ABSTRACT

SYSTEM AND METHOD TO DESIGN ARTIFACTS ASSOCIATED WITH A WEB PAGE

A system, method and computer program product to design one or more artifacts associated with a web page are disclosed. A user is allowed to select values of the artifacts and to specify boundary conditions for an html pattern. Style pattern is identified based on the html pattern of the screen and the boundary conditions. Editing operations to be performed on the artifacts are controlled by controlling an activation and deactivation of the editing operations by referring to style specifications as specified by the user. Based on the style pattern, the values of the artifacts and the boundary conditions, the editing operations are performed on the artifacts and design of the web page is obtained.

To be published with figure 5

WE CLAIM:

1. A system to design artifacts associated with a web page, the system comprising:

   a processor;

   a user interface configured to allow a user to:

   customize the web page by selecting values of the artifacts; and

   specify boundary conditions for the design of the artifacts; and

   a memory coupled to the processor, wherein the processor is capable of executing a plurality of modules stored in the memory, and wherein the plurality of modules comprising:

   a screen modeling module configured to identify a style pattern based on an html pattern of the web page and the boundary conditions specified by the user;

   a control module configured to control one or more editing operations to be performed on the artifacts by referring to a style specification specified by the user, wherein the style specification is stored in a style modeling repository;

   an editing module configured to perform the one or more editing operations on the artifacts based on the values of the artifacts, the style pattern and the boundary conditions; and

   a generating module configured to render the design of the artifacts based upon the editing operations.

2. The system of claim 1, wherein the artifacts further comprises of text pattern, color, font, or a combination thereof.
3. The system of claim 1, wherein the boundary conditions may include a web control that is surrounded by a span/dive tag that additionally has one more label tag for its label placement or Form section having a header and the web controls.

4. The system of claim 1, wherein the control module controls the editing operation by controlling an activation and deactivation of the editing module.

5. The system of claim 1, wherein the editing operations are performed irrespective of a browser type by enabling a conversion of values of the artifacts.

6. The system of claim 1, wherein the editing module is configured to display a preview of the design of the web page with respect to the editing operations in real time.

7. A method to design artifacts associated with a web page, the method comprising:

allowing a user to customize the web page by selecting values of the artifacts and specify one or more boundary conditions for design of the artifacts;

identifying a style pattern based on html pattern of the web page and the boundary conditions specified by the user;

controlling one or more editing operations to be performed on the artifacts by referring to one or more style specifications specified by the user; wherein the style specifications are stored in a style modeling repository;

performing one or more editing operations on the artifacts based on the values of the one or more artifacts, the style pattern and boundary conditions as specified by the user; and

rendering the design of the artifacts based on editing operations so performed;

wherein the allowing, the identifying, the performing and the controlling are performed by means of a processor.

8. The method of claim 7, wherein the artifacts further comprises of text pattern, color, font, or a combination thereof.

9. The method of claim 7, wherein the boundary conditions further comprises of may include a web control that is surrounded by a span/dive tag that additionally has one more label tag for its label placement or Form section having a header and the web controls.

10. The method of claim7, wherein the performance of the editing operations are further controlled by activating and deactivating the editing operations.

11. The method of claim 7, wherein the editing operations are performed irrespective of a browser by enabling a conversion of values of the artifacts as specified by the user.

12. The method of claim 7, wherein the design of the web page is displayed with respect to the editing operations in real time.

13. A computer program product having embodied thereon a computer program to design artifacts associated with a web page, the method comprising:

a program code allowing a user to customize the web page by selecting values of the artifacts and specify one or more boundary conditions for design of the artifacts;

a program code identifying a style pattern based on html pattern of the web page and the boundary conditions specified by the user;

a program code controlling one or more editing operations to be performed on the artifacts by referring to one or more style specifications specified by the user; wherein the style specifications are stored in a style modeling repository;

a program code performing one or more editing operations on the artifacts based on the values of the one or more artifacts, the style pattern and boundary conditions as specified by the user;

and

a program code rendering the design of the artifacts based on editing operations so performed.

Dated this 05th Day of September, 2013

Deepak Pawar
Agent for Applicant
IN-PA-2052

FIELD OF THE INVENTON

[001] The present invention in general relates to a system and method for designing one or more artifacts. More particularly, the present invention relates to the system and method for designing of the artifacts associated with a web page in a web development environment.

BACKGROUND OF THE INVENTION

[002] In the development of web applications, many of the features are dependent on code structures. Most of the things are based on writing codes, modifying codes. These days, designing of web pages is getting more advance as lot of features may be customized as per a user's requirements. Moreover, a user may control designs of various artifacts and can modify them as per the requirement. However in most of the features while designing the web application has to be generalized code generation is involved and thus creates complexity.

[003] Conventional solutions to provide user-customized style specifications provide methodology to update code in order to achieve a user specific web application. However, these methods also involve certain drawbacks. Code updating bloat the code generators, it also conflict with the generic nature of code generation strategies leading to maintenance problems for product lines. Code updation process also needs some additional time to be exhausted in testing. Another big problem associated with web development is browser compatibility. These issues are also classified as scripting and styling issues.

[004] Therefore, there is a need to provide a solution capable of addressing the styling of web application without involving any code complexity and addressing difficult areas like reusability, maintainability & browser compatible.

SUMMARY OF THE INVENTION

[005] The present invention discloses a system to design artifacts associated with a web page. The system comprises of a processor, a user interface configured to allow a user to customize the web page by selecting portion of the artifacts and specify boundary conditions for the design of the artifacts and a memory coupled to the processor. The processor is capable of executing a plurality of modules stored in the memory. The plurality of modules comprises of a screen modeling module configured to identify a style pattern based on an html pattern of the web page and the boundary conditions specified by the user, a control module configured to control one or more editing operations to be performed on the artifacts by referring to a style specification specified by the user and an editing module configured to perform the one or more editing operations on the artifacts based on the values of the artifacts, the style pattern and the boundary conditions. The modules further comprises of a generating module configured to render the design of the artifacts based upon the editing operations.

[006] The present invention also discloses a method to design artifacts associated with a web page. The method comprises of allowing a user to customize the web page by selecting values of the artifacts and specify one or more boundary conditions for design of the artifacts, identifying a style pattern based on html pattern of the web page and the boundary conditions specified by the user and controlling one or more editing operations to be performed on the artifacts by referring to one or more style specifications specified by the user. The method further comprises of performing one or more editing operations on the artifacts based on the values of the one or more artifacts, the style pattern and boundary conditions as specified by the user and rendering the design of the artifacts based on editing operations so performed. The allowing, the identifying, the performing and the controlling are performed by means of a processor.

[007] The present invention also discloses a computer program product having embodied thereon a computer program to design artifacts associated with a web page. The computer program product comprises of a program code allowing a user to customize the web page by selecting values of the artifacts and specify one or more boundary conditions for design of the artifacts, a program code identifying a style pattern

based on html pattern of the web page and the boundary conditions specified by the user and a program code controlling one or more editing operations to be performed on the artifacts by referring to one or more style specifications specified by the user; wherein the style specifications are stored in a style modeling repository. The computer program product further comprises of a program code performing one or more editing operations on the artifacts based on the values of the one or more artifacts, the style pattern and boundary conditions as specified by the user and a program code rendering the design of the artifacts based on editing operations so performed.

BRIEF DESCRIPTION OF DRAWINGS

[008] Figure 1 illustrates to a network implementation of a system to design artifacts in accordance with an embodiment of the invention.

[009] Figure 2 illustrates the details of modules to design artifacts in accordance with an embodiment of the invention.

[0010] Figure 3 illustrates the step wise methodology to design artifacts in accordance with an embodiment of the invention.

[0011] Figure 4 illustrates the process flow of designing the web artifacts in accordance with an embodiment of the invention.

[0012] Figure 5 illustrates flow of data while designing the artifacts of web page in accordance with an embodiment of the invention.

DETAILED DESCRIPTION

[0013] System, method and computer program product to design artifacts associated with a web page are disclosed. A user may customize a web page by selecting one or more values of the artifacts. The user may also specify certain boundary conditions for the design of the artifacts. The boundary conditions and an html pattern of the web page are used to specify a style pattern. Later, the style pattern, the boundary conditions and the values of the artifacts are processed to execute one or more editing operations on the artifacts. The editing operations are also controlled by referring to a style specification as specified by the user.

[0014] While aspects of described system and method to design artifacts of a web page may be implemented in any number of different computing systems, environments, and/or configurations, the embodiments are described in the context of the following exemplary system.

[0015] Referring now to Figure 1, a network implementation 100 of system 102 to design artifacts of a web page is shown. A user may customize the web page as per his requirement by designing various artifacts of the web page. The input in a form of boundary conditions, values of artifacts is supplied to the system 102. Editing operations are performed over the artifacts and changes in design by way of a preview are displayed in real-time to the user. The design of the artifacts may be customized at any stage by the user.

[0016] Although the present subject matter is explained considering that the system 102 is implemented as an application on a server, it may be understood that the system 102 may also be implemented in a variety of computing systems, such as a laptop computer, a desktop computer, a notebook, a workstation, a mainframe computer, a server, a network server, and the like. It will be understood that the system 102 may be accessed by multiple users through one or more user devices 104-1, 104-2...104-N, collectively referred to as user 104 hereinafter, or applications residing on the user devices 104. Examples of the user devices 104 may include, but are not limited to, a portable computer, a personal digital assistant, a handheld device, and a workstation. The user devices 104 are communicatively coupled to the system 102 through a network 106.

[0017] In one implementation, the network 106 may be a wireless network, a wired network or a combination thereof. The network 106 can be implemented as one of the different types of networks, such as intranet, local area network (LAN), wide area network (WAN), the internet, and the like. The network 106 may either be a dedicated network or a shared network. The shared network represents an association of the different types of networks that use a variety of protocols, for example, Hypertext Transfer Protocol (HTTP), Transmission Control Protocol/Internet Protocol (TCP/IP), Wireless Application Protocol (WAP), and the like, to communicate with one another. Further the network 106 may include a variety of network devices, including routers, bridges, servers, computing devices, storage devices, and the like.

[0018] Referring now to Figure 2, the system 102 is illustrated in accordance with an embodiment of the present subject matter. In one embodiment, the system 102 may include at least one processor 202, an input/output (I/O) interface 204, and a memory 206. The at least one processor 202 may be implemented as one or more microprocessors, microcomputers, microcontrollers, digital signal processors, central processing units, state machines, logic circuitries, and/or any devices that manipulate signals based on operational instructions. Among other capabilities, the at least one processor 202 is configured to fetch and execute computer-readable instructions stored in the memory 206.

[0019] The I/O interface 204 may include a variety of software and hardware interfaces, for example, a web interface, a graphical user interface, and the like. The I/O interface 204 may allow the system 102 to interact with a user directly or through the client devices 104. Further, the I/O interface 204 may enable the system 102 to communicate with other computing devices, such as web servers and external data servers (not shown). The I/O interface 204 can facilitate multiple communications within a wide variety of networks and protocol types, including wired networks, for example, LAN, cable, etc., and wireless networks, such as WLAN, cellular, or satellite. The I/O interface 204 may include one or more ports for connecting a number of devices to one another or to another server.

[0020] The memory 206 may include any computer-readable medium known in the art including, for example, volatile memory, such as static random access memory (SRAM) and dynamic random access memory (DRAM), and/or non-volatile memory, such as read only memory (ROM), erasable programmable ROM, flash memories, hard disks, optical disks, and magnetic tapes. The memory 206 may include modules 208 and data 210.

[0021] The modules 208 include routines, programs, objects, components, data structures, etc., which perform particular tasks or implement particular abstract data types. In one implementation, the modules 208 may include a screen modeling module

212, a control module 214, an editing module 216, and a generating module 218 and other modules 219. The other modules 219 may include programs or coded instructions that supplement applications and functions of the system 102.

[0022] The data 210, amongst other things, serves as a repository for storing data processed, received, and generated by one or more of the modules 208. The data 210 may also include a database 222, a style modeling repository 214 and other data 130. The other data 130 may include data generated as a result of the execution of one or more modules in the other module 219.

[0023] Referring to figure 4 and 5 in combination, the system 102 is implemented as a Cascading Style Sheet (CSS) by using an MVC (Model View Controller) framework. The user interface 204 is configured to allow a user to select values of artifacts. In order to change or design the artifacts to further design the web page or style specifications of the web page, the user may press key up and down to swap between the available values of the artifacts. By means of the user interface 204, the user may also provide or customize one or more boundary conditions of the artifacts to design the web page. The boundary conditions enable the designing of the web page according to an html pattern of screen. The user designs or changes the artifacts in order to design or customize the web page with respect to his requirement.

[0024] The artifacts further comprises of a text pattern, color, font, Background image, background color , borders, margins & padding height, width etc.

[0025] The screen modeling module 212 identifies a style pattern based on an html pattern of the web page and boundary conditions as specified by the user. These boundary conditions are specified as a boundary listener (215). The artifacts may be designed only for specific html patterns, therefore, once the html pattern is identified, the user may specify the boundary conditions and may work on any html patterns. The system 102 provides may boundary conditions like text box, dropdown etc. The boundary conditions may include a web control that is surrounded by span/dive tag that additionally has one more label tag for its label placement or Form section having a header and the web controls in it. So for these, user can define some those as boundaries for the editing module 216 activation. When user clinks on these portions then only editing module 216 will be activated otherwise editing module will not be shown at all.

[0026] The control module 214 is configured to control editing operations to be performed on the artifacts. The control module refers to a style specification specified by the user. The style specifications are stored in a style modeling repository 214. The editing operations are controlled by controlling an activation and deactivation process of the editing operations.

[0027] The screen modeling module 212 also assists in the functioning of the editing module 216. The user has to define or specify boundary listener 215 by providing boundary conditions based on the html pattern on a screen. The editing module understands the patterns based on the html pattern and the boundary conditions and will start its editing operations.

[0028] The editing module 216 may be activated on a particular section of the screen. The editing module 216 may be presented as small sections based on the html pattern of the screen. The presence of the editing module 216 on the various sections of the screen may also be specified by the user by specifying the boundary conditions.

[0029] The editing module 216 is a responsive editor. When the user changes the style specifications by giving values of the artifacts and boundary conditions, then the editing module 216 generates a preview in real time to display the changes accordingly on the screen.

[0030] The editing module 216 also supports the browser compatibility. When a user specifies one kind of style (values for artifacts) which is specific to the editing module 216 of that particular browser, the editing module 216 will ensure these changes may work on any browser 213. The browser compatibility is supported by Custom jquery plugin (Javascript Library). The editing module 216 may address browser compatibility when user specifies the style changes. Editing module itself may store the specification in terms of all the browser related specifications so that they may be generated according to target browsers in the Generating module 218. This will be addressed in the editing module 216 as all the browser specifications are injected while designing the editor.

[0031] The generating module 218 renders the design of the artifacts as an output based on the editing operations performed by the editing module 216. The output may be obtained in a form of a CSS file and may be modified as per the user's requirements.

[0032] Still referring to figure 4, by way of specific example, the user may design a wireframe of the screen by using the system 102. This is performed by means of various modules. These modules may also be implemented in terms of various models on the screen. The system 102 may translate language to provide default look and feel of the screen (step 400). The boundary conditions are provided as an input to the editing module 216. For example, the editing module or editor has to be activated only on a specific html pattern then user has a flexibility of specifying the boundary conditions of the artifacts for that html pattern. The system 102 provides various controls like text box, dropdown etc in order to enable all control boundaries so that the system 102 may work on any html pattern.

[0033] The user may decorate (step 402 and 404) the screen by using the editing module 216. We may also refer to this as actual decoration section or model. This will be the location of the screen where user may start his decoration. A start/stop switch may also be provided in order to activate or deactivate the editing operations of the editing module 216. The start/stop switch (which is the control module 214) activates the editing module based on the boundary conditions and the style specifications as specified by the user.

[0034] As a part of an output (step 406), the generating module may provide an output format. Based on the output specification translator's may be written to convert the style specification to required format. The specification of the output may be specified in the Generating module 218, this specification may be according to css specifications of a js(java script) object where some js code may understand and interpret. An example may be if a html pattern is having a div with a text box then div will have some class with defined some styling. As it is uniquely identified from its textbox the style may be specified as a relative position to the text box and store in the screen modeling repository 224. As the pattern is specified in the boundary condition this will automatically understood and show the specified style specification

[0035] Once the decorator style is converted to required output format, the screen may start rendering with its new style specifications. This may work on all the browsers with multi-skin support (step 408).

[0036] The order in which the method 300 is described is not intended to be construed as a limitation, and any number of the described method blocks can be combined in any order to implement the method 300 or alternate methods. Additionally, individual blocks may be deleted from the method 300 without departing from the spirit and scope of the subject matter described herein. Furthermore, the method may be implemented in any suitable hardware, software, firmware, or combination thereof. However, for ease of explanation, in the embodiments described below, the method 300 may be considered to be implemented in the above described system 102.

[0037] At block 302, a user is allowed to customize the web page by selecting values of the artifacts and by specifying boundary conditions of the artifacts for an html pattern.

[0038] At block 304, style pattern based on the html pattern of the web page and the boundary conditions is specified.

[0039] At block 306, editing operations to be performed on the artifacts are controlled by referring to one or more style specifications as specified by the user.

[0040] At block 308, editing operations are performed based on the values of the artifacts, the style pattern and the boundary conditions.

[0041] At block 310, design of the artifacts is obtained based on the editing operations so performed.

[0042] The written description describes the subject matter herein to enable any person skilled in the art to make and use the embodiments of the invention. The scope of the subject matter embodiments is defined by the claims and may include other modifications that occur to those skilled in the art. Such other modifications are intended to be within the scope of the claims if they have similar elements that do not differ from the literal language of the claims or if they include equivalent elements with insubstantial differences from the literal language of the claims.

WE CLAIM:

1. A system to design artifacts associated with a web page, the system comprising:

    a processor;

    a user interface configured to allow a user to:

        customize the web page by selecting values of the artifacts; and

        specify boundary conditions for the design of the artifacts; and

    a memory coupled to the processor, wherein the processor is capable of executing a plurality of modules stored in the memory, and wherein the plurality of modules comprising:

        a screen modeling module configured to identify a style pattern based on an html pattern of the web page and the boundary conditions specified by the user;

        a control module configured to control one or more editing operations to be performed on the artifacts by referring to a style specification specified by the user, wherein the style specification is stored in a style modeling repository;

        an editing module configured to perform the one or more editing operations on the artifacts based on the values of the artifacts, the style pattern and the boundary conditions; and

        a generating module configured to render the design of the artifacts based upon the editing operations.

2. The system of claim 1, wherein the artifacts further comprises of text pattern, color, font, or a combination thereof.
3. The system of claim 1, wherein the boundary conditions may include a web control that is surrounded by a span/dive tag that additionally has one more label tag for its label placement or Form section having a header and the web controls.

4. The system of claim 1, wherein the control module controls the editing operation by controlling an activation and deactivation of the editing module.

5. The system of claim 1, wherein the editing operations are performed irrespective of a browser type by enabling a conversion of values of the artifacts.

6. The system of claim 1, wherein the editing module is configured to display a preview of the design of the web page with respect to the editing operations in real time.

7. A method to design artifacts associated with a web page, the method comprising:

allowing a user to customize the web page by selecting values of the artifacts and specify one or more boundary conditions for design of the artifacts;

identifying a style pattern based on html pattern of the web page and the boundary conditions specified by the user;

controlling one or more editing operations to be performed on the artifacts by referring to one or more style specifications specified by the user; wherein the style specifications are stored in a style modeling repository;

performing one or more editing operations on the artifacts based on the values of the one or more artifacts, the style pattern and boundary conditions as specified by the user; and

rendering the design of the artifacts based on editing operations so performed;

wherein the allowing, the identifying, the performing and the controlling are performed by means of a processor.

8. The method of claim 7, wherein the artifacts further comprises of text pattern, color, font, or a combination thereof.

9. The method of claim 7, wherein the boundary conditions further comprises of may include a web control that is surrounded by a span/dive tag that additionally has one more label tag for its label placement or Form section having a header and the web controls.

10. The method of claim7, wherein the performance of the editing operations are further controlled by activating and deactivating the editing operations.

11. The method of claim 7, wherein the editing operations are performed irrespective of a browser by enabling a conversion of values of the artifacts as specified by the user.

12. The method of claim 7, wherein the design of the web page is displayed with respect to the editing operations in real time.

13. A computer program product having embodied thereon a computer program to design artifacts associated with a web page, the method comprising:

a program code allowing a user to customize the web page by selecting values of the artifacts and specify one or more boundary conditions for design of the artifacts;

a program code identifying a style pattern based on html pattern of the web page and the boundary conditions specified by the user;

a program code controlling one or more editing operations to be performed on the artifacts by referring to one or more style specifications specified by the user; wherein the style specifications are stored in a style modeling repository;

a program code performing one or more editing operations on the artifacts based on the values of the one or more artifacts, the style pattern and boundary conditions as specified by the user;

and

a program code rendering the design of the artifacts based on editing operations so performed.

Dated this 05th Day of September, 2013

Deepak Pawar
Agent for Applicant
IN-PA-2052