Title: DYNAMICALLY OPTIMIZED CONTENT DISPLAY

Abstract: Provided are systems and methods for dynamic optimization of content on a client device. In one example, a method for dynamic optimization of content is based on receiving the content and parameters of a display area associated with the content from an enterprise system. The enterprise system is pre-configured based on predetermined business rules. The content is analyzed to determine properties associated with the content and selectively optimized based on the parameters of the display area and the properties associated with the content. Thereby, optimized content is produced. The optimized content is selectively rendered on the client device, thereby providing for selective optimization and rendering of the content without reconfiguring the enterprise system.
Dynamically Optimized Content Display

CROSS-REFERENCE TO RELATED APPLICATIONS


FIELD

[0002] This application relates generally to enterprise software systems and more specifically to enterprise resotirce planning (ERP) systems and plugins associated with ERP systems for selective optimization and rendering of content.

BACKGROUND

[0003] ERP systems were originally developed in the 1980s and the 1990s to support access by desktop terminals and/or personal computers (PCs). These systems are quite complex, difficult to implement, and require heavy customization by customers. Oftentimes, ERP systems have specific business
rules as well as regulatory and security functionalities, which are difficult to create and to reconfigure once created.

[0004] Existing ERP systems cannot provide users with optimal experience for every device; optimization of ERP systems for a specific device can require extensive customization, purchasing of additional hardware, and various system integration.

SUMMARY

[0005] This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

[0006] The present disclosure is related to dynamic optimization of content. Specifically, according to one example embodiment of the present disclosure, a method for dynamic optimization of content includes receiving the content from an enterprise system and receiving parameters of a display area associated with the content. The enterprise system is pre-configured based on specific business rules. The content is analyzed to determine properties associated with the content and selectively optimized, based on the parameters of display area and the properties associated with the content, to produce optimized content. The optimized content is selectively rendered on a client device, thereby providing selective optimization and rendering of the content without reconfiguring the enterprise system.

[0007] According to another approach of the present disclosure, there is provided a system for dynamic optimization of content comprising a
communication module, an analyzing module, an optimization module, and
a rendering module. The communication module is operable to receive the
content from an enterprise system and parameters of a display area
associated with the content. The enterprise system is pre-configured based
on specific business rules. The analyzing module is operable to determine
properties associated with the content. The optimization module is operable
to selectively optimize the content, based on the parameters of the display
area and the properties associated with the content, thereby producing
optimized content. The rendering module is operable to selectively render
the optimized content on a client device. Thus, the selective optimization of
the content is provided without reconfiguring the enterprise system.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Embodiments are illustrated by way of example and not limitation
in the figures of the accompanying drawings, in which like references
indicate similar elements.

[0009] FIG. 1 is a schematic high level representation of an ERP system
including a system for dynamic optimization of content on client devices.

[0010] FIG. 2 is a block diagram illustrating a system for dynamic
optimization of content.

[0011] FIG. 3 is a schematic representation of various interactions
between components of the ERP system and a client device.

[0012] FIG. 4 is a flow diagram illustrating various interactions between
components of the ERP system, a system for dynamic optimization of
content, and a client device.
[0013] FIG. 5 is a block diagram showing handling of a client request by the ERP system with a system for dynamic optimization of content.

[0014] FIG. 6 is a process flow chart representing a method for dynamic optimization of content.

[0015] FIG. 7 is a diagrammatic representation of an exemplary machine in the form of a computer system, within which a set of instructions for causing the machine to perform any one or more of the methodologies discussed herein may be executed.
DETAILED DESCRIPTION

[0016] ERP systems can integrate internal and external management information across an entire organization, embracing various relationships within the organization, such as finance/accounting, manufacturing, sales and service, and customer relationship management. ERP systems automate many of these activities with an integrated software application. The purpose of ERP systems is to facilitate the flow of information between all business functions inside the boundaries of the organization and manage the connections to outside stakeholders. ERP systems can run on a variety of computer hardware and network configurations and typically employ a database as a repository for information.

[0017] Many ERP systems are deployed as web-based applications, allowing various functions of such ERP systems to be accessed and run from remote locations via web browsers installed on tablet computers, laptops, mobile telephones, or desktop computers. However, user interfaces created for using web-applications are not optimized for specific devices. The system for dynamic optimization of content described herein enables interactions with client devices of different types without additional customizations by extending instead of replacing ERP system output. The system can perform dynamic optimization of content from an ERP system by working with the user interface generated by the ERP system. More specifically, the optimization is performed based on parameters of a display area of a client device. The parameters of the display area of the client device can be associated with user activity related to the client device. User activity can include user manipulations on the display area (e.g., by a mouse or a finger). The manipulations include, for example, changing a size of the
display area by pressing and holding down the mouse button while pulling
the mouse cursor to the desired end of the display area. Using this approach
a user can transform the content layout according to user preferences,
thereby improving the usability and user experience.

[0018] In some embodiments, the system for dynamic optimization of
content enables customizations and upgrades of an ERP system. This
approach allows organizations to avoid custom modifications to client
devices as the system for dynamic optimization of content renders the
content efficiently for any client device. In some embodiments, the system
for dynamic optimization of content is a plug-in that is integrated into the
web server. The system for dynamic optimization of content can leverage
the output of the ERP system as the starting point for marking it up for client
device access. The output is analyzed by the system to read the
code/definitions of the ERP system to understand relationships between
elements in the user interface, parameters of display area received from the
client device, and the meaning of fields and controls on the user interface.
The existing application infrastructure can be leveraged, for example, by
using a J2EE web server plug-in (servlet filter) to inject the optimization logic
into the processing stream. Interactions between the ERP system and client
devices (requests, sessions, and requested content) can be captured by the
system for dynamic optimization of content to allow rules for controlling
security and content rendering.

[0019] In some embodiments, the system for dynamic optimization of
content enables optimizing a user interface based on parameters of display
area received from the client device. The system for dynamic optimization of
content can read the content and processing done on the web server of the
ERP system. Furthermore, the system can modify the content based on predetermined rules and in accordance with the parameters of the display area as received from the client device. The modifying can include restructuring the content, injecting additional content, removing at least a part of the content, and so forth. The dynamic optimization of the content is provided without reconfiguring the ERP system.

[0020] Referring now to the drawings, FIG. 1 is a schematic high level representation of an ERP system comprising an ERP application database 110, an ERP application server 120, and a web server 130 with a system 200 for dynamic optimization of content for interacting with various types of client devices 150. Client devices 150 include any type of computer system capable of supporting a web browser and having some input and output means. Some examples of client devices 150 include desktops, laptops, notebooks, ultrabooks, tablet computers, handheld computers, mobile phones (e.g., smartphones), portable media players, E-book readers, game consoles, and head mounted displays. These devices may have a variety of screen sizes and input options.

[0021] Client devices 150 are connected to web server 130 using a network (not specifically shown in FIG. 1). The network can take any suitable form, such as a wide area network (WAN) or Internet and/or one or more local area networks (LANs). The network includes any suitable number and type of devices (e.g., routers and switches) for forwarding commands, content, and/or web object requests from each client to the online community application and responses back to the clients.

[0022] The systems and methods described herein can also be practiced in a wide variety of network environments including, for example,
Transmission Control Protocol (TCP)/IP-based networks, telecommunications networks, wireless networks, and the like. In addition, the computer program instructions are stored in any type of computer-readable media. The program may be executed according to a variety of computing models including a client/server model, a peer-to-peer model, and a stand-alone computing device, or according to a distributed computing model in which various functionalities described herein are effected or employed at different locations.

[0023] As mentioned above, client devices 150 are capable of supporting web browsers to generate user interfaces. Web browsers allow users and administrators to access various applications and administrative tools of an ERP system. Each client device 150 used for communicating with an ERP system has browser software installed therein. Generally, no additional software (specific to the ERP system) needs to be installed on client devices 150 to implement described methods and systems.

[0024] Web server 130 includes a Java-enabled web server for supporting browser transaction requests and the application messaging technology. Web server 130 is used to install a collection of servlets (e.g., Java servlets) designed to handle a wide range of ERP transactions. Some examples of a web server include an Apache web server, Internet Information Services (IIS) web server, nginx web server, Google web server, and Resin web server. The main function of web server 130 is to deliver web pages based on requests from client devices 150. This means delivery of HTML documents and any additional content that can be included by a document, such as images, style sheets, and scripts. The web browser described above typically initiates communication by making a request for a specific resource using Hypertext
Transfer Protocol (HTTP), and web server 130 responds with the content of that resource or an error message if unable to do so. Web server 130 is also used to receive content from client devices 150. This feature is used for submitting web forms, including the uploading of files. Web server 130 supports server-side scripting using Active Server Pages (ASP), PHP, or other scripting languages. Additional features of web server 130 specific to ERP functions are described below.

[0025] ERP application server 120 is another part of an ERP system. ERP application server 120 executes business logic. It may also be configured to issue SQL to ERP application database 110. In other words, ERP application server 120 serves as a link between web server 130 and ERP application database 110. ERP application server 120 includes various services and server processes that handle transaction requests. ERP application server 120 is also responsible for maintaining the SQL connection to ERP application database 110 for browser requests. Two examples of middleware platforms that are used on ERP application server 120 include TUXEDO (Transactions for Unix Extended for Distributed Operations) and Jolt. For example, TUXEDO is used to manage database transactions, while Jolt is used to facilitate transaction requests issued from the Internet.

[0026] ERP application database 110, which is sometimes referred to as a database server, is yet another component of an ERP system. ERP application database 110 contains one or more database engines and databases. These, in turn, include object definitions, system tables, application tables, and data. ERP application database 110 may be running one of the supported relational database management system (RDBMS)/operating system combinations. The relationship between ERP
application database 110 and ERP application server 120 may be according to a one-to-many model. That is, a single ERP application database 110 can have multiple ERP application servers 120 connecting to it. ERP application database 110 is configured to simultaneously handle multiple application server connections, development environment connections, and batch programs. A component of an ERP system that is not shown in FIG. 1 is a batch server. A batch server includes process schedulers, batch programs (e.g., application engine programs), and COBOL and SQR executables.

As stated above, a system 200 for dynamic optimization of content on web server 130 is configured to communicate with client devices 150. In certain embodiments, the system 200 for dynamic optimization of content is implemented as a web server plug-in, which is a set of software components that adds specific functionalities to a larger software application provided on web server 130. This plug-in enables customizing the functionality of web server 130.

To provide this functionality, system 200 for dynamic optimization of content includes multiple modules, as illustrated in FIG. 2. Specifically, system 200 for dynamic optimization of content includes communication module 202, analyzing module 204, optimization module 206, and rendering module 208, and, optionally, security module 210, logging module 212, and extracting module 214.

In an example embodiment, communication module 202 is operable to receive the content from an ERP system. The ERP system is pre-configured based on specific business rules. Communication module 202 is further operable to receive parameters of a display area associated with the content. The parameters of the display area include one or more of the
following: width of the display area, height of the display area, arrangement of the content on the display area, elements to be included into the content, elements to be excluded from the content, and so forth. The parameters of the display area are associated with user activity related to the client device.

User activity can include user manipulations on the display area (e.g., by a mouse). Manipulation includes, for example, changing a size of the display area by pressing and holding down the mouse button while pulling the mouse cursor to the desired end of the display area.

[0030] In an example embodiment, analyzing module 204 is operable to determine properties associated with the content. The properties associated with the content include definition and structure of the enterprise system.

[0031] In an example embodiment, optimization module 206 is operable to selectively optimize the content. The content is optimized based on the parameters of the display area and the properties associated with the content. The content can be further optimized based on configuration of the client device. In such a case, the system 200 for dynamic optimization of content comprises extracting module 214. Extracting module 214 is operable to read and cache configuration of the client device on a web server of the enterprise system.

[0032] The optimization module 206 is integrated into a web server of the ERP system. As a result of optimization, optimized content is produced. Optimization of the content can include one or more of the following: restructuring the content, injecting additional content, removing at least a part of the content, and the like.

[0033] In an example embodiment, rendering module 208 is operable to selectively render the optimized content on a client device. The client device
includes one or more of a desktop, a laptop, a notebook, an ultrabook, a smartphone, a tablet computer, and so forth. As a result, the selective optimization of the content is provided without reconfiguring the ERP system.

[0034] In a further example embodiment, security module 210 is operable to provide security. The security is based on one or more of the following: a location, a network, rules-based authentication, multiple-factor authentication, time-based authentication, a proxy associated with routing of the content, application data, screens associated with the content, functionality associated with the content, and so forth.

[0035] In another example embodiment, logging module 212 is operable to log information related to one or more of the following: an identity of a user and actions performed under the user in the enterprise system, an identity of the data accessed, a source of a location associated with the client device, and so forth.

[0036] In some embodiments, in order to ensure optimization of the content based on configuration of the client device, the functionality of the system 200 for dynamic optimization of content is represented by a admin services component, a navigation component, a user interface component, a client device extensions component, a security component, and a metadata services component.

[0037] The admin services component controls the overall processing of the system for dynamic optimization of content. Because it is intended to function as a seamless part of the ERP application infrastructure, it is designed to work in conjunction with the processing already done on the web server of the ERP system. As such, it makes decisions about displaying
client device rendering, allowing access to content, level of logging performed, type of content to be served, and/or necessity to reload access rules from the database. In order to make these decisions, the system for dynamic optimization of content understands a profile of the user/device accessing the system, state of the plug-in, and type of request being made to the system.

[0038] The admin services component uses the following attributes of the user/device to control access to the system: the browser/operating system making the request to the system (used to determine whether a client device is accessing the system and make a decision as to whether to perform rendering) and the location from which the user is accessing the system (this is used to determine whether to grant or deny access to content based on location).

[0039] In some embodiments, the type of content requested from the system is used to control the request processing. If the request is for HTML content for the system, the HTML generated by the application will be used as the starting point for rendering. If the request is for navigation menus, then the plug-in will request the raw menu information from the system and generate its own rendering. If the request is for portal content, then the plug-in will either start with the HTML generated by the application or request the raw data, depending on whether the request is related to navigation or content.

[0040] In some embodiments, the navigation component reads navigation rules of the ERP system and repurposes content for client device access. Services performed by the navigation component include reading the hierarchy of menu titles and descriptions to be presented in a format of the
client device as users drill through the menus; pruning navigation to parts of the system that are disabled either due to location security or device access rules; and keeping track of the breadcrumbs and maintaining a breadcrumb menu to allow easy access without hitting the back button. The latter includes navigating through menu relationships in the navigation definitions of the application; identifying navigation collections (pages defined for the purpose of navigation) and displaying them in a menu form; and identifying when the system was accessed from an external system and embedding the breadcrumb for it in the breadcrumb menu. To perform these services, the infrastructure of the navigation component includes JQuery menu and navigation components; role-based caching of navigation menus to minimize the need to request information from the ERP system about who can access what content; and metadata services to allow the web server to access the menus, rules, and the like that drive the presentation and navigation within the system.

[0041] In some embodiments, the user interface component takes the user interface generated by the ERP application and modifies it to work properly on client devices. This includes marking up the HTML generated by the ERP system. Marking up the HTML involves several operations. The first operation is marking the HTML as valid HTML5 syntax. Then, it involves identifying the meaning of the content on the generated user interface by looking at the HTML and Asynchronous Javascript And XML (AJAX XML) generated by the application. The metadata services component is used in combination with the structure of the user interface (UI) to decide on an association of page elements, such as page containers (group boxes), labels, fields, and page controls; movement of groups of entities within the user
interface, so that pages can be displayed in narrow mode and still be meaningful; tagging of fields as to the type of field (i.e., numeric, date, time, alphanumeric); honoring of multilingual translations within the ERP system; changing controls to touch-optimized controls using JQuery (or similar library); and removing extraneous HTML markup (such as extra tables and width tags) intended to optimize UI for wide screen monitors.

Another operation that is a part of marking up the HTML is managing JavaScript generated by the ERP system. This consists of stripping out extraneous JavaScript not needed on the client device and embedding JavaScript to perform processing on the device to minimize the number of server trips and improve performance (such as navigating between multiple items within a page).

Also, marking up the HTML includes rules-based templating infrastructure to minimize code serving to process UI services, thereby allowing rapid iterations and deployments without requiring re-compilation of code; and deciding on logic for making modifications to user interface without requiring expensive processing to walk the document object model (DOM) for each page being rendered.

In some embodiments, the client device extensions component provides a capability for injecting customer-defined logic to the end-product of the user interface component. This allows the customer to make additional UI tweaks to the ERP system without requiring modification of either the core ERP application or the core code of the plug-in. The UI tweaks include injecting modifications without changing the plug-in at different locations within the generated UI. This is done based on understanding properties of the session/user/and content in the ERP system,
leveraging the same rules-based templating infrastructure in the UI services to allow robust processing rules based on content, user, session, or application data; and allowing calls to any JavaScript library that is enabled in the system. The modifications are defined and administered in relational database pages that are deployed within the user interface of the ERP system.

[0045] In some embodiments, the security component provides device and location security features to allow easy control, management, and auditing of access to a client device. This includes rules-based means of identifying whether to allow or block access to parts of the ERP system; generating detailed logs of all traffic, including date/time, IP address, user identity, content accessed, data accessed, and actions taken; and automatic pruning of navigation UI based on rules.

[0046] In some embodiments, the metadata services component represents the bridge between the product processing on the server and the rules by which it performs this processing. The metadata services component provides services that extract rules that define how the ERP system is defined and structured, and services that extract rules that drive how the plug-in should operate on top of the ERP system. By understanding the meaning of the different parts of the ERP system to be rendered, the plug-in is able to automatically transform the UI of the ERP application. For all of these services, the metadata services component reads the definitions from relational tables and application programming interfaces (APIs) in the ERP system; stores the definitions in a cache on the web server to be processed locally; and manages the reading and storing of the definitions at either web server start-up time or as content is requested.
[0047] The metadata services component extracts the meaning of all of the pieces of the user interface generated by the ERP system. Because the plug-in understands at a granular level the meaning of all pieces of the ERP system, whether developed by the original software developer or modified by the customer, it is able to automate the optimized rendering of the user interface. The types of metadata read from the ERP system to be used as part of the plug-in include page element structure, element properties, control properties, navigation rules, and user identity and permissions.

[0048] As the metadata services component is configured and administered from pages within the ERP system delivered by the plug-in, it also includes services to read and cache the client device configuration on the web server (i.e., branding rules, client device extensions, additional CSS and JavaScript rules).

[0049] In some embodiments, because the plug-in contains location-based security capabilities, the rules that define the access control and logging are also managed as part of the metadata services.

[0050] FIG. 3 is a schematic representation of various interactions between components of an ERP system, a system 200 for dynamic optimization of content, and a client device 150. Client device 150 is equipped with a web browser, which uses HTTP protocol. The browser sends requests to web server 130, which forwards the request to ERP application server 120 and further to ERP application database 110. Servlet 302 installed on web server 130 facilitates all browser connections. Configuration data is transferred to configuration cache 306 and further to the system 200 for dynamic optimization of content. User and device data and activity tracking is transferred to activity monitor 304.
FIG. 4 is a process flow chart representing the handling of a client request by an ERP system with a system for dynamic optimization of content. Method 400 commences with receiving a request from a client device at a web server including a system for dynamic optimization of content during operation 402. The request includes parameters of a display area associated with the content to be represented on the client device.

Based on the received request or previously received information, an access level available for this type of client device is determined during operation 404. The previously received information includes another request, login information, and/or a cookie. For example, the request is generated either by a non-mobile computer system or a mobile computer system. Depending on the type of computer system, different access levels are given to the device, and the same type of request may be processed in different manners. Method 400 then proceeds with processing the request at an application server of the ERP system based on the access level during operation 406. This processing operation generates an HTML output. The HTML output involves retrieval of ERP information from an application database of the ERP system.

Method 400 proceeds with processing the HTML output at the ERP interface based on the received parameters of the display area associated with the content to generate a client specific HTML output during operation 408, and transmitting the client specific HTML output to the client device during operation 410.

FIG. 5 shows exemplary optimization of content for a client device. Interface items 506 and corresponding controls 508 rendered on a display area 502 may be optimized to produce a display area 504 with optimized...
arrangement of interface items 510 and corresponding controls 512. The optimization can be performed based on received parameters of the display area 504.

[0055] FIG. 6 is a process flow chart showing a detailed method for dynamic optimization of content. The method 600 is performed by processing logic that comprises hardware (e.g., dedicated logic, programmable logic, microcode, etc.), software (such as that which is run on a general-purpose computer system or a dedicated machine), or a combination of both. In one exemplary embodiment, the processing logic resides at the system 200 illustrated in FIG. 2.

[0056] The method 600 commences at operation 602 with the communication module receiving the content from an ERP system. The ERP system is pre-configured based on specific business rules.

[0057] Parameters of a display area associated with the content are received at operation 604. The parameters of the display area include one or more of the following: width of the display area, height of the display area, arrangement of the content on the display area, elements to be included into the content, elements to be excluded from the content, and so forth.

[0058] At operation 606, the content is analyzed to determine properties associated with the content. The properties associated with the content include definition and structure of the enterprise system.

[0059] After receiving the parameters of the display area and determining properties associated with the content, the content is selectively optimized at operation 608. As a result, optimized content is produced. The optimization is based on the parameters of the display area and the properties associated with the content. The optimization of the content includes one or more of
the following: restructuring the content, injecting additional content, removing at least a part of the content, and the like.

[0060] In an example embodiment, the optimization of the content is further based on configuration of a client device. The client device includes one or more of a desktop, a laptop, a notebook, an ultrabook, a smartphone, a tablet computer, and so forth. In such a case, configuration of the client device is read and cached on a web server of the ERP system.

[0061] Thereafter, the optimized content is selectively rendered on the client device at operation 610. Thus, selective optimization of the content is provided without reconfiguring the ERP system.

[0062] In an example embodiment, method 600 optionally comprises providing security. The security is based on one or more of the following: a location, a network, rules-based authentication, multiple-factor authentication, time-based authentication, a proxy associated with routing of the content, application data, screens associated with the content, functionality associated with the content, and so forth.

[0063] In a further example embodiment, method 600 optionally comprises logging information related to one or more of the following: an identity of a user and actions performed under the user in the enterprise system, an identity of the data accessed, a source of a location associated with the client device, and the like.

[0064] FIG. 7 is a diagrammatic representation of an exemplary machine in the form of a computer system 700, within which a set of instructions for causing the machine to perform any one or more of the methodologies discussed herein may be executed. In various exemplary embodiments, the machine operates as a standalone device or are connected (e.g., networked).
to other machines. In a networked deployment, the machine operates in the capacity of a server or a client machine in a server-client network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The machine includes a PC, a tablet PC, a set-top box (STB), a cellular telephone, a portable music player (e.g., a portable hard drive audio device such as an Moving Picture Experts Group Audio Layer 3 (MP3) player), a web appliance, a network router, switch or bridge, or any machine capable of executing a set of instructions (sequential or otherwise) that specify actions to be taken by that machine. Further, while only a single machine is illustrated, the term "machine" shall also be taken to include any collection of machines that individually or jointly execute a set (or multiple sets) of instructions to perform any one or more of the methodologies discussed herein.

[0065] The exemplary computer system 700 includes a processor or multiple processors 702 (e.g., a central processing unit (CPU), a graphics processing unit (GPU), or both), and a main memory 708 and static memory 714, which communicate with each other via a bus 728. The computer system 700 further includes a video display 706 (e.g., a liquid crystal display (LCD)). The computer system 700 also includes an alphanumeric input device 712 (e.g., a keyboard), a cursor control device 716 (e.g., a mouse), a voice recognition or biometric verification unit (not shown), a disk drive unit 720, a signal generation device 726 (e.g., a speaker) and a network interface device 718. The computer system 700 further includes a data encryption module (not shown) to encrypt data.
The disk drive unit 720 includes a computer-readable medium 722 on which is stored one or more sets of instructions and data structures (e.g., instructions 710) embodying or utilizing any one or more of the methodologies or functions described herein. The instructions 710 also reside, completely or at least partially, within the main memory 708 and/or within the processors 702 during execution thereof by the computer system 700. The main memory 708 and the processors 702 also constitute machine-readable media.

The instructions 710 are further transmitted or received over a network 724 via the network interface device 718 utilizing any one of a number of well-known transfer protocols (e.g., HTTP).

While the computer-readable medium 722 is shown in an exemplary embodiment to be a single medium, the term "computer-readable medium" should be taken to include a single medium or multiple media (e.g., a centralized or distributed database and/or associated caches and servers) that store the one or more sets of instructions. The term "computer-readable medium" shall also be taken to include any medium that is capable of storing, encoding, or carrying a set of instructions for execution by the machine and that causes the machine to perform any one or more of the methodologies of the present application, or that is capable of storing, encoding, or carrying data structures utilized by or associated with such a set of instructions. The term "computer-readable medium" shall accordingly be taken to include, but not be limited to, solid-state memories, optical and magnetic media, and carrier wave signals. Such media also include, without
limitation, hard disks, floppy disks, flash memory cards, digital video disks, random access memory (RAM), read only memory (ROM), and the like.

[0069] The exemplary embodiments described herein may be implemented in an operating environment comprising software installed on a computer, in hardware, or in a combination of software and hardware.

[0070] Thus, methods and systems for dynamic optimization of content have been disclosed. Although embodiments have been described with reference to specific exemplary embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the system and method described herein. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.
CLAIMS

What is claimed is:

1. A system for dynamic optimization of content, the system comprising:
   a communication module operable to:
   receive the content from an enterprise system, the enterprise
   system being pre-configured based on predetermined business rules,
   receive parameters of a display area associated with the
   content;
   an analyzing module operable to:
   determine properties associated with the content;
   an optimization module operable to selectively optimize the content,
   based on the parameters of display area and the properties associated with
   the content, thereby producing optimized content; and
   a rendering module operable to selectively render the optimized
   content on a client device, thereby providing for selective optimization of the
   content without reconfiguring the enterprise system.

2. The system of claim 1, wherein the parameters of display area include
   one or more of the following: a width of the display area, a height of the
   display area, an arrangement of the content on the display area, an
   element to be included into the content, and an element to be excluded
   from the content.

3. The system of claim 1, wherein the client device includes one or more of
   a desktop, a laptop, a notebook, an ultrabook, a smartphone, and a
   tablet computer.
4. The system of claim 1, wherein the optimization module is integrated into a web server associated with the enterprise system.

5. The system of claim 1, further comprising a security module operable to provide security, wherein the security is based on one or more of the following: a location, a network, rules-based authentication, multiple-factor authentication, time-based authentication, a proxy associated with routing of the content, application data, screens associated with the content, and functionality associated with the content.

6. The system of claim 1, further comprising a logging module operable to log information related to one or more of the following: an identity of a user and actions performed under the user in the enterprise system, an identity of the data accessed, and a source of a location associated with the client device.

7. The system of claim 1, wherein the properties associated with the content include definition and structure of the enterprise system.

8. The system of claim 1, wherein the optimizing of the content includes one or more of the following: restructuring the content, injecting additional content, and removing at least a part of the content.

9. The system of claim 1, wherein the parameters of the display area are associated with user activity related to the client device.
10. The system of claim 1, wherein the optimizing of the content is further based on configuration of the client device.

11. The system of claim 1, further comprising an extracting module operable to read and cache a configuration of the client device on a web server of the enterprise system.

12. A method for dynamic optimization of content, the method comprising:
   - receiving the content from an enterprise system, the enterprise system being pre-configured based on predetermined business rules;
   - receiving parameters of a display area associated with the content;
   - analyzing the content to determine properties associated with the content;
   - selectively optimizing the content, based on the parameters of the display area and the properties associated with the content, thereby producing optimized content; and
   - selectively rendering the optimized content on a client device, thereby providing for the selective optimization of the content without reconfiguring the enterprise system.

13. The method of claim 12, wherein the parameters of display area include one or more of the following: a width of the display area, a height of the display area, an arrangement of the content on the display area, an
element to be included into the content, and an element to be excluded from the content.

14. The method of claim 12, wherein the client device includes one or more of a desktop, a laptop, a notebook, an ultrabook, a smartphone, and a tablet computer.

15. The method of claim 12, further comprising providing security, wherein the security is based on one or more of the following: a location, a network, rules-based authentication, multiple-factor authentication, time-based authentication, a proxy associated with routing of the content, application data, screens associated with the content, and functionality associated with the content.

16. The method of claim 12, wherein the optimizing of the content includes one or more of the following: restructuring the content, injecting additional content, and removing at least a part of the content.

17. The method of claim 12, further comprising logging information related to one or more of the following: an identity of a user and actions performed under the user in the enterprise system, an identity of the data accessed, and a source of a location associated with the client device.

18. The method of claim 12, wherein the properties associated with the content include definition and structure of the enterprise system.
19. The method of claim 12, further comprising reading and caching a configuration of the client device on a web server.

20. A non-transitory processor-readable medium having instructions stored thereon, which when executed by one or more processors, cause the one or more processors to implement a method for dynamic optimization of content, the method comprising:
   receiving the content from a enterprise system, the enterprise system being pre-configured based on predetermined business rules;
   receiving parameters of a display area associated with the content;
   determining properties associated with the content;
   selectively optimizing the content, based on the parameters of display area and the properties associated with the content, thereby producing optimized content; and
   selectively rendering the optimized content on a client device, thereby providing for the selective optimization of the content without reconfiguring the enterprise system.
SYSTEM FOR DYNAMIC OPTIMIZATION OF CONTENT

COMMUNICATION MODULE 202
ANALYZING MODULE 204
OPTIMIZATION MODULE 206
RENDERING MODULE 208
SECURITY MODULE 210
LOGGING MODULE 212
EXTRACTING MODULE 214

FIG. 2
Start

Receive Request to Reconfigure Layout of Content 402

Determine Access Level 404

Process Request to Generate HTML Output 406

Process HTML Output to Generate Client Specific HTML Output 408

Transmit Client Specific HTML Output 410

Done

FIG. 4
Start

Receive Content from an Enterprise System 602

Receive Parameters of Display Area Associated with the Content 604

Analyze the Content to Determine Properties Associated with the Content 606

Selectively Optimize the Content, Based on the Parameters of Display Area and the Properties Associated with the Content 608

Selectively Render the Optimized Content on a Client Device 610

Done

FIG. 6
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 17/30 (2014.01)
CPC - G06F 17/30905 (2014.11)

B. FIELDS SEARCHED
Minimum documentation searched (classification system followed by classification symbols)
IPC(8) - G06F: 15/16, 17/00, 17/30, 21/00; G06Q 10/00, 10/06 (2014.01)
USPC - 705/7; 709/203; 217, 232; 715/234

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
CPC - G06Q 10/06; G06F 17/30905; H04L 67/02, 67/2823, 67/2842 (2014.11) (keyword delimited)

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>A</td>
<td>US 2012/01 10067 A1 (BIRSA et al) 03 May 2012 (03.05.2012) entire document</td>
<td>1-20</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C.

* Special categories of cited documents:
  "A" document defining the general state of the art which is not considered to be of particular relevance
  "E" earlier application or patent but published on or after the international filing date
  "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  "O" document referring to an oral disclosure, use, exhibition or other means
  "P" document published prior to the international filing date but later than the priority date claimed
  "V" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
  "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
  "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
  "&" document member of the same patent family

Date of the actual completion of the international search
22 December 2014

Date of mailing of the international search report
05 FEB 2015

Name and mailing address of the ISA/US
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450
Facsimile No. 571-273-3201

Authorized officer:
Blaine R. Copenhagen
PCT Helpdesk: 571-272-4390
PCT OSP: 571-272-7774

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