

(19) 日本国特許庁(JP)

## (12) 公開特許公報(A)

(11) 特許出願公開番号

特開2004-259276  
(P2004-259276A)

(43) 公開日 平成16年9月16日(2004.9.16)

(51) Int.C1.<sup>7</sup>

G06F 13/00

G06F 12/00

F 1

G06F 13/00

G06F 12/00

560A

546T

テーマコード(参考)

5B082

審査請求 未請求 請求項の数 24 O L (全 24 頁)

(21) 出願番号 特願2004-48656 (P2004-48656)  
 (22) 出願日 平成16年2月24日 (2004. 2. 24)  
 (31) 優先権主張番号 10/373,362  
 (32) 優先日 平成15年2月24日 (2003. 2. 24)  
 (33) 優先権主張国 米国(US)

(71) 出願人 391055933  
 マイクロソフト コーポレイション  
 MICROSOFT CORPORATION  
 アメリカ合衆国 ワシントン州 9805  
 2-6399 レッドモンド ワン マイ  
 クロソフト ウェイ (番地なし)  
 (74) 代理人 100077481  
 弁理士 谷 義一  
 (74) 代理人 100088915  
 弁理士 阿部 和夫  
 (72) 発明者 マウリシオ ロメリンーストウピニャン  
 アメリカ合衆国 98121 ワシントン  
 州 シアトル エリオット アベニュー  
 2800

最終頁に続く

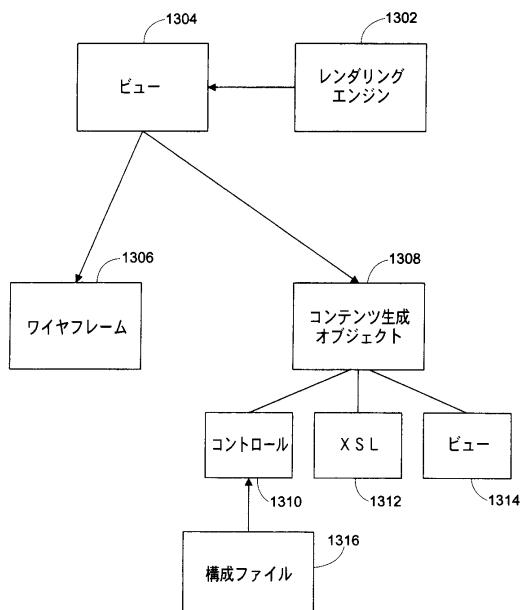
(54) 【発明の名称】コンテンツ項目を生成するためのシステム、方法および媒体

## (57) 【要約】

【課題】 コンテンツの様々なコンポーネントをモジュール化するウェブコンテンツを提示するためのアーキテクチャを提供する。

【解決手段】 ワイヤフレームは、コンテンツのページの空間領域(スロット)を定義する。ビューは、ワイヤフレーム内のスロット間のマッピングまたはバインディング、および、コンテンツを生成するオブジェクトを定義する。レンダリングエンジンはスロットを、スロットにバインドされたオブジェクトに基づいたコンテンツで埋める。スロットをコントロール、拡張可能スタイルシート言語(XSL)タグまたは他のビューにバインドさせることができる。コントロールは、スロットを埋めるためのコンテンツを生成する実行可能または解釈可能コードである。XSLプロセッサはそのようなコンテンツを、XSLタグに基づいて生成することもできる。

【選択図】 図13



**【特許請求の範囲】****【請求項 1】**

コンテンツ項目を生成するためのシステムであって、  
前記コンテンツ項目の複数の領域を定義する第1のワイヤフレームデータ構造と、  
コンテンツ生成オブジェクトの第1のセットと、  
前記コンテンツ生成オブジェクトの第1のセットを、前記第1のワイヤフレームによって定義された前記領域にマップする第1のビューと、  
前記第1のワイヤフレームによって定義された各領域について、  
前記ビューが前記領域にマップする前記コンテンツ生成オブジェクトに基づいてコンテンツを生成し、および  
前記生成されたコンテンツを前記領域に挿入する  
レンダリングエンジンと  
を備えたことを特徴とするシステム。

**【請求項 2】**

前記コンテンツ項目は、ハイパーテキストマークアップ言語または拡張可能ハイパーテキストマークアップ言語ウェブページを備えたことを特徴とする請求項1に記載のシステム。

**【請求項 3】**

前記ビューが前記領域にマップする前記コンテンツ生成オブジェクトは、実行可能または解釈可能コードを備えたコントロールであり、前記生成されたコンテンツが、前記実行可能または解釈可能コードの実行によって生成されることを特徴とする請求項1に記載のシステム。

**【請求項 4】**

構成ファイルをさらに備え、  
前記実行可能または解釈可能コードによって生成された前記コンテンツは、少なくとも部分的には前記構成ファイルに含まれたデータに基づくことを特徴とする請求項3に記載のシステム。

**【請求項 5】**

前記ビューは、前記実行可能または解釈可能コードが動作するモードを指定し、前記実行可能または解釈可能コードによって生成された前記コンテンツは、少なくとも部分的には前記指定されたモードに基づくことを特徴とする請求項3に記載のシステム。

**【請求項 6】**

前記ビューが前記領域にマップする前記コンテンツ生成オブジェクトは、拡張可能スタイルシート言語タグを備え、前記レンダリングエンジンは、拡張可能スタイルシート言語プロセッサを使用して前記拡張可能スタイルシート言語タグに基づいて、コンテンツを生成することを特徴とする請求項1に記載のシステム。

**【請求項 7】**

前記ビューが前記領域にマップする前記コンテンツ生成オブジェクトは、第2のワイヤフレームをコンテンツ生成オブジェクトの第2のセットに関連付けることによってコンテンツを指定する第2のビューを備え、前記生成されたコンテンツが、前記レンダリングエンジンが前記第2のビューを評価することによって生成されることを特徴とする請求項1に記載のシステム。

**【請求項 8】**

前記第1のワイヤフレームによって定義された前記領域の少なくとも1つが、前記第1のビューによっていずれのコンテンツ生成オブジェクトにもマップされず、前記レンダリングエンジンは、前記コンテンツ項目に、前記第1のワイヤフレームによって定義された前記領域の前記1つにおいてコンテンツがないことを反映させることを特徴とする請求項1に記載のシステム。

**【請求項 9】**

コンテンツ項目を生成する方法であって、

コンテンツ項目の領域を定義するワイドフレームのスロットのセットとコンテンツ生成オブジェクトのセットとの間の関連付けを指定するビューにアクセスすること、および

前記スロットの1つの各々について、前記ビューが前記スロットの1つに関連付ける前記コンテンツ生成オブジェクトに基づいて、前記スロットの1つにコンテンツを挿入すること

を含むことを特徴とする方法。

【請求項 10】

前記コンテンツ項目は、ハイパーテキストマークアップ言語または拡張可能ハイパーテキストマークアップ言語ウェブページを備えたことを特徴とする請求項9に記載の方法。

【請求項 11】

前記ビューが前記スロットの1つに関連付ける前記コンテンツ生成オブジェクトは、実行可能または解釈可能コードを含み、前記スロットの1つに挿入される前記コンテンツは、前記実行可能または解釈可能コードを実行することによって生成されることを特徴とする請求項9に記載の方法。

【請求項 12】

前記実行可能または解釈可能コードは、生成する前記コンテンツを決定するに際して、構成ファイルに含まれたデータを考慮に入れることを特徴とする請求項11に記載の方法。

【請求項 13】

前記実行可能または解釈可能コードは、生成する前記コンテンツを決定する際に、前記ビューで指定されたモードを考慮に入れることを特徴とする請求項11に記載の方法。

【請求項 14】

前記ビューが前記スロットの1つに関連付ける前記コンテンツ生成オブジェクトは、拡張可能スタイルシート言語タグを備え、前記方法は、

拡張可能スタイルシート言語プロセッサを使用して、前記拡張可能スタイルシート言語タグに基づいてコンテンツを生成することをさらに含むことを特徴とする請求項9に記載の方法。

【請求項 15】

前記ビューが前記スロットの1つに関連付ける前記コンテンツ生成オブジェクトはサブビューを備え、前記方法は、

前記サブビューを評価して、前記スロットの1つに挿入されるコンテンツを生成することをさらに含むことを特徴とする請求項9に記載の方法。

【請求項 16】

コンピュータ実行可能命令により、ウェブページを生成する方法を実行するように符号化されたコンピュータ可読媒体であって、該方法は、

ウェブページの複数の領域を備え、前記領域の各々が空またはコンテンツ生成オブジェクトの第1のセットの1つに対応している第1のワイドフレームと、前記コンテンツ生成オブジェクトの第1のセットとの間の対応を指定する第1のビューを受信すること、

前記第1のワイドフレーム内の前記領域の1つの各々について、

前記領域の1つがコンテンツ生成オブジェクトに対応している場合に、前記領域の1つに、前記コンテンツ生成オブジェクトに基づいて、コンテンツをウェブページに追加すること、および

前記領域の1つがコンテンツ生成オブジェクトに対応していない場合に、前記領域の1つにコンテンツがないことを反映するように、前記ウェブページを構築させることを含むことを特徴とするコンピュータ可読媒体。

【請求項 17】

前記コンテンツ生成オブジェクトのセットは、  
コンテンツを生成するように動作する実行可能または解釈可能コードと、  
拡張可能スタイルシート言語プロセッサに、生成または検索されるコンテンツを指定する拡張可能スタイルシート言語タグと、

10

20

30

40

50

第 2 のワイヤフレームとコンテンツ生成オブジェクトの第 2 のセットとの間の対応を指定する第 2 のビューと

のうちの 1 つまたは複数を備えたことを特徴とする請求項 16 に記載のコンピュータ可読媒体。

【請求項 18】

前記実行可能または解釈可能コードが生成する前記コンテンツは、構成ファイルに含まれたデータの関数であることを特徴とする請求項 17 に記載のコンピュータ可読媒体。

【請求項 19】

前記実行可能または解釈可能コードが生成する前記コンテンツは、前記第 1 のビュー内で前記実行可能または解釈可能コードについて指定されたモードの関数であることを特徴とする請求項 17 に記載のコンピュータ可読媒体。 10

【請求項 20】

コンテンツ項目の生成を容易にするためのデータ構造上で符号化しているコンピュータ可読媒体であって、前記データは、

前記コンテンツ項目の複数の領域を定義する第 1 のワイヤフレームの第 1 の識別子と、

前記第 1 のワイヤフレームによって定義された前記領域の少なくとも 1 つについて、前記領域の 1 つに関連付けられるコンテンツ生成オブジェクトを指定し、前記領域の 1 つを、前記領域の 1 つに関連付けられた第 2 の識別子によって識別し、さらに前記コンテンツ生成オブジェクトを、前記コンテンツ生成オブジェクトに関連付けられた第 3 の識別子によって識別するバインディングと 20

を備えたことを特徴とするコンピュータ可読媒体。

【請求項 21】

前記各第 1、第 2 および第 3 の識別子はテキスト文字列を含むことを特徴とする請求項 20 に記載のコンピュータ可読媒体。

【請求項 22】

前記コンテンツ生成オブジェクトは、前記領域の 1 つに挿入されるコンテンツを生成する実行可能または解釈可能コードを備えたことを特徴とする請求項 20 に記載のコンピュータ可読媒体。

【請求項 23】

前記コンテンツ生成オブジェクトは、拡張可能スタイルシート言語プロセッサに、生成または検索されるコンテンツを指定する拡張可能スタイルシート言語タグを備えたことを特徴とする請求項 20 に記載のコンピュータ可読媒体。 30

【請求項 24】

前記コンテンツ生成オブジェクトは、第 2 のワイヤフレームの領域をコンテンツ生成オブジェクトのセットに関連付ける第 2 のビューを備えたことを特徴とする請求項 20 に記載のコンピュータ可読媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は一般にコンピュータ技術の分野に関する。より詳細には、本発明は、ポータルサイトおよび検索エンジンによって作成されたウェブページのような、ウェブコンテンツを動的に作成するために使用することができるインフラストラクチャを提供する。 40

【背景技術】

【0002】

従来のウェブページは、ハイパーテキストマークアップ言語 (H T M L) または拡張可能ハイパーテキストマークアップ言語 (X H T M L) などのマークアップ言語で記述され、ウェブブラウザによってレンダリングするのに適したコンテンツを備える。いくつかのウェブページは静的であり、すなわち、一定で不变の H T M L または X H T M L コンテンツの一部を、クライアントマシンにダウンロードしてブラウザで表示することができるファイルに格納することができる。しかし、検索サイト、ポータルサイト、電子商取引サイ 50

トによって提供されたコンテンツなど、大部分の商用ウェブコンテンツは静的でなく、むしろ動的に生成され、ユーザに合わせてパーソナライズすることができるか、あるいはユーザからのある入力に応答して生成することができる。

#### 【0003】

動的に生成されるコンテンツは、その目的のために設計されたプログラムによって生成される。このようなプログラムは、ルールまたはプロシージャのセットに従ってコンテンツを作成する実行可能または解釈可能コードのモジュールである。たとえば、ユーザがクエリを検索エンジンサイトに送信するとき、検索エンジンは様々なソース（たとえば、既知のウェブページのデータベース）を調べて結果を生成し、次いで、検索エンジンサイトにあるコンテンツ生成プログラムが、結果を含むHTMLコンテンツのページを生成し、そのページをユーザに送信する。明らかに、このような結果ページは動的に生成されなければならず、これは結果ページのコンテンツが、どのようなクエリをユーザが出したかに応じて異なるからである。

#### 【0004】

既存のソフトウェアはウェブコンテンツを動的に生成することに有効であるが、このようなソフトウェアは、プログラムが生成する必要のあるコンテンツに創造的な変更があるとき、柔軟性の欠乏を欠点として有する。たとえば、検索エンジンは、その検索エンジンが探し出したウェブページのタイトルのリストとして検索結果を表示するプログラムを使用することができる。しかし、検索エンジンのオペレータが、これらの結果を提示する新しい方法（たとえば、このリストを、探し出したウェブページのサムネイル画像により向上させること）を導入することを望む場合、一般にこの変更を実現するための唯一の方法は、プログラムを書き換えること（または少なくとも、プログラムに、サムネイル画像の包含をサポートするコードを追加すること）である。

#### 【0005】

大部分のウェブコンテンツを、より小さなコンテンツ部品のモジュールである「ビルディングブロック」から構築されるものとみなすことができる。たとえば、検索結果ページは、コンテンツの様々な別々の部品からなる（たとえば、ロゴ、著作権表示、最新検索の結果、別の検索クエリを入力するための検索ボックス、広告など）。理論上は、ウェブコンテンツのモジュールの特性は、コンテンツ設計者が実行時間に提供するモジュールのビルディングブロックからコンテンツを構築することによって、単一のプログラムがコンテンツ生成プロセスを駆動できることを示唆する。コンテンツへの創造的な変更は、このようなプログラムへの変更を必要とせず、このプログラムが単に、異なるビルディングブロックを使用して異なるコンテンツを生成するように企図することができるからである。しかし、従来のコンテンツ生成ソフトウェアは、ソフトウェア自体を変更することを伴わず、どのような種類の創造的な変更をコンテンツに対して行なうことができるかに関して、極めて制限されている。

#### 【0006】

前述に鑑みて、従来技術の欠点を克服するシステムの必要性がある。

#### 【発明の開示】

##### 【発明が解決しようとする課題】

#### 【0007】

本発明は、ウェブページなど、コンテンツを生成するためのインフラストラクチャを提供する。このインフラストラクチャは、コンテンツのページを、別々に設計、実行および修正することができるより小さなコンポーネントから構築することができるという考えに基づいている。本発明により、これらのより小さなコンポーネント、ならびにこれらが組み込まれる方法を、コンテンツ生成ソフトウェアに実行時間で指定することができる。

#### 【課題を解決するための手段】

#### 【0008】

本発明によれば、コンテンツの作成は、「ワイヤフレーム」、「ビュー」およびコンテンツ生成オブジェクトの組み合わせによって指定される。ワイヤフレームは、コンテンツ

10

20

30

40

50

部品の空間領域を定義するデータ構造である。たとえば、ワイアフレームは、4つの矩形領域を有し、ページの最上部から最下部まで垂直に実行するように、ウェブページを定義することができる。これらの各領域は「スロット」と呼ばれる。「ビュー」は、ワイアフレーム内の各スロットをコンテンツ生成オブジェクト、または、コンテンツ生成オブジェクトのシーケンスにバインドする（あるいはまた、特定のスロットを空となるように指定する）データ構造である。

#### 【0009】

ビューをレンダリングするために、レンダリングエンジンは、ビュー内で識別されたワイアフレームにアクセスし、そのワイアフレーム内の各スロットについて、そのスロットにバインドされたコンテンツ生成オブジェクトにコンテンツを生成させる。次いで、レンダリングエンジンは、オブジェクトによって生成されたコンテンツをスロットに入れる。このプロセスは、ワイアフレーム内の各スロットについて繰り返される。レンダリングエンジンの出力は、コンテンツ部品（たとえば、HTMLウェブページ）であり、これはワイアフレームの適切な領域において各コンテンツ生成オブジェクトによって生成されたコンテンツを含む。

#### 【0010】

コンテンツ生成オブジェクトには、「コントロール」、拡張可能スタイルシート言語（XSL）タグ、およびビューが含まれる。コントロールは、スロットに入れられるHTMLまたはXHTMLコンテンツを生成する実行可能または解釈可能コードである。XSLタグは、XSL処理ソフトウェアによって使用され、HTMLまたはXHTMLコンテンツを生成することができるデータである。ビューを評価することで、結果としてコンテンツの生成となるので、ビュー自体がコンテンツ生成オブジェクトである。したがって、コンテンツを、実際には最上位ビューのサブビュー、サブサブビューなどであるビューから、階層的に構築することができる。レンダリングエンジンが、サブビューにバインドされるスロットに遭遇すると、レンダリングエンジンはそれ自体を再帰的に呼び出して、そのスロットのためのコンテンツをそのビューに基づいて生成することができる。

#### 【0011】

本発明の1つの特徴によれば、コントロールは、構成ファイルに含まれたパラメータに基づいて、異なる振舞を提示することができる。したがって、所与のコントロールを、どのパラメータが構成ファイル内で提供されるかに基づいて、コンテンツ部品の異なるバリエーションを生成するように設計することができる。

#### 【0012】

本発明の他の特徴を以下で説明する。

前述した概略、ならびに、以下の好ましい実施形態の詳細な説明は、添付の図面と共に読まれるとき、よりよく理解される。本発明を例示するため、図面において本発明の例示的構成を示すが、本発明は、開示された特定の方法および手段に限定されない。

#### 【発明を実施するための最良の形態】

#### 【0013】

##### 概要

多くの商用ウェブコンテンツは、定義された方法でグラフィカルに組み込まれる、より小さなコンテンツの部品から構築される。本発明は、コンテンツプロバイダが、どのようにコンテンツが構築されるかを、これらのより小さな部品に基づいて指定できるようにするインフラストラクチャを提供する。このインフラストラクチャは、レンダリングエンジンを含み、コンテンツをコンテンツプロバイダの仕様に基づいて動的に生成する。

#### 【0014】

##### 例示的コンピューティング環境

図1は、本発明の態様を実施することができる例示的コンピューティング環境を示す。コンピューティング環境100は、適切なコンピューティング環境の一例でしかなく、本発明の用途または機能性の範囲についてのいかなる限定も示唆するように意図されていない。コンピューティング環境100はまた、例示的オペレーティング環境100に例示し

10

20

30

40

50

たコンポーネントの任意の1つまたは組み合わせに関するいかなる依存性または必須要件も有するように解釈されるべきではない。

#### 【0015】

本発明は、多数の他の汎用または専用コンピューティングシステム環境または構成により動作可能である。本発明と共に使用するのに適しているだろう周知のコンピューティングシステム、環境および/または構成の例には、限定するものではないが、パーソナルコンピュータ、サーバコンピュータ、ハンドヘルドまたはラップトップデバイス、マルチプロセッサシステム、マイクロプロセッサベースのシステム、セットトップボックス、プログラマブルコンシューマエレクトロニクス、ネットワークPC、ミニコンピュータ、メインフレームコンピュータ、組込みシステム、上記のシステムまたはデバイスなどのいずれかを含む分散コンピューティング環境などが含まれる。

#### 【0016】

本発明を一般に、プログラムモジュールのような、コンピュータによって実行されるコンピュータ実行可能命令に関連して説明することができる。一般に、プログラムモジュールにはルーチン、プログラム、オブジェクト、コンポーネント、データ構造などが含まれ、これらは特定のタスクを実行するか、あるいは特定の抽象データ型を実装する。本発明はまた分散コンピューティング環境において実施することもでき、この環境ではタスクが、通信ネットワークまたは他のデータ伝送媒体を通じてリンクされるリモート処理デバイスによって実行される。分散コンピューティング環境では、プログラムモジュールおよび他のデータが、記憶デバイスを含むローカルおよびリモートのコンピュータ記憶媒体内に位置することができる。

#### 【0017】

図1を参照すると、本発明を実施するための例示的システムは、コンピュータ110の形態で、汎用コンピューティングデバイスを含む。コンピュータ110のコンポーネントには、それだけに限定されないが、処理装置120、システムメモリ130、および、システムメモリを含む様々なシステムコンポーネントを処理装置120に結合するシステムバス121を含むことができる。システムバス121は、いくつかのタイプのバス構造のいずれかにすることでき、これには、様々なバスアーキテクチャのいずれかを使用するメモリバスまたはメモリコントローラ、周辺バスおよびローカルバスが含まれる。例として、限定ではなく、このようなアーキテクチャには、ISA(Industrial Standard Architecture)バス、MCA(Micro Channel Architecture)バス、EISA(Extended Industrial Standard Architecture)バス、VESA(Video Electronics Standards Association)ローカルバス、およびPCI(Peripheral Components Interconnect)バス(メザニンバスとしても知られる)が含まれる。

#### 【0018】

コンピュータ110は通常、様々なコンピュータ可読媒体を含む。コンピュータ可読媒体は、コンピュータ110によってアクセスすることができる任意の使用可能な媒体とすることでき、これには、揮発性および不揮発性媒体、リムーバブルおよび非リムーバブル媒体が含まれる。例として、限定ではなく、コンピュータ可読媒体は、コンピュータ記憶媒体および通信媒体を含むことができる。コンピュータ記憶媒体は、揮発性および不揮発性、リムーバブルおよび非リムーバブルの媒体を含み、これらはコンピュータ可読命令、データ構造、プログラムモジュールまたは他のデータなど、情報の格納のためのいずれかの方法または技術において実施される。コンピュータ記憶媒体には、それだけに限定されないが、RAM(Random Access Memory)、ROM(Read Only Memory)、EEPROM(Electronically Erasable and Programmable Read Only Memory)、フラッシュメモリもしくは他のメモリ技術、CD-ROM、DVD(Digital Versatile Disk)もしくは他の光ディスク記憶、磁気カセット、磁気テープ、磁気ディスク記憶もしくは他の磁気記憶デバイス、または、所望の情報を格納するために使用することができ、コンピュータ110によってアクセスすることができる他のいかなる媒体もが含まれる。通信媒体は典型的に、コンピュータ可読命令、データ構造、プログラムモジュール、または他のデータを

10

20

30

40

50

、搬送波または他の伝送メカニズムなどの変調データ信号中に具現化する任意の情報配信媒体をも含む。「変調データ信号」という用語は、信号中に情報を符号化するような方法でその特性の1つまたは複数の組を有するか、または変化されている信号を意味する。例として、限定ではなく、通信媒体には、有線ネットワークまたは直接配線接続などの有線媒体、ならびに、音響、RF (Radio Frequency)、赤外線および他の無線媒体などの無線媒体が含まれる。上記のいずれの組み合わせも、コンピュータ可読媒体の範囲内に含まれるべきである。

#### 【0019】

システムメモリ130は、揮発性および/または不揮発性メモリの形態におけるコンピュータ記憶媒体を含み、これは読み取り専用メモリ(ROM)131およびランダムアクセスメモリ(RAM)132などである。BIOS(Basic Input/Output System)133は、起動中など、コンピュータ110内の複数の要素の間で情報を転送する助けとなる基本ルーチンを含み、通常はROM131に格納される。RAM132は通常、処理装置120によって即時アクセス可能および/または現在動作中であるデータおよび/またはプログラムモジュールを含む。例として、限定ではなく、図1は、オペレーティングシステム134、アプリケーションプログラム135、他のプログラムモジュール136およびプログラムデータ137を例示する。

#### 【0020】

コンピュータ110はまた、他のリムーバブル/非リムーバブル、揮発性/不揮発性のコンピュータ記憶媒体も含むことができる。例としてのみ、図1は、非リムーバブル、不揮発性の磁気媒体に対する読み書きを行なうハードディスクドライブ141、リムーバブル、不揮発性磁気ディスク152に対する読み書きを行なう磁気ディスクドライブ151および、CD-ROMまたは他の光媒体など、リムーバブル、不揮発性の光ディスク156に対する読み書きを行なう光ディスクドライブ155を例示する。例示的オペレーティング環境で使用することができる他のリムーバブル/非リムーバブル、揮発性/不揮発性のコンピュータ記憶媒体には、それだけに限定されないが、磁気テープカセット、フラッシュメモリカード、デジタル多用途ディスク、デジタルビデオテープ、半導体素子(solid state)RAM、半導体素子ROMなどが含まれる。ハードディスクドライブ141は通常システムバス121に、インターフェイス140などの非リムーバブルメモリインターフェイスを通じて接続され、磁気ディスクドライブ151および光ディスクドライブ155は通常システムバス121に、インターフェイス150などのリムーバブルメモリインターフェイスによって接続される。

#### 【0021】

上述し、図1に例示したドライブおよびそれらの関連付けられたコンピュータ記憶媒体は、コンピュータ110用のコンピュータ可読命令、データ構造、プログラムモジュールおよび他のデータの記憶装置を提供する。図1では、たとえば、ハードディスクドライブ141が、オペレーティングシステム144、アプリケーションプログラム145、他のプログラムモジュール146およびプログラムデータ147を格納するものとして例示される。これらのコンポーネントを、オペレーティングシステム134、アプリケーションプログラム135、他のプログラムモジュール136およびプログラムデータ137と同じものにも異なるものにもすることができますことに留意されたい。オペレーティングシステム144、アプリケーションプログラム145、他のプログラムモジュール146およびプログラムデータ147にはここで異なる番号が与えられて、最低でも、それらが異なるコピーであることが例示される。ユーザはコマンドおよび情報をコンピュータ110へ、キーボード162、および、一般にマウス、トラックボールまたはタッチパッドと呼ばれるポインティングデバイス161などの入力デバイスを通じて入力することができる。他の入力デバイス(図示せず)には、マイクロフォン、ジョイスティック、ゲームパッド、衛星放送受信アンテナ、スキャナなどが含まれる可能性がある。これらおよび他の入力デバイスがしばしば処理装置120へ、システムバスに結合されるユーザ入力インターフェイス160を通じて接続されるが、これを、パラレルポート、ゲームポートまたはユニ

10

20

30

40

50

バーサルシリアルバス(USB)など、他のインターフェイスおよびバス構造によって接続することができる。モニタ191または他のタイプの表示デバイスもシステムバス121へ、ビデオインターフェイス190などのインターフェイスを介して接続される。モニタに加えて、コンピュータはまた、スピーカ197およびプリンタ197など、出力周辺インターフェイス195を通じて接続することができる他の周辺出力デバイスを含むこともできる。

#### 【0022】

コンピュータ110はネットワーク環境において、リモートコンピュータ180など、1つまたは複数のリモートコンピュータへの論理接続を使用して動作することができる。リモートコンピュータ180は、パーソナルコンピュータ、サーバ、ルータ、ネットワークPC、ピアデバイスまたは他の共通ネットワークノードにすることができ、通常は、コンピュータ110に関連して上述した要素の多数またはすべてを含むが、記憶デバイス181のみを図1に例示している。図1に示す論理接続は、ローカルエリアネットワーク(LAN)171およびワイドエリアネットワーク(WAN)173を含むが、他のネットワークも含むことができる。このようなネットワーキング環境は、オフィス、企業全体のコンピュータネットワーク、インターネットおよびインターネットにおいて一般的である。

#### 【0023】

LANネットワーキング環境において使用するとき、コンピュータ110がLAN171へ、ネットワークインターフェイスまたはアダプタ170を通じて接続される。WANネットワーキング環境において使用するとき、コンピュータ110は通常、モデム172、またはインターネットなどのWAN173を介して通信を確立するための他の手段を含む。モデム172を内部または外部にすることができ、システムバス121へ、ユーザ入力インターフェイス160または他の適切なメカニズムを介して接続することができる。ネットワーク環境では、コンピュータ110に関連して示したプログラムモジュールまたはその一部を、リモート記憶デバイスに格納することができる。例として、限定ではなく、図1はリモートアプリケーションプログラム185を記憶デバイス181上に存在するものとして例示する。図示のネットワーク接続は例示的であり、通信リンクを複数のコンピュータの間で確立する他の手段を使用できることは理解されよう。

#### 【0024】

例示的ウェブブラウザ  
図2は、例示的ウェブブラウザ200のビジュアルインターフェイスを示す。当技術分野で知られているように、ウェブブラウザは、ユーザがあるタイプのコンテンツ(たとえば、HTMLコンテンツ)と対話し、このようなコンテンツをネットワークから検索して取ってくることができるようになる、1つのソフトウェアである。ブラウザ200は、たとえば、コンピュータ110(図1)上に格納され、処理装置120(図1)上で実行する、ソフトウェアを構成することができる。ブラウザ200は、インターネットなど、コンピュータ110が接続されるワイドエリアネットワーク173(図1)からのコンテンツにアクセスすることができる。典型的に、ブラウザ200はまた、コンピュータ110上にローカルに格納されているコンテンツにアクセスすることもできる。

#### 【0025】

図2の例示的ブラウザは様々な情報をユーザに表示する。詳細には、ブラウザ200はナビゲーションバー202をユーザに表示する。ナビゲーションバー202は、ボックス204を含み、この中にユーザがURL(Uniform Resource Locator)を入力することができ、これはブラウザを特定のコンテンツ項目に向けるためである。図2の例では、ユーザがURLのhttp://search.msn.comをボックス204に入力しており、それにより、ユーザがそのURLによって識別されたコンテンツにアクセスすることを望むことを示す。ブラウザ200は、このコンテンツを、インターネット上でそれが位置することができるどこからでも検索し、コンテンツをビューア206に表示する。図2の例では、コンテンツがHTMLまたはXHTMLで配信される。(URL

10

20

30

40

50

内の「`http`」という句は「ハイパーテキスト転送プロトコル」を意味し、URL内のその表現は、基礎となるコンテンツがHTMLまたはXHTMLコンテンツであることを示す。) ブラウザ200は、HTMLおよび/またはXHTMLインタプリタを含むか、そうでない場合はそれにアクセスし、ブラウザ200によって受信されたHTMLまたはXHTMLコンテンツをレンダリングする。図2の例では、ブラウザ200は、コンテンツのタイトル(「MSN Search - More Useful Every day」)をタイトルバー208に表示する。

#### 【0026】

ブラウザ200のような、典型的なブラウザは、表示されたコンテンツを印刷、メール送信または保存すること、「お気に入り」または「ブックマーク」のリストを使用して他のコンテンツにナビゲートすること、テキストのコンテンツ用のデフォルトフォントを変更することなど、ユーザが様々な機能を実行することができるようになる。この機能性はユーザに、メニューバー210および/またはボタン212を用いて並べられている。

#### 【0027】

以下で述べるように、本発明は、ビュイングエリア206に表示されるコンテンツのような、コンテンツを生成するためのインフラストラクチャを提供する。

#### 【0028】

##### コンテンツの提示を作成するためのデータ構造

本発明は、ウェブページに利用できるXHTMLコンテンツのような、コンテンツが作成されるプロセスを汎用化および抽象化するインフラストラクチャを提供する。(以降、ブラウザでレンダリング可能なコンテンツの例は、そのコンテンツをXHTMLコンテンツであるとして指すものとする。しかし、本発明は、HTML、ワイヤレスマークアップ言語(WML)、音声拡張マークアップ言語(VXML)など、他のタイプのコンテンツを生成するために使用できることは理解されよう。)

本発明のインフラストラクチャは4種類のデータ構造、すなわち、ワイヤフレーム、ビュー、コントロールおよび構成ファイルを使用する。

#### 【0029】

「ワイヤフレーム」は、コンテンツのページの空間領域を定義するデータ構造である。各領域は「スロット」と呼ばれる。

#### 【0030】

「コントロール」は、ワイヤフレームによって定義された特定のスロット内に入れられるXHTMLコンテンツを動的に生成する、実行可能または解釈可能コードを備える。

#### 【0031】

「ビュー」は、特定のスロットと特定のコントロール(または特定のコントロールのシーケンス)との間のマッピングを作成するデータ構造である。ビューはまた、拡張可能スタイルシート言語(XSL)タグまたは別のビューをスロットにマップすることもでき、(ビューはまた、XSLタグまたはビューのシーケンスをスロットにマップすることもできる)、XSLタグおよびビューの、スロットへのマッピングを、以下でより詳細に述べる。

#### 【0032】

「構成ファイル」は、コントロールによってそれらの振舞に影響を与えるために使用されるデータを含む。たとえば、コントロールは、コンテンツの特定の部分の2つ以上の異なるバリエーションを生成することができ、構成ファイルは、これらの異なるバリエーションのどれが生成されるべきかを指示するパラメータを含むことができる。

#### 【0033】

図3は例示的ワイヤフレーム300を示す。ワイヤフレーム300は、コンテンツの概念ページを、上述のように「スロット」と呼ばれる領域に分割する。図3の例では、4つのスロット、304(1)、304(2)、304(3)および304(4)がある。各スロットは名前を有し、この例では、スロット304(1)から304(4)の名前はそれぞれ「slot\_header」、「slot\_results」、「slot\_sidebar」および「slot\_footer」で

10

20

30

40

50

ある。ワイヤフレーム 300 自体も名前を有する（この例では「Wireframe\_Site」）。後述のように、名前は、ビューデータ構造でワイヤフレームおよび各スロットを識別するために使用される。

【0034】

典型的に、ワイヤフレームは X H T M L 構造として表現される。したがって、図 3 の例示的ワイヤフレームを以下のように定義することができる。

【0035】

【表 1】

```

<wireframe name="wireframe_site">
  <div><slot name="slot_header" /></div>
  <table>
    <tr>
      <td><slot name="slot_results" /></td>
      <td><slot name="slot_sidebar" /></td>
    </tr>
  </table>
  <div><slot name="slot_footer" /></div>
</wireframe>

```

10

20

30

40

【0036】

上記を見るとわかるように、上記の X H T M L 定義はコンテンツの 4 つの領域、すなわち、スロット名「slot\_header」を有する最上部の領域、スロット名「slot\_footer」を有する最下部の領域、ならびに、最上部および最下部の領域の間の 2 列のテーブルをレイアウトし、左側の列の名前は「slot\_results」であり、右側の列の名前は「slot\_sidebar」である。

【0037】

図 4 は、X H T M L コンテンツ 402 を生成する例示的コントロール 400 を示す。コントロール 400 は名前（この例では「Control\_Footer」）を有し、後述するように、名前はビューデータ構造においてコントロールを識別するために使用される。コントロール 400 を使用して、ワイヤフレーム 300 内の「slot\_footer」という名前のスロットに挿入されるコンテンツを生成することができる。

【0038】

コントロール 400 は、実行中に X H T M L コンテンツ 402 を動的に生成するように構成された実行可能または解釈可能コードを備える。コントロール 400 は様々な方法で実施されることができる。たとえば、コントロール 400 は、マシン実行可能コード（.e x e または .d l l ファイルなど）、解釈可能スクリプト（J a v a（登録商標）または C# 言語におけるスクリプトなど）、仮想マシンコード（J a v a（登録商標）バイトコードなど）などを備えることができる。

【0039】

以下は、コントロール 400 の関数を記述する擬似コードの一例である。（以下の擬似コードを上述の実行可能または解釈可能コンポーネントのいずれにおいても実施できることは、容易に理解されよう。）

【0040】

## 【表2】

```

XHTML Control_Footer()
{
    String strFooter;

    strFooter = _10
        "<TABLE WIDTH='770' HEIGHT='16' CELLPADDING='0'"
        "CELLSPACING='0' BORDER='0'>
        <TR><TD WIDTH='770' HEIGHT='1' BGCOLOR='#bdbebd'>
        <SPACER TYPE='block' HEIGHT='1' /></TD></TR>
        <TR><TD HEIGHT='15' class='nsf'>&nbsp; &nbsp;
        "
    strFooter = strFooter & file_read(copyright_notice.txt);

    strFooter = _20
        "</TD></TR></TABLE>

    return strFooter;
}

```

## 【0041】

上記の擬似コードは、コントロール400を、XHTMLコンテンツを生成して戻す「Control\_Footer」と名前付けされた関数として定義する。この例では、Control\_Footerは、特定のサイズおよび背景色を有し、かつ「Copyright 2003 Microsoft Corporation - All Rights Reserved」などのテキスト文字列（または、何であれファイル「copyright\_notice.txt」内に含まれるテキスト文字列）を含む、テーブルのためのXHTMLコードを戻す。

## 【0042】

上述のように、コントロールをワイアフレームのスロットに、「ビュー」と呼ばれる構造を用いてマップ（または「バインド」）させることができる。図5は、ワイアフレーム300のスロットへのコントロールの例示的マッピングを示す。この例では、コントロール420（「Control\_Header」という名前）がスロット304（1）にマップし、コントロール410（「Control\_Results」）がスロット304（2）にマップし、コントロール430（「Control\_Sidebar」）がスロット304（3）にマップし、コントロール400（図4に関連して上述した「Control\_Footer」）がスロット304（4）にマップする。このマッピングは、XHTMLコンテンツのページをワイアフレーム300に基づいて定義し、スロット304（1）、304（2）、304（3）および304（4）によって定義された各領域が、それぞれコントロール420、410、430および400によって生成されたXHTMLコンテンツで満たされる。このマッピングはビュー500を表現する。

## 【0043】

図5に示すビュー500を、以下のデータ構造として表現することができる。

## 【0044】

10

20

30

40

## 【表3】

```

<view name="view_Classic" wireframe="wireframe_site">
    <binding slot="slot_header">
        <ctl:Control_Header />
    </binding>
    <binding slot="slot_results">
        <ctl:Control_Results />
    </binding>
    <binding slot="slot_sidebar">
        <ctl:Control_Sidebar />
    </binding>
    <binding slot="slot_footer">
        <ctl:Control_Footer />
    </binding>
</view>

```

10

20

## 【0045】

上に示したデータ構造は、名前「view\_Classic」をビューに割り当てる。典型的なシステムはいくつかのビューを有し、名前は、ビューがいくつかのビューの中から識別されることを可能とする。見てわかるように、このデータ構造は、各スロットについての「バインディング」を含み、スロットに「バインド」される特定のコントロールを示す。

## 【0046】

上記の例は単一のコントロールが各スロットにバインドされることを示すが、複数のコントロールを単一のスロットにバインドさせて、ある順序で実行されるようにすることができることに留意されたい。したがって、「slot\_header」についてのバインディングを以下のように読むことができる。

30

## 【0047】

## 【表4】

```

<binding slot="slot_header">
    <ctl:Control_A />
    <ctl:Control_B />
    <ctl:Control_C />
</binding>

```

40

## 【0048】

このバインディングは、slot\_headerをコンテンツで満たすために、「Control\_A」、「Control\_B」および「Control\_C」という名前のコントロールが、示された順序で実行されることを示す。

## 【0049】

加えて、上記の例は、コントロールのみがスロットにバインドされることを示すが、前述のように、本発明はまたビューまたはXSLタグをスロットにバインドさせることも可能にする。スロットへのコントロールのバインディングは、最も単純なバインディングの例である。しかし、スロットバインディングの概念を汎用化することができ、図6～12

50

はこのような汎用化の一例を示す。

【0050】

ビューを使用してコンテンツを指定する例

上述のように、コントロールは、XHTMLコンテンツを生成するソフトウェアオブジェクトである。コントロールをスロットに「マップ」または「バインド」することは、そのコントロールによって生成されている XHTMLコンテンツでそのスロットが満たされることを意味する。しかし、コントロールは、XHTMLコンテンツを生成する唯一の種類のオブジェクトではない。詳細には、XSLタグを評価して XHTMLコンテンツを生成できることが当技術分野で知られている。加えて、ビューが評価されて適切なコンテンツがワイヤフレームのスロットに配置されるとき、この評価の結果は XHTMLコンテンツであり、そのためビューは、XHTMLコンテンツを生成するさらにもう1つのオブジェクトである。これらの事実を考慮すると、XSLタグおよびビュー、ならびに、XHTMLコンテンツを生成するために評価することができる他のいかなる種類のオブジェクトも、スロットにマップさせることができることがわかる。

【0051】

図6～11は、ワイヤフレーム300における「slot\_header」という名前のスロット(図3)にバインドさせることができるビューを構築する方法を示す。前の例のように、ビューは、特定のワイヤフレームについてのスロットバインディングを含み、ビューで使用されるワイヤフレームを定義することが必要である。図6は、「WireFrame\_Header」という名前のワイヤフレーム600を示す。ワイヤフレーム600は4つのスロット602(1)、602(2)、602(3)および602(4)を有し、それぞれ「slot\_big7」、「slot\_banner」、「slot\_sharkfin」および「slot\_querybox」という名前である。このワイヤフレームを、以下のデータ構造によって定義することができる。

【0052】

【表5】

```
<wireframe name="WireFrame_Header">
  <div><slot name="slot_big7" /></div>
  <div><slot name="slot_banner" /></div>
  <div><slot name="slot_sharkfin" /></div>
  <div><slot name="slot_querybox" /></div>
</wireframe>
```

【0053】

図7～10は、ワイヤフレーム600内のスロット602(1)から602(4)にバインドされる様々なオブジェクトを示す。

【0054】

図7は、「Big7」という名前のXSLタグ700を示す。XSLタグ700は、XSLデータに関連付けられ、適切なXSL処理ソフトウェアを用いて、search.msn.comウェブサイトの米国バージョンの最上部に現れる「ビッグ7」リンク702を含むXHTMLコンテンツを生成するために使用することができる。これらの「ビッグ7」リンクは、「MSNホーム」、「マイMSN」、「Hotmail」、「検索」、「ショッピング」、「マネー」および「ピープル&チャット」である。

【0055】

図8は、もう1つのXSLタグ800を示し、「Banner」という名前である。XSLタグ800をXSL処理ソフトウェアによって、XSLタグ700と同じ方法で評価することができる。しかし、XSLタグ800はバナー広告802を生成する。図8の例では、このバナー広告は、レドモンド大学のための広告である。

【0056】

10

20

30

40

50

図9は、「Control\_Sharkfin」という名前のコントロール900を示す。コントロール900は、曲線のグラフィック(参照符号902)の隣にMSNという文字を示すXHTMLコンテンツを生成し、これはsearch.msn.comウェブサイトに現れる。(このグラフィックの名前は、曲線が示唆するサメの背びれに由来する。)

【0057】

図10は、「View\_QueryBox」という名前のビュー1000を示す。ビュー500(図5に関連して上述)のように、ビュー1000は、ワイヤフレームの複数のスロットの間のマッピングまたはバインディング、および、これらのスロットのためのコンテンツを生成するオブジェクト(たとえば、コントロール、XSLタグまたはビュー)を表現する。図10の例では、ワイヤフレーム1002が2つのスロット1004(1)および1004(2)を有し、それぞれ「slot\_searchbox」および「slot\_polead」という名前である。ビュー1000は、スロット1004(1)「slot\_searchbox」を、その中にユーザがクエリを打ち込むことができるボックス1006、および、ユーザがクリックしてクエリを検索エンジンに送信することができる「go」ボタン1008のためのXHTMLコンテンツを生成するオブジェクトにバインドする。スロット1004(2)('slot\_polead')は、広告1010のためのXHTMLコンテンツを生成するオブジェクトにバインドされる。

【0058】

XSLタグ700および800、コントロール900、およびビュー1000を、ワイヤフレーム600(図6)のスロットにバインドするビューを作成することができる。図11は、このようなバインディングを示す。XSLタグ700('Big7')がスロット602(1)('slot\_big7')にバインドされる。XSLタグ800('Banner')がスロット602(2)('slot\_banner')にバインドされる。コントロール900('Control\_Sharkfin')がスロット602(3)('slot\_sharkfin')にバインドされる。ビュー1000('View\_QueryBox')がスロット602(4)('slot\_querybox')にバインドされる。このバインディングによって定義されたビュー1100を、以下のデータ構造として表現することができる。

【0059】

【表6】

```

<view name="view_header" wireframe="WireFrame_Header">
  <binding slot="slot_big7">
    <xsl:Big7 />    'XSL yields Big7 HTML
  </binding>
  <binding slot="slot_banner">
    <xsl:Banner />  'XSL yields Banner Ad creative
  </binding>
  <binding slot="slot_sharkfin">
    <ctl:Control_Sharkfin />
  </binding>
  <binding slot="slot_querybox">
    <view:View_QueryBox />
  </binding>
</view>

```

【0060】

10

20

30

40

50

上記のデータ構造は、ビュー 1100 に「view\_header」という名前を与え、図 11 に示すバインディングを作成する。上の例では、スロットにバインドされるオブジェクトの各タイプの前に、そのタイプについての指定が来る。たとえば「xsl:Banner」で「Banner」が XSL タグであることを示し、同様に「ctl:Control\_Sharkfin」および「view:View\_Querybox」がコントロールおよびビューを示す。このタイプの指定を使用することは好ましく、これはコントロール、XSL タグおよびビューが異なるタイプの処理を必要とし、この指定によりビューレンダリングエンジン（図 13～14 に関連して後述）が、どのタイプのオブジェクトを処理する必要があるかを容易に識別できるようになるからである。

#### 【0061】

上述のように、図 6～11 は、ビュー（ビュー 1100）がワイヤフレーム 300 内のスロット 304(1)（「slot\_header」）にバインドされる一例を示す。（ワイヤフレーム 300 のスロットを、これらのスロットを満たすコンテンツ生成オブジェクトにバインドするために、もう 1 つのビューが作成されなければならない。このビューには、スロット 304(1) とビュー 1100 の間のバインディングが含まれる。このようなビューの詳細は、これまでの考察から容易に理解することができる。）図 12 は、このバインディングに基づいて作成することができる、結果として生じるコンテンツを示す。図 12 はワイヤフレーム 300 を、その 4 つのスロット 304(1)、304(2)、304(3) および 304(4) と共に示す。スロット 304(1)（「slot\_header」）がビュー 1100 にバインドされるので、ワイヤフレーム 600（これにビュー 1100 が基づく）は、スロット 304(1) をオーバーレイするとして示される。ワイヤフレーム 600 は、4 つのスロット 602(1)、602(2)、602(3) および 602(4) を有する。これらの 4 つのスロットのうち、スロット 602(4) がビュー 1000（図 10）にバインドされ、ビュー 1000 はワイヤフレーム 1002 に基づく。したがって、ワイヤフレーム 1002 は図 12 ではスロット 602(4) をオーバーレイするとして示される。

#### 【0062】

図 12 から理解できるように、この図に示すコンテンツは階層を形成しており、ワイヤフレーム 1002 はワイヤフレーム 600 のスロットの内部にあり、ワイヤフレーム 600 はワイヤフレーム 300 のスロットの内部にある。図 12 のコンテンツを階層の最下部から上に解析すると、ビュー 1000 が以下のコンテンツ、すなわち、ボックス 1006 および「go」ボタン 1008（ワイヤフレーム 1002 のスロット 1004(1) 内）、ならびに広告 1010（スロット 1004(2) 内）を生成することがわかる。このコンテンツはワイヤフレーム 600 のスロット 602(4) 内に位置する。ワイヤフレーム 600 は、コンテンツ項目 702、802 および 902 も含み、図 7～9 で上述したように、XSL タグおよびコントロールから生成されたものである。したがって、ビュー 1100 は、ワイヤフレーム 600 のスロット内に含まれたコンテンツを生成する。ビュー 1100 は、ワイヤフレーム 300 のスロット 304(1) にバインドされているので、ビュー 1100 によって生成されたコンテンツがスロット 304(1) 内に配置される。図 12 は、スロット 304(2)～(4) も示し、これらのスロットをコントロール、XSL タグおよびビューに、上述の方法でバインディングするビューを指定することができる。図 6～12 の例は、各スロットが单一のコントロール、XSL タグまたはビューにバインディングされることを示すが、上述のように、スロットを複数のコントロール、XSL タグまたはビューに（またはそれらの任意の組み合わせにも）バインディングさせることができ、これらをすべて評価して所与のスロットを満たすことができる。

#### 【0063】

図 12 では、破線を使用してワイヤフレームおよびそれらのスロットのアウトラインを示すが、これらのラインは実際には、ビューから構築されるコンテンツに含まれないことに留意されたい。これらのラインは単に図 12 において、本発明によるコンテンツを構築することができる階層構造を明示するために示される。

#### 【0064】

10

20

30

40

50

コンテンツをビューから生成するためのアーキテクチャ

図13は、本発明によるコンテンツを生成するために使用される様々なコンポーネントの概要を示す。

【0065】

レンダリングエンジン1302は、ビュー1304を使用して、ブラウザ（図2のブラウザ200など）上でレンダリングするために適するXHTMLコンテンツを生成する、ソフトウェアコンポーネントである。レンダリングエンジンは、たとえば、検索エンジンサーバ上に存在することができ、これを使用して、ユーザクエリに応答して結果のページを動的に生成することができる。

【0066】

上述のように、ビュー1304などのビューは、一方ではワイヤフレーム1306の複数のスロットの間のマッピングまたはバインディングを備え、他方ではコンテンツ生成オブジェクト1308を備える。また上述のように、コンテンツ生成オブジェクト1308は様々な形態を取ることができ、コントロール1310、XSLタグ1312およびビュー1314は、それにコンテンツ生成を基づかせることができるオブジェクトの例である。

【0067】

コントロール1310は、上述のように、コンテンツを動的に生成する実行可能または解釈可能コードを備える。コントロール1310をプログラムとして実施することができる、これらは、それらの入力に基づいて異なる振舞を提示することができる。構成ファイル1316は、コントロールのためのこの入力を提供する。たとえば、「shark fin」コントロール（図9の参照符号900）を、様々な状況に応じて異なるバージョンのサメのひれのグラフィック（たとえば、より幅広いか、あるいはより狭いバージョン）を生成するように適合させることができる。たとえば、search.msn.comウェブサイトは、そのウェブサイトがナビゲーションバー202（図2）を通じて直接アクセスされているか、ビューアイングエリア206を2つの狭いサブエリアに分割する検索補助機能を通じてアクセスされているかに応じて、幅広いかあるいは狭いバージョンのサメのひれを生成することができる。構成ファイル1316は、どのバージョンのサメのひれが生成されるかを指定するためにshark finコントロールによって使用されるパラメータを含むことができる。shark finコントロールが、構成ファイル内のパラメータに応答するように設計されている場合、異なるバージョンのサメのひれを、单一のコントロールおよび多数の構成ファイルにより作成することができ、どのバージョンのサメのひれのグラフィックが生成されるかに応じて、異なる構成ファイルをコントロールに提供することができる。加えて、構成情報をビュー内で、スロットのバインディングの一部として指定することができる。たとえば、以下のバインディングを使用して、「header」コントロールがデフォルトモードで使用されることを指定することができる。

【0068】

【表7】

```

<binding slot="slot_a">
  <control_name="header" mode="default">
</binding>

```

【0069】

また、以下の類似のバインディングを使用して、「header」コントロールが「searchpane」モードで使用されることを指定することができる。

【0070】

10

20

30

40

## 【表8】

```

<binding slot="slot_a">
  <control_name="header" mode="searchpane">
</binding>

```

## 【0071】

異なるモードを指定するこの方法は、異なる構成ファイルを使用してコントロールについての異なるパラメータを指定することに類似の方法で動作する。

## 【0072】

構成ファイルの使用のもう1つの例として、本明細書で説明するインフラストラクチャを検索エンジンによって使用して、クエリおよび結果ページを生成することができる。異なるユーザが検索の結果を異なるフォーマットで見ることを望む可能性があり、たとえば、あるユーザは検索結果のテキスト説明のみを受信することを望む可能性があり、他のユーザはテキスト説明と共にサムネイル画像を見ることを望む可能性がある。単一の「results」コントロールを書くことができ（たとえば、図3の「slot\_results」スロット304（2）へのバインディングのため）、これは結果を様々なフォーマットで生成することができる。「results」コントロールによって生成された特定のフォーマットは、構成ファイル1316に含まれたパラメータによって決まる。結果が提示される実際のフォーマットを、単に構成ファイルを変更することによって変更することができる。一例では、いくつかの構成ファイル（それぞれが異なるパラメータを有する）を提供することができ、ユーザに結果の提示を変更する機会を与えることができる。たとえば、ユーザが、検索結果を含むページを受信するとき、ユーザはページ上のメニューを使用して、ユーザが結果を受信することを望む、異なる形態を選択することができる。次いで、ユーザの選択が検索エンジンサーバに、サーバが異なる構成ファイルを使用してページを再生成する要求と共に送信される。

## 【0073】

構成ファイル1316を従来の「ファイル」（すなわち、ファイルシステムに格納された、名前付きオブジェクト）として実施することができるが、この実施形態に限定されないことに留意されたい。一般に「構成ファイル」は、どのようにこのデータが格納されるかにかかわらず、どこかに格納され、かつコントロールによってアクセスすることができるデータを表す。

## 【0074】

コンテンツをビューに基づいて生成するプロセス

コンテンツをビューに基づいて生成するプロセスを、以下の擬似コードによって記述することができる。

## 【0075】

10

20

30

## 【表9】

```

  RendererEval(View)
  {
    For each Slot in the View's WireFrame
    {
      Find Slot binding in View.Bindings
      If binding exists
        If binding is a Control
          Slot = Evaluate the control
        If binding is an XSL Tag
          Slot = Process XSL
        If binding is a View
          Slot = RendererEval(sub view)
        If binding does not exist
          Slot = empty
    }
  }

```

## 【0076】

上述の「RendererEval()」メソッドは、たとえば、レンダリングエンジン1302によって実行される。このメソッドは、ビューをパラメータとしてみなし、ビューのワイヤフレーム内のスロット中を進み、各スロットのための適切なXHTMLコンテンツを、どのオブジェクトがそのスロットにバインドされるかに基づいて生成する。スロットがコントロールにバインドされる場合、そのコントロールが実行され、スロットが、そのコントロールによって生成された XHTMLコンテンツで満たされる。スロットがXSLタグにバインドされる場合、XSL処理ソフトウェアがそのタグに適用されて、適切な XHTMLコンテンツが生成される。スロットがビューにバインドされる場合、そのビューが評価されて XHTMLコンテンツが生成される。上記の擬似コードでは、「サブビュー」が、「RendererEval()」関数が現在評価中であるスロットにバインドされるビューである。上記の擬似コードを見るとわかるように、RendererEval()はそれ自体を再帰的に呼び出してこのサブビューを評価することができる。

## 【0077】

図14は、レンダリングエンジン1302によって実行されるプロセスの流れ図を示す。レンダリングエンジンはビューを得る（ブロック1402）。上記の擬似コードの例では、このステップは、ビューがレンダリングエンジンの最上位のメソッドにパラメータとして渡されるとき、実施される。次いで、レンダリングエンジンは最初のスロットバインディングを検査する（ブロック1404）。バインディングが空である場合（ブロック1405）、スロットに入れるべきコンテンツはなく、そのためレンダリングエンジンはブロック1414に進行して、さらにスロットがあるかどうかを判断する。

## 【0078】

しかし、スロットがバインディングを有している場合、レンダリングエンジンは、スロットにバインドされたオブジェクトを評価するように進行し、スロットを、そのオブジェクトに基づいて適切なコンテンツで満たす。上述のように、バインディングは、スロットがコントロールにバインドされるか、XSLタグにバインドされるか、ビューにバインド

されるかを指示することが好ましい。レンダリングエンジンは、どのタイプのオブジェクトがスロットにバインドされるかに基づいて分岐する（ブロック 1406）。スロットがコントロールにバインドされる場合、このコントロールが実行されて XHTM L コンテンツが生成され、XHTM L コンテンツがスロットに入れられる（ブロック 1408）。スロットが XSL タグにバインドされる場合、この XSL タグが処理されて XHTM L コンテンツが生成され、そのコンテンツがスロットに入れられる（ブロック 1410）。スロットが別のビューにバインドされる場合、この他のビューが評価されて XHTM L コンテンツが生成され、そのコンテンツがスロットに入れられる（ブロック 1412）。上述のように、スロットバインディングにおいてビューに出会うとき、そのビューを評価するプロセスは本質的に、最上位ビューを評価するプロセスと同じであり、ブロック 1412 を実際には、図 14 のプロセスを実施するメソッドへの再帰的呼び出しとして実施することができる。10

#### 【0079】

現在のスロットが処理された後、レンダリングエンジンは、いずれかの追加のスロットが現在のビュー内にあるかどうかを判断する（ブロック 1414）。このようなスロットがない場合、プロセスは終了する。追加のスロットがある場合、レンダリングエンジンは次のスロットに進行し（ブロック 1416）、ブロック 1405 に戻って次のスロットを処理する。

#### 【0080】

前述の例は、説明のためにのみ提供されており、決して本発明の限定として解釈されるべきでないことに留意されたい。本発明を、様々な実施形態を参照して説明したが、本明細書で使用された言葉は、限定の言葉ではなく、説明および例示の言葉であることを理解されたい。さらに、本発明を本明細書で、特定の手段、材料および実施形態を参照して説明したが、本発明は、本明細書で開示された明細に限定されるように意図されず、むしろ本発明は、付属の特許請求の範囲内であるものなど、すべての機能的に等価な構造、方法および使用にまで及ぶ。本明細書の教示を利用する当業者は、多数の修正をこれに実施することができ、その態様において本発明の範囲および精神から逸脱することなく、変更を行なうことができる。20

#### 【図面の簡単な説明】

#### 【0081】

【図 1】本発明の態様を実施することができる例示的コンピューティング環境のブロック図である。

【図 2】例示的ウェブブラウザのユーザインターフェイスのブロック図である。

【図 3】第 1 の例示的ワイアフレームのブロック図である。

【図 4】第 1 の例示的コントロールのブロック図である。

【図 5】コントロールを第 1 の例示的ワイアフレームにバインドする、第 1 の例示的ビューのブロック図である。

【図 6】第 2 の例示的ワイアフレームのブロック図である。

【図 7】例示的拡張可能スタイルシート言語（XSL）タグのブロック図である。

【図 8】例示的拡張可能スタイルシート言語（XSL）タグのブロック図である。

【図 9】第 2 の例示的コントロールのブロック図である。40

【図 10】第 2 の例示的ビューのブロック図である。

【図 11】第 3 の例示的ビューのブロック図である。

【図 12】本発明の態様により生成された例示的コンテンツのブロック図である。

【図 13】本発明の態様によるコンテンツを生成するための例示的インフラストラクチャのブロック図である。

【図 14】本発明の態様によるコンテンツを生成するための例示的プロセスの流れ図である。

#### 【符号の説明】

#### 【0082】

10

20

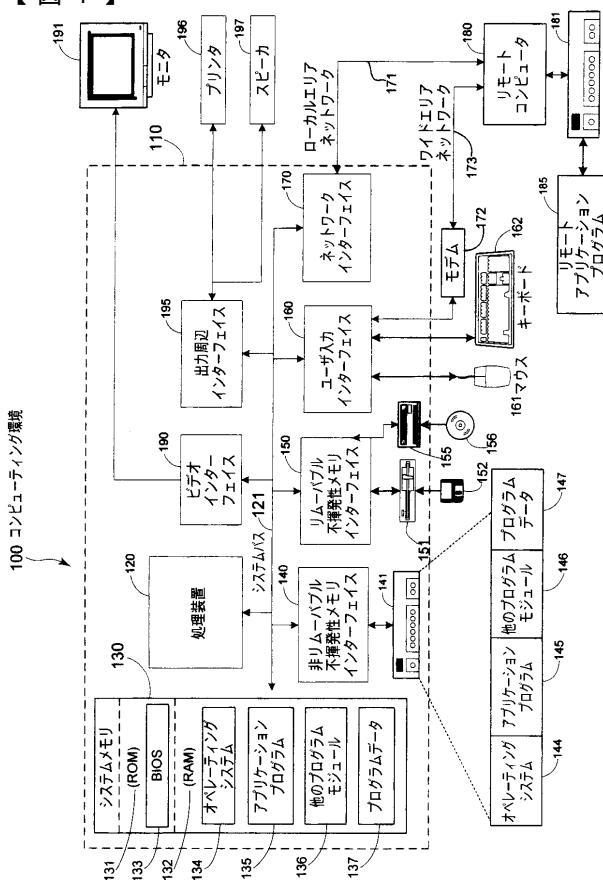
30

40

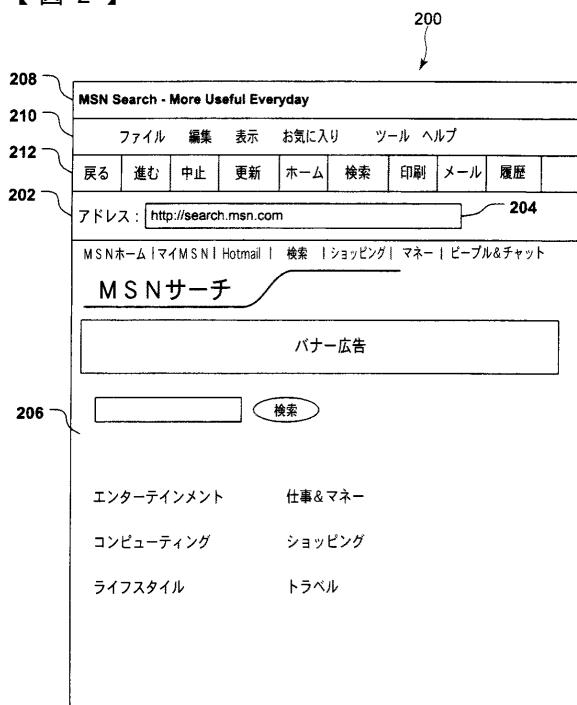
50

1 3 0 2	レンダリングエンジン
1 3 0 4	ビュー
1 3 0 6	ワイヤフレーム
1 3 0 8	コンテンツ生成オブジェクト
1 3 1 0	コントロール
1 3 1 2	XSL
1 3 1 4	ビュー
1 3 1 6	構成ファイル

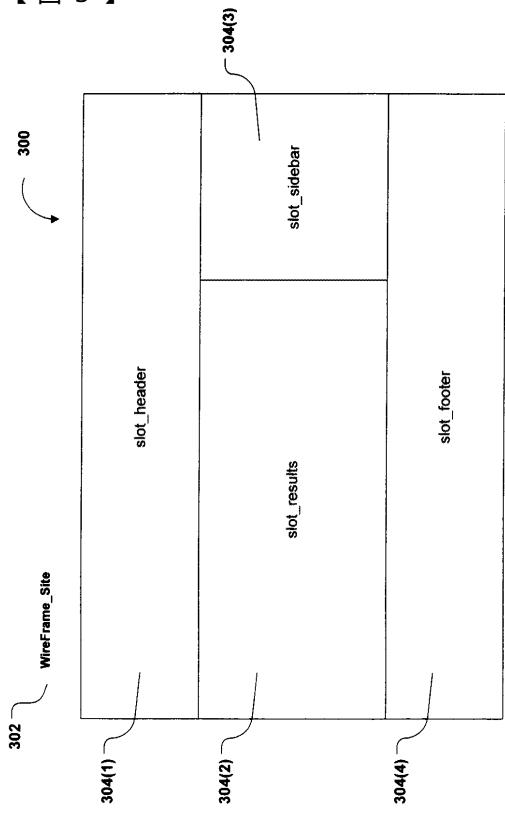
( 义 1 )



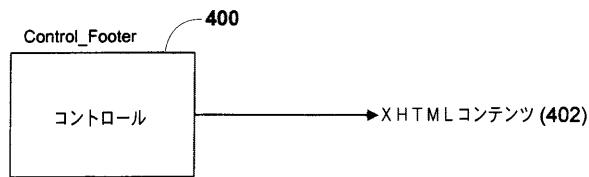
〔 2 〕



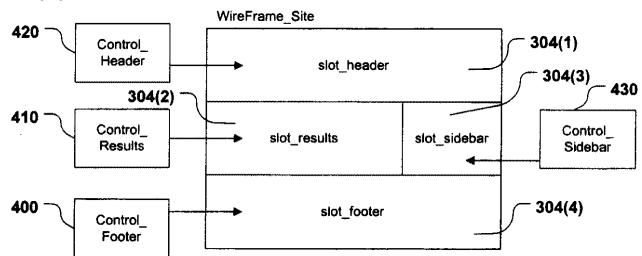
【図3】



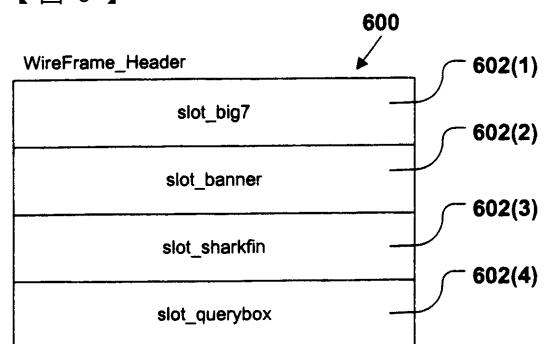
【図4】



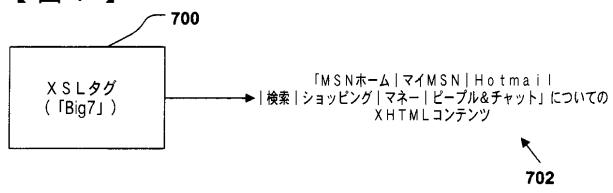
【図5】



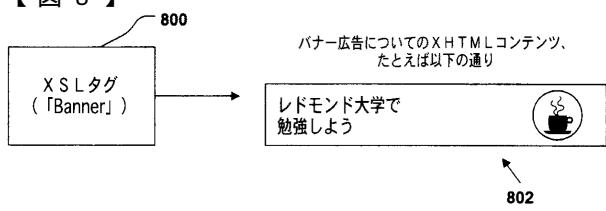
【図6】



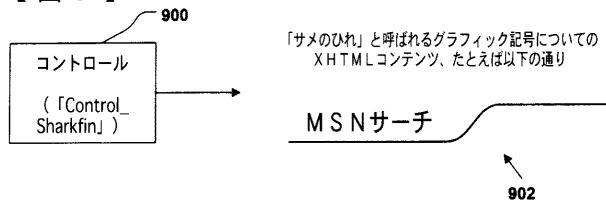
【図7】



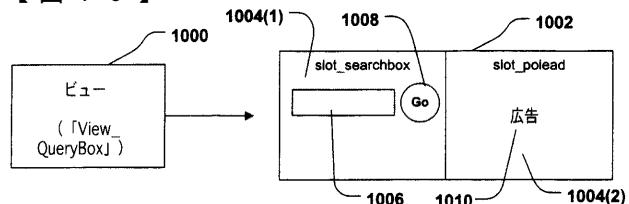
【図8】



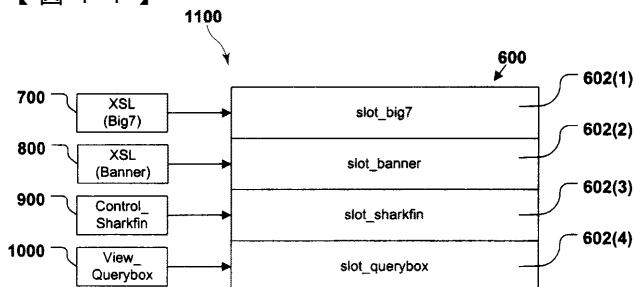
【図9】



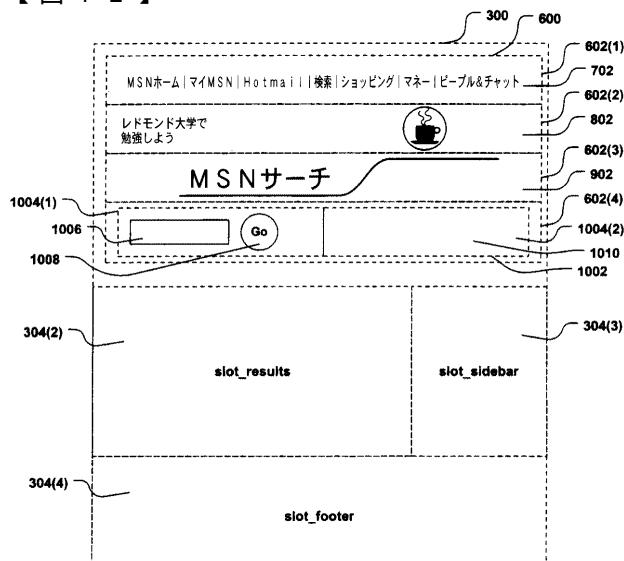
【図10】



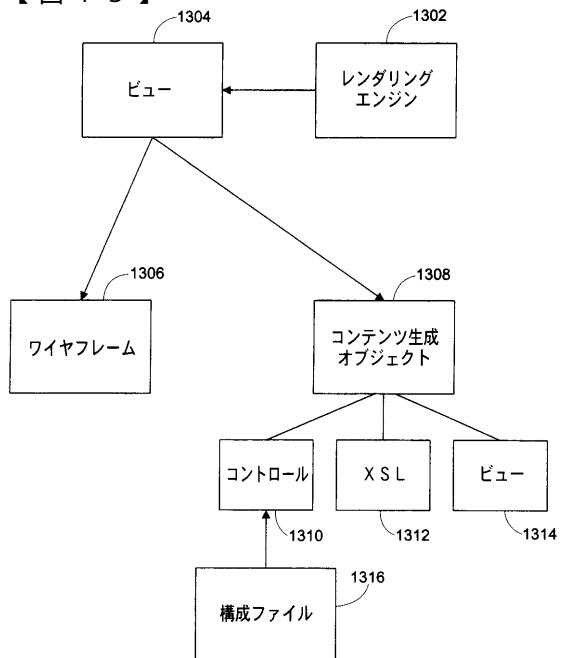
【図11】



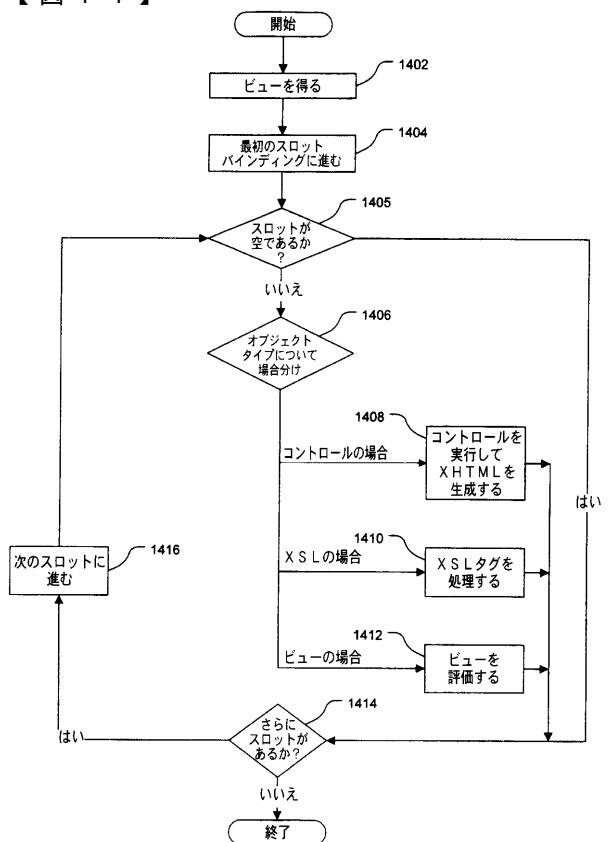
【図12】



【図13】



【図14】



---

フロントページの続き

(72)発明者 マルセロ アイゼンスタイン ファーマン カルバッチ  
アメリカ合衆国 98052 ワシントン州 レッドモンド 177 コート ノースイースト  
10711  
F ターム(参考) 5B082 GA06