/00-3/0489, 3/14-3/153, 9/00-9/54, 13/00, 15/00-15/82, 17/00-17/40, G06T 1/00-1/60, 3/00-3/60,
G06Q 20/00, 20/30-20/42, 30/00, 30/02, 30/06, 90/00, G09G 5/00-5/42, H04M 3/00, 3/42,
H04N 5/00-5/956, 9/00-9/898, 19/00-19/98, 21/00-21/858

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

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

PatSearch (RUPTO internal), USPTO, PAI, Esp@cenet, Information Retrieval System of FIPS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category*</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>US 2005/0188066 A1 (THOMAS PAPANIKOLAOU et al.) 25.08.2005, abstract, paragraphs [0029], [0049], [0056]-[0057], [0063], [0065]-[0066], [0068]-[0069], [0076]</td>
<td>1, 6-9, 12-14, 19, 21-23</td>
</tr>
<tr>
<td>Y</td>
<td>US 2010/01 14857 A1 (JOHN EDWARDS et al.) 06.05.2010, par. [0026], [0033], [0046], [0058], [0075]</td>
<td>2-5, 10-11, 15-18, 20</td>
</tr>
<tr>
<td>A</td>
<td>WO 01/33416 A1 (KIDATA AG) 10.05.2001</td>
<td>1-23</td>
</tr>
</tbody>
</table>

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

Date of the actual completion of the international search
16 April 2014 (16.04.2014)

Date of mailing of the international search report
17 April 2014 (17.04.2014)

Name and mailing address of the ISA - HPS
Russia, 123995, Moscow, G-59, GSP-5, Berezkhnovskaya nab., 30-1
Facsimile No. +7 (499) 243-33-37

Authorized officer
S. Ilyasov
Telephone No. (499) 240-25-91

Form PCT/ISA/210 (second sheet) (July 2009)