

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3762687号  
(P3762687)

(45) 発行日 平成18年4月5日(2006.4.5)

(24) 登録日 平成18年1月20日(2006.1.20)

|                   |                  |            |      |  |
|-------------------|------------------|------------|------|--|
| (51) Int. Cl.     |                  | F I        |      |  |
| <b>G06F 15/00</b> | <b>(2006.01)</b> | G06F 15/00 | 31OR |  |
| <b>G06F 12/00</b> | <b>(2006.01)</b> | G06F 12/00 | 546A |  |
| <b>G06F 13/00</b> | <b>(2006.01)</b> | G06F 13/00 | 550B |  |

請求項の数 6 (全 17 頁)

|              |                               |           |   |
|--------------|-------------------------------|-----------|---|
| (21) 出願番号    | 特願2001-336308 (P2001-336308)  | (73) 特許権者 | 390009531   |
| (22) 出願日     | 平成13年11月1日(2001.11.1)         |           | インターナショナル・ビジネス・マシー<br>ズ・コーポレーション                    |
| (65) 公開番号    | 特開2002-207697 (P2002-207697A) |           | INTERNATIONAL BUSIN<br>ESS MACHINES CORPO<br>RATION |
| (43) 公開日     | 平成14年7月26日(2002.7.26)         |           | アメリカ合衆国10504 ニューヨーク<br>州 アーモンク ニュー オーチャード<br>ロード    |
| 審査請求日        | 平成13年11月1日(2001.11.1)         | (74) 代理人  | 100086243   |
| (31) 優先権主張番号 | 2327159                       |           | 弁理士 坂口 博  |
| (32) 優先日     | 平成12年11月30日(2000.11.30)       | (74) 代理人  | 100091568   |
| (33) 優先権主張国  | カナダ(CA)                       |           | 弁理士 市位 嘉宏   |

最終頁に続く

(54) 【発明の名称】 HTMLフォーム・エレメントを動的に表示するシステムおよび方法

(57) 【特許請求の範囲】

【請求項1】

サーバ・アプリケーション・プログラムがWebサーバに対して使用可能であるときに、マークアップ言語フォーム・エレメントをWebページ上に選択的に表示するために、前記Webサーバのコンピュータに、

(a) 前記マークアップ言語フォーム・エレメントを前記サーバ・アプリケーション・プログラムと関係付ける第1の構成ファイルをインストールするステップと、

(b) 前記サーバ・アプリケーション・プログラムが前記Webサーバに対して使用可能であるか否かを示す第2の構成ファイルをインストールするステップと、

(c) 前記第1の構成ファイル及び前記第2の構成ファイルを構文解析して、前記サーバ・アプリケーション・プログラムが前記Webサーバに対して使用可能である場合には、前記マークアップ言語フォーム・エレメントを含むマークアップ言語ドキュメントを生成するステップと、

(d) 前記マークアップ言語ドキュメントをWebクライアントへ送信するステップとを  
実行させるコンピュータ・プログラム。

【請求項2】

ステップ(c)は、前記サーバ・アプリケーション・プログラムが前記Webサーバに対して使用可能でない場合には、前記マークアップ言語フォーム・エレメントを含まないマークアップ言語ドキュメントを生成するステップを、さらに含む請求項1に記載のコンピュータ・プログラム。

10

20

## 【請求項 3】

ステップ(c)は、前記サーバ・アプリケーション・プログラムが前記Webサーバに対して使用可能でない場合には、使用不可のマークアップ言語フォーム・エレメントを含むマークアップ言語ドキュメントを生成するステップを、さらに含む請求項1に記載のコンピュータ・プログラム。

## 【請求項 4】

関連したサーバ・アプリケーション・プログラムがWebサーバに対して使用可能であるときに、マークアップ言語フォーム・エレメントをWebページ上に選択的に表示するコンピュータ・システムであって、

(a) 前記サーバ・アプリケーション・プログラムが前記Webサーバに対して使用可能であるか否かを表現する値を格納する記憶手段と、

(b) 前記サーバ・アプリケーション・プログラムが前記Webサーバに対して使用可能であるか否かを前記値に基づいて判別し、前記サーバ・アプリケーション・プログラムが前記Webサーバに対して使用可能である場合には前記マークアップ言語フォーム・エレメントを含むマークアップ言語ドキュメントを生成する、前記記憶手段に接続された処理手段と

を含むコンピュータ・システム。

## 【請求項 5】

前記処理手段は、前記サーバ・アプリケーション・プログラムが前記Webサーバに対して使用可能でない場合には、前記マークアップ言語フォーム・エレメントを含まないマークアップ言語ドキュメントをさらに生成する請求項4に記載のコンピュータ・システム

## 【請求項 6】

前記処理手段は、前記サーバ・アプリケーション・プログラムが前記Webサーバに対して使用可能でない場合には、使用不可のマークアップ言語フォーム・エレメントを含むマークアップ言語ドキュメントをさらに生成する請求項4に記載のコンピュータ・システム

## 【発明の詳細な説明】

## 【0001】

## 【発明の属する技術分野】

本発明は、一般的には、クライアント・サーバ・ベースのコンピュータ・ネットワークに関し、特に、クライアントに対してコンテンツを動的に与えるシステムおよび方法に関する。

## 【0002】

## 【従来の技術】

インターネットは、ユーザがネットワークを通じて対話するおよび情報を共用することを可能とするネットワーキング・プロトコルを実行するコンピュータによって相互接続される地理的に分散したコンピュータ・ネットワークのシステムである。特に、ワールド・ワイド・ウェブ(“WWW”)は、Webクライアントによって表示するためのいわゆる“Webページ”としてフォーマットされたドキュメントをサポートするWebサーバの集合である。WWW通信パラダイムは、一般的なクライアント・サーバ・モデルに基づき、インターネットおよびWebプロトコル(例えば、TCP/IPおよびHTTP)をサポートするWebサーバの使用によって促進される。

## 【0003】

Webクライアントは、その時WWWを通じてWebクライアントに対して提供されているWebサーバにWebページ・ドキュメントを要求することができる。Webページは、Webサーバを運用するホスト上のコンピュータ・データ・ファイルである。Webページは、情報の単一行もしくは複数ページを含むことができ、いかなるメッセージ、名前、ワード、音、ピクチャ、あるいはこれらのエレメントの組み合わせをも含むことができる。ユーザは、一般的に、Webクライアントにインストールされたソフトウェア・プログ

10

20

30

40

50

ラムであり、ユーザがインターネットをアクセスおよびサーチすることを可能にする、Webブラウザとして知られるツールを用いてWebページを表示する。Webブラウザは、グラフィカル（例えばNetscapeコーポレーションのNavigator<sup>TM</sup>、もしくはMicrosoftコーポレーションのInternet Explorer<sup>TM</sup>）、あるいはテキスト・ベース（例えばGopher）とすることができる。Webブラウザにおいて、情報は、他のWWWサイトを識別するハイパーテキスト・リンクと一緒に表示される。

【0004】

Webページは、ハイパーテキスト・マークアップ言語（“HTML”）と呼ばれる標準のコーディング言語を用いて作成される。拡張マークアップ言語（“XML”）等に加えてHTMLは、“マークアップ言語”と称されるコンピュータ言語のファミリのメンバーである。マークアップ言語は、プラットフォームに依存しないテキスト・ドキュメントのフォーマットを定義するコンピュータ言語である。HTMLのようなマークアップ言語の主な目的は、テキスト、イメージおよび他のサポート・メディアを含むドキュメントのコンテンツを表示する方法をWebブラウザに指示することである。

【0005】

HTMLドキュメントは、適用されるべきフォーマットあるいはドキュメント・テキストに関連して起こるべき他のアクションを指示するテキスト・タグを含む。これらのタグは、使用されている特定マークアップ言語における意味を定義した文字ストリングである。したがって、HTMLは、Webページの様々なコンポーネントを定義する、プラットフォーム独立タグの集合である。HTMLドキュメントは、一般的に静的である、すなわち、ドキュメントが手動で変更される場合を除いて、コンテンツが長期にわたって変わらない。したがって、異なるHTMLドキュメントを提供するためには、HTMLドキュメントを生成することができるサーバ側プログラムをインストールすることが必要である。HTMLフォームは、1以上のテキスト入力ボックス、クリックできるボタン、複数の選択チェック・ボックス、およびプルダウン・メニューおよびイメージ・マップから成る。1以上のHTMLフォームをHTMLドキュメントに含めることができる。各HTMLフォームは、テキストおよびイメージを含む、HTMLドキュメントと同じコンテンツを含むことができる。

【0006】

典型的に、Webサーバ・アドミニストレータは、そこから一定のアプリケーション・プログラムがWebサーバに対するインストールのために具体的に選択され得るサーバ・アプリケーション・プログラムのスーツ（集合）（例えば、IBMのWebsphere<sup>TM</sup> コマース・スーツ・バージョン4.1）を販売される。いったん、これらのサーバ・アプリケーション・プログラムがインストールされると、それらは、WebサーバによってホストされるWebサイトの機能を拡張するために使用可能である。これらのサーバ・アプリケーション・プログラムへのアクセスは、Webブラウザによって表示されるボタンまたはスクロールダウン・メニューのようなHTMLフォーム・エレメントの使用によってユーザに提供可能である。

【0007】

Webサーバ・アドミニストレータは、また、Webサーバにインストールされている様々なサーバ・アプリケーション・プログラムに対する対話式アクセスをWebクライアントに提供するHTMLドキュメントを生成する標準のWebサーバ・プログラムを（典型としてJava（R）で）与えられる。Webサーバ・アドミニストレータが、Webサーバに対して特定のサーバ・アプリケーション・プログラムをインストールまたはアンインストールしようとする場合には、彼らは、ユーザに対して表示されるWebページに関連したHTMLフォーム・エレメント（例えば、ボタンを使用可能にするアプリケーション）を与えるか、または削除するかのいずれかを行うことを望み得る。そうするためには、一般的には、Webサーバ・アドミニストレータが、Webクライアントに対して適切なHTMLドキュメントを提供するためにWebサーバ・プログラムを変更する必要があ

10

20

30

40

50

る。このタイプのプログラム・ファイル・メンテナンスは、とりわけ、Webサーバに対して日常的にインストールおよびアンインストールされる多数のサーバ・アプリケーションが存在する場合には、極めて面倒であり、かつ時間を浪費する。

【0008】

【発明が解決しようとする課題】

したがって、関連したサーバ・アプリケーション・プログラムがWebサーバにインストールされているか否かに依存して、WebクライアントのWebページ上でのHTMLフォーム・エレメントの選択的表示に備えるシステムおよび方法の必要性が存在する。

【0009】

【課題を解決するための手段】

したがって、1つの態様における本発明の目的は、オブジェクトがWebサーバに対して使用可能であるときに、マークアップ言語フォーム・エレメントをWebページ上に選択的に表示する方法であって、前記方法は、(a)前記マークアップ言語フォーム・エレメントを前記オブジェクトと関係付けるステップと、(b)前記オブジェクトが前記Webサーバに対して使用可能であるか否か判別するステップと、(c)前記オブジェクトが前記Webサーバに対して使用可能である場合には、前記マークアップ言語フォーム・エレメントを含むマークアップ言語ドキュメントを生成するステップと、(d)前記マークアップ言語ドキュメントが前記マークアップ言語フォーム・エレメントを含む場合には、前記マークアップ言語フォーム・エレメントを前記Webページ上に表示するステップとを含む方法を提供することである。

10

20

【0010】

他の態様において本発明は、関連したオブジェクトがWebサーバに対して使用可能であるときに、マークアップ言語フォーム・エレメントをWebページ上に選択的に表示するコンピュータ・システムであって、前記コンピュータ・システムは、(a)マークアップ言語ドキュメントを生成するサーバ、前記サーバは、(i)前記オブジェクトが前記Webサーバに対して使用可能であるか否かを表現する値を格納する記憶手段と、(ii)(A)前記オブジェクトが前記Webサーバに対して使用可能であるか否か判別し、(B)前記オブジェクトが前記Webサーバに対して使用可能である場合には前記マークアップ言語フォーム・エレメントを含むマークアップ言語ドキュメントを生成する、前記記憶手段に接続された第1の処理手段とを含み、(b)前記サーバに接続された通信ネットワーク、(c)前記マークアップ言語ドキュメントを読み取り表示する第2の処理手段であって、前記通信ネットワークを経て前記サーバに接続された第2の処理手段、を備えるコンピュータ・システムを提供する。

30

【0011】

記憶装置に格納され、あるいは例えばコンピュータ・システムにおいて使用される変調キャリア信号上の、もしくはコンピュータ・システムにおいて使用されるインターネットのようなネットワーク上の信号として伝送され得るコンピュータ・プログラムにおいて本発明を実施できるということが当業者によって理解され得る。

【0012】

【発明の実施の形態】

本発明のより良い理解のために、そしてどのようにしてそれを実施に移すことができるのかをより明確に示すために、例として、本発明の好適な実施の形態を示す添付の図面を参照されたい。

40

【0013】

その内部で本発明を実施することができる周知のインターネット・コンピュータ・ネットワーク10の概念図を示す図1を最初に参照する。インターネット10は、クライアント・サーバ・モデルに基づき、そして通信ネットワーク16を通じて複数のWebクライアント14によってアクセス可能なWebサーバ12の大規模ネットワークを含む。ワールド・ワイド・ウェブ(“WWW”)は、HTTPを使用するWebサーバ12の集合である。HTTPは、標準のページ記述言語または“マークアップ言語”、例えばハイパーテ

50

キスト・マークアップ言語(“HTML”)もしくは拡張マークアップ言語(“XML”)を使用する(テキスト, グラフィック, イメージ, 音, ビデオ等のような異なるフォーマットであり得る)ファイルへのアクセスをユーザに提供する既知のアプリケーション・プロトコルである。

【0014】

図2は、通信ネットワーク16を通じて接続されたWebサーバ12およびWebクライアント14を含むコンピュータ・システム19を示す。Webサーバ12およびWebクライアント14のハードウェア・コンポーネントがより詳細に示される。

【0015】

Webサーバ12は、例えば、プロセッサ20, メモリ22(例えばRAM), プロセッサ20およびメモリ22を接続するバス24, 入出力・コントローラ28によってプロセッサ20およびメモリ22へ接続された大容量記憶装置26(例えば磁気または光ディスク)および一般的なモデムのようなネットワーク・インターフェース29等の一般的なコンポーネントを含む。Webサーバ12は、典型的に、通信ネットワーク16(例えばインターネット, イン트라ネットまたは他のネットワーク)へ接続され、Webサーバ・ソフトウェアがインストールされたサーバ・コンピュータ(例えばUNIX(R)またはWindows(R) NTベースのコンピュータ)より成る。

10

【0016】

Webクライアント14は、Webブラウザ・プログラム15を実行できる他のコンピュータのいかなるパーソナル・コンピュータであってもよく、例えば、プロセッサ30, メモリ32(例えばRAM), プロセッサ30およびメモリ32を接続するバス34, 入出力・コントローラ38によってプロセッサ30およびメモリへ接続された大容量記憶装置36(例えば磁気ハード・ディスクまたは光記憶ディスク)およびネットワーク・インターフェース39(例えば一般的なモデム)等の一般的なコンポーネントを含む。Webクライアント14のオペレーティング・システムは、Windows(R), OS/2, またはLinuxとすることができる。メモリ32は、HTTPに従うWebブラウザ15(例えば、カリフォルニア州マウンテンビューのNetscape CommunicationsによるNetscape<sup>TM</sup> Navigatorあるいはワシントン州レドモンドのMicrosoftコーポレーションによるInternet Explorer<sup>TM</sup>等)を含む多数のインターネット・アクセス・ツールをサポートする。述べたように、Webブラウザ15は、Webクライアント14がWebサーバ12と通信して、HTMLドキュメントへのアクセスおよびその表示をできるようにすることを可能にするソフトウェア・プログラムである。

20

30

【0017】

Webサーバ12は、Netscape, MicrosoftあるいはApacheによって提供されるようなWebサーバ・ソフトウェアを実行できるコンピュータである。Webサーバ12は、HTMLドキュメントおよびページの形式のファイルをサポートする“Webサイト”を運用する。Webサーバ・ソフトウェアは、Webサーバ12のホスト名とWebサーバ12によって格納されたHTMLドキュメントのロケーションと他のサーバ上に格納されたそれらのものとで構成される。

40

【0018】

HTMLドキュメント40のロケーションは、URL(“Uniform Resource Locator”)と呼ばれるアドレスによって定義される。Webブラウザ15は、特定URLを使用してインターネットおよびWWWからリソースを検索し取り出す。HTMLコードに加えて、HTMLドキュメント40は、典型的にはWebサーバ12に存在するが、通信ネットワーク16へ接続された他のコンピュータ上に存在してもよい他の情報コンテンツ42、例えば、イメージ, 音声, ビデオ, 実行可能プログラム等(“イメージ”42として示される)を組み込むことが可能である。好適な実施の形態において、HTMLドキュメント40およびイメージ42は、Webサーバ12のファイル・システム内のファイルとして格納される。HTMLドキュメント40は、ファイルあるいは他

50

のインターネット・リソースのロケーションを指定するHTMLタグを使用するイメージ42を組み込む。

【0019】

したがって、Webページをアクセスするために、Webページ要求が、Webクライアント14上のWebブラウザ15を通してUniform Resource Locator (“URL”)を指定するユーザによって入力される。ユーザは、Webページ・リンクをマウスを用いて“クリック”することによってWWWを通じてURLを送信することが可能であり、あるいは、ユーザは、全体のURLアドレスをWebブラウザ15へ手動で入力することが可能である。Webクライアント14のユーザがWebクライアント14からURLを送信する場合、このURLは送信され、(通信ネットワーク16を通じて)WWW中を回り、URLにおいて指定されるWebサーバ12にコンタクトする。具体的には、Webクライアント14は、Webサーバ12に対してTCP/IP要求を行う(すなわち、URLがHTTPを用いてWebサーバ12へ送信される)。アプリケーションによって与えられるURLは、Webサーバ12上に格納されるHTMLドキュメント40を指示する。

10

【0020】

Webサーバ12は、要求されたHTMLドキュメント40を探し出し、それをWebクライアント14上のブラウザに対してリターンすることによって応答する。Webブラウザ15は、HTMLドキュメント40を受信し、HTMLコードを解釈し、それらをWebページ(すなわち、HTMLにしたがってフォーマットされたドキュメント)に変換し、Webクライアント14上にWebページを表示する。HTMLドキュメント40を表示するために、ブラウザ制御は、ハイパーテキスト・ビューア(図示されない)に要請して、ドキュメント・ビューアのインスタンスを作成する。このドキュメント・ビューアは、それを通してWebブラウザ15が指定されたフレーム(図示されない)におけるドキュメントの表示を管理するプログラミング・インターフェース(特にドキュメント・オブジェクト・インターフェース)を有する。Webブラウザ15からの要求に応答して、ハイパーテキスト・ビューアは、HTMLコードを構文解析してレンダラーし、そしてHTMLドキュメント40および関連したグラフィカル・コンテンツをフレーム内部に表示する。

20

【0021】

図3は、13として概略的に示された、Webブラウザ15に対する典型的なユーザ・インターフェースの画面取りである。Webクライアント14上に生成されるユーザ・インターフェース13は、HTMLフォーム・エレメント21a, 21b, 21cおよび21dの使用によって、Webサーバ12上にインストールされている多数のアプリケーション・プログラムに対するアクセスをユーザに提供する。ユーザは、通常、Webブラウザ15を利用して、WWWに対して行われたURL要求から受信した全てのデータ・コンテンツ(すなわち、HTMLドキュメント40)をブラウズできる。

30

【0022】

図2および3を参照すると、Webサーバ12のアドミニストレータは、典型としてWebサーバ12に対するインストールのためのサーバ・アプリケーション・プログラムのバンドルを与えられる。これらのサーバ・アプリケーション・プログラムは、したがってWWWを通じてWebクライアント14に提供され得る。Webサーバ・アドミニストレータは、また、Webクライアント14による表示のためのHTMLフォーム・エレメント(例えばボタン)を含むHTMLドキュメントを生成し、ユーザがサーバ・アプリケーション・プログラムをアクセスすることを可能にする標準のWebサーバ・プログラム(例えばJava(R)プログラム)を与えられる。Webサーバ・アドミニストレータが、アプリケーション・プログラムのバンドルのサブセットをWebサーバ12にインストールするだけの場合には、一定のアプリケーション・プログラムは、Webクライアント14による実行に利用できず、対応するHTMLフォーム・エレメントはユーザに対して表示されないこととなる。

40

50

## 【0023】

例示として、多数のサーバ・アプリケーション・プログラム、すなわち、DELETE ROW, CREATE ROW, MODIFY ROWおよびGRAPH ROWがWebサーバ12にインストールされている状態を考慮されたい。図3に示されるように、関連したHTMLフォーム・エレメント21a, 21b, 21cおよび21dは、ユーザ・インターフェース13によって表示される。具体的には、delete rowアプリケーション・プログラムは、DELETE ROW ボタン21aとして表示され、create rowアプリケーション・プログラムは、CREATE ROW ボタン21bとして表示され、modify rowアプリケーション・プログラムは、MODIFY ROW ボタン21cとして表示され、graph rowアプリケーション・プログラムは、GRAPH ROW ボタン21dとして表示される。ユーザは、単に適切なHTMLフォーム・エレメント(例えばボタン)を選択することにより、これらのサーバ・アプリケーション・プログラムのうちのどれでも1つを実行することが可能である。

10

## 【0024】

graph row サーバ・アプリケーション・プログラムがWebサーバ12にインストールされていない場合には、サーバ・アプリケーション・プログラムは実行に利用できないので、HTMLフォーム・エレメント(すなわちGRAPH ROW ボタン)はユーザに対して示されないこととなる。それがユーザに対して実際に示される場合には、Webブラウザ15によるその表示は、混乱を引き起こし、グラフィカル・ユーザ・インターフェースの全体的な効果を損なう結果となり得る。HTMLフォーム・エレメントは、どんなタイプのフォーム・エレメント(例えばプルダウン・メニュー・エレメント, スクロール・リスト等)でもよく、ボタン・エレメントは、例示のために限って本説明において使用されるということを理解されたい。

20

## 【0025】

図4を参照すると、その内部で、HTMLフォーム・エレメントの動的表示が本発明の好適な実施の形態に従って達成可能であるコンピュータ・システム50が示される。Webサーバ12にインストールされた表示サーバ・プログラム52は、Webクライアント14のWebブラウザ15による処理のためのHTMLドキュメント40を生成するために使用される。HTMLフォーム・エレメント構成ファイル54aおよびコンポーネント構成ファイル54bは、実行時に表示サーバ・プログラム52によって構文解析される。

30

## 【0026】

HTMLフォーム・エレメント構成ファイル54aは、HTMLフォーム・エレメントおよびそれらの関連したコンポーネント(すなわち、それらが表すサーバ・アプリケーション・プログラム)の記述を含む。コンポーネント構成ファイル54bは、問題となっているサーバ・アプリケーション・プログラムのインストール状況(すなわち、インストールされているあるいはアンインストールされている)を含む。本発明は、IBMのWebSphere<sup>TM</sup> コマース・スーツ・バージョン5.1内部で実現され、特定のサーバ・アプリケーション・プログラムのインストール状況を対応するHTMLフォーム・エレメントと関係付けることによってHTMLフォーム・エレメントの動的表示に備える。

## 【0027】

表示サーバ・プログラム52は、HTMLドキュメントを生成できるあらゆるタイプのソフトウェア・プログラム言語(例えば、Sun MicrosystemsのJava(R) Server Pages<sup>TM</sup>テクノロジー(“JSP”), MicrosoftのActive Server Page<sup>TM</sup>言語(“ASP”), C++言語等)内で実現可能である。例示として、表示サーバ・プログラム52は、JSPプログラムとして書かれていると仮定されたい。JSPプログラムは、典型的に、XMLのようなタグとJava(R)プログラム言語で書かれたスクリプトレットとを含み、ページのためのコンテンツを生成する論理をカプセル化する。

40

## 【0028】

HTMLフォーム・エレメントおよびコンポーネント構成ファイル54aおよび54bは

50

、あらゆるタイプのマークアップ言語（例えばHTML, XML等）内で実現可能である。しかしながら、例示として、それらはXMLで書かれていると仮定されたい。述べたように、HTMLフォーム・エレメント構成ファイル54aは、HTMLフォーム・エレメントの記述を含み、コンポーネント構成ファイル54bは、問題となっているサーバ・アプリケーション・プログラムのインストール状況（すなわち、インストールされているあるいはアンインストールされている）を含む。

【0029】

説明するように、本発明の方法は、サーバ・アプリケーション・プログラムがインストールされるかまたはアンインストールされるかのいずれかの場合には、Webサーバ・アドミニストレータが、HTMLフォーム・エレメントおよびコンポーネント構成ファイル54aおよび54bを適切に修正変更することによって様々なサーバ・アプリケーション・プログラムのインストール状況を反映する現行のHTMLドキュメントを生成することを可能にする。このアクションは、新しい構成ファイルを手動でインストールするか、あるいは適切なサーバ・システム・ユーティリティのような自動手段を用いるかのいずれかによって達成可能である。

10

【0030】

HTMLフォーム・エレメント構成ファイル54aは、表示のためにコンピュータ・システム50によって使用される特定HTMLフォーム・エレメントを記述するデータ構造を含む。以下の実施例XML擬似コードによって説明されるように、HTMLフォーム・エレメント（例えばボタン）は、名前属性（すなわちボタン名）および関連したコンポーネント（すなわち関連したサーバ・アプリケーション・プログラム）を有すると定義される。

20

【0031】

【表1】

\\name = the name of the button  
 \\component = the component associated with the button  
 \\action = the action to perform when the button is pressed

<action>

<button name = "sampleButtonName"  
 component = "sampleComponent"  
 action = "sampleButtonAction2()" />

30

</action>

【0032】

コンポーネント構成ファイル54bは、Webサーバ12にインストールされていてもよく、されていなくてもよい特定コンポーネント（すなわちサーバ・アプリケーション・プログラム）を記述するデータ構造を含む。以下の実施例XML擬似コードによって説明されるように、コンポーネント・エレメントは、名前属性（すなわちアプリケーション・プログラム名）および使用可能属性（すなわち、インストールされている場合には“真”そしてインストールされていない場合には“偽”の値を用いて）を有すると定義される。

40

【0033】

【表2】



`\\name = the name of the component`

`\\enabled = true/false depending on whether the component is enabled or not`

```
<components>
  <component name = "sampleComponent"

    enabled = "true"/>
</components>
```

10

#### 【 0 0 3 4 】

図 4 に示されるコンピュータ・システム 5 0 のオペレーションを、コンピュータ・システム 5 0 の Web サーバ 1 2 および Web クライアント 1 4 に存在するプロセスによって実行される、図 5 および 6 において説明される高レベル動作処理ステップに関して説明する。これらのアプリケーションは、サーバ・アプリケーション・プログラムが Web サーバ 1 2 にインストールされているか否かを表すデータを、対応する HTML フォーム・エレメントと関係付け、どのサーバ・アプリケーション・プログラムが Web サーバ 1 2 にインストールされているかを反映する適切な HTML ドキュメント 4 0 を生成する。

#### 【 0 0 3 5 】

図 4 および 5 を最初に参照すると、コンピュータ・システム 5 0 の Web サーバ 1 2 および Web クライアント 1 4 によって実行される全体の処理ステップ 1 0 0 が説明される。

20

#### 【 0 0 3 6 】

ステップ 1 0 2 において、Web サーバ 1 2 は、Web ページ要求が Web クライアント 1 4 から受信されたか否かを判別する。Web クライアント 1 4 のユーザが、URL を Web クライアント 1 4 からコンタクトしている Web サーバ 1 2 へ送信した場合には、ステップ 1 0 4 において、表示アプリケーション・プログラム 5 2 が実行される（すなわち、JSP がサプレットへコンパイルされて実行される）。ステップ 1 0 6 において、実行時における表示アプリケーション・プログラム 5 2 は、HTML フォーム・エレメントおよびコンポーネント構成ファイル 5 4 a および 5 4 b を構文解析する。ステップ 1 0 8 において、表示アプリケーション・プログラム 5 2 は、次に、HTML フォーム・エレメントおよびコンポーネント構成ファイル 5 4 a および 5 4 b 内に記録されたコンポーネントの状況を反映し得る新しい HTML ドキュメント 4 0 を生成する。最後に、ステップ 1 1 0 において、ユーザ Web ページ要求に回答して、Web サーバ 1 2 は、新たに生成された HTML ドキュメント 4 0 を Web クライアント 1 4 に対して送信する。ステップ 1 1 2 において、Web クライアント 1 4 にインストールされている Web ブラウザ 1 5 は、HTML ドキュメント 4 0 を読み取り、ユーザに対して表示する。

30

#### 【 0 0 3 7 】

図 4 および 6 を参照すると、コンピュータ・システム 5 0 の Web サーバ 1 2 によって実行されて自動化された方法で現行 HTML フォーム・エレメントおよびコンポーネント構成ファイル 5 4 a および 5 4 b を保守する一般処理ステップ 2 0 0 が説明される。処理ステップ 2 0 0 は、タイム・スケジュール・ベースのシステムに回答して実行され（すなわち、5 分ごとの検査）、あるいは“保守検査”イベントに回答して実行可能である（すなわち、システムは、新しいアプリケーション・ソフトウェアがインストールまたはアンインストールされる時はいつでも“ウェイクアップ”して検査するようにプログラムされる）。

40

#### 【 0 0 3 8 】

ステップ 2 0 2 において、Web サーバ 1 2 は、サーバ・アプリケーション・プログラムが最近インストールされたかあるいはアンインストールされたか否かを判別する。すなわち、インストールされているサーバ・アプリケーション・プログラムがアンインストールされた、あるいはアンインストールされているサーバ・アプリケーション・プログラムが

50

インストールされたか否かである。そうでない場合には、処理は終了する。そうである場合には、ステップ204が実行される。ステップ204において、アンインストールされているサーバ・アプリケーション・プログラムがWebサーバ12に最近インストールされたか否かが判別される。そうである場合には、ステップ206において、XMLコンポーネント・エレメントの使用可能属性に値“真”を割当てて。そうでない場合には、ステップ208において、XMLコンポーネント・エレメントの使用可能属性に値“偽”を割当てて。ステップ207において、Webサーバ12は、処理すべきそれ以上の変更点（すなわち、それ以上のインストールあるいはアンインストール）が存在するか否かを判別する。そうである場合には、ステップ204が実行される。そうでない場合には、処理は終了する。

10

**【0039】**

図7および8は、上述したHTMLフォーム・エレメント構成ファイル54aおよびコンポーネント構成ファイル54bによって与えられるデータ構造と共同して表示サーバ・プログラム52によって実行される一般処理ステップを説明する。

**【0040】**

図7を最初に参照すると、表示サーバ・プログラム52によって実行される全体の処理ステップ300が示される。具体的には、ステップ302において、表示サーバ・プログラム52は、HTMLフォーム・エレメント構成ファイル54aを読み取る。ステップ304において、HTMLフォーム・エレメントと関連したコンポーネントが検索され、一時変数“C”に格納される。ステップ306において、HTMLフォーム・エレメントと関連したコンポーネントが使用可能であるかまたは使用不可であるかを判別するために、コンポーネント使用可能ルーチンが実行される。

20

**【0041】**

ステップ308において、結果が“真”の場合には、対応するHTMLコードが、Webブラウザ15による表示のためにHTMLドキュメント40内部に生成される。そうでない場合には、そのHTMLフォーム・エレメントに関してはHTMLドキュメント40内部に含まれるものは何もない。ステップ312において、検索されるコンポーネントが存在するか否かが判別される。コンポーネントが存在しない場合には、処理は終了する。コンポーネントが存在する場合には、ステップ304、306、308（および場合により310）が繰り返される。このようにして、HTMLフォーム・エレメント構成ファイル54a内部にリストされる全てのHTMLフォーム・エレメントが検索され、関連したコンポーネントの使用可能性状況が検査される。

30

**【0042】**

図8を参照すると、表示サーバ・プログラム52のコンポーネント使用可能サブルーチンによって実行される処理ステップ400が示される。上述したように、コンポーネント使用可能サブルーチンは、全体の処理300（図7）の中からステップ306において呼び出され、問題となっているHTMLフォーム・エレメントと関連するコンポーネントが表示に対して使用可能か否かを評価し、“真”または“偽”ブール値のいずれかをリターンする。

**【0043】**

具体的には、ステップ402において、表示サーバ・プログラム52は、コンポーネント構成ファイル54bを読み取る。ステップ404において、表示サーバ・プログラム52は、その内部にコンポーネント・データを格納するコンポーネント・ハッシュ・テーブルを作成する。ステップ406において、コンポーネント・エレメントの名前および使用可能属性の値はハッシュされ、ハッシュ・テーブル内部に格納される。ステップ408において、各コンポーネント・エレメントについて、使用可能属性の値が検査され評価される（ステップ410において）。使用可能属性の値が“真”の場合には、“真”がリターンされる（ステップ412において）。使用可能属性の値が“偽”の場合には、“偽”がリターンされる（ステップ414において）。コンポーネントが定義されていない場合には、コンポーネントは使用可能値を割当てられることはなく、結果“真”がリターンされる

40

50

こととなるということを留意されたい。これは、Webサーバ・アドミニストレータが不注意によってコンポーネントを定義するのを忘れる場合には、関連したHTMLフォーム・エレメントは、依然としてデフォルトに従って表示されることとなるということを保証する。

【0044】

例示として、表示サーバ・プログラム52の実施例をJSP擬似コードで以下に与える。特に、下記のもは、どのようにしてコンポーネント使用可能ルーチンを実施できるかを説明する。

【0045】

【表3】

10

```
// this part of the JSP program defines the isEnabled element
* Determines if the component comp is enabled or disabled
* These components are defined in XML and either enabled or disabled
//
```

```
public static boolean isEnabled(String comp)
{
```

```
    Hashtable components = new Hashtable();
```

```
// Pseudo Code start
```

```
Load XML file which defines the components and whether they are enabled or disabled
```

```
Put XML data into components Hashtable as name value pairs
```

```
name = 'Name of Component'
```

```
value = 'True or False depending on whether the component is enabled or not'
```

```
// Pseudo Code end
```

```
// retrieve component comp from xml and return true or false
```

```
// depending on the value set in xml
```

```
Boolean b = (Boolean)components.get(comp);
```

```
if (b != null)
```

```
{
```

```
    return b.booleanValue();
```

```
}
```

```
else
```

```
{
```

```
    return true;
```

```
}
```

```
}
```

20

30

【0046】

下記のもは、WebクライアントのWebページ上での表示のためにHTMLドキュメントを生成する時に、図4の表示サーバ・プログラムによって読み取られるHTMLフォーム・エレメント構成ファイル54aの実施例抜粋、実施例1である(XMLコードで実施されている)。

40

【0047】

【表4】

## //EXAMPLE 1- Sample HTML Form Element Configuration XML File

```

<action>
    <button name="DELETE ROW" component="delete row"/>
    <button name="CREATE ROW" component="create row"/>
    <button name="MODIFY ROW" component="modify row"/>
    <button name="GRAPH ROW" component="graph row"/>
</action>

```

10

## 【 0 0 4 8 】

示されているように、HTMLフォーム・エレメント構成ファイル54aは、HTMLフォーム・エレメント（例えば様々なボタン）すなわち、ボタン名属性および関連したコンポーネント属性の値の組のXML記述を含む。図7の処理フローチャートに関して述べたように、HTMLフォーム・エレメント構成ファイル54aは、HTMLフォーム・エレメント（例えば表示ボタン）のセットを識別するために表示サーバ・プログラム52によって使用される。各HTMLフォーム・エレメントについて、表示サーバ・プログラム52は、関連したコンポーネント・エレメントが使用可能か否かを判別する。

## 【 0 0 4 9 】

下記のは、WebクライアントのWebページ上での表示のためにHTMLドキュメントを生成する時に、図4の表示サーバ・プログラムによって読み取られるコンポーネント構成ファイルの実施例抜粋、実施例2である（XMLコードで実施されている）。

20

## 【 0 0 5 0 】

## 【表5】

## //EXAMPLE 2 - Sample Component Configuration XML File

```

<components>
    <component name="delete row"
    enabled="true"/>
    <component name="create row"
    enabled="true"/>
    <component name="modify row"
    enabled="true"/>
    <component name="graph row"
    enabled="false"/>
</components>

```

30

## 【 0 0 5 1 】

示されているように、コンポーネント構成ファイル54bは、コンポーネント・エレメント、すなわち、コンポーネント名属性および使用可能属性の値の組のXML記述を含む。図8の処理フローチャートに関して述べたように、コンポーネント構成ファイル54bは、上述したHTMLフォーム・エレメント構成ファイル54aによって規定される様々なHTMLフォーム・エレメントに関連するコンポーネント・エレメント（すなわち、様々なサーバ・アプリケーション・プログラム）のインストール状況を検査する（すなわち、使用可能属性の値を検査することによって）ために表示サーバ・プログラム52によって使用される。

40

## 【 0 0 5 2 】

最後に、実施例1および実施例2のHTMLフォーム・エレメント構成ファイル54aおよびコンポーネント構成ファイル54bを用いて実行される時に、表示サーバ・プログラム52によって生成され得るHTMLコードは、DELETE ROW, CREATE

50

ROWおよびMODIFY ROWボタンだけを、Webクライアント14のユーザ・インターフェース13(図3)上に表示させ得る。“graph row”コンポーネント・エレメントの使用可能属性の値は偽である(すなわち、データをグラフで表すためのサーバ・アプリケーション・プログラムはWebサーバ12上にインストールされていない)ので、HTMLコードは、関連したHTMLフォーム・エレメントに関して生成されない(すなわち、“GRAPH ROW”ボタンに関するHTMLコードは生成されず、HTMLドキュメント40内部に含まれない)。

#### 【0053】

したがって、図4のコンピュータ・システム50と図5,6,7および8において説明された関連する方法とは、関連したサーバ・アプリケーション・プログラムがWebサーバ12にインストールされているか否かに依存するWebクライアント14のWebページ上でのHTMLフォーム・エレメントの選択的表示に備える。このようにして、本発明は、Webサーバ管理者に、Webページ上に表示されるHTMLフォーム・エレメントがWebサーバ12にインストールされているサーバ・アプリケーション・プログラムと直接関連するということを保証するWebページ保守ツールの使いやすさを提供する。

#### 【0054】

サーバ・アプリケーション・プログラムがHTMLフォーム・エレメントとの1:1の関係を有する状態に関して本発明を説明してきたが、サーバ・アプリケーション・プログラムを、複数のHTMLフォーム・エレメントと関係付けることが可能であり、あるいは反対に、HTMLフォーム・エレメントを複数のサーバ・アプリケーション・プログラムと関係付けることが可能であるということが意図される。これらの関連付けは、追加的なエントリをHTMLフォーム・エレメントおよびコンポーネント構成ファイル54aおよび54b内部に加えて追加的な関連付けを設定することによって簡単な方法で本発明のシステムおよび方法の範囲内で実現可能である。表示サーバ・プログラム52を、それがHTMLフォーム・エレメントおよびコンポーネント構成ファイル54aおよび54b内部の複数のエントリを認識して処理するようにわずかに修正可能である。

#### 【0055】

HTMLドキュメント40内部にHTMLフォーム・エレメントを単に表示するまたは隠す代わりに、あるHTMLタグの使用不可属性(例えば<button>タグの使用不可属性)を使用することによってHTMLフォーム・エレメントを単に“使用不可にする”オプションをWebサーバ管理者に提供することが可能であるということも意図される。すなわち、HTMLフォーム・エレメントをサーバ・アプリケーション・プログラムと関係付け、構成ファイル内のHTMLフォーム・エレメントおよびコンポーネント(すなわち、サーバ・アプリケーション・プログラム・インストール・データ)を保守する手法を使用して、“使用不可”HTMLフォーム・エレメントを有するHTMLドキュメントを生成することが可能である。結果は、サーバ・アプリケーション・プログラムが実行に利用できない時に、使用不可のHTMLフォーム・エレメント(例えば陰影をつけたボタン)を表示するWebページ画面となることになる。関連したアプリケーションが利用できない場合の使用不可のHTMLフォーム・エレメントの表示は、ユーザに混乱を生じさせ、ユーザ・インターフェースの全体的な効果を損ねることが知られているが、このアプローチは、ある状況において好ましいことがある。

#### 【0056】

アプリケーション・プログラムに基づく使用不可または使用可機能として本発明を説明してきたが、当業者は、Webサーバ12が利用できるあらゆるオブジェクトを同種の方法で与え得るということを理解できる。例えば、インストールされていてもよく、されていなくてもよいMP3あるいはWAVファイルである。インストールされていることに関しては、Webサーバ12が利用できることを意味し、必ずしもWebサーバ12に存在する必要はない。発明者は、インストールされているアプリケーション・プログラムだけの認識に本発明を限定することを意図しない。

#### 【0057】

10

20

30

40

50

本発明，添付の特許請求の範囲において定義される本発明の範囲から外れることなしに、ここに説明され例示された好適なおよび代替の実施の形態に対して様々な変更を行うことが可能であるということを理解されたい。

【図面の簡単な説明】

【図 1】その内部で本発明が実施されるコンピュータ・ネットワークのブロック図である。

【図 2】サーバおよびクライアント間の対話を説明する図 1 の Web サーバおよび Web クライアントの概略図である。

【図 3】図 2 の Web サーバにインストールされた多数のアプリケーション・プログラムに対するアクセスをユーザに提供する図 2 の Web サーバによって生成される HTML ドキュメントを表示する Web ブラウザの画面取りである。 10

【図 4】Web サーバにインストールされた表示サーバ・プログラムおよび構成ファイルを含む本発明の好適な実施の形態によるコンピュータ・システムの概略図である。

【図 5】Web ページ上に HTML フォーム・エレメントを選択的に表示するために図 4 の表示サーバ・プログラムおよび構成ファイルを用いる本発明の好適な方法を説明するフローチャートである。

【図 6】Web ページ上に HTML フォーム・エレメントを選択的に表示するために図 4 の表示サーバ・プログラムおよび構成ファイルを用いる本発明の好適な方法を説明するフローチャートである。

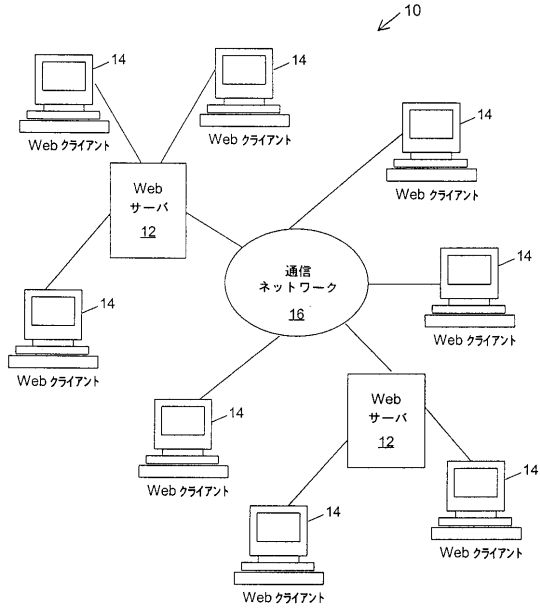
【図 7】Web クライアントの Web ページ上の表示のための HTML ドキュメントを生成する図 4 の表示サーバ・プログラムの処理ステップを説明するフローチャートである。 20

【図 8】Web クライアントの Web ページ上の表示のための HTML ドキュメントを生成する図 4 の表示サーバ・プログラムの処理ステップを説明するフローチャートである。

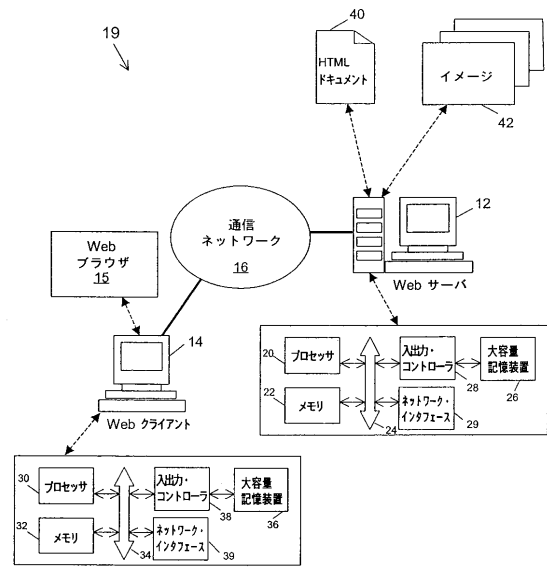
【符号の説明】

- 1 0 インターネット
- 1 2 Web サーバ
- 1 3 ユーザ・インターフェース
- 1 4 Web クライアント
- 1 5 Web ブラウザ・プログラム
- 1 6 通信ネットワーク 30
- 1 9 コンピュータ・システム
- 2 0 , 3 0 プロセッサ
- 2 1 a , 2 1 b , 2 1 c , 2 1 d ボタン
- 2 2 , 3 2 メモリ
- 2 4 , 3 4 バス
- 2 6 , 3 6 大容量記憶装置
- 2 8 , 3 8 入出力・コントローラ
- 2 9 , 3 9 ネットワーク・インターフェース
- 4 0 HTML ドキュメント
- 4 2 イメージ 40
- 5 0 コンピュータ・システム
- 5 2 表示サーバ・プログラム
- 5 4 a HTML フォーム・エレメント構成ファイル
- 5 4 b コンポーネント構成ファイル

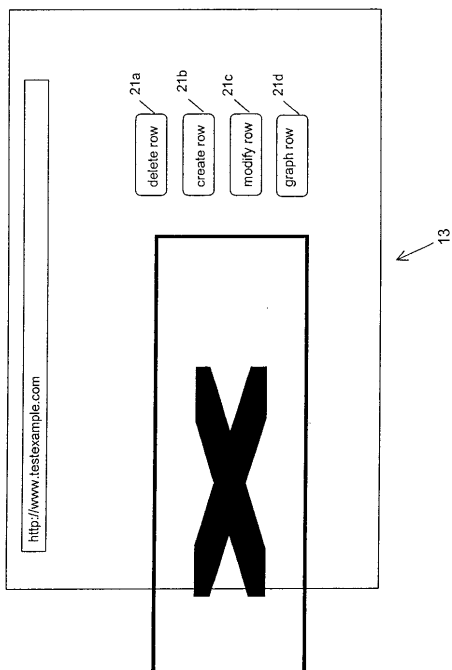
【 図 1 】



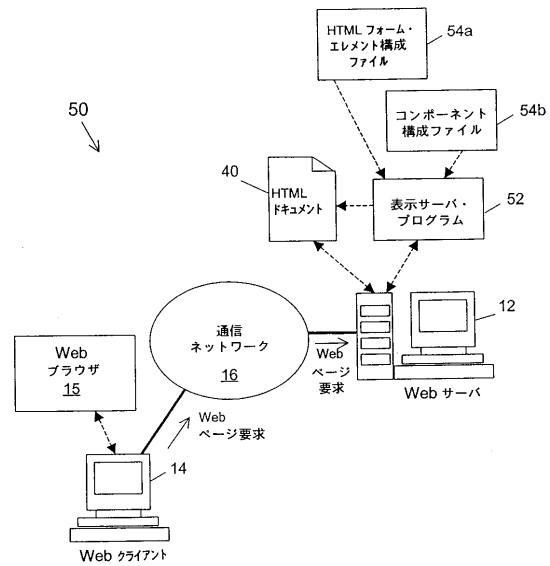
【 図 2 】



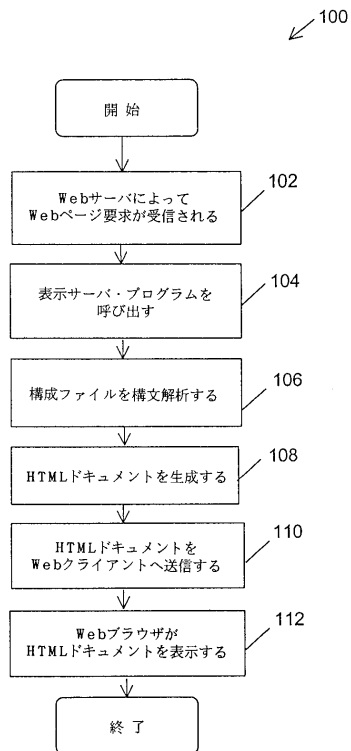
【 図 3 】



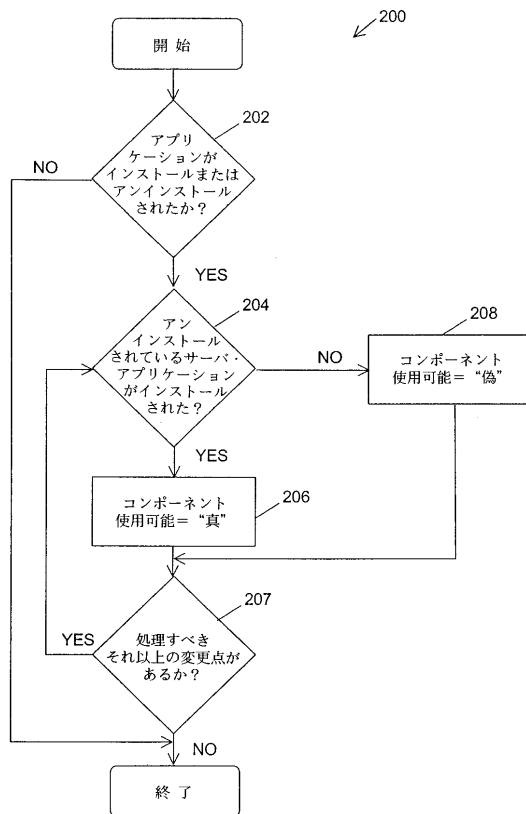
【 図 4 】



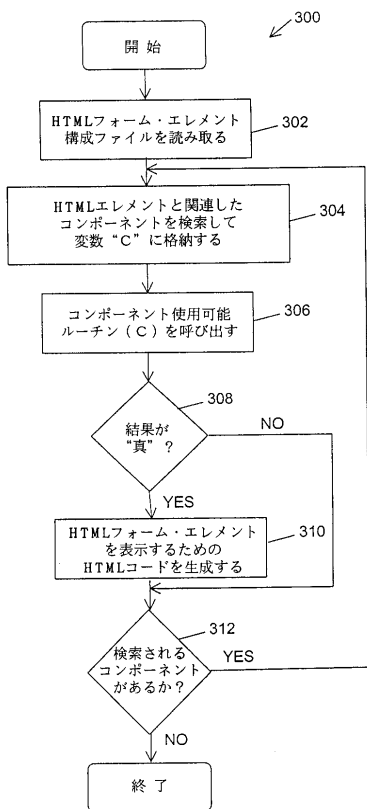
【図5】



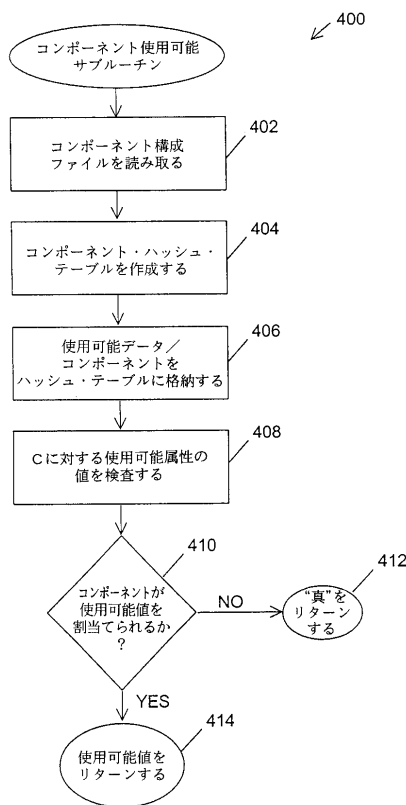
【図6】



【図7】



【図8】





---

フロントページの続き

(72)発明者 ディーン・エム・ヒルデブランド

カナダ M5X 1W7 オンタリオ州 トロント エイピーティール 2119 シェルブルヌ  
ストリート 565

審査官 宮司 卓佳

(56)参考文献 特開2000-298646(JP,A)

米国特許第6064977(US,A)

特開平9-223062(JP,A)

(58)調査した分野(Int.Cl., DB名)

G06F15/00,G06F17/30

G06F12/00,G06F13/00