

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6142620号  
(P6142620)

(45) 発行日 平成29年6月7日(2017.6.7)

(24) 登録日 平成29年5月19日(2017.5.19)

(51) Int.Cl. F 1  
G 0 6 F 3/048 (2013.01) G 0 6 F 3/048

請求項の数 6 (全 31 頁)

(21) 出願番号	特願2013-68952 (P2013-68952)	(73) 特許権者	000005223 富士通株式会社
(22) 出願日	平成25年3月28日 (2013. 3. 28)		神奈川県川崎市中原区上小田中4丁目1番 1号
(65) 公開番号	特開2014-191756 (P2014-191756A)	(74) 代理人	100092978 弁理士 真田 有
(43) 公開日	平成26年10月6日 (2014. 10. 6)	(74) 代理人	100112678 弁理士 山本 雅久
審査請求日	平成27年12月4日 (2015. 12. 4)	(72) 発明者	大木 憲二 神奈川県川崎市中原区上小田中4丁目1番 1号 富士通株式会社内
		審査官	梅岡 信幸

最終頁に続く

(54) 【発明の名称】 表示変更プログラム、表示変更方法及び表示変更装置

(57) 【特許請求の範囲】

【請求項1】

コンピュータに、

前記コンピュータである端末又は前記端末に備えられるブラウザの表示画面幅を条件として規定された複数のコンテンツ画面のスタイルのうち、前記端末又はブラウザの表示画面幅の条件を満たす条件充足スタイルを適用して前記端末にコンテンツ画面が表示された状態で、前記端末に備えられる取得部が、前記端末がコンテンツ画面のスタイル変更指示を受けた場合に、前記複数のコンテンツ画面のスタイルのそれぞれに対する適用条件を書き換えるための書換情報を取得し、前記端末に備えられる書換部が、前記書換情報に基づいて、前記複数のコンテンツ画面のスタイルのうち、少なくとも一つの条件充足スタイルに対する適用条件を「偽」に書き換える、又は、端末又はブラウザの表示画面幅の条件を満たさない少なくとも一つの条件非充足スタイルに対する適用条件を「真」に書き換える、

処理を実行させることを特徴とする表示変更プログラム。

【請求項2】

前記端末がスタイル保存指示を受けた場合に、前記端末に備えられる保存処理部が、その時に適用されている書換情報とWebサイトのアドレスとを対応づけて前記端末に備えられる記憶部に格納する処理を前記コンピュータに実行させることを特徴とする、請求項1に記載の表示変更プログラム。

【請求項3】

10

20

Webサイトのアドレスが前記記憶部に格納されていた場合に、前記取得部が、前記記憶部から前記Webサイトのアドレスに基づいて前記書換情報を取得し、前記書換部が、前記書換情報に基づいて、前記複数のコンテンツ画面のスタイルのうち、少なくとも一つの条件充足スタイルに対する適用条件を「偽」に書き換える、又は、前記端末又はブラウザの表示画面幅の条件を満たさない少なくとも一つの条件非充足スタイルに対する適用条件を「真」に書き換える処理を前記コンピュータに実行させることを特徴とする、請求項2に記載の表示変更プログラム。

【請求項4】

前記端末に備えられる描画部が、前記適用条件が「真」に書き換えられたスタイルを適用してコンテンツ画面を表示させる処理を前記コンピュータに実行させることを特徴とする、請求項1～3のいずれか1項に記載の表示変更プログラム。

10

【請求項5】

コンピュータが、

前記コンピュータである端末又は前記端末に備えられるブラウザの表示画面幅を条件として規定された複数のコンテンツ画面のスタイルのうち、前記端末又はブラウザの表示画面幅の条件を満たす条件充足スタイルを適用して前記端末にコンテンツ画面が表示された状態で、前記端末に備えられる取得部が、前記端末がコンテンツ画面のスタイル変更指示を受けた場合に、前記複数のコンテンツ画面のスタイルのそれぞれに対する適用条件を書き換えるための書換情報を取得し、前記端末に備えられる書換部が、前記書換情報に基づいて、前記複数のコンテンツ画面のスタイルのうち、少なくとも一つの条件充足スタイルに対する適用条件を「偽」に書き換える、又は、端末又はブラウザの表示画面幅の条件を満たさない少なくとも一つの条件非充足スタイルに対する適用条件を「真」に書き換える、

20

処理を実行することを特徴とする表示変更方法。

【請求項6】

端末又はブラウザの表示画面幅を条件として規定された複数のコンテンツ画面のスタイルのうち、端末又はブラウザの表示画面幅の条件を満たす条件充足スタイルを適用してコンテンツ画面が表示された状態で、コンテンツ画面のスタイル変更指示を受けた場合に、前記複数のコンテンツ画面のスタイルのそれぞれに対する適用条件を書き換えるための書換情報を取得する取得部と、

30

前記書換情報に基づいて、前記複数のコンテンツ画面のスタイルのうち、少なくとも一つの条件充足スタイルに対する適用条件を「偽」に書き換える、又は、端末又はブラウザの表示画面幅の条件を満たさない少なくとも一つの条件非充足スタイルに対する適用条件を「真」に書き換える書換部とを備えることを特徴とする表示変更装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、表示変更プログラム、表示変更方法及び表示変更装置に関する。

【背景技術】

【0002】

従来、マルチデバイス対応のWebサイトの開発を行なう際には、サーバサイドで、スマートフォン、タブレット、パーソナルコンピュータ(PC: personal computer)などの各端末向けに、それぞれ異なる固定レイアウトのHTML(HyperText Markup Language)などのコンテンツ(リソース)を用意し、ユーザエージェントを見て、各端末に適したコンテンツを提供する方法が採られていた。これをSSD(Server-Side Detection)という。

40

【0003】

近年、端末の種類が急速に増加してきており、全ての端末のそれぞれに最適化することが開発、運用の両面で困難になってきている。

そこで、サーバサイドからは同一のコンテンツを提供し、それぞれの端末側で、表示画

50

面幅に基づいてコンテンツのスタイル（レイアウトやデザイン）を動的に変更する方法が採られるようになってきている。これをレスポンシブWebデザイン（RWD：Responsive Web Design）という。

【先行技術文献】

【特許文献】

【0004】

【特許文献1】特開2004-110427号公報

【特許文献2】特開2010-152797号公報

【特許文献3】特開2010-191739号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

ところで、上述のレスポンシブWebデザインでは、コンテンツ開発者が、スマートフォン向け、タブレット向け、PC向けなど数種類のスタイル（スタイルシート）を用意し、かつ、それぞれのスタイルを適用する条件として表示画面幅を設定する。そして、それぞれの端末側で、表示画面幅の条件を満たすスタイルを適用してコンテンツ画面を表示させる。

【0006】

しかしながら、コンテンツ開発者がそれぞれのスタイルを適用する条件として設定した表示画面幅が実態にそぐわない場合がある。例えば、既存のスマートフォンよりも幅の広いスマートフォンが新しく登場した場合に、タブレットと認識されてしまい、タブレット向けのスタイルが適用されてしまうことがある。また、例えば、タブレットにおいて標準ブラウザ以外の独自のブラウザを使用する場合に、インラインフレームの使用などで他のUI（user interface）コンポーネントに幅を取られ、表示画面幅が狭くなり、スマートフォンと認識されてしまい、スマートフォン向けのスタイルが適用されてしまうことがある。

【0007】

特に、コンテンツ開発者は、既存の端末又はブラウザを想定して、それぞれのスタイルを適用する条件として表示画面幅を設定することは可能であるが、コンテンツ開発後に続々と登場する新しい端末又はブラウザに対応するのは困難であり、条件として設定された表示画面幅が実態にそぐわないことが起こりうる。

また、端末が表示画面幅の条件を満たすスタイルを適用するようになっているため、条件として設定された表示画面幅が実態にそぐわず、利用者がそのスタイルが適していないと感じる場合であっても、表示画面幅の条件を満たすスタイル以外のスタイル、即ち、表示画面幅の条件を満たさないスタイルに変更することはできない。

【0008】

そこで、条件として設定された表示画面幅が実態にそぐわず、利用者がそのスタイルが適していないと感じる場合などに、表示画面幅の条件を満たさないスタイルに変更できるようにすることを目的とする。

【課題を解決するための手段】

【0009】

本表示変更プログラムは、コンピュータに、コンピュータである端末又は端末に備えられるブラウザの表示画面幅を条件として規定された複数のコンテンツ画面のスタイルのうち、端末又はブラウザの表示画面幅の条件を満たす条件充足スタイルを適用して端末にコンテンツ画面が表示された状態で、端末に備えられる取得部が、端末がコンテンツ画面のスタイル変更指示を受けた場合に、複数のコンテンツ画面のスタイルのそれぞれに対する適用条件を書き換えるための書換情報を取得し、端末に備えられる書換部が、書換情報に基づいて、複数のコンテンツ画面のスタイルのうち、少なくとも一つの条件充足スタイルに対する適用条件を「偽」に書き換える、又は、端末又はブラウザの表示画面幅の条件を満たさない少なくとも一つの条件非充足スタイルに対する適用条件を「真」に書き換える

10

20

30

40

50

、処理を実行させることを要件とする。

【0010】

本表示変更方法は、コンピュータが、コンピュータである端末又は端末に備えられるブラウザの表示画面幅を条件として規定された複数のコンテンツ画面のスタイルのうち、端末又はブラウザの表示画面幅の条件を満たす条件充足スタイルを適用して端末にコンテンツ画面が表示された状態で、端末に備えられる取得部が、端末がコンテンツ画面のスタイル変更指示を受けた場合に、複数のコンテンツ画面のスタイルのそれぞれに対する適用条件を書き換えるための書換情報を取得し、端末に備えられる書換部が、書換情報に基づいて、複数のコンテンツ画面のスタイルのうち、少なくとも一つの条件充足スタイルに対する適用条件を「偽」に書き換える、又は、端末又はブラウザの表示画面幅の条件を満たさない少なくとも一つの条件非充足スタイルに対する適用条件を「真」に書き換える、処理を実行することを要件とする。

10

【0011】

本表示変更装置は、端末又はブラウザの表示画面幅を条件として規定された複数のコンテンツ画面のスタイルのうち、端末又はブラウザの表示画面幅の条件を満たす条件充足スタイルを適用してコンテンツ画面が表示された状態で、コンテンツ画面のスタイル変更指示を受けた場合に、複数のコンテンツ画面のスタイルのそれぞれに対する適用条件を書き換えるための書換情報を取得する取得部と、書換情報に基づいて、複数のコンテンツ画面のスタイルのうち、少なくとも一つの条件充足スタイルに対する適用条件を「偽」に書き換える、又は、端末又はブラウザの表示画面幅の条件を満たさない少なくとも一つの条件非充足スタイルに対する適用条件を「真」に書き換える書換部とを備えることを要件とする。

20

【発明の効果】

【0012】

したがって、表示画面幅の条件を満たさないスタイルに変更できるという利点がある。

【図面の簡単な説明】

【0013】

【図1】本実施形態にかかる表示変更装置を備える端末の構成を示す図である。

【図2】本実施形態にかかる表示変更装置を備える端末におけるスタイル変更の概要を示す模式図である。

30

【図3】(A)～(C)は、本実施形態にかかる表示変更装置を備える端末が取得するコンテンツに対して用意されているスタイルを示す図であって、(A)はスマートフォン向けのスタイルを示しており、(B)はタブレット向けのスタイルを示しており、(C)はPC向けのスタイルを示している。

【図4】本実施形態にかかる表示変更装置を備える端末が取得するコンテンツのCSSファイルに含まれるメディアクエリを示す図である。

【図5】(A)～(C)は、本実施形態にかかる表示変更装置を備える端末におけるスタイル変更方法を説明するための模式図である。

【図6】(A)、(B)は、本実施形態にかかる表示変更装置を備える端末におけるスタイル保存方法を説明するための模式図である。

40

【図7】本実施形態にかかる表示変更装置を備える端末のハードウェア構成を示す図である。

【図8】本実施形態にかかる表示変更装置を備える端末における画面表示処理を説明するためのフローチャートである。

【図9】本実施形態にかかる表示変更装置を備える端末におけるルール抽出処理を説明するためのフローチャートである。

【図10】(A)、(B)は、本実施形態にかかる表示変更装置を備える端末によるルール抽出処理におけるルールに含まれる条件の取得方法を説明するための図である。

【図11】本実施形態にかかる表示変更装置を備える端末のルール抽出処理における「media」、「min-width」、「max-width」の抽出方法を説明するための図である。

50

【図12】本実施形態にかかる表示変更装置を備える端末のルール抽出処理において作成されるルールテーブルを示す図である。

【図13】本実施形態にかかる表示変更装置を備える端末における書換情報生成処理を説明するためのフローチャートである。

【図14】本実施形態にかかる表示変更装置を備える端末における書換情報生成処理によって作成される書換情報テーブルを示す図である。

【図15】本実施形態にかかる表示変更装置を備える端末が取得するコンテンツのCSSファイルに含まれるメディアクエリの他の例を示す図である。

【図16】本実施形態にかかる表示変更装置を備える端末のルール抽出処理において作成されるルールテーブルの他の例を示す図である。

10

【図17】本実施形態にかかる表示変更装置を備える端末における書換情報生成処理によって作成される書換情報テーブルの他の例を示す図である。

【図18】本実施形態にかかる表示変更装置を備える端末におけるルール書換処理を説明するためのフローチャートである。

【図19】(A)、(B)は、本実施形態にかかる表示変更装置を備える端末におけるルール書換処理によってどのようにルールが書き換えられるかを説明するための図であって、(A)は書き換え前にどのようになっているかを示しており、(B)は、書き換え後にどのようになるかを示している。

【図20】本実施形態にかかる表示変更装置を備える端末によるルール書換処理におけるルールの書換方法を説明するための図である。

20

【図21】本実施形態にかかる表示変更装置を備える端末における画面変更表示処理を説明するためのフローチャートである。

【図22】本実施形態にかかる表示変更装置を備える端末におけるスタイル変更処理の一例を説明するためのフローチャートである。

【図23】本実施形態にかかる表示変更装置を備える端末におけるスタイル変更処理の他の例を説明するためのフローチャートである。

【図24】本実施形態にかかる表示変更装置を備える端末における保存処理を説明するためのフローチャートである。

【図25】本実施形態にかかる表示変更装置を備える端末による保存処理における保存方法を説明するための図である。

30

【図26】本実施形態にかかる表示変更装置を備える端末における保存処理によって作成される再アクセス時書換情報テーブルを示す図である。

【図27】本実施形態にかかる表示変更装置を備える端末における再アクセス時書換処理を説明するためのフローチャートである。

【図28】本実施形態にかかる表示変更装置の端末への実装方法を説明するための図である。

【図29】本実施形態にかかる表示変更装置の端末への実装方法を説明するための図である。

【図30】本実施形態にかかる表示変更装置の端末への他の実装方法を説明するための図である。

40

【発明を実施するための形態】

【0014】

以下、図面により、本発明の実施の形態にかかる表示変更プログラム、表示変更方法及び表示変更装置について、図1～図30を参照しながら説明する。

本実施形態にかかる表示変更装置は、図2に示すように、例えばスマートフォン、タブレット、PCなどの端末1に備えられ、例えばHTMLファイル及びメディアクエリ(Media Query)を含むCSS(Cascading Style Sheet)ファイルからなるレスポンシブWebデザイン対応のコンテンツを提供するWebサイト(Webサーバ)70からコンテンツを取得する場合に、条件として設定された表示画面幅が実態にそぐわず、利用者がメディアクエリにしたがって適用されたスタイル(レイアウト)が適していないと感じたとき

50

に、端末1を操作して、動的にスタイルを切り替えられるようにするものである。つまり、コンテンツ開発時ではなく、利用者の閲覧時に、適したスタイルを選択できるようにするものである。なお、コンテンツを、Webコンテンツともいう。また、コンテンツには、例えばWebアプリケーションが提供するものも含まれる。

#### 【0015】

本実施形態では、レスポンスWebデザイン対応のコンテンツは、例えば、HTMLファイルとCSSファイルとからなり、CSSファイルが、CSS3で仕様として規定された、表示画面幅などの条件によって適用するスタイル(スタイルシート)を切り替えるために用いられるメディアクエリを含んでいる。このメディアクエリでは、表示画面幅を条件として複数のスタイルが規定されている。つまり、メディアクエリでは、スマートフォン向け、タブレット向け、PC向けなど数種類のスタイル(スタイルシート)が、それぞれのスタイルを適用する条件としての表示画面幅に対応づけて規定されている。このメディアクエリは、ルール(Rule)の集合( $R_0, \dots, R_n$ )として構成される。これはJavaScript(登録商標)オブジェクトとして取得可能である。また、メディアクエリの各ルールは、「真」(True)又は「偽」(False)のいずれかに評価される式であって、メディアタイプや表示画面幅などの条件(式)、スタイル(スタイルルール; CSS; CSSルール)を含む。そして、このようなレスポンスWebデザイン対応のコンテンツを取得した端末に備えられているブラウザは、メディアクエリの各ルールの真偽(True or False)を評価し、「真」(True)と評価したスタイルを適用し、「偽」(False)と評価したスタイルは適用しないで、コンテンツ画面を表示させる。つまり、レスポンスWebデザイン対応のコンテンツを取得した端末に備えられているブラウザは、表示画面幅の条件を満たすスタイルを適用して、コンテンツ画面を表示させる。なお、ブラウザをWebブラウザともいう。また、コンテンツ画面をWebページともいう。

#### 【0016】

例えば図3に示すように、スマートフォン向けのスタイルS0、タブレット向けのスタイルS1、PC向けのスタイルS2の3つのスタイルを用意しているレスポンスWebデザイン対応のコンテンツでは、CSSファイルが、例えば図4に示すようなメディアクエリを含む。

ここでは、メディアクエリは、@media以降に記述されているルールの集合、即ち、スマートフォン向けのスタイルS0に対するルール、タブレット向けのスタイルS1に対するルール、PC向けのスタイルS2に対するルールの集合として構成されている。なお、ここでは、@media規則でメディアクエリを記述する場合を例に挙げて説明しているが、@import規則やHTML文書でのlink要素などで用いられるmedia属性値などの他のメディア指定できる箇所でメディアクエリを記述しても良い。

#### 【0017】

このうち、スマートフォン向けのスタイルS0に対するルールは、メディアタイプとして「screen」、条件として「(min-width:0px)and(max-width:320px)」、スタイルとして「スマートフォン向けCSS」を含む。このルールは、ブラウザに対し、ブラウザの表示画面幅が0ピクセル以上320ピクセル以下の場合に、スマートフォン向けCSSを適用するように指示する。また、タブレット向けのスタイルS1に対するルールは、メディアタイプとして「screen」、条件として「(min-width:321px)and(max-width:768px)」、スタイルとして「タブレット向けCSS」を含む。このルールは、ブラウザに対し、ブラウザの表示画面幅が321ピクセル以上768ピクセル以下の場合に、タブレット向けCSSを適用するように指示する。また、PC向けのスタイルS2に対するルールは、メディアタイプとして「screen」、条件として「(min-width:769px)」、スタイルとして「PC向けCSS」を含む。このルールは、ブラウザに対し、ブラウザの表示画面幅が769ピクセル以上の場合に、PC向けCSSを適用するように指示する。この例では、コンテンツ開発者は、0ピクセル以上320ピクセル以下の表示画面幅を持つものをスマートフォンと想定し、321ピクセル以上768ピクセル以下の表示画面幅を持つものをタブレットと想定し、769ピクセル以上の表示画面幅を持つものをPCと想定し、320ピクセ

10

20

30

40

50

ル、768ピクセルを、それぞれ、ブレイクポイント(閾値)として、スタイルを切り替えるようにしている。なお、ブラウザの表示画面を、ブラウザの表示領域又はブラウザのウィンドウともいう。また、スマートフォンやタブレットなどの端末では、ブラウザ(モバイルブラウザ)の表示画面のサイズが固定であり、任意にリサイズすることができないため、ブラウザの表示画面幅と端末の画面幅とは一致する。また、ここでは、ブラウザの表示画面幅に基づいてスタイルを切り替えるようにしているが、ブラウザの表示画面幅の代わりに端末の画面幅を用いても良い。

【0018】

そして、例えば、このようなレスポンシブWebデザイン対応のコンテンツを取得した端末に備えられているブラウザの表示画面幅が330ピクセルであるとする、これは321ピクセル以上768ピクセル以下の条件を満たす。このため、ブラウザは、タブレット向けCSSを適用して、コンテンツ画面を表示させる。つまり、ブラウザは、タブレット向けのスタイルS1に対するルールを「真」(True)と評価し、スマートフォン向けのスタイルS0に対するルール及びPC向けのスタイルS2に対するルールを「偽」(False)と評価し、「真」(True)と評価したタブレット向けのスタイルS1を適用して、コンテンツ画面を表示させる。

10

【0019】

しかしながら、コンテンツ開発者がそれぞれのスタイルを適用する条件として設定した表示画面幅が実態にそぐわず、利用者が上述のようにして表示したコンテンツ画面のスタイルが適していないと感じる場合がある。

20

特に、コンテンツ開発者は、既存の端末又はブラウザを想定して、それぞれのスタイルを適用する条件として表示画面幅を設定することは可能であるが、コンテンツ開発後に続々と登場する新しい端末又はブラウザに対応するのは困難であり、条件として設定された表示画面幅が実態にそぐわず、利用者がそのスタイルが適していないと感じる場合がある。

【0020】

また、端末が表示画面幅の条件を満たすスタイルを適用するようになっているため、条件として設定された表示画面幅が実態にそぐわず、利用者がそのスタイルが適していないと感じる場合であっても、表示画面幅の条件を満たすスタイル以外のスタイル、即ち、表示画面幅の条件を満たさないスタイルに変更することはできない。

30

例えば、スマートフォンやタブレットなどの端末では、ブラウザの表示画面幅(又は端末の画面幅)に基づいて、表示画面幅の条件を満たすスタイルが一義的に適用されてしまう。このため、利用者がそのスタイルが適していないと感じる場合であっても、表示画面幅の条件を満たすスタイル以外のスタイル、即ち、表示画面幅の条件を満たさないスタイルに変更することはできない。

【0021】

このようなスマートフォンやタブレットなどの端末では、利用者は、ブラウザの表示画面のサイズ(ウィンドウサイズ)を任意に変えることができないのに対し、PCなどの端末では、利用者は、ブラウザの表示画面のサイズ(ウィンドウサイズ)を任意に変えることができる。このため、利用者がブラウザの表示画面のサイズを変えた場合に、適用されるスタイルが切り替わるようにすることも可能である。つまり、利用者が、ブラウザの表示画面のサイズを端末の画面サイズに一致させている場合には、PC向けのスタイルが適用され、ブラウザの表示画面のサイズを端末の画面サイズよりも小さくした場合には、その程度に応じて、タブレット向けのスタイルやスマートフォン向けのスタイルが適用されるようにすることも可能である。しかしながら、このような場合も、ブラウザの表示画面幅の条件を満たすスタイルが適用されるにすぎず、利用者がそのスタイルが適していないと感じる場合に、表示画面幅の条件を満たすスタイル以外のスタイル、即ち、表示画面幅の条件を満たさないスタイルに変更することはできない。

40

【0022】

そこで、本実施形態では、条件として設定された表示画面幅が実態にそぐわず、利用者

50

がメディアクエリにしたがって適用されたスタイルが適していないと感じる場合に、端末1を操作して、他のスタイル、即ち、表示画面幅の条件を満たさないスタイルに変更できるようにしている。

ここでは、図5(A)に示すように、端末画面1A上に上述のようにしてコンテンツ画面が表示された状態で、例えばフリック(画面を指ではじく動作)などの利用者からの指示を受けた場合、図5(B)に示すように、例えば「Change」ボタンや「Save」ボタンを含むダイアログボックス1Bを表示させる。そして、利用者が現在のスタイルが適していないと感じ、ダイアログボックス1B内の「Change」ボタンを押した場合、即ち、スタイル変更指示を受けた場合、図5(C)に示すように、他のスタイル、即ち、表示画面幅の条件を満たさないスタイルに切り替えるようにしている。ここでは、上述のようにして、表示画面幅の条件を満たすスタイルとして、タブレット向けのスタイルS1が適用されたコンテンツ画面から、例えばPC向けのスタイルS2が適用されたコンテンツ画面(スタイル変更画面)に切り替えるようにしている。

10

#### 【0023】

そして、再度、利用者が「Change」ボタンを押した場合、PC向けのスタイルS2が適用されたコンテンツ画面から、例えばスマートフォン向けのスタイルS0が適用されたコンテンツ画面(スタイル変更画面)に切り替えるようにしている。

このように、利用者が「Change」ボタンを押すたびに、PC向けのスタイルS2、スマートフォン向けのスタイルS0、タブレット向けのスタイルS1を順に適用し、スタイルを切り替えるようにしている。

20

#### 【0024】

なお、ここでは、ボタン押下をスタイル切替イベント(スタイル変更イベント)として用いているが、これに限られるものではなく、端末1のブラウザが認識できるイベントを用いれば良い。例えば、タップやフリックなどを用いても良いし、端末1の加速度変更時に呼ばれるdevicemotionイベントをトリガとして端末1を振る動作でスタイルを切り替えるようにしても良い。

#### 【0025】

また、本実施形態では、利用者が、図6(A)に示すように、端末画面1A上に例えばPC向けのスタイルS2が適用されたコンテンツ画面が表示された状態で、図6(B)に示すように、ダイアログボックス1Bを表示させ、ダイアログボックス1B内の「Save」ボタンを押した場合、即ち、スタイル保存指示を受けた場合、その時に適用されているスタイルにするために必要な情報(後述の書換情報)とWebサイトのアドレスとを対応付けてローカルストレージ(ハードディスク;記憶部)に格納するようにしている。

30

#### 【0026】

そして、再度、利用者がその端末1でそのWebサイトにアクセスした場合、そのWebサイトのアドレスに基づいて、ローカルストレージから保存指示を受けたスタイルにするために必要な情報(後述の書換情報)を取得し、これに基づいてスタイルを適用してコンテンツ画面を表示させるようにしている。

これらを実現すべく、本実施形態では、例えばスマートフォン、タブレット、PCなどの端末1を、後述の表示変更装置を備えるものとしている。なお、端末1を、利用端末、通信端末、情報端末などともいう。

40

#### 【0027】

ここでは、後述の表示変更装置を備える端末1は、以下のように構成される。

まず、端末1のハードウェア構成について、図7を参照しながら説明する。

端末1は、コンピュータであり、そのハードウェア構成は、例えば図7に示すように、CPU(Central Processing Unit)102、メモリ101、通信制御部109、入力装置106、表示制御部103、表示装置104、記憶装置105、可搬型記録媒体108のドライブ装置107を備え、これらがバス110によって相互に接続された構成になっている。なお、本装置のハードウェア構成はこれに限られるものではない。例えば可搬型記録媒体108のドライブ装置107を備えないものであっても良い。

50



## 【 0 0 2 8 】

ここで、CPU 102は、コンピュータ全体を制御するものであり、プログラムをメモリ101に読み出して実行し、表示変更装置を備える端末1に必要な処理を行なうものである。

メモリ101は、例えばRAMなどの主記憶装置であり、プログラムの実行、データの書き換え等を行なう際に、プログラム又はデータを一時的に格納するものである。

## 【 0 0 2 9 】

通信制御部109（通信インターフェース）は、例えばLANやインターネットなどのネットワークを介して、他の装置と通信するために用いられるものである。この通信制御部109は、コンピュータに元から組み込まれていても良いし、後からコンピュータに取り付けられたNIC（Network Interface Card）でも良い。

入力装置106は、例えば、タッチパネル、プッシュボタン、マウスなどのポインティングデバイス、キーボードなどである。

## 【 0 0 3 0 】

表示装置104は、例えば液晶ディスプレイなどの表示装置である。

表示制御部103は、例えばコンテンツ画面などを表示装置104に表示させるための制御を行なうものである。

記憶装置105は、例えばハードディスクドライブ（HDD）やSSDなどの補助記憶装置であり、各種のプログラム及び各種のデータが格納されている。特に、本実施形態では、記憶装置105には、ブラウザプログラムや表示変更プログラムが格納されている。なお、メモリ101として、例えばROM（Read Only Memory）を備え、これに各種のプログラムや各種のデータを格納しておいても良い。

## 【 0 0 3 1 】

ドライブ装置107は、例えばフラッシュメモリ等の半導体メモリ、光ディスク、光磁気ディスク等の可搬型記録媒体108の記憶内容にアクセスするためのものである。

このようなハードウェア構成を備える端末1において、CPU102が、例えば記憶装置105に格納されているブラウザプログラムをメモリ101に読み出して実行することで、ブラウザの各機能、即ち、図1に示す画面表示部2が実現される。また、端末1において、CPU102が、例えば記憶装置105に格納されている表示変更プログラムをメモリ101に読み出して実行することで、表示変更装置の各機能、即ち、図1に示すスタイル変更部3が実現される。なお、ここでは、表示変更プログラムは、ブラウザが解釈できる言語（ここではJavaScript）で記述されている。そして、ブラウザはJavaScriptエンジンを備える。

## 【 0 0 3 2 】

つまり、本実施形態では、端末1は、図1に示すように、画面表示部（ブラウザ）2と、スタイル変更部（表示変更装置）3とを備え、これらによって画面表示処理及び表示変更処理（表示変更方法）が行なわれるようになっている。ここで、画面表示処理は、CPU102がメインメモリ101に読み込まれたブラウザプログラムに従って実行する処理であり、表示変更処理は、CPU102がメインメモリ101に読み込まれた表示変更プログラムに従って実行する処理（表示変更方法）である。

## 【 0 0 3 3 】

ここで、画面表示部2は、解析部4と、描画部5とを備え、それぞれにおいて、以下のような処理が行なわれる。

解析部4は、Webサイトから取得したコンテンツを解析する。ここでは、解析部4は、Webサイトから取得したHTMLファイル及びCSSファイルを解析し、DOM（Document Object Model）ツリー及びスタイルルールとしてメモリ101上に保持する（図8のステップS10参照）。ここで、メモリ101上のDOMツリーを保持している部分をDOMツリー保持部101Aといい、メモリ101上のスタイルルールを保持している部分をスタイルルール保持部101Bという。なお、DOMツリーやスタイルルールの仕様は、W3CのDocument Object Model Level 2 Style Specification（DOM2SS）

10

20

30

40

50

に規定されている。

【 0 0 3 4 】

描画部 5 は、メモリ 1 0 1 上に保持された D O M ツリー及びスタイルルールに基づいてコンテンツ画面を描画する（図 8 のステップ S 2 0 参照）。つまり、描画部 5 は、D O M ツリーにスタイルルールを適用してコンテンツ画面を描画する。なお、描画部 5 は、スタイルルールが書き換えられると、再度、書き換えられたスタイルルールを D O M ツリーに適用してコンテンツ画面を描画する（図 8 のステップ S 3 0、S 4 0 参照）。

【 0 0 3 5 】

本実施形態では、C S S ファイルは、例えば上述のようなメディアクエリ（図 4 参照）を含む。なお、ここでは、端末 1 に備えられているブラウザの表示画面幅は例えば 3 3 0

10

ピクセルとする。  
このため、画面表示部 2 は、C S S ファイルに含まれるメディアクエリを解析し、メモリ 1 0 1 上に保持されているスタイルルールを D O M ツリーに適用する際に、ブラウザの表示画面幅である 3 3 0 ピクセルは、3 2 1 ピクセル以上 7 6 8 ピクセル以下という条件を満たすとして、タブレット向け C S S を適用して、コンテンツ画面を表示させる。つまり、画面表示部 2 は、タブレット向けのスタイル S 1 に対するルールを「真」（True）と評価し、スマートフォン向けのスタイル S 0 に対するルール及び P C 向けのスタイル S 2 に対するルールを「偽」（False）と評価し、「真」（True）と評価したタブレット向けのスタイル S 1 を適用して、コンテンツ画面を表示させる。

【 0 0 3 6 】

20

このように、画面表示部 2 は、表示画面幅を条件として規定された複数のスタイルのうち、表示画面幅の条件を満たす条件充足スタイル（ここではタブレット向けのスタイル S 1）を適用してコンテンツ画面を表示させる。

スタイル変更部 3 は、スタイル変更指示を受けた場合に、適用されるスタイルが変更されるようにする。つまり、スタイル変更部 3 は、表示画面幅を条件として規定された複数のスタイルのうち、表示画面幅の条件を満たす条件充足スタイルを適用してコンテンツ画面が表示された状態で、スタイル変更指示を受けた場合に、条件充足スタイル以外のスタイルが適用されるように複数のスタイルのそれぞれに対する条件を「真」又は「偽」に書き換える。

【 0 0 3 7 】

30

このため、スタイル変更部 3 は、ルール抽出部 6 と、書換情報生成部 7 と、ルール書換部 8 とを備え、それぞれにおいて、以下のような処理が行なわれる。

ルール抽出部 6 は、メモリ 1 0 1（ここではスタイルルール保持部 1 0 1 B）から、C S S ファイルのメディアクエリを解析して得られた全てのスタイルに対するルールに含まれる条件を取得し、表示画面幅に関する条件を抽出し、ルールテーブル（図 1 2 参照）に格納する。

【 0 0 3 8 】

ここでは、まず、ルール抽出部 6 は、メモリ 1 0 1（ここではスタイルルール保持部 1 0 1 B）から、C S S ファイルのメディアクエリを解析して得られた全てのスタイルに対するルールに含まれる条件を取得する（図 9 のステップ A 1 0 参照）。ここで、W 3 C の D O M 2 S S において C S S のオブジェクトモデルやインタフェースが規定されており、メディアクエリを扱える端末 1 のブラウザにはこの仕様が実装されているため、メモリ 1 0 1 上に保持されている全てのスタイルに対するルールに含まれる条件は、この仕様に基づいたアクセスによって取得可能である。ここで、C S S ファイルのメディアクエリが、図 1 0（A）に示すようなスマートフォン向けのスタイル S 0 に対するルールを含む場合、この C S S ファイルのメディアクエリに含まれているスマートフォン向けのスタイル S 0 に対するルールを解析して得られたルールに含まれる条件（ここでは D O M 2 S S で規定されている type が MEDIA\_RULE である c s s R u l e オブジェクト集合の MediaText）は、図 1 0（B）に示すようなコードによって取得可能である。ここでは、C S S ファイルのメディアクエリに含まれているスマートフォン向けのスタイル S 0 に対するルールを解

40

50

析して得られたルールに含まれる条件として、「only screen and (min-width:0px) and (max-width:320px)」が取得される。なお、図10(B)では、参考のために、下側にログ出力も表示している。同様に、CSSファイルのメディアクエリに含まれているタブレット向けのスタイルS1に対するルールを解析して得られたルールに含まれる条件及びPC向けのスタイルS2に対するルールを解析して得られたルールに含まれる条件も取得可能である。ここでは、CSSファイルのメディアクエリに含まれているタブレット向けのスタイルS1に対するルールを解析して得られたルールに含まれる条件として、「only screen and (min-width:321px) and (max-width:768px)」が取得され、PC向けのスタイルS2に対するルールを解析して得られたルールに含まれる条件として、「only screen and (min-width:769px)」が取得される。また、これらのルールに含まれる条件は、インデックス番号(cssRuleオブジェクト集合のインデックス番号; rule\_num)に対応づけて取得される(図9のステップA10参照)。ここでは、スマートフォン向けのスタイルS0に対するルールを解析して得られたルールに含まれる条件は、インデックス番号「0」に対応づけて取得される。また、タブレット向けのスタイルS1に対するルールを解析して得られたルールに含まれる条件は、インデックス番号「1」に対応づけて取得される。また、PC向けのスタイルS2に対するルールを解析して得られたルールに含まれる条件は、インデックス番号「2」に対応づけて取得される。このようにして、CSSファイルのメディアクエリを解析して得られた全てのスタイルに対するルールに含まれる条件を取得することができる。つまり、CSSファイルのメディアクエリにおいて表示画面幅を条件として規定された複数のスタイルに対するルールの全て、即ち、表示画面幅に基づいて切り替え可能なスタイルに対するルールの全てを取得することができる。

#### 【0039】

次いで、ルール抽出部6は、上述のようにして取得された全てのスタイルに対するルールに含まれる条件の中から、正規表現を用いて、「media」、「min-width」、「max-width」のそれぞれの値(属性値)を抽出する(図9のステップA20参照)。例えば、上述のようにして取得されたスマートフォン向けのスタイルに対するルールに含まれる条件の中から、「media」、「min-width」、「max-width」のそれぞれの値を、図11に示すようなコードによって抽出可能である。なお、図11では、参考のために、下側にログ出力も表示している。

#### 【0040】

次に、ルール抽出部6は、上述のようにして全てのスタイルに対するルールに含まれる条件の中からそれぞれ抽出された「media」の値が「screen」であるか否かを順番に判定し(図9のステップA30参照)、「media」の値が「screen」であると判定した場合は、YESルートへ進み、「rule\_num」、「min-width」、「max-width」のそれぞれの値をルールテーブルに格納する(図9のステップA40参照)。一方、「media」の値が「screen」でないと判定した場合は、これらの値のルールテーブルへの格納は行なわない。このようにして、図12に示すように、インデックス番号(rule\_num)と表示画面幅に関する条件(最小幅「min-width」、最大幅「max-width」)とが対応づけられたルールテーブルが作成される。

#### 【0041】

書換情報生成部7は、図1に示すように、様々な表示画面幅に対して、それぞれの表示画面幅に関する条件を満たすか否かの評価(「真」又は「偽」)の組み合わせとして起こり得るものを全て抽出し、これを用いて、メモリ101上のスタイルルール保持部101Bに保持されているルールに含まれる表示画面幅に関する条件を書き換える書換コードを生成し、メモリ101上の書換情報テーブル101Cに格納する。なお、書換情報テーブル101Cを書換情報保持部ともいう。

#### 【0042】

まず、書換情報生成部7は、上述のようにして作成されたルールテーブル(図12参照)を用いて、様々な表示画面幅に対して、それぞれの表示画面幅に関する条件を満たすか否かの評価(「真」又は「偽」)の組み合わせ(ルールセット;ルール集合)として起こ

10

20

30

40

50

り得るものを全て抽出する（図13のステップB10参照）。つまり、上述のようにして作成されたルールテーブル（図12参照）に格納されたインデックス番号「0」～「2」のそれぞれに対応づけられている表示画面幅に関する条件の集合のべき集合、即ち、様々な表示画面幅に対して、インデックス番号「0」～「2」のそれぞれに対応づけられている表示画面幅に関する条件を満たすか否かの評価（「真」又は「偽」）の $2^3$ 通りの組み合わせの中で起こり得るものを全て抽出する。

#### 【0043】

例えば、まず、上述のようにして作成されたルールテーブル（図12参照）から、インデックス番号「0」、「1」、「2」のそれぞれの最小幅「min-width」及び最大幅「max-width」として設定されている値を取得し、表示画面幅 $x$ （変数）の値を変えながら、インデックス番号「0」、「1」、「2」のそれぞれの最小幅「min-width」として設定されている値と最大幅「max-width」として設定されている値とによって規定される論理式を満たす、即ち、最小幅として設定されている値以上であり、かつ、最大幅として設定されている値以下であるという条件を満たすか否かの評価を行なうことで、様々な表示画面幅に対して、それぞれの表示画面幅に関する条件を満たすか否かの評価（「真」又は「偽」）の組み合わせとして起こり得るものを全て抽出することができる。

#### 【0044】

本実施形態では、それぞれの表示画面幅に関する条件を同時に満たす表示画面幅は存在しない。このため、（1）インデックス番号「0」に対応づけられている表示画面幅に関する条件を満たし（即ち、評価「真」）、かつ、インデックス番号「1」、「2」のそれぞれに対応づけられている表示画面幅に関する条件を満たさない（即ち、評価「偽」）、（2）インデックス番号「1」に対応づけられている表示画面幅に関する条件を満たし（即ち、評価「真」）、かつ、インデックス番号「0」、「2」のそれぞれに対応づけられている表示画面幅に関する条件を満たさない（即ち、評価「偽」）、（3）インデックス番号「2」に対応づけられている表示画面幅に関する条件を満たし（即ち、評価「真」）、かつ、インデックス番号「0」、「1」のそれぞれに対応づけられている表示画面幅に関する条件を満たさない（即ち、評価「偽」）、という3つの組み合わせ（集合族）が抽出される。これらの3つの組み合わせは、それぞれ、評価「真」のインデックス番号（rule\_num）を用いて、{0}、{1}、{2}と表記する。ここで、{0}、{1}、{2}は、評価「真」のインデックス番号の集合を意味し、ルールセットのインデックスという。そして、このようにして抽出された3つの組み合わせ{0}、{1}、{2}を、それぞれ、メモリ101上に保持されている書換情報テーブル101C（図14参照）のインデックス番号（set\_index）「0」、「1」、「2」で指定される行の「rule\_index（ルールセットのインデックス）」の欄に格納する（図13のステップB20参照）。なお、ここでは、上述のようにして抽出したルールセットのインデックス（rule\_index）{0}、{1}、{2}を書換情報テーブル101Cに格納するようにしているが、これらは後述の書換コードを生成するために用いるだけであるため、書換コードを生成した後は保持しておかなくても良い。

#### 【0045】

次に、書換情報生成部7は、上述のようにして抽出したルールセットのインデックス（rule\_index）{0}、{1}、{2}を用いて、メモリ101上のスタイルルール保持部101Bにインデックス番号（rule\_num）に対応づけられて保持されているルールに含まれる表示画面幅に関する条件を書き換える書換コードを生成する（図13のステップB30参照）。つまり、上述のようにして抽出したルールセットのインデックス（rule\_index）{0}、{1}、{2}のそれぞれの中に記述されているインデックス番号（rule\_num）「0」、「1」、「2」を用いて、これらに対応づけられている表示画面幅に関する条件を「真」に、記述されていないインデックス番号（rule\_num）に対応づけられている表示画面幅に関する条件を「偽」に書き換える書換コードを生成する。ここでは、図14に示すように、書換コードとして、上述のようにして抽出したルールセットのインデックス（rule\_index）{0}に記述されているインデックス番号（rule\_num）「0」を用いて、

10

20

30

40

50

applyMQ({{0:true},{1:false},{2:false}})を生成し、ルールセットのインデックス (rule\_index) { 1 } に記述されているインデックス番号 (rule\_num) 「 1 」を用いて、applyMQ({{0:false},{1:true},{2:false}})を生成し、ルールセットのインデックス (rule\_index) { 2 } に記述されているインデックス番号 (rule\_num) 「 2 」を用いて、applyMQ({{0:false},{1:false},{2:true}})を生成する。そして、このようにして生成された書換コードを、それぞれ、書換情報テーブル (図 1 4 参照) のインデックス番号 (set\_index) 「 0 」、「 1 」、「 2 」で指定される行の「書換コード」の欄に格納する (図 1 3 のステップ B 4 0 参照)。なお、書換コードを、適用コード又は実行コードともいう。

#### 【 0 0 4 6 】

次に、書換情報生成部 7 は、現在の状態を示す情報、即ち、どのルールセットのインデックス (rule\_index) が有効になっているかを示す情報である真偽情報 (true or false) を、書換情報テーブル (図 1 4 参照) のインデックス番号 (set\_index) 「 0 」、「 1 」、「 2 」で指定される行の「状態」の欄に格納する (図 1 3 のステップ B 5 0 参照)。ここで、書換情報生成部 7 が現在の状態を示す情報を書換情報テーブル (図 1 4 参照) に格納するのは、最初にコンテンツ画面を表示させた段階である。そして、本実施形態では、最初にコンテンツ画面を表示させる場合、上述のように、画面表示部 2 は、タブレット向けのスタイル S 1 に対するルールを「真」 (True) と評価し、スマートフォン向けのスタイル S 0 に対するルール及び PC 向けのスタイル S 2 に対するルールを「偽」 (False) と評価し、「真」 (True) と評価したタブレット向けのスタイル S 1 を適用して、コンテンツ画面を表示させる。このため、本実施形態では、図 1 4 に示すように、書換情報テーブルのインデックス番号 (set\_index) 「 1 」で指定される行の「状態」の欄に、現在の状態を示す情報として、「真」 (true) を格納し、書換情報テーブルのインデックス番号 (set\_index) 「 0 」、「 2 」で指定される行の「状態」の欄に、現在の状態を示す情報として、「偽」 (false) を格納する。

#### 【 0 0 4 7 】

なお、ここでは、CSS ファイルが図 4 に示すようなメディアクエリを含む場合を例に挙げて説明しているため、それぞれの表示画面幅に関する条件を同時に満たす表示画面幅は存在せず、抽出されたルールセットのインデックス (rule\_index) は、{ 0 }、{ 1 }、{ 2 } のように、それぞれ 1 つずつのインデックス番号 (rule\_num) を含むものとなっており、生成された書換コードは、それぞれ 1 つずつ true を含むものとなっているが、これに限られるものではない。

#### 【 0 0 4 8 】

例えば、CSS ファイルが図 1 5 に示すようなメディアクエリを含む場合もある。このメディアクエリでは、いずれのスタイルに対するルールにおいても max-width が設定されていない。つまり、このメディアクエリでは、スマートフォン向けのスタイル S 0 に対するルールは、メディアタイプとして「screen」、条件として「(min-width:0px)」、スタイルとして「スマートフォン向け CSS」を含む。このルールは、ブラウザに対し、ブラウザの表示画面幅が 0 ピクセル以上の場合に、スマートフォン向け CSS を適用するように指示する。また、タブレット向けのスタイル S 1 に対するルールは、メディアタイプとして「screen」、条件として「(min-width:320px)」、スタイルとして「タブレット向け CSS」を含む。このルールは、ブラウザに対し、ブラウザの表示画面幅が 3 2 0 ピクセル以上の場合に、タブレット向け CSS を適用するように指示する。また、PC 向けのスタイル S 2 に対するルールは、メディアタイプとして「screen」、条件として「(min-width:768px)」、スタイルとして「PC 向け CSS」を含む。このルールは、ブラウザに対し、ブラウザの表示画面幅が 7 6 8 ピクセル以上の場合に、PC 向け CSS を適用するように指示する。この場合、上述の実施形態と同様に、このようなメディアクエリを解析して得られた全てのスタイルに対するルールに含まれる条件を取得し、表示画面幅に関する条件を抽出し、ルールテーブルに格納すると、ルールテーブルは、図 1 6 に示すようになる。そして、上述の実施形態と同様に、このルールテーブルを用いて、様々な表示画面幅に対して、それぞれの表示画面幅に関する条件を満たすか否かの評価の組み合わせとして

起こり得るものを全て抽出すると、インデックス番号「2」に対応づけられている表示画面幅に関する条件を満たす場合は必ずインデックス番号「1」に対応づけられている表示画面幅に関する条件を満たし、インデックス番号「1」に対応づけられている表示画面幅に関する条件を満たす場合は必ずインデックス番号「0」に対応づけられている表示画面幅に関する条件を満たすため、抽出されたルールセットのインデックス (rule\_index) は、{0}、{0, 1}、{0, 1, 2} のようになる。これらのルールセットのインデックス (rule\_index) {0}、{0, 1}、{0, 1, 2} を用いて書換コードを生成すると、それぞれ、applyMQ({{0:true},{1:false},{2:false}})、applyMQ({{0:true},{1:true},{2:false}})、applyMQ({{0:true},{1:true},{2:true}}) となる。そして、書換情報テーブルは例えば図17に示すようになる。このように、抽出されたルールセットのインデックス (rule\_index) がそれぞれ1つずつのインデックス番号 (rule\_num) を含むものとならず、生成された書換コードがそれぞれ1つずつtrueを含むものとならない場合もある。なお、2つ又は3つの表示画面幅に関する条件を満たす場合、メディアクエリに記述されているスタイルは上から順に適用され (ここではスマートフォン向けCSS、タブレット向けCSS、PC向けCSSの順に適用され)、上書きされて、これらのスタイルを組み合わせ合わせたスタイルが適用されることになる。

#### 【0049】

ルール書換部8は、図1に示すように、表示画面幅を条件として規定された複数のスタイルのうち、表示画面幅の条件を満たす条件充足スタイルを適用してコンテンツ画面が表示された状態で、スタイル変更指示を受けた場合に、書換情報を取得し、条件充足スタイル以外のスタイルが適用されるように書換情報に基づいて複数のスタイルのそれぞれに対する条件を「真」又は「偽」に書き換える。つまり、本実施形態では、表示画面幅を条件として規定された複数のスタイルのうち、表示画面幅の条件を満たす条件充足スタイルを適用してコンテンツ画面が表示された状態で、スタイル変更指示を受けた場合に、書換情報を取得する取得部8Aと、条件充足スタイル以外のスタイルが適用されるように書換情報に基づいて複数のスタイルのそれぞれに対する条件を「真」又は「偽」に書き換える書換部8Bとを備える。

#### 【0050】

本実施形態では、ルール書換部8は、スタイル変更指示を受けた場合に、書換情報として、上述の書換情報テーブル (図14参照) からインデックス番号 (set\_index) を用いて書換コードを取得し、この書換コードをブラウザに備えられるJavascriptエンジンで実行して、メモリ101 (スタイルルール保持部101B) 上に保持されている、CSSファイルのメディアクエリを解析して得られた全てのスタイルに対するルールに含まれる条件を「真」又は「偽」に書き換える。なお、書換コードは、複数のスタイルのそれぞれに対する条件 (適用条件) を書き換える書換情報である。また、ここでは、書換情報として書換コードを用いているが、書換情報はこれに限られるものではなく、複数のスタイルのそれぞれに対する条件を書き換えるのに必要な情報であれば良い。

#### 【0051】

つまり、CSSファイルのメディアクエリを解析して得られた全てのスタイルに対するルールに含まれる条件として、メモリ101 (スタイルルール保持部101B) 上に、図19(A)に示すような条件が保持されている。ここでは、CSSファイルのメディアクエリに含まれているスマートフォン向けのスタイルS0に対するルールを解析して得られたルールに含まれる条件として、メモリ101 (スタイルルール保持部101B) 上に、document.styleSheet[0].cssRule[0]の値として、「only screen and (min-width:0px) and (max-width:320px)」という条件が保持されている。また、CSSファイルのメディアクエリに含まれているタブレット向けのスタイルS1に対するルールを解析して得られたルールに含まれる条件として、メモリ101 (スタイルルール保持部101B) 上に、document.styleSheet[0].cssRule[1]の値として、「only screen and (min-width:321px) and (max-width:768px)」という条件が保持されている。また、PC向けのスタイルS2に対するルールを解析して得られたルールに含まれる条件として、メモリ101 (スタイル

10

20

30

40

50

ルール保持部 101B) 上に、document.styleSheet[0].cssRule[2]の値として、「only screen and (min-width:769px)」という条件が保持されている。

【0052】

また、本実施形態では、最初にコンテンツ画面を表示させる場合、上述のように、画面表示部 2 は、タブレット向けのスタイル S 1 に対するルールを「真」(True)と評価し、スマートフォン向けのスタイル S 0 に対するルール及び PC 向けのスタイル S 2 に対するルールを「偽」(False)と評価し、「真」(True)と評価したタブレット向けのスタイル S 1 を適用して、コンテンツ画面を表示させる。このため、本実施形態では、図 14 に示すように、書換情報テーブルのインデックス番号 (set\_index) 「1」で指定される行の「状態」の欄に、「真」(true)が格納されており、書換情報テーブルのインデックス番号 (set\_index) 「0」、「2」で指定される行の「状態」の欄に、「偽」(false)が格納されている。

10

【0053】

このような状態で、スタイル変更指示を受けた場合、ルール書換部 8 は、インデックス番号 (set\_index) 「1」以外のインデックス番号 (set\_index) である「0」又は「2」を用いて書換情報テーブル (図 14 参照) から書換コードを取得する (図 18 のステップ C 10)。ここでは、例えば、インデックス番号 (set\_index) 「0」を用いて、書換情報テーブル (図 14 参照) から、書換コードとして applyMQ({{0:true},{1:false},{2:false}}) を取得する。そして、この書換コードをブラウザに備えられる Javascript エンジンで実行して各スタイルに対するルールに含まれる条件を書き換える (図 18 のステップ C 20)。ここで、図 20 は、applyMQ メソッドの例を示している。これにより、メモリ 101 (スタイルルール保持部 101B) 上に、CSS ファイルのメディアクエリに含まれているスマートフォン向けのスタイル S 0 に対するルールを解析して得られたルールに含まれる条件として、document.styleSheet[0].cssRule[0]の値として保持されている、「only screen and (min-width:0px) and (max-width:320px)」(図 19 (A) 参照) が、図 19 (B) に示すように、「true」に書き換えられる。また、CSS ファイルのメディアクエリに含まれているタブレット向けのスタイル S 1 に対するルールを解析して得られたルールに含まれる条件として、document.styleSheet[0].cssRule[1]の値として保持されている、「only screen and (min-width:321px) and (max-width:768px)」(図 19 (A) 参照) が、図 19 (B) に示すように、「false」に書き換えられる。また、PC 向けのスタイル S 2 に対するルールを解析して得られたルールに含まれる条件として、document.styleSheet[0].cssRule[2]の値として保持されている、「only screen and (min-width:769px)」(図 19 (A) 参照) が、図 19 (B) に示すように、「false」に書き換えられる。なお、ここでは、メモリ 101 上のスタイルルール保持部 101B 上に保持されているルールに含まれる条件を書き換えるようにしているが、これに限られるものではなく、メディアクエリに含まれている各スタイルに対するルールを解析して得られた各スタイルに対する条件 (メディアクエリ情報) であってメモリ上に保持されているものを「真」又は「偽」に書き換えるようにすれば良い。

20

30

【0054】

そして、ルール書換部 8 は、現在の状態を示す情報、即ち、どのルールセットのインデックス (rule\_index) が有効になっているかを示す情報である真偽情報 (true or false) を、書換情報テーブルのインデックス番号 (set\_index) 「0」、「1」、「2」で指定される行の「状態」の欄に格納する (図 18 のステップ C 30)。つまり、ルール書換部 8 は、書換情報テーブルのインデックス番号 (set\_index) 「0」、「1」、「2」で指定される行の「状態」の欄に格納されている真偽情報を書き換える。ここでは、書換情報テーブルのインデックス番号 (set\_index) 「0」で指定される行の「状態」の欄に、現在の状態を示す情報として、「真」(true)を格納し、書換情報テーブルのインデックス番号 (set\_index) 「1」、「2」で指定される行の「状態」の欄に、現在の状態を示す情報として、「偽」(false)を格納する。

40

【0055】

50

このようにして、表示画面幅を条件として規定された複数のスタイルのうち、表示画面幅の条件を満たす条件充足スタイルを適用してコンテンツ画面が表示された状態で、スタイル変更指示を受けた場合に、ルール書換部 8 が、上述のようにして CSS ファイルのメディアクエリを解析して得られた各スタイルに対するルールに含まれる条件を書き換えると、上述の描画部 5 が、再度、書き換えられたスタイルルールを DOM ツリーに適用してコンテンツ画面を描画する。

**【 0 0 5 6 】**

つまり、本実施形態では、表示画面幅を条件として規定された複数のスタイルのうち、表示画面幅の条件を満たす条件充足スタイルを適用してコンテンツ画面が表示された状態で、スタイル変更指示を受けた場合に、ルール書換部 8 が、複数のスタイルのそれぞれに対する条件を書き換える書換情報を取得し、この書換情報に基づいて、表示画面幅の条件を満たす条件充足スタイル（ここではタブレット向けのスタイル S 1）に対する条件を「偽」に、表示画面幅の条件を満たさない少なくとも一つの条件非充足スタイル（ここではスマートフォン向けのスタイル S 0）に対する条件を「真」に書き換える。つまり、ルール書換部 8 は、複数のスタイルのそれぞれに対する条件を書き換える書換情報を取得する取得部 8 A と、この書換情報に基づいて、表示画面幅の条件を満たす条件充足スタイル（ここではタブレット向けのスタイル S 1）に対する条件を「偽」に、表示画面幅の条件を満たさない少なくとも一つの条件非充足スタイル（ここではスマートフォン向けのスタイル S 0）に対する条件を「真」に書き換える書換部 8 B とを備える。そして、描画部 5 が、複数のスタイル（ここでは 3 つのスタイル S 0 ~ S 2）のうち、少なくとも一つの条件非充足スタイルを適用して、コンテンツ画面のスタイルを変更したスタイル変更画面を表示させる。つまり、描画部 5 が、条件が「真」に書き換えられたスタイルを適用してコンテンツ画面を表示させる。このため、ルール書換部 8 及び描画部 5 をまとめて画面変更表示部という。

**【 0 0 5 7 】**

このように、本実施形態では、画面表示部 2 が、表示画面幅を条件として規定された複数のスタイルのうち、表示画面幅の条件を満たす条件充足スタイルを適用してコンテンツ画面を表示させ（図 2 1 のステップ D 1 0 参照）、画面変更表示部が、スタイル変更指示を受けた場合に、書換情報を取得し、書換情報に基づいて複数のスタイルのそれぞれに対する条件を「真」又は「偽」に書き換え、条件が「真」に書き換えられたスタイルを適用してコンテンツ画面を表示させる（図 2 1 のステップ D 2 0、D 3 0 参照）。

**【 0 0 5 8 】**

なお、CSS ファイルが図 1 5 に示すようなメディアクエリを含む場合には、表示画面幅を条件として規定された複数のスタイルのうち、表示画面幅の条件を満たす条件充足スタイルを適用してコンテンツ画面が表示された状態で、スタイル変更指示を受けた場合に、ルール書換部 8 が、複数のスタイルのそれぞれに対する条件を書き換える書換情報を取得し、書換情報に基づいて、少なくとも一つの条件充足スタイルに対する条件を「偽」に書き換える、又は、表示画面幅の条件を満たさない少なくとも一つの条件非充足スタイルに対する条件を「真」に書き換えるようにすれば良い。そして、描画部 5 が、複数のスタイル（ここでは 3 つのスタイル）のうち、少なくとも一つの条件非充足スタイルを適用して、又は、少なくとも一つの条件充足スタイルを適用せずに、コンテンツ画面のスタイルを変更したスタイル変更画面を表示させれば良い。

**【 0 0 5 9 】**

例えば、図 1 7 に示す書換情報テーブルのインデックス番号 (set\_index) 「1」から「0」に切り替えられる場合、インデックス番号 (rule\_num) 「0」、「1」に対応づけられている条件、即ち、スマートフォン向けのスタイル S 0 及びタブレット向けのスタイル S 1 に対する条件が「真」になっており、インデックス番号 (rule\_num) 「2」に対応づけられている条件、即ち、PC 向けのスタイル S 2 に対する条件が「偽」になっている状態から、インデックス番号 (rule\_num) 「1」に対応づけられている条件、即ち、タブレット向けのスタイル S 1 に対する条件が「偽」に書き換えられる。つまり、少なくとも

10

20

30

40

50



一つの条件充足スタイルに対する条件が「偽」に書き換えられる。そして、描画部が、複数のスタイル（ここでは3つのスタイル）のうち、少なくとも一つの条件充足スタイルを適用せずに、コンテンツ画面のスタイルを変更したスタイル変更画面が表示される。

【0060】

また、例えば、図17に示す書換情報テーブルのインデックス番号（set\_index）「1」から「2」に切り替えられる場合、インデックス番号（rule\_num）「0」、「1」に対応づけられている条件、即ち、スマートフォン向けのスタイルS0及びタブレット向けのスタイルS1に対する条件が「真」になっており、インデックス番号（rule\_num）「2」に対応づけられている条件、即ち、PC向けのスタイルS2に対する条件が「偽」になっている状態から、インデックス番号（rule\_num）「2」に対応づけられている条件、即ち、PC向けのスタイルS2に対する条件が「真」に書き換えられる。つまり、少なくとも一つの条件非充足スタイルに対する条件が「真」に書き換えられる。そして、描画部5が、複数のスタイル（ここでは3つのスタイルS0～S2）のうち、少なくとも一つの条件非充足スタイルを適用して、コンテンツ画面のスタイルを変更したスタイル変更画面が表示される。

10

【0061】

このため、本実施形態では、表示画面幅を条件として規定された複数のスタイルのうち、表示画面幅の条件を満たす条件充足スタイルを適用してコンテンツ画面が表示された状態で、スタイル変更指示を受けた場合に、複数のスタイルのそれぞれに対する条件を書き換える書換情報を取得する取得部8Aと、書換情報に基づいて、少なくとも一つの条件充足スタイルに対する条件を「偽」に書き換える、又は、表示画面幅の条件を満たさない少なくとも一つの条件非充足スタイルに対する条件を「真」に書き換える書換部8Bとを備えることになる。つまり、本実施形態では、表示画面幅を条件として規定された複数のスタイルのうち、表示画面幅の条件を満たす条件充足スタイルを適用してコンテンツ画面が表示された状態で、スタイル変更指示を受けた場合に、書換情報を取得する取得部8Aと、条件充足スタイル以外のスタイルが適用されるように、書換情報に基づいて、複数のスタイルのそれぞれに対する条件を「真」又は「偽」に書き換える書換部8Bとを備えることになる。

20

【0062】

本実施形態では、利用者が、表示画面幅の条件を満たすスタイルが適していないと感じ、ダイアログボックス1B内の「Change」ボタンを押した場合（図5参照）、即ち、スタイル変更指示を受けた場合に、スタイル変更イベントハンドラ（スタイル切替イベントハンドラ）を介して、上述のルール書換部8による処理、描画部5による処理が実行され、表示画面幅の条件を満たすスタイル以外のスタイル、即ち、表示画面幅の条件を満たさないスタイルに変更される。

30

【0063】

なお、スタイル変更部3は、表示画面幅を条件として規定された複数のスタイルのうち、表示画面幅の条件を満たす条件充足スタイルを適用してコンテンツ画面が表示された状態で、スタイル変更指示を受けた場合に、書換情報を取得し、条件充足スタイル以外のスタイルが適用されるように書換情報に基づいて複数のスタイルのそれぞれに対する条件を「真」又は「偽」に書き換える処理を行なえば良く、少なくとも、上述のルール書換部8による処理を行なえば良い。例えば、スタイル変更部3は、スタイル変更指示を受ける前に、例えば最初にコンテンツ画面を表示させる際に、予め、上述のルール抽出部6による処理、及び、上述の書換情報生成部7による処理を行なっておき、スタイル変更指示を受けた場合に、上述のルール書換部8による処理を行なうようにしても良い。つまり、スタイル変更部3によるスタイル変更処理としては、まず、上述のルール抽出部6による処理を行ない（図22のステップE10参照）、次に、上述の書換情報生成部7による処理を行ない（図22のステップE20参照）、次いで、スタイル変更指示を受けたかを判定し（図22のステップS30参照）、スタイル変更指示を受けた場合に、上述のルール書換部8による処理（図22のステップE40参照）を行なうようにしても良い。この場合、

40

50

最初にコンテンツ画面を表示させる際に、ロード完了時に自動的に呼び出されるイベントハンドラを介して、上述のルール抽出部 6 による処理（図 2 2 のステップ E 1 0 参照）、及び、上述の書換情報生成部 7 による処理（図 2 2 のステップ E 2 0 参照）が実行されるようにすれば良い。また、スタイル変更部 3 は、スタイル変更指示を受けたかを判定し（図 2 3 のステップ F 1 0 参照）、スタイル変更指示を受けた場合に、上述のルール抽出部 6 による処理（図 2 3 のステップ F 2 0 参照）、上述の書換情報生成部 7 による処理（図 2 3 のステップ F 3 0 参照）、及び、上述のルール書換部 8 による処理（図 2 3 のステップ F 4 0 参照）の全てを行なうようにしても良い。

#### 【 0 0 6 4 】

ところで、本実施形態では、スタイル変更部 3 は、さらに、保存処理部 9 と、再アクセス時書換部 1 0 とを備える。

10

ここで、保存処理部 9 は、スタイル保存指示を受けた場合に、その時に適用されている書換情報と Web サイトのアドレスとを対応づけてハードディスク（記憶部）1 0 5 に格納する処理を行なう。

#### 【 0 0 6 5 】

本実施形態では、保存処理部 9 は、利用者が、あるスタイルが適用されたコンテンツ画面が表示された状態で、ダイアログボックス 1 B 内の「Save」ボタンを押した場合、（図 6 参照）即ち、スタイル保存指示を受けた場合に、スタイル保存イベントハンドラを介して、その時に適用されている書換情報としての書換コードと Web サイトのアドレスとしての URL とを対応づけて記憶部 1 0 5 としてのローカルストレージ（ハードディスク；ブラウザの記憶領域）に格納する処理を行なう。

20

#### 【 0 0 6 6 】

例えば、表示画面幅が 3 3 0 ピクセルのブラウザを備える端末で、URL「http://example.com/index.html」の Web サイトにアクセスし、PC 向けのスタイル S 2 が適用されたコンテンツ画面が表示された状態で、利用者が、ダイアログボックス 1 B 内の「Save」ボタンを押した場合（図 6 参照）、即ち、スタイル保存指示を受けた場合、保存処理部 9 は、次のような処理を行なう。つまり、保存処理部 9 は、まず、スタイル保存指示を受けたかを判定し（図 2 4 のステップ G 1 0 参照）、スタイル保存指示を受けた場合に、メモリ 1 0 1 上の書換情報テーブル 1 0 1 C（図 1 4 参照）を参照し、現在の有効になっている、即ち、「状態」の欄が「true」になっているルールセットのインデックス（rule\_index）（ここでは { 2 }）及び書換コード [ここでは applyMQ({{0:false},{1:false},{2:true}})] を取得する（図 2 4 のステップ G 2 0 参照）。次に、保存処理部 9 は、ブラウザから、現在アクセスしている Web サイトの URL（ここでは「http://example.com/index.html」）及びブラウザの表示画面幅（ここでは 3 3 0 ピクセル）を取得する（図 2 4 のステップ G 3 0 参照）。そして、保存処理部 9 は、上述のようにして取得したルールセットのインデックス（rule\_index）（ここでは { 2 }）、書換コード [ここでは applyMQ({{0:false},{1:false},{2:true}})]、Web サイトの URL（ここでは「http://example.com/index.html」）及びブラウザの表示画面幅（ここでは 3 3 0 ピクセル）を用いて、例えば図 2 5 に示すようなコードによって保存処理（書込処理）を実行して、例えば図 2 6 に示すように、ローカルストレージとしてのハードディスク 1 0 5 上の再アクセス時書換情報テーブル 1 0 5 A の URL の欄に「http://example.com/index.html」、ブラウザの表示画面幅（width）の欄に 3 3 0、ルールセットのインデックス（rule\_index）の欄に { 2 }、書換コードの欄に applyMQ({{0:false},{1:false},{2:true}}) を格納する（図 2 4 のステップ G 4 0 参照）。このようにしてローカルストレージとしてのハードディスク 1 0 5 に格納された再アクセス時書換情報テーブル 1 0 5 A は、ブラウザを閉じた後も保持される。なお、再アクセス時書換情報テーブル 1 0 5 A を、再アクセス時書換情報保持部又は保存情報ともいう。

30

40

#### 【 0 0 6 7 】

なお、スタイル変更指示を受けた際に書換コードを取得するのに用いたインデックス番号（set\_index）を保持している場合には、そのインデックス番号（set\_index）を用いて

50

、ルールセットのインデックス (rule\_index) 及び書換コードを取得しても良い。また、ここでは、ローカルストレージとしてのハードディスク 105 に格納されている再アクセス時書換情報テーブル 105 A を、ブラウザの表示画面幅 (width) の欄、及び、ルールセットのインデックス (rule\_index) の欄を含むものとしているが、これらを含まないものとしても良い。

【0068】

再アクセス時書換部 10 は、図 1 に示すように、Web サイトのアドレスがハードディスク (記憶部) 105 に格納されていた場合に、ハードディスク 105 から Web サイトのアドレスに基づいて書換情報 (ここでは書換コード) を取得し、この書換情報に基づいて複数のスタイルのそれぞれに対する条件を「真」又は「偽」に書き換える処理を行なう。

10

【0069】

本実施形態では、再度、利用者が同じ端末 (ブラウザ) で同じ Web サイトにアクセスした場合、その Web サイトのアドレスに基づいて、記憶部としてのローカルストレージ 105 に格納されている再アクセス時書換情報テーブル 105 A から書換情報としての書換コードを取得し、この書換コードに基づいて各スタイルに対する条件を「真」又は「偽」に書き換える処理を行なう。

【0070】

例えば、再度、表示画面幅が 330 ピクセルのブラウザを備える端末 1 で、URL 「http://example.com/index.html」の Web サイトにアクセスした場合、再アクセス時書換部 10 は、次のような処理を行なう。

20

つまり、まず、再アクセス時書換部 10 は、ブラウザから、現在アクセスしている Web サイトの URL (ここでは「http://example.com/index.html」) 及びブラウザの表示画面幅 (ここでは 330 ピクセル) を取得する (図 27 のステップ H10 参照)。次に、再アクセス時書換部 10 は、ローカルストレージ 105 上の再アクセス時書換情報テーブル 105 A から URL 及びブラウザの表示画面幅を取得する (図 27 のステップ H20 参照)。

【0071】

次に、再アクセス時書換部 10 は、これらが一致するかを判定する (図 27 のステップ H30 参照)。この判定の結果、一致するものがあった場合は、再アクセス時書換部 10 は、その URL (ここでは「http://example.com/index.html」) 及びブラウザの表示画面幅 (ここでは 330 ピクセル) を用いて、ローカルストレージ 105 上の再アクセス時書換情報テーブル 105 A (図 26 参照) から書換コード [ここでは applyMQ({0:false}, {1:false}, {2:true})] を取得し (図 27 のステップ H40 参照)、その書換コード [ここでは applyMQ({0:false}, {1:false}, {2:true})] をブラウザに備えられる JavaScript エンジンで実行する。これにより、メモリ 101 (スタイルルール保持部 101 B) 上に、CSS ファイルのメディアクエリに含まれているスマートフォン向けのスタイル S0 に対するルールを解析して得られたルールに含まれる条件として、document.styleSheet[0].cssRule[0] の値として保持されている、「only screen and (min-width:0px) and (max-width:320px)」が「false」に書き換えられ、CSS ファイルのメディアクエリに含まれているタブレット向けのスタイル S1 に対するルールを解析して得られたルールに含まれる条件として、document.styleSheet[0].cssRule[1] の値として保持されている、「only screen and (min-width:321px) and (max-width:768px)」が「false」に書き換えられ、PC 向けのスタイル S2 に対するルールを解析して得られたルールに含まれる条件として、document.styleSheet[0].cssRule[2] の値として保持されている、「only screen and (min-width:769px)」が「true」に書き換えられる (図 27 のステップ H50 参照)。なお、ここでは、URL だけでなく、ブラウザの表示画面幅も取得し、これも一致するか否かの判定に用いているが、URL だけを取得し、URL だけを一致するか否かの判定に用いるようにしても良い。

30

40

【0072】

50

そして、再アクセス時書換部 10 は、現在の状態を示す情報、即ち、どのルールセットのインデックス (rule\_index) が有効になっているかを示す情報である真偽情報 (true or false) を、書換情報テーブル 101C (図 14 参照) のインデックス番号 (set\_index) 「0」、「1」、「2」で指定される行の「状態」の欄に格納する (図 27 のステップ H60 参照)。つまり、再アクセス時書換部 10 は、書換情報テーブル 101C のインデックス番号 (set\_index) 「0」、「1」、「2」で指定される行の「状態」の欄に格納されている真偽情報を書き換える。ここでは、書換情報テーブル 101C のインデックス番号 (set\_index) 「2」で指定される行の「状態」の欄に、現在の状態を示す情報として、「真」(true) を格納し、書換情報テーブル 101C のインデックス番号 (set\_index) 「0」、「1」で指定される行の「状態」の欄に、現在の状態を示す情報として、「偽」(false) を格納する。

10

**【0073】**

このようにして、利用者が同じ端末 (ブラウザ) で同じ Web サイトにアクセスした場合に、再アクセス時書換部 10 が、CSS ファイルのメディアクエリを解析して得られた各スタイルに対するルールに含まれる条件を書き換えると、描画部 5 が、書き換えられたスタイルルールを DOM ツリーに適用してコンテンツ画面を描画する。これにより、利用者が同じ端末 (ブラウザ) で同じ Web サイトにアクセスした場合に、以前にアクセスした際に利用者が適していると判断したスタイルが適用されたコンテンツ画面が直ちに表示されることになる。

**【0074】**

20

したがって、本実施形態にかかる表示変更装置及び表示変更方法によれば、条件として設定された表示画面幅が実態にそぐわず、利用者がそのスタイルが適していないと感じる場合に、他のスタイル、即ち、表示画面幅の条件を満たさないスタイルに変更できるという利点がある。

なお、本発明は、上述した実施形態に記載した構成に限定されるものではなく、本発明の趣旨を逸脱しない範囲で種々変形することが可能である。

**【0075】**

例えば、上述の実施形態では、ブラウザプログラムが実行されることによって実現される機能 (ブラウザ; 画面表示部) と、表示変更プログラムが実行されることによって実現される機能 (表示変更装置; スタイル変更部) とを分けて説明しているが、これに限られるものではなく、例えば表示変更プログラムが実行されることによって実現される機能を備えるブラウザとして構成することもできる。この場合、端末は、ブラウザプログラムが実行されることによって実現される機能として画面表示部及びスタイル変更部を備え、これらによって画面表示処理及び表示変更処理 (表示変更方法) が行なわれることになる。

30

**【0076】**

また、例えば、上述の実施形態では、コンピュータである端末 1 が、上述の実施形態における各処理をコンピュータに実行させる表示変更プログラムを備えるものとして説明しているが、この表示変更プログラムの実装方法としては、以下のように、種々の方法がある。

例えば、図 28 に示すように、ブラウザ 50 からのリクエストやサーバ 70 からのレスポンスを中継する中継サーバ 60 (プロキシサーバなど) が、プログラム挿入機能 (プログラム挿入部) 60A によって、表示変更プログラムをレスポンスコンテンツ中に挿入するようになれば良い。具体的には、利用者が中継サーバ 60 を介してレスポンス Web デザイン対応の Web サイト 70 にアクセスした場合に、中継サーバ 60 が、レスポンス HTML の構造を解析し、例えば図 29 に示すように、このレスポンス HTML 内に、表示変更プログラムが格納されているサーバのアドレス (アクセス先 URL) を含む script タグを挿入するようになれば良い。なお、script タグの挿入に代えて、表示変更プログラム (スクリプト) を HTML 内に直接埋め込むようにしても良い。これを中継サーバ方式という。

40

**【0077】**

50

また、例えば、図30に示すように、ブラウザ50が、プラグインなどのブラウザ拡張機能としてのプログラム挿入機能(プログラム挿入部)50Aによって、レスポンスコンテンツを受け取った後に、表示変更プログラムをレスポンスコンテンツ中に挿入するようによっても良い。これをブラウザプラグイン方式という。

これらの方式によって、コンテンツに依存しないで、表示変更プログラムを組み込むことができ、既存のWebサイトが提供するコンテンツに対しても適用可能である。

【0078】

このほか、例えば、表示変更プログラム(又は表示変更プログラムを含むブラウザ)は、例えば伝送媒体としてのネットワーク(例えばインターネット、公衆回線や専用回線等の通信回線等)を介して提供される場合もある。

例えば、プログラム提供者が例えばサーバなどの他のコンピュータ上で提供している表示変更プログラムを、例えばインターネットやLAN等のネットワーク及び通信インタフェースを介して、記憶装置にインストール又はダウンロードしても良い。これにより、上述の実施形態で説明した表示変更装置及び表示変更方法が実現され、上述の実施形態の場合と同様に、記憶装置にインストール又はダウンロードされた表示変更プログラムを、CPUがメモリ上に読み出して実行することで、上述の実施形態の各処理が行なわれることになる。

【0079】

また、例えば、表示変更プログラム(又は表示変更プログラムを含むブラウザ)は、コンピュータ読取可能な記録媒体に格納した状態で提供される場合もある。

ここで、記録媒体には、例えば半導体メモリなどのメモリ、磁気ディスク、光ディスク[例えばCD(Compact Disc)-ROM, DVD(Digital Versatile Disk),ブルーレイディスク等]、光磁気ディスク(MO:Magneto optical Disc)等のプログラムを記録することができるものが含まれる。なお、磁気ディスク、光ディスク、光磁気ディスク等を可搬型記録媒体ともいう。

【0080】

この場合、ドライブ装置を介して、可搬型記録媒体から表示変更プログラムを読み出し、読み出された表示変更プログラムを記憶装置にインストールすることになる。これにより、上述の実施形態で説明した表示変更装置及び表示変更方法が実現され、上述の実施形態の場合と同様に、記憶装置にインストールされた表示変更プログラムを、CPUがメモリ上に読み出して実行することで、上述の実施形態の各処理が行なわれることになる。なお、コンピュータは、可搬型記録媒体から直接プログラムを読み取り、そのプログラムに従った処理を実行することもできる。

【0081】

以下、上述の実施形態及び変形例に関し、更に、付記を開示する。

(付記1)

コンピュータに、

表示画面幅を条件として規定された複数のスタイルのうち、表示画面幅の条件を満たす条件充足スタイルを適用してコンテンツ画面が表示された状態で、スタイル変更指示を受けた場合に、書換情報を取得し、前記条件充足スタイル以外のスタイルが適用されるように前記書換情報に基づいて前記複数のスタイルのそれぞれに対する条件を「真」又は「偽」に書き換える、

処理を実行させることを特徴とする表示変更プログラム。

【0082】

(付記2)

スタイル保存指示を受けた場合に、その時に適用されている書換情報とWebサイトのアドレスとを対応づけて記憶部に格納する処理を前記コンピュータに実行させることを特徴とする、付記1に記載の表示変更プログラム。

(付記3)

Webサイトのアドレスが前記記憶部に格納されていた場合に、前記記憶部から前記W

10

20

30

40

50

e bサイトのアドレスに基づいて前記書換情報を取得し、前記書換情報に基づいて前記複数のスタイルのそれぞれに対する条件を「真」又は「偽」に書き換える処理を前記コンピュータに実行させることを特徴とする、付記2に記載の表示変更プログラム。

【0083】

(付記4)

前記条件が「真」に書き換えられたスタイルを適用してコンテンツ画面を表示させる処理を前記コンピュータに実行させることを特徴とする、付記1～3のいずれか1項に記載の表示変更プログラム。

(付記5)

コンピュータが、

表示画面幅を条件として規定された複数のスタイルのうち、表示画面幅の条件を満たす条件充足スタイルを適用してコンテンツ画面が表示された状態で、スタイル変更指示を受けた場合に、書換情報を取得し、前記条件充足スタイル以外のスタイルが適用されるように前記書換情報に基づいて前記複数のスタイルのそれぞれに対する条件を「真」又は「偽」に書き換える、

処理を実行することを特徴とする表示変更方法。

【0084】

(付記6)

スタイル保存指示を受けた場合に、その時に適用されている書換情報とWebサイトのアドレスとを対応づけて記憶部に格納する処理を前記コンピュータが実行することを特徴とする、付記5に記載の表示変更方法。

(付記7)

Webサイトのアドレスが前記記憶部に格納されていた場合に、前記記憶部から前記Webサイトのアドレスに基づいて前記書換情報を取得し、前記書換情報に基づいて前記複数のスタイルのそれぞれに対する条件を「真」又は「偽」に書き換える処理を前記コンピュータが実行することを特徴とする、付記6に記載の表示変更方法。

【0085】

(付記8)

前記条件が「真」に書き換えられたスタイルを適用してコンテンツ画面を表示させる処理を前記コンピュータが実行することを特徴とする、付記5～7のいずれか1項に記載の表示変更方法。

(付記9)

表示画面幅を条件として規定された複数のスタイルのうち、表示画面幅の条件を満たす条件充足スタイルを適用してコンテンツ画面が表示された状態で、スタイル変更指示を受けた場合に、書換情報を取得する取得部と、

前記条件充足スタイル以外のスタイルが適用されるように前記書換情報に基づいて前記複数のスタイルのそれぞれに対する条件を「真」又は「偽」に書き換える書換部とを備えることを特徴とする表示変更装置。

【0086】

(付記10)

スタイル保存指示を受けた場合に、その時に適用されている書換情報とWebサイトのアドレスとを対応づけて記憶部に格納する保存処理部を備えることを特徴とする、付記9に記載の表示変更装置。

(付記11)

Webサイトのアドレスが前記記憶部に格納されていた場合に、前記記憶部から前記Webサイトのアドレスに基づいて前記書換情報を取得し、前記書換情報に基づいて前記スタイルに対する条件を「真」又は「偽」に書き換える再アクセス時書換部を備えることを特徴とする、付記10に記載の表示変更装置。

【0087】

(付記12)

10

20

30

40

50

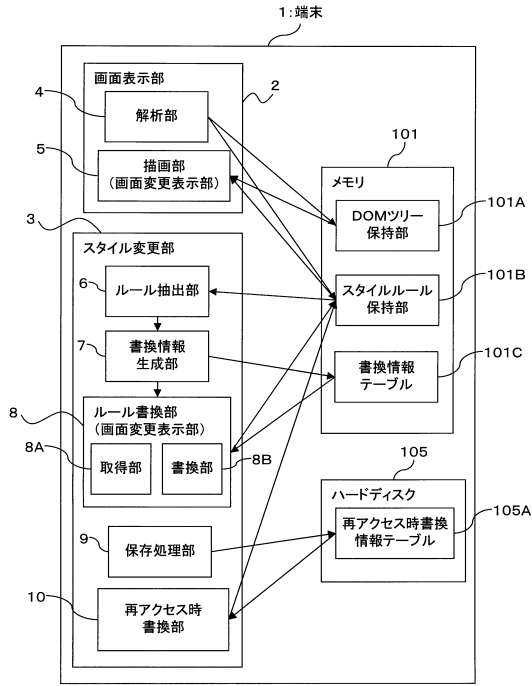
前記条件が「真」に書き換えられたスタイルを適用してコンテンツ画面を表示させる描画部を備えることを特徴とする、付記9～11のいずれか1項に記載の表示変更装置。

【符号の説明】

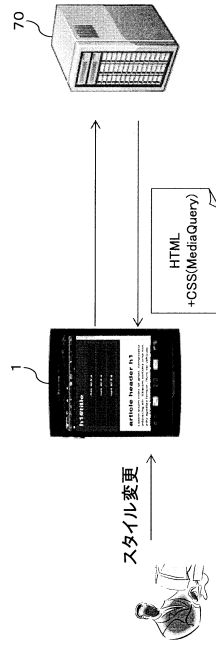
【0088】

- 1 端末
- 1 A 端末画面
- 1 B ダイアログボックス
- 2 画面表示部（ブラウザ）
- 3 スタイル変更部（表示変更装置）
- 4 解析部 10
- 5 描画部
- 6 ルール抽出部
- 7 書換情報生成部
- 8 ルール書換部
- 8 A 取得部
- 8 B 書換部
- 9 保存処理部
- 10 再アクセス時書換部
- 50 ブラウザ
- 50 A プログラム挿入機能（プログラム挿入部） 20
- 60 中継サーバ
- 60 A プログラム挿入機能（プログラム挿入部）
- 70 Webサイト（Webサーバ）
- 101 メモリ
- 101 A DOMツリー保持部
- 101 B スタイルルール保持部
- 101 C 書換情報テーブル
- 102 CPU
- 103 表示制御部
- 104 表示装置 30
- 105 記憶装置（ローカルディスク；ハードディスク；記憶部）
- 105 A 再アクセス時書換情報テーブル
- 106 入力装置
- 107 ドライブ装置
- 108 可搬型記録媒体
- 109 通信制御部
- 110 バス

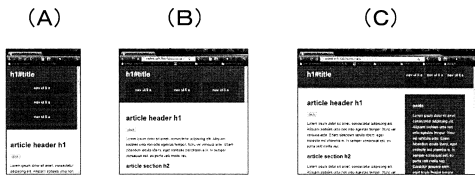
【図1】



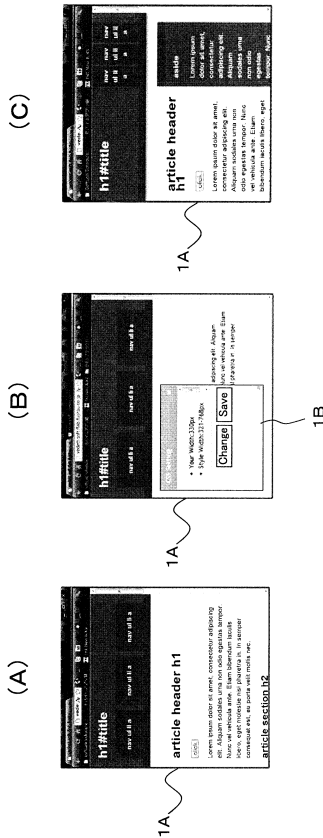
【図2】



【図3】



【図5】



【図4】

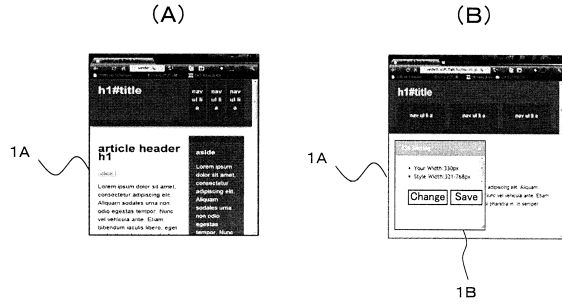
```

@media only screen and (min-width: 0px) and (max-width: 320px) {
/*S0: スマートフォン向けCSS */
}
@media only screen and (min-width: 321px) and (max-width: 769px) {
/*S1: タブレット向けCSS */
}
@media only screen and (min-width: 769px) {
/* S2: PC向けCSS */
}

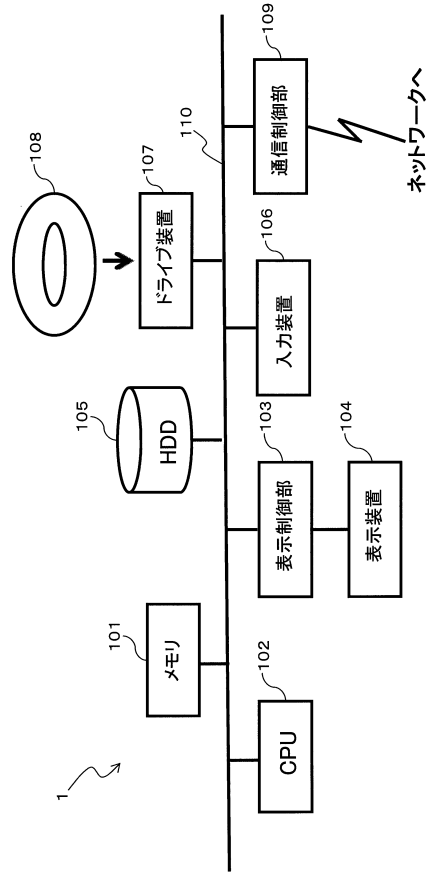
```



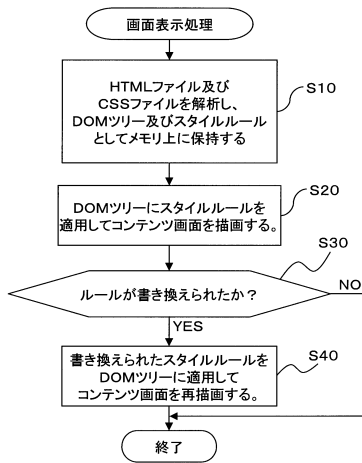
【図6】



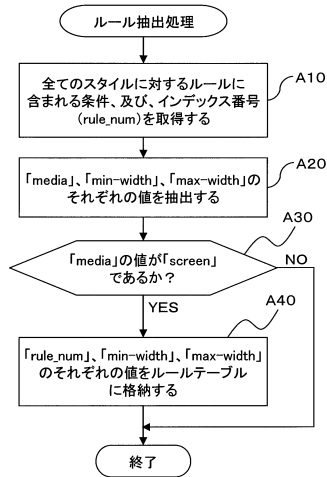
【図7】



【図8】



【図9】



【図 10】

```

(A) @media only screen and (min-width: 0px) and (max-width: 320px) {
    /*SO: スマートフォン向けCSS */
}

(B) var query = document.styleSheets[0].cssRules[0].media.mediaText;
    console.log(query);
    // "only screen and (min-width: 0px) and (max-width: 320px)"
    
```

【図 11】

```

var data = {
  media: query.match(/(only|s+)?([a-zA-Z]+)(#sand)?/ ) && RegExp.$2,
  minw: query.match(/(min*-width:[#s]*([#s]*[0-9]+px)[#s]**)/ ) && parseFloat( RegExp.$1 ),
  maxw: query.match(/(max*-width:[#s]*([#s]*[0-9]+px)[#s]**)/ ) && parseFloat( RegExp.$1 )
};
console.log(data); //{"maxw: 320, media: "screen", minw: 0}
    
```

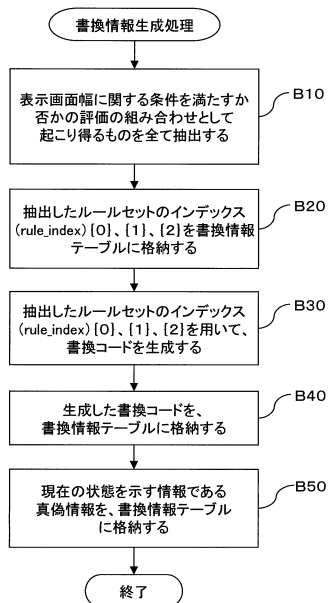
【図 12】

rule_num	min	max
0	0	320
1	321	768
2	769	*

【図 14】

set_index	rule_index	書換コード	状態
0	{0}	applyMQ({0: true},{1: false},{2: false});	false
1	{1}	applyMQ({0: false},{1: true},{2: false});	true
2	{2}	applyMQ({0: false},{false},{2: true});	false

【図 13】



【図 15】

```

@media only screen and (min-width: 0px) {
/*S0: スマートフォン向けCSS */
}
@media only screen and (min-width: 320px) {
/*S1: タブレット向けCSS */
}
@media only screen and (min-width: 768px) {
/* S2: PC向けCSS */
}
    
```

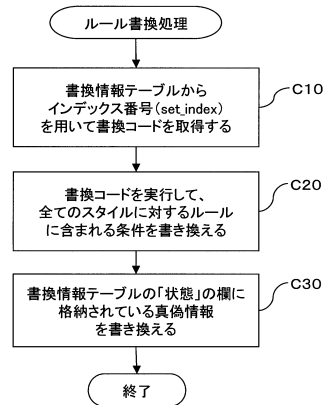
【図 16】

rule_num	min	max
0	0	*
1	320	*
2	768	*

【図 17】

set_index	rule_index	書換コード	状態
0	{0}	applyMQ{{{0: true},{1: false},{2: false}}};	false
1	{0,1}	applyMQ{{{0: true},{1: true},{2: false}}};	true
2	{0,1,2}	applyMQ{{{0: true},{1: true},{2: true}}};	false

【図 18】



【図19】

```

document.styleSheets[0].cssRule[0] の値 : only screen and (min-width: 0px) and
(max-width: 320px)
document.styleSheets[0].cssRule[1] の値 : only screen and (min-width: 321px) and
(max-width: 768px)
document.styleSheets[0].cssRule[2] の値 : only screen and (min-width: 769px)

(A)

document.styleSheets[0].cssRule[0] の値 : true
document.styleSheets[0].cssRule[1] の値 : false
document.styleSheets[0].cssRule[2] の値 : false

(B)

```

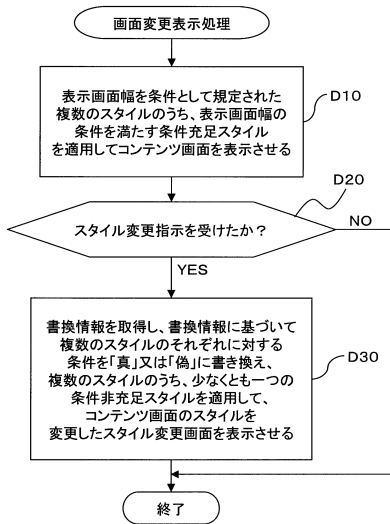
【図20】

```

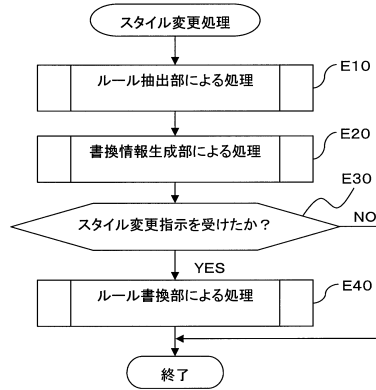
var applyMQ = function(obj) {
  for (var data in obj) {
    //MQに保持されている各スタイルに対する条件を書き換える
    document.styleSheets[0].cssRule[data].media.mediaText = obj[data];
  }
};

```

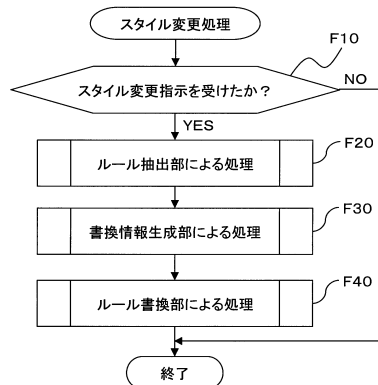
【図21】



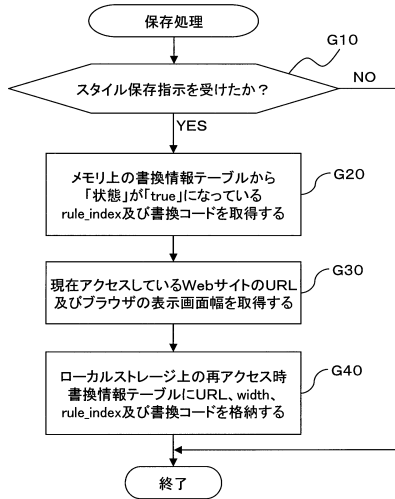
【図22】



【図23】



【図24】



【図25】

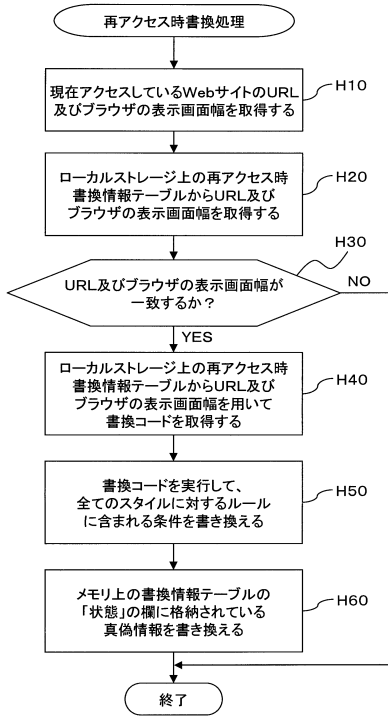
```

window.localStorage.setItem("data1.url",
"http://example.com/index.html");
window.localStorage.setItem("data1.width", 330);
window.localStorage.setItem("data1.rule_index", "2");
window.localStorage.setItem("data1.code",
"applyMQ({{0: false},{1:false},{2: true}});");
  
```

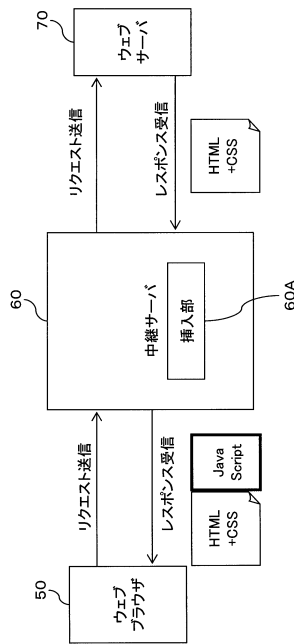
【図26】

書換コード	applyMQ({{0: false},{1:false},{2: true}});
rule_index	{2}
width	330
URL	http://example.com/index.html

【図27】



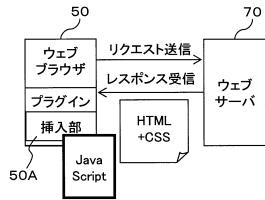
【図28】



【 図 29 】

```
<!doctype html>  
<html><head>  
...  
<script src="http://server/script.js">  
</script>  
</head>  
<body>  
...  
</body></html>
```

【 図 30 】



---

フロントページの続き

- (56)参考文献 特開2005-044304(JP,A)  
特開2001-142673(JP,A)  
特開2010-181971(JP,A)  
特開2004-164655(JP,A)  
国際公開第2012/141183(WO,A1)

- (58)調査した分野(Int.Cl., DB名)  
G06F 3/048