

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6026108号
(P6026108)

(45) 発行日 平成28年11月16日(2016.11.16)

(24) 登録日 平成28年10月21日(2016.10.21)

(51) Int.Cl. F I
G O 6 F 21/53 (2013.01) G O 6 F 21/53

請求項の数 15 (全 12 頁)

(21) 出願番号	特願2011-540748 (P2011-540748)	(73) 特許権者	314015767
(86) (22) 出願日	平成21年11月11日 (2009.11.11)		マイクロソフト テクノロジー ライセン
(65) 公表番号	特表2012-511776 (P2012-511776A)		シング, エルエルシー
(43) 公表日	平成24年5月24日 (2012.5.24)		アメリカ合衆国 ワシントン州 9805
(86) 国際出願番号	PCT/US2009/064034		2 レッドモンド ワン マイクロソフト
(87) 国際公開番号	W02010/077443		ウェイ
(87) 国際公開日	平成22年7月8日 (2010.7.8)	(74) 代理人	100107766
審査請求日	平成24年11月2日 (2012.11.2)		弁理士 伊東 忠重
審査番号	不服2014-19296 (P2014-19296/J1)	(74) 代理人	100070150
審査請求日	平成26年9月26日 (2014.9.26)		弁理士 伊東 忠彦
(31) 優先権主張番号	12/330, 528	(74) 代理人	100091214
(32) 優先日	平成20年12月9日 (2008.12.9)		弁理士 大貫 進介
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 プラグインコードによりホストされたアプリケーションの分離方法

(57) 【特許請求の範囲】

【請求項 1】

コンピューティング環境における方法であって、
ホストアプリケーションにおいて他のプラグインをインスタンス化する分離の境界を表す前記ホストアプリケーションに対応するデータ構造にタグエレメントを設けるステップと、

前記ホストアプリケーションをブラウザにロードするステップであって、前記ホストアプリケーションはプラグインを含む、ステップと、

前記ブラウザにおいて実行されている前記ホストアプリケーションにおいて、被ホストアプリケーションをホストするステップであって、プログラムにより分離する方法で前記ブラウザから発見できないようにされた前記他のプラグインをインスタンス化して、前記被ホストアプリケーションを前記他のプラグインにロードすることを含む、ステップと、

前記被ホストアプリケーションが前記ブラウザのデータ、他の被ホストアプリケーションのデータ、及び前記ホストアプリケーションからのデータにアクセスするのを防ぐステップと

を含むことを特徴とする方法。

【請求項 2】

前記タグエレメントは、前記ホストアプリケーションが前記被ホストアプリケーションと通信できるようにする一以上のアプリケーションプログラミングインタフェースを有する、請求項 1 に記載の方法。

【請求項 3】

前記被ホストアプリケーションがデータにアクセスすることを防ぐステップは、
前記被ホストアプリケーションが前記タグエレメントを介して前記ホストアプリケーションにアウトプットを提供するよう限定するステップと、
前記被ホストアプリケーションの、前記ホストアプリケーションのデータ構造及び前記ブラウザのドメインオブジェクトモデルへのアクセスを無効にするステップと
を含む、請求項 1 に記載の方法。

【請求項 4】

前記ホストアプリケーションに対応するユーザーインターフェースの一部がもはやレンダリングされないことを前記タグエレメントが示すことに応じて、前記被ホストアプリケーションを破壊するステップをさらに含む、請求項 1 に記載の方法。

10

【請求項 5】

前記タグエレメントは前記被ホストアプリケーションのアウトプットを処理する、請求項 1 に記載の方法。

【請求項 6】

前記ホストアプリケーションからのアウトプットを前記被ホストアプリケーションからのアウトプットと合成するステップをさらに含む、請求項 1 に記載の方法。

【請求項 7】

前記ホストアプリケーションから前記被ホストアプリケーションへ変数を前記タグエレメントを介して通信するステップ、または前記被ホストアプリケーションが電力の消費を制御することを試みる行為を制御するステップ、または前記ホストアプリケーションから前記被ホストアプリケーションへ変数を前記タグエレメントを介して通信するステップおよび前記被ホストアプリケーションが電力の消費を制御することを試みる行為を制御するステップの双方をさらに含む、請求項 1 に記載の方法。

20

【請求項 8】

前記ホストアプリケーションから前記被ホストアプリケーションへ前記変数を通信するステップは、前記被ホストアプリケーションが少なくとも 1 の関係する広告を選択できる情報を提供するステップを含む、請求項 7 に記載の方法。

【請求項 9】

前記被ホストアプリケーションが少なくとも 1 の特権とされた操作を実行することを許可するステップをさらに含む、請求項 1 に記載の方法。

30

【請求項 10】

コンピューティング環境におけるシステムであって、
少なくとも 1 のプロセッサと、
前記少なくとも 1 のプロセッサに通信可能に接続されたメモリと、
前記少なくとも 1 のプロセッサにより実行されブラウザにおいて実行される第 1 のプラグインと、

前記第 1 のプラグインに関連するデータ構造であって、前記データ構造はエレメントのツリーを含み、前記エレメントのツリーは前記第 1 のプラグインのユーザーインターフェースのエレメントを含むデータ構造と、

40

プログラムにより分離する方法で前記ブラウザから発見できないようにされた前記第 1 のプラグインによりインスタンスが作成された第 2 のプラグインであって、ブラウザのエレメント、他のブラウザ被ホストアプリケーションエレメント、及び前記第 1 のプラグインのエレメントへのアクセスが無効にされ、前記第 2 のプラグインはエレメントのツリーにおけるホストエレメントに対応し、前記ホストエレメントは前記分離の境界を画定し、前記第 2 のプラグインにロードされた被ホストアプリケーションに関連する、第 2 のプラグインと
を備えるシステム。

【請求項 11】

前記第 1 のプラグインは `extensible application marku`

50

pl language (XAML)コードであり、前記第2のプラグインは、XAMLコードを構文解析している間に前記第2のプラグインに対応する前記ホストエレメントが検出されるときに、インスタンスが作成される、請求項10に記載のシステム。

【請求項12】

前記第1のプラグインに関連する1またはそれより多くの非公開のapplication programming interface (APIs)をさらに備え、前記第2のプラグインは前記第1のプラグインによりインスタンスを作成され、前記非公開のAPIを介して前記第1のプラグインと通信する、請求項10に記載のシステム。

【請求項13】

コンピュータにより実行されると、前記コンピュータに、
ブラウザにおいて、第1のアプリケーションを第1のプラグインとして実行するステップであって、前記第1のアプリケーションのエレメントを示すエレメントツリーを作成するステップを含むステップと、

前記第1のプラグイン内に第2のアプリケーションをホストするステップであって、前記エレメントツリー内の前記第1のプラグインのためのホストエレメントを含むステップと、プログラムにより分離する方法で前記ブラウザから発見できないようにされた、前記ブラウザのデータ、前記第1のプラグインのデータ、及び他のブラウザ被ホストアプリケーションエレメントのデータへのアクセスが無効とされた第2のプラグインのインスタンスを作成するステップと、前記第2のプラグインにおいて実行するための第2のアプリケーションをロードするステップであって、前記第2のアプリケーションは前記第1のアプリケーションに前記ホストエレメントを介してユーザーインターフェースエレメントを提供するステップとを含む、ステップと、

前記第1のアプリケーションのユーザーインターフェースエレメントと前記第2のアプリケーションのユーザーインターフェースエレメントとを処理することにより目に見えるアウトプットをレンダリングするステップと
を実行させるコンピュータプログラム。

【請求項14】

前記第2のアプリケーションは、広告に対応しており、前記コンピュータにより実行されると、前記コンピュータに、前記ホストエレメントを介して前記第2のアプリケーションの前記ブラウザのページコンテンツに対応する1または複数のキーワードを提供するステップをさらに実行させる、請求項13に記載のコンピュータプログラム。

【請求項15】

前記コンピュータにより実行されると、前記コンピュータに、前記第2のアプリケーションがウェブサイトのコンテンツを表示するためのブラウザのウィンドウを開くことを許可するステップをさらに実行させる、請求項13に記載のコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

この発明は、ブラウザにおいて実行しているアプリケーションが他のアプリケーションをホストするための技術に関する。

【背景技術】

【0002】

現代のブラウザはプラグインを許可し、何らかの要求された機能を提供するためにホスティングブラウザ/アプリケーションと相互作用するホストされたソフトウェアコードを一般的に含んでいる。このようなプラグインのひとつは、本件出願人の発売するシルバライト(商標)で、アニメーション、ベクターグラフィックおよび/または再生装置で使用できる媒体(例えばビデオ・オーディオ)を含むことが多い高価なウェブアプリケーションの開発およびホスティングを許可するプラットフォームを、提供するものである。プラグインは、第三者のアプリケーションをホストできる。

【0003】

10

20

30

40

50

一般的に、これは現代のアプリケーションの開発の重要な機能、すなわちコンポーネント化と結び付けられる。コンポーネント化の原因となっている一般的な考えは、大きなアプリケーションを分割して、より大きなタスクのより小さい部分を実行し、再利用可能であり、周知のインターフェースを有する、より小さなコンポーネントにすることである。外部関係者（例えば、第二者または第三者）が既存のアプリケーションのためのコンポーネントを作成する能力は、コンポーネント化により生じ、例えば現代のウェブに基づいたこの概念の具体物はウェブマッシュアップである。

【0004】

しかしプラグインによりホストされたアプリケーション等のアプリケーションにおいて、当事者以外のコンポーネントが使用されたとき、コンポーネントとアプリケーションの間の信頼の限度が重要になってくる。ホスティングアプリケーションが特別にこのために設計され期待された場合は別として、コンポーネントは、ユーザーのインターフェース（UI）コードの外観および機能を変更し、またはアプリケーションデータを監視する等を可能とすべきでない。

10

【発明の概要】

【0005】

この概要は、「発明を実施するための形態」においてさらに詳細に説明された簡潔な様式に代表される概念の選択を紹介するために提供される。この概要は、「特許請求の範囲」に記載の主題の重要な特徴または本質的な特徴を識別することを意図するものではなく、「特許請求の範囲」に記載の主題の範囲を限定する任意の方法において使用されることを意図するものでもない。

20

【0006】

簡潔にいうと、本明細書で説明されたこの主題の様々な態様は、ブラウザにおいて実行されている1のアプリケーションがホストされたアプリケーションがホストアプリケーションデータまたは他のホストされたアプリケーションのデータにアクセスするのを防ぐ方法により、他のアプリケーションをホストできる技術へと向けられている。一態様において、ホストアプリケーションは、ブラウザに隠された他方のプラグインのインスタンスを作成するブラウザのプラグイン（例えば、本件出願人の発売するシルバーライト（商標））アプリケーションであり、ブラウザ/ホストされたアプリケーションデータ（例えば、エレメント）または他のアプリケーションデータへのアクセスが無効にされた状態で他のプラグインのインスタンスを作成することによりホストされたアプリケーションが、データにアクセスするのを防ぐ。

30

【0007】

一態様において、タグエレメント（tag element）は、ホストアプリケーションに関連するコード（例えばXML）内においてホストされたアプリケーションを識別する。ホストエレメント（host element）は、ホストされたアプリケーションを示すためのホストアプリケーションのエレメントツリー（element tree）を追加する。タグエレメントは、ホストされたアプリケーションへのレンダリングの範囲を特定することができる。ホストされたアプリケーションのコンテンツは、ホストアプリケーションからのコンテンツと連動して（例えば、合成されて）、レンダリングされる。

40

【0008】

1の態様において、ホストされたアプリケーションは、広告となりうる。ホストアプリケーションはページコンテンツに対応しているもの等のキーワードをホストされたアプリケーションに提供し、これによりホストされたアプリケーションは関連する広告を選択することが可能となる。ホストされたアプリケーションは、ホストされたアプリケーションがブラウザのウィンドウを開いてディスプレイのウェブサイトのコンテンツ上に表示することを許可する等のいくつかの特権を与えられた操作（例えば、ホストされたアプリケーションのレンダリングの範囲内でユーザーが広告をクリックするとウェブサイトを表示する）を実行することを許可されうる。

【0009】

50

他の利点は、図面と連動して、以下の詳細な説明から表されるようになる。

【図面の簡単な説明】

【0010】

本発明は、例として図示され、同様の参照する番号が類似のエレメントを示すような添付する図面に限定されない。

【図1】ホスト（ブラウザプラグイン）アプリケーションによりホストされたアプリケーションを分離するコンポーネントの例を示すブロック図である。

【図2】ホストされたアプリケーションの分離を行うステップの例を示すフロー図である。

【図3】ホストアプリケーションと分離されたプラグインアプリケーションとの間の追加の通信のコンポーネントの例を示すブロック図である。

【図4】本発明の様々な態様が合併されるコンピューティング環境の例を示す図である。

【発明を実施するための形態】

【0011】

本明細書で説明される技術のさまざまな態様は、一般的に、分離された方法において第1のアプリケーション（例として、ブラウザのプラグイン等）が第2の（典型的に信頼性がない、第三者の）アプリケーションをホストする分離のモデルへ向けられている。一般的にこれは、メモリ内の個別のプラグインインスタンスをインスタンス化して第2のアプリケーションを分割することにより達成され、プラグインインスタンスはウェブブラウザにとって未知である。同時に、例えば、目に見えるアウトプットがホストするアプリケーションの目に見えるアウトプットに沿ってレンダリングされるため、第2のアプリケーション（例えば、広告）は、コンピュータユーザーにとってホストされたアプリケーションの一部として実行されるように見える。しかし、理解されるように、ホスティングアプリケーションが何らかの情報を提供することを要求しない限り、ホストされたアプリケーションはホスティングアプリケーションまたはウェブブラウザに関連する情報へのアクセスを有さない。

【0012】

本件出願人の発売するシルバーライト（商標）（クロスプラットフォーム、クロスブラウザのプラグイン）は、ホストするアプリケーションの例として使用されるが、ここで説明されるいかなる例も制限されないものである。そういうわけで、本発明は、どのような特定の実施例、態様、概念、構造、機能またはここに示された例にも限定されない。むしろ実施例、態様、概念、構成、機能またはここに説明された例のいずれも限定的なものではなく、本発明は、一般的にコンピューティングおよびホスティングアプリケーションの利益および利点を提供する様々な方法を使用できる。

【0013】

図1は、ここに説明されるものとしてアプリケーションの分離に関連した様々な態様を示すものである。一般的に、ブラウザ102は、本件出願人の発売するシルバーライト（商標）のプラグイン等のホストアプリケーション104とともにロードされる。一実装において、ホストアプリケーション104は、XAML（拡張可能なアプリケーションのマークアップ言語）に書き込まれたもの等の、ユーザーインターフェースのアプリケーションコード106に対応し、アプリケーションコード106はパーサ108によりツリー110に対応する様々なエレメントへ構文解析される。周知のように、これらのエレメントはブラウザのページの目に見えるアウトプットへのレンダリングへと進む。

【0014】

一実装において、第三者の広告等のアプリケーション111は、隠された（分割され、ブラウザにおいて発見できない）プラグイン112の一部として、ブラウザ102およびホストするアプリケーションデータから隠されたプラグイン112をプログラムにより分離する方法で、ホストアプリケーション104によりホストされるが、それにもかかわらず、ホストされたアプリケーション111が、ホストアプリケーションのレンダリング、インプットおよびレイアウトに参加することを許可する。レンダリング、インプットおよ

10

20

30

40

50

びレイアウトは、隠されたプラグイン 1 1 2 により管理されるため、例えば、目に見えるアウトプットは、ホストされたアプリケーションのアウトプットに確保されるユーザーインターフェースの範囲に限定される。隠されたプラグイン 1 1 2 が他方のシルバーライト（商標）のプラグイン、すなわちホストシルバーライトのプラグインにより（ブラウザに知られることなく）非公式で生成された他方のインスタンスとなりうることに注意されたい。さらに、他のプラグインのタイプは、HTML ブラウザとのいかなる直接通信も行わないようにインスタンス化または他の方法で構成するために使用することができ、例えばプラグインは、ブラウザの DOM（ドキュメントオブジェクトモデル）にアクセスしないようにインスタンス化されることに注意されたい。

【 0 0 1 5 】

この目的を達成するために、実装の一例において、ホストされたアプリケーションのソースを識別し、ホストされた（分離された）アプリケーション 1 1 1 をロードする隠されたプラグイン 1 1 2 を インスタンス化 する分離の境界を表示するタグエレメントが（例えば XAML アプリケーションコード 1 0 6 において）提供される。

```
<Canvas> <!-- part of the host application-->
  <XapHost Source = " http://www.advertisements.com/ad.xap " Height = " 3
00 "
    Width = " 350 " /> <!-- the isolation boundary-->
</Canvas>
```

【 0 0 1 6 】

タグエレメントは、XapHost のエレメント 1 1 4 により図 1 において表される。数ある操作の中で、XapHost のエレメント 1 1 4 はメモリ内、つまりブラウザ 1 0 2 から発見できない記憶場所において、隠されたプラグイン 1 1 2 のインスタンス化および初期化を担う。これは API またはそれと同様のものを通して達成されることが可能である。ホストされたアプリケーション 1 1 1 は、隠されたプラグイン 1 1 2 を分離するプロキシとして動作する XapHost のエレメント 1 1 4 を介さなければホストアプリケーション 1 0 4 のツリー 1 1 0 に参加しない。たとえばホストアプリケーション 1 1 1 は、そのアプリケーションのルートノードに対応しているツリーを自身の最上位のノード以外、走査することができない。それは XapHost のエレメントへのアクセスを有しておらず、XapHost のエレメントはホスティングアプリケーションのツリーの中に存在することに注意されたい。代わりに、ホストされたアプリケーション 1 1 1 は XapHost のエレメント 1 1 4 を介してレンダリングするためのアウトプット 1 1 6 を提供することにのみ限定される。（一方で、XapHost のエレメント 1 1 4 は初期化変数等の追加のデータを提供する、および/または図 3 を参照して以下に説明されるようないくつかの特権をホストされたアプリケーション 1 1 1 に許可する。）これおよび XapHost のエレメント 1 1 4 から隠されたプラグイン 1 1 2 / ホストされたアプリケーション 1 1 1 に通信された他の任意のデータは、インプット 1 1 7 として図 1 において表現される。

【 0 0 1 7 】

図 2 の例となるステップを使用して概略的に示すように、XAML のアプリケーションコード 1 0 6 が構文解析されたときまたは XapHost のエレメント 1 1 4 がプログラムでツリー 1 1 0 に追加されたとき（ステップ 2 0 2）、ステップ 2 0 4 および 2 0 6 において XapHost のエレメント 1 1 4 は隠されたプラグイン 1 1 2 を初期化し「ソース」属性により特定された対応するホストされたアプリケーション 1 1 1（例えば、XAP）をロードする（上述した XAML の言語の例において、アプリケーション 1 1 1 は http://advertisements.com/ad.xap として特定される）。ホストされたアプリケーション 1 1 1 は、ブラウザの HTML ドメインオブジェクトモデルへのアクセスを有しておらず（例えば、EnableHtmlAccess プロパティが偽に設定される）、例えば（共通の言語ランタイム、または CLR AppDomain）グローバルの静的な変数等へのそれ自身のアクセスにより、それ自身のアプリケーションのドメインにロードされ

10

20

30

40

50

る。シルバーライト（商標）が、それ自身のミニCLR118を含んでいることに注意されたい。ドメインオブジェクトモデル（DomBridge）のアクセスなしに、ホストされたアプリケーション111はページそれ自体または他のプラグインを含む、ページ上の他の任意のコンテンツへのアクセスを有さない。

【0018】

ステップ208および210によって図2にさらに示すように、を介して、隠されたプラグイン112のインスタンス化および初期化、およびアプリケーション111のロードに加えて、XapHostのエLEMENT114は、ホストするアプリケーションからホストされたアプリケーションへのインプットおよびレイアウトの通知を指示する。さらに、XapHostのエLEMENT114は、ホストされたアプリケーション111のアウトプットを処理し、その結果ホスティングアプリケーションのプラグインによりそのアウトプットを構成できるようにする（音声および/または触覚等の他の任意のアウトプットが同様に混合されることに注意されたい）。ステップ212は、適切な時間、例えばシャットダウンにおいて、またはXapHostのエLEMENT114における対応する参照カウンタが0になったとき（例えば、ユーザーのインターフェースの部分がもはやレンダリングされなくなったとき）に、ホストされたアプリケーションの破壊を表現する。

10

【0019】

一態様において、一般的に図3で示されるようにXapHostのエLEMENT114は、ホストアプリケーション104に、例えば初期化変数332を渡す等、ホストされたアプリケーション111との通信を許可する1または複数のインターフェース（例えば、上述の他の非公開のAPIの中のAPI330）を含むことができる。例えば広告のシナリオにおいて、初期化変数は、ホストしているページに存在するキーワードの集合体を含んでおり、ホストするページに、広告のアプリケーションが1または複数のキーワードに対応する関連の広告を供給できる。

20

【0020】

また一般的に図3において示される他方の態様においても、API330を介して、XapHostのエLEMENT114はホストされたアプリケーションがある特権を与えられた操作を実行すること（または、適切な呼334において、その代わりに操作を実行すること）を許可する。ひとつの例は、クリックされた広告に対するウェブサイトを表示する等、ユーザーの起動する行為に回答して新しいブラウザのウィンドウを開くことである。

30

【0021】

他方の利点は電力の消費を制御する能力である。例えば、広告は現代のコンピュータ使用において電力消費の大きな源となりうる。プラグインのホスティングモデル（例えば、XapHost）により、省電力モードは、アニメーションや他の電力を消費する動作を無効にすること等により、電力を消費しないようにできる。

【0022】

操作環境の例

図4は、図1から3の例を実装することができる適切なコンピューティングおよびネットワーク環境400の例を示している。コンピューティングシステム環境400は適切なコンピューティング環境の一例に過ぎず、本発明の使用および機能の範囲についてのいかなる限界をも提案することを意図していない。コンピューティング環境400は、操作環境400の一例において示されたコンポーネントの一つまたは組み合わせに関連するいかなる従属および要求を有しているとしても解釈されるべきでない。

40

【0023】

本発明は、多くの他の一般的な目的または特定の目的のコンピューティングシステム環境または構成において操作可能である。本発明と共に使用するために適した周知のコンピューティングシステム、環境、および/または構成の例は、パーソナルコンピュータ、サーバーコンピュータ、携帯またはラップトップ機器、タブレット機器、マルチプロセッサシステム、マイクロプロセッサに基づくシステム、セットトップボックス、プログラム可能な家庭用電化機器、ネットワークPC、マイクロコンピュータ、メインフレームコンピュ

50

ータ、上記システムまたはデバイスの任意のものを含む分散コンピューティング環境、またはそれと同様のものを含むが、それに限定されない。

【 0 0 2 4 】

本発明は、コンピュータにより実施されるプログラムモジュール等のコンピュータ実行可能命令の一般的なコンテキストにおいて説明されうる。一般的に、プログラムモジュールは、特定のタスクを実行するか、特定の理論上のデータタイプを実装するルーティン、プログラム、オブジェクト、コンポーネント、データ構造等を含む。本発明はまた、コミュニケーションネットワークを介してリンクされた遠隔処理装置によりタスクが実行される分離されたコンピューティング環境においても実現されうる。分配されたコンピューティング環境において、プログラムモジュールは、ローカルおよび/またはメモリ記録装置を含むリモートコンピュータの記録媒体に配置されうる。

10

【 0 0 2 5 】

図4を参照して、本発明の様々な態様を実装するための例示のシステムは、コンピュータ410の形式における一般的な目的のコンピューティングデバイスを含むことができる。コンピュータ410のコンポーネントは、処理ユニット420、システムメモリ430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と処理ユニット420を結合するシステムバス421を含むことができるがそれに限定されない。システムバス421は、メモリバスまたは様々なバスアーキテクチャーの任意のものを使用するメモリコントローラ、周辺機器バスおよびローカルバスを含むバス構造のいくつかのタイプの任意のものである。例として、そのようなアーキテクチャーはISA (Industry Standard Architecture) バス、MCA (Micro Channel Architecture) バス、EISA (Enhanced ISA) バス、VESA (Video Electronics Standards Association) ローカルバスおよびメザニンバスとしても知られているPCI (Peripheral Component Interconnect) バスを含むがそれに限定されない。

20

【 0 0 2 6 】

コンピュータ410は典型的に様々なコンピュータ可読媒体を含んでいる。コンピュータ可読媒体は、コンピュータ410によりアクセスされることができる任意の利用可能な媒体とすることができ、揮発性および不揮発性の媒体双方、および固定式および取り外し可能な媒体双方を含む。例として、コンピュータ可読媒体は、コンピュータ記録媒体および通信媒体を含むがそれに限定されない。コンピュータ記録媒体は、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等の情報の記録の方法および技術において実装される揮発性および不揮発性、取り外し可能または固定式の媒体を含む。コンピュータ記録媒体は、RAM、ROM、EEPROM、フラッシュメモリまたは他のメモリ技術、CD-ROM、DVD (digital versatile disks) または他の光ディスクストレージ、磁気カセット、磁気テープ、磁気ディスクストレージまたは他の磁気ストレージデバイス、または要求される情報を記録するのに使用しコンピュータ410によりアクセスできる他の媒体を含むがそれに限定されない。通信媒体は、典型的に搬送波や他の伝達機構等の変調されたデータ信号における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により具体化され、任意の配信媒体の情報を含む。「変調されたデータ信号」という用語は、1またはそれ以上の特徴の組を有し、信号内に情報をエンコードするような方法により変更された信号を意味する。例として、通信媒体は、有線ネットワークまたは直接の配線等の有線媒体、および音響、RF、赤外線または他の無線媒体等の無線媒体を含むがそれに限定されない。上記の任意の組み合わせはまたコンピュータ可読媒体の範囲内に含まれる。

30

40

【 0 0 2 7 】

システムメモリ430は、ROM431またはRAM432等の揮発性および/または不揮発性メモリの形式においてコンピュータ記録媒体を含む。BIOS (基本入出力システム) 433は、起動中等のコンピュータ410内のエレメントの間の情報の転送を助ける基本のルーティンを含んでおり、典型的にROM431に記録される。RAM432は、典型的に処理ユニット420によりすぐにアクセス可能なおよび/または現在操作さて

50

いれるデータおよび/またはプログラムモジュールを含んでいる。例として、図4はオペレーティングシステム434、アプリケーションプログラム435、他のプログラムモジュール436およびプログラムデータ437を図示しているが、これらに限定されない。

【0028】

コンピュータ410はまた、他の取り外し可能/固定式の、揮発性/不揮発性のコンピュータ記録媒体を含んでいる。単なる例として、図4は、固定式、不揮発性の磁気媒体から読み込む、または書き込むハードディスクドライブ441、取り外し可能、不揮発性の磁気ディスク452から読み込む、または書き込む磁気ディスクドライブ451、およびCD-ROMまたは他の光媒体等の取り外し可能、不揮発性の光ディスク456から読み込む、または書き込む光ディスクドライブ455のみ図示している。他の例示の操作環境において使用されることができる取り外し可能/固定式の、揮発性/不揮発性のコンピュータ記録媒体は、磁気テープカセット、フラッシュメモリカード、DVD、デジタルビデオテープ、ソリッドステートRAM、ソリッドステートROM等を含むがそれに限定されない。ハードディスクドライブ441は、インターフェース440等の固定式メモリのインターフェースを通して典型的にシステムバス421に接続され、磁気ディスクドライブ451および光ディスクドライブ455は、インターフェース450等の取り外し可能なメモリのインターフェースによりシステムバス421に典型的に接続される。

【0029】

ドライブおよびそれらに関連するコンピュータ記録媒体は、上述され図4で説明されており、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コンピュータ410の他のデータのストレージを提供する。例えば図4において、ハードディスクドライブ441はオペレーティングシステム444、アプリケーションプログラム445、他のプログラムモジュール446およびプログラムデータ447の記録として図示されている。これらのコンポーネントはオペレーティングシステム434、アプリケーションプログラム435、他のプログラムモジュール436およびプログラムデータ437と同一または異なるものにできることに注意されたい。オペレーティングシステム444、アプリケーションプログラム445、他のプログラムモジュール446、プログラムデータ447は、少なくとも異なるコピーであることを示すため、ここでは異なる番号が与えられる。ユーザーは、タブレットまたは電子デジタイザー464、マイクロホン463、キーボード462およびポインティングデバイス446等の一般的にマウス、トラックボールまたはタッチパッドと称されるインプットデバイスを通してコンピュータ410にコマンドおよび情報を入力できる。図4に示されていない他のインプットデバイスはジョイスティック、ゲームパッド、衛星アンテナ、スキャナ、またはそれと同様のものを含んでもよい。これらおよび他のインプットデバイスはシステムバスと接続されたユーザーインプットインターフェース460を通してしばしば処理ユニット420に接続されるが、パラレルポート、ゲームポート、またはUSB等の他のインターフェースおよびバス構造によっても接続される。モニター491または他のタイプのディスプレイ装置はまた、ビデオインターフェース490等のインターフェースを通して、システムバス421と接続される。モニター491は、タッチスクリーンパネルまたはそれと同様のものと統合してもよい。モニターおよび/またはタッチスクリーンパネルはタブレット型パーソナルコンピュータのようにコンピューティングデバイス410を組み込んだ、筐体に物理的に結合することができることに注意されたい。加えて、コンピューティングデバイス410等のコンピュータはまた、周辺のアウトプットインターフェース494等を通して接続されうるスピーカー495およびプリンター496等の他の周辺のアウトプット機器を含むことができる。

【0030】

コンピュータ410はネットワーク化された環境においてリモートコンピュータ480等の1または複数のリモートコンピュータへの理論接続を使用して動作しうる。リモートコンピュータ480はパーソナルコンピュータ、サーバー、ルータ、ネットワークPC、ピアデバイスまたは他の一般的なネットワークノードとすることができ、典型的にコンピュータ410に関連して上述された多くのまたはすべてのエレメントを含むが、メモリ記

10

20

30

40

50

録装置 481 だけが図 4 において図示されている。図 4 に描かれている理論接続は、1 または複数の LAN (Local area network) 471 および 1 または複数の WAN (Wide area network) 473 を含むが、また、他のネットワークも含むことができる。そのようなネットワーク化環境はオフィス、企業規模のネットワーク、イントラネットおよびインターネットにおいて一般的である。

【0031】

LAN のネットワーク化環境において使用する場合、コンピュータ 410 はネットワークインターフェースやアダプター 470 を通して LAN 471 に接続される。WAN のネットワーク化環境において使用する場合、コンピュータ 410 は典型的にモデム 472 またはインターネット等の WAN 473 を通した通信を構築するための他の手段を含む。モデム 472 は、内臓または外付けとすることができ、インプットインターフェース 460 または他の適切な機構を介してシステムバス 421 に接続できる。インターフェースやアンテナを含んでいるような無線ネットワークコンポーネント 474 は、アクセスポイントまたはピアコンピュータ等の適切な装置を通して WAN または LAN に結合されうる。ネットワーク化された環境において、コンピュータ 410 に関して描かれたプログラムモジュール、またはその一部は、リモートのメモリ記録装置に記録される。例として、図 4 は、リモートアプリケーションプログラム 485 をメモリデバイス 481 内に存在するものとして図示しているが、これに限定されない。示されているネットワーク接続は例示のものであり、コンピュータの間の通信リンクを確立する他の手段が使用されることが理解できよう。

10

20

【0032】

補助のサブシステム 499 (例えばコンテンツの補助ディスプレイ) は、コンピュータの主要部分が省電力状態にある場合でさえ、プログラムコンテンツ、システム状態および事象通知等のデータをユーザーに提供するように許可するためにユーザーインターフェース 460 を介して接続される。補助のサブシステム 499 は、主処理装置 420 が省電力状態にある間にこれらのシステム間で通信することを許可するため、モデム 472 および/またはネットワークインターフェース 470 に接続される。

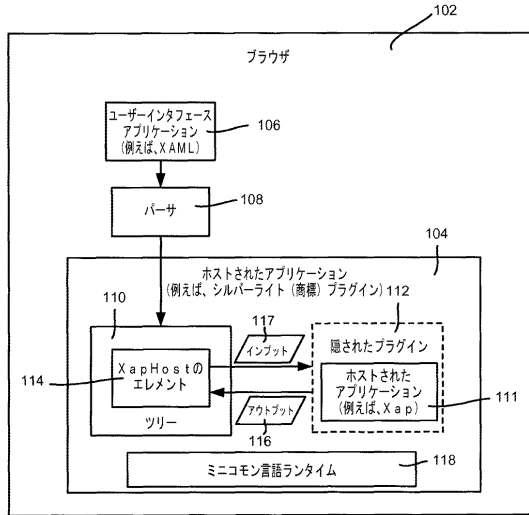
【0033】

結論

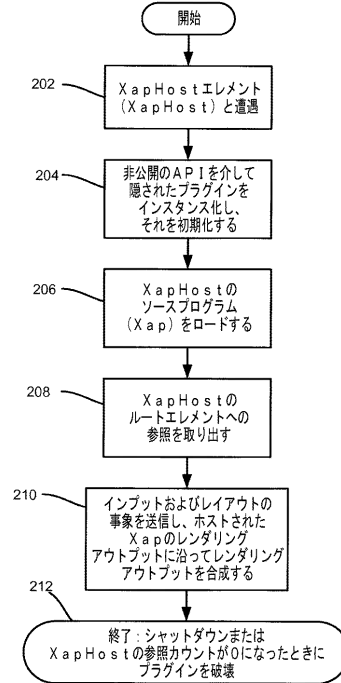
本発明は様々な修正や構成の変更が可能であるが、本発明の特定の例示の実施形態が、図に示され、詳細に上述されてきた。しかし、本発明を開示された特定の形式へ限定することを意図しておらず、反対に、本発明の趣旨および範囲内に収まるすべての修正、構成の変更および等価物に及ぶように意図していることが理解できよう。

30

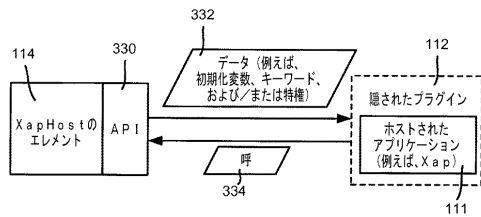
【図 1】



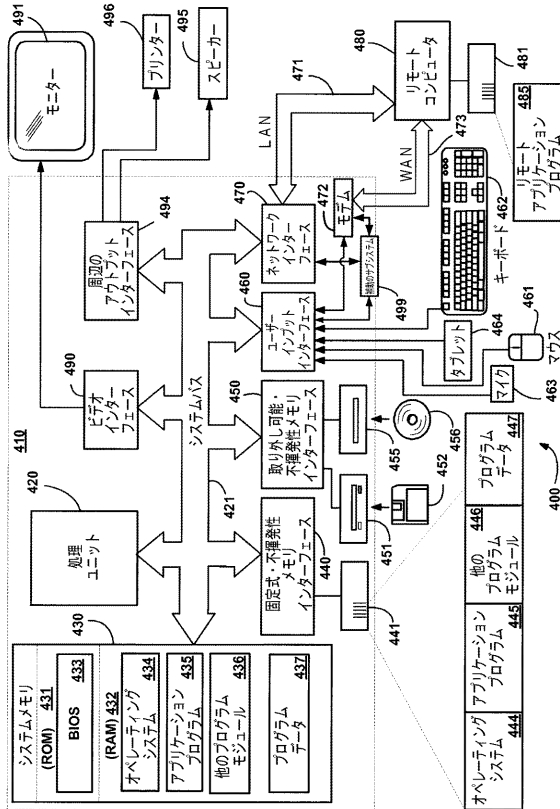
【図 2】



【図 3】



【図 4】



フロントページの続き

- (72)発明者 サム ジョージ
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション エルシーイー - インターナショナル パテント内
- (72)発明者 アキレッシュ カザ
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション エルシーイー - インターナショナル パテント内
- (72)発明者 マイケル アール . ハーシュ
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション エルシーイー - インターナショナル パテント内

合議体

審判長 辻本 泰隆
審判官 須田 勝巳
審判官 石井 茂和

- (56)参考文献 米国特許出願公開第2002/0007393(US, A1)
米国特許第6691230(US, B1)
米国特許出願公開第2003/0101292(US, A1)
特開2004-118866(JP, A)
Charles Reis, Steven D. Gribble, Henry M. L
evy, Architectural Principles for Safe Web P
rograms, [online], 米国, ACM SIGCOMM, 2007年11月, [平
成26年10月15日検索], URL, <http://conferences.sigcomm.org/hotnets/2007/papers/hotnets6-final68.pdf>

(58)調査した分野(Int.Cl., DB名)

G06F 21/53
G06F 21/60