

(19) 日本国特許庁(JP)

## (12) 公表特許公報(A)

(11) 特許出願公表番号

特表2008-535083  
(P2008-535083A)

(43) 公表日 平成20年8月28日(2008.8.28)

(51) Int.Cl.	F 1	テーマコード (参考)
<b>G06F 3/048</b> (2006.01)	G06F 3/048	655A 5B069
<b>G06F 3/14</b> (2006.01)	G06F 3/14	350A 5C082
<b>G09G 5/14</b> (2006.01)	G09G 5/14	A 5E501
<b>G09G 5/00</b> (2006.01)	G09G 5/00	530A
<b>G09G 5/36</b> (2006.01)	G09G 5/36	520G

審査請求 未請求 予備審査請求 未請求 (全 28 頁)

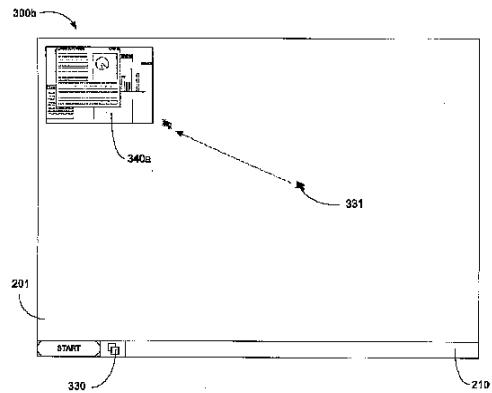
(21) 出願番号	特願2008-504008 (P2008-504008)	(71) 出願人	500046438 マイクロソフト コーポレーション アメリカ合衆国 ワシントン州 9805 2-6399 レッドモンド ワン マイ クロソフト ウェイ
(86) (22) 出願日	平成17年8月25日 (2005.8.25)	(74) 代理人	100077481 弁理士 谷 義一
(85) 翻訳文提出日	平成19年8月30日 (2007.8.30)	(74) 代理人	100088915 弁理士 阿部 和夫
(86) 國際出願番号	PCT/US2005/030312	(72) 発明者	チャールズ ダブリュ. スタブ 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 コーポレーション内
(87) 國際公開番号	W02006/107324		
(87) 國際公開日	平成18年10月12日 (2006.10.12)		
(31) 優先権主張番号	11/094,140		
(32) 優先日	平成17年3月31日 (2005.3.31)		
(33) 優先権主張国	米国(US)		

最終頁に続く

(54) 【発明の名称】アプリケーションウィンドウのグループ化および管理のための方法および装置

## (57) 【要約】

ディスプレイ画面上に複数のウィンドウを表示するための方法が説明される。本方法は、ディスプレイ画面(要素300b)上に第1のアプリケーションウィンドウ(要素340a)と、第1のアプリケーションウィンドウと独立しており外部に存在する第2のアプリケーションウィンドウ(要素331)とを表示するステップと、第1のアプリケーションウィンドウと第2のアプリケーションウィンドウとを1つのグループとして宣言するステップと、第1のアプリケーションウィンドウと第2のウィンドウとを1つのグループ(330)へとグループ化するステップと、ディスプレイ画面(330)上の表現としてそのグループを表示するステップとを含んでいる。新しいアプリケーションウィンドウが開かれるときに、それらは、そのグループに追加することができ、かつ/またはそれらは、第2のグループのアプリケーションウィンドウのメンバになることができる。ユーザは、様々なグループ間で切り換えることができる。



**【特許請求の範囲】****【請求項 1】**

複数のアプリケーションウィンドウをグループ化するための方法であって、

第1のアプリケーションウィンドウと、前記第1のアプリケーションウィンドウと独立してあり外部に存在する第2のアプリケーションウィンドウとをディスプレイ画面上に表示するステップと、

前記第1のアプリケーションウィンドウと前記第2のアプリケーションウィンドウとを1つのグループとして宣言するステップと、

前記第1のアプリケーションウィンドウと前記第2のウィンドウとを1つのグループへとグループ化するステップと、

前記グループを前記ディスプレイ画面上で表現として表示するステップと  
を含むことを特徴とする方法。

**【請求項 2】**

前記グループ上でオペレーションを実行するステップをさらに含むことを特徴とする請求項1に記載の方法。

**【請求項 3】**

実行する前記ステップは、入力デバイスからのコマンドの受け取りに基づくことを特徴とする請求項2に記載の方法。

**【請求項 4】**

動作を実行する前記ステップは、前記グループの前記アプリケーションウィンドウのサブセット上で実行されることを特徴とする請求項2に記載の方法。

**【請求項 5】**

前記第1のアプリケーションウィンドウは、第1のアプリケーションプログラムに対応し、前記第2のアプリケーションウィンドウは、第2のアプリケーションプログラムに対応することを特徴とする請求項1に記載の方法。

**【請求項 6】**

前記第1のアプリケーションプログラムと前記第2のアプリケーションプログラムとは、異なるアプリケーションプログラムであることを特徴とする請求項5に記載の方法。

**【請求項 7】**

前記第1のアプリケーションウィンドウと前記第2のアプリケーションウィンドウとを前記グループから取り出す入力を受け取るステップと、

前記第1のアプリケーションウィンドウと前記第2のアプリケーションウィンドウとを前記ディスプレイ画面上で元の状態へと回復させるステップと

をさらに含むことを特徴とする請求項1に記載の方法。

**【請求項 8】**

宣言する前記ステップは、前記第1のアプリケーションウィンドウと前記第2のアプリケーションウィンドウとをグループ化する入力を入力デバイスから受け取るステップを含むことを特徴とする請求項1に記載の方法。

**【請求項 9】**

前記表現は、前記ディスプレイ画面のデスクトップスペース中に表示される前記第1のアプリケーションウィンドウと前記第2のアプリケーションウィンドウとの縮小されたバージョンのサムネイル画像であることを特徴とする請求項1に記載の方法。

**【請求項 10】**

前記表現は、前記ディスプレイ画面上のタスクバー区域内のタスクバーボタンであることを特徴とする請求項1に記載の方法。

**【請求項 11】**

少なくとも1つの第3のアプリケーションウィンドウを表示するステップと、

前記少なくとも1つの第3のアプリケーションウィンドウを1つのグループとして宣言するステップと、

前記少なくとも1つの第3のアプリケーションウィンドウを第2のグループへとグル

10

20

30

40

50

化するステップと、

前記第2のグループを前記ディスプレイ画面上で第2の表現として表示するステップとをさらに含むことを特徴とする請求項1に記載の方法。

【請求項12】

前記第1のアプリケーションウィンドウと前記第2のアプリケーションウィンドウとを前記グループから取り出す入力を受け取るステップと、

前記第1のアプリケーションウィンドウと前記第2のアプリケーションウィンドウとを前記ディスプレイ画面上で元の状態に回復させるステップとをさらに含むことを特徴とする請求項11に記載の方法。

【請求項13】

前記少なくとも1つの第3のアプリケーションウィンドウを前記第2のグループから取り出す入力を受け取るステップと、

前記少なくとも1つの第3のアプリケーションウィンドウを前記ディスプレイ画面上で元の状態に回復させるステップとをさらに含むことを特徴とする請求項11に記載の方法。

【請求項14】

少なくとも1つの第3のアプリケーションウィンドウを表示するステップと、

前記第1のアプリケーションウィンドウと前記第2のアプリケーションウィンドウとを前記グループから取り出す入力を受け取るステップと、

前記第1のアプリケーションウィンドウと前記第2のアプリケーションウィンドウとを前記ディスプレイ画面上で元の状態に回復させるステップと、

前記入力を受け取ることに応じて前記少なくとも1つの第3のアプリケーションウィンドウを第2のグループへとグループ化するステップと、

前記第2のグループを前記ディスプレイ画面上で第2の表現として表示するステップとをさらに含むことを特徴とする請求項1に記載の方法。

【請求項15】

複数のアプリケーションウィンドウをグループ化するための一連のコンピュータ実行可能命令を含むコンピュータ読み取り可能媒体であって、前記命令は、

第1のアプリケーションウィンドウと、前記第1のアプリケーションウィンドウと独立しており外部に存在する第2のアプリケーションウィンドウとをディスプレイ画面上で表示するステップと、

前記第1のアプリケーションウィンドウと前記第2のアプリケーションウィンドウとを1つのグループとして宣言するステップと、

前記第1のアプリケーションウィンドウと前記第2のウィンドウとを1つのグループへとグループ化するステップと、

前記グループを前記ディスプレイ画面上で表現として表示するステップとを実行することを特徴とするコンピュータ読み取り可能媒体。

【請求項16】

前記命令は、さらに

前記第1のアプリケーションウィンドウと前記第2のアプリケーションウィンドウとを前記グループから取り出す入力を受け取るステップと、

前記第1のアプリケーションウィンドウと前記第2のアプリケーションウィンドウとを前記ディスプレイ画面上で元の状態へと回復させるステップとを実行することを特徴とする請求項15に記載のコンピュータ読み取り可能媒体。

【請求項17】

前記命令は、さらに

少なくとも1つの第3のアプリケーションウィンドウを表示するステップと、

前記少なくとも1つの第3のアプリケーションウィンドウを1つのグループとして宣言するステップと、

前記少なくとも1つの第3のアプリケーションウィンドウを第2のグループへとグル

10

20

30

40

50

化するステップと、

前記第2のグループを前記ディスプレイ画面上で第2の表現として表示するステップとを実行することを特徴とする請求項15に記載のコンピュータ読み取り可能媒体。

#### 【請求項18】

前記命令は、さらに

少なくとも1つの第3のアプリケーションウィンドウを表示するステップと、

前記第1のアプリケーションウィンドウと前記第2のアプリケーションウィンドウとを前記グループから取り出す入力を受け取るステップと、

前記第1のアプリケーションウィンドウと前記第2のアプリケーションウィンドウとを前記ディスプレイ画面上で元の状態に回復させるステップと、

前記入力を受け取ることに応じて前記少なくとも1つの第3のアプリケーションウィンドウを第2のグループへとグループ化するステップと、

前記第2のグループを前記ディスプレイ画面上で第2の表現として表示するステップとを実行することを特徴とする請求項15に記載のコンピュータ読み取り可能媒体。

#### 【請求項19】

グループ化されたウィンドウを表すデータを処理するためのソフトウェアアーキテクチャであって、

第1のアプリケーションウィンドウと第2のアプリケーションウィンドウとを前記第1のアプリケーションウィンドウと前記第2のアプリケーションウィンドウとのサムネイル画像を表すグループへとグループ化するように構成される少なくとも1つのコンポーネントと、

前記コンポーネントにアクセスする少なくとも1つのアプリケーションプログラムインターフェースと

を備えることを特徴とするソフトウェアアーキテクチャ。

#### 【請求項20】

ログアウトセッション中に前記グループの保存することをプログラムにより制御するように構成される少なくとも1つのコンポーネントをさらに備えることを特徴とする請求項19に記載のソフトウェアアーキテクチャ。

#### 【発明の詳細な説明】

##### 【技術分野】

##### 【0001】

本発明の態様は、一般にオペレーティングシステム中におけるウィンドウ構成を対象としている。より詳細には、本発明の態様は、オペレーティングシステム中における複数のウィンドウをグループ化し、何らかのインテリジェントな動作が諸ウィンドウ間に存在することができるようにするための方法およびシステムを対象としている。

##### 【背景技術】

##### 【0002】

労働においても私生活においてもコンピュータの使用が増大するにつれて、コンピュータのより簡単な使用を可能にしたいという要望がもたれている。今日では多数のオペレーティングシステムが、ウィンドウベースのコンフィギュレーションのアプリケーションプログラムを利用している。情報は、数枚の書類があるよう見えるようにしてディスプレイスクリーン上に表示される。諸ウィンドウとインターフェースすることにより、ユーザは、まるで1枚の書類をつかむかのように任意のウィンドウにアクセスすることができる。ウィンドウベースのコンフィギュレーションにより、ユーザは、ディスプレイ画面上に同時に複数のウィンドウを開くことができるようになる。

##### 【0003】

アプリケーションウィンドウは、すべてのグラフィックユーザインターフェース(graphical user interface) (GUI) システムのためのユーザインターフェース機能である。アプリケーションウィンドウは、システム間で外観が異なることもあるが、それらのアプリケーションウィンドウは、サイズ変更し位置を変更する機

10

20

30

40

50

能、異なる複数のアプリケーションに関連する他の複数のアプリケーションウィンドウの間に存在させる機能など、多数の共通の属性を共有する。複数のアプリケーションウィンドウは、一緒に、互いに他方の上に層状に重ねられ、一般的に各ウィンドウがユーザによって最後にアクセスされた順序によって表わされて画面上に同時に現れる。

#### 【0004】

現代のG U Iに共通のユーザ対話のシナリオは、共通の画面領域 (s c r e e n r e a l e s t a t e) を共用する複数の同時に開かれたアプリケーションウィンドウを伴う。複数の同時に開かれたアプリケーションウィンドウについてのサポートは、現代のG U Iの魅力および能力の一部分であるが、これは、しばしばアプリケーションウィンドウのオーバーラップと互いに見えなくすることをもたらし、ユーザが特定のアプリケーションウィンドウを見出し、それにナビゲートすることを困難にする。このタイプのシナリオと関連するソリューションは、一般にウィンドウ管理の問題とソリューションと称される。

10

#### 【0005】

ユーザは一般的に、アプリケーションウィンドウを電子メールなどのタスクに関連付け、さらに複数のアプリケーションウィンドウと単一のタスクまたはプロジェクトとの間の精神的な関連付けを作成することができる。例えば、スライドショープレゼンテーション (s l i d e s h o w p r e s e n t a t i o n) を生成するために、ユーザは、おそらく電子メール、グラフィックアプリケーション、スプレッドシートアプリケーションなど、1つまたは複数のアプリケーションからの3つまたは4つの異なるアプリケーションウィンドウからのコンテンツを必要とすることもある。複数のアプリケーションウィンドウが、画面上で目に見えることもあり、各アプリケーションウィンドウは、1つまたは複数のタスクに関連付けられる。ユーザは、これらの異なるアプリケーションウィンドウの間で所在を見出しナビゲートして、コンテンツにアクセスし、またはコンテンツを交換して、タスクを完了する必要がある。

20

#### 【0006】

多数のG U I中には共通のウィンドウ管理ソリューションが存在し、それによってアプリケーションウィンドウは、一次作業画面スペース中において最小にし、ビューから取り除くことができる。最小化されたアプリケーションウィンドウは、一般的にかなり小さなU I要素によって表される。ワシントン州、レッドモンド市のマイクロソフト (登録商標) (M i c r o s o f t<sup>(R)</sup> C o r p o r a t i o n) によるW i n d o w s (登録商標) X Pにおいては、最小化されたアプリケーションウィンドウは、タスクバー・コントロール (T a s k B a r c o n t r o l) など、コントロールバー (c o n t r o l b a r) 上のボタン (b u t t o n) として表される。カリフォルニア州、クバチーノ市のアップルコンピュータ社 (A p p l e C o m p u t e r, I n c.) によるM a c

30

O S Xにおいては、最小化されたアプリケーションウィンドウは、ドック (D o c k) 中の小型サムネイルボタン (m i n i a t u r e t h u m b n a i l b u t t o n) として表される。アプリケーションウィンドウを最小化することにより、ユーザは、画面スペースから不必要的アプリケーションウィンドウを取り除くことができ、それによってユーザは、アプリケーションウィンドウのより小さな組に集中することができる。しかし、アプリケーションウィンドウを最小化することは、個別に実行することができるだけである。同様に、アプリケーションウィンドウは、最小化状態から取り出し、目に見えるアプリケーションウィンドウの組に対して再導入することができるが、これは、個別のアプリケーションウィンドウに基づいて行うことができるにすぎない。

40

#### 【発明の開示】

#### 【発明が解決しようとする課題】

#### 【0007】

現在では、独立した外部のウィンドウは、一緒にしてグループ化することができず、それ故に異種のウィンドウは、一緒に管理することができない。オペレーティングシステムは、現在では互いに一緒にグループ化すべき複数のウィンドウを許可せず、その結果、1

50

つのウィンドウ上で実行されるオペレーションは、第2の独立した外部のアプリケーションウィンドウ上で自動的に実行することができない。

【課題を解決するための手段】

【0008】

任意の複数のアプリケーションウィンドウを一緒にグループ化し、それ故に異種ウ 10 ィンドウを、一緒に管理することができる機能が必要になっている。ディスプレイ画面上で第1のアプリケーションウィンドウと、第1のアプリケーションウィンドウと独立しており外部に存在する第2のアプリケーションウィンドウとを表示し、第1のアプリケーションウィンドウと第2のアプリケーションウィンドウとを1つのグループとして宣言し、第1のアプリケーションウィンドウと第2のアプリケーションウィンドウとを1つのグループへとグループ化し、ディスプレイ画面上で表現としてそのグループを表示するための方法が必要になっている。

【0009】

本発明の態様は、複数のアプリケーションウィンドウを1つのグループとして管理する際の助けとなるウィンドウ管理システムを提供する。本発明の態様は、アプリケーションウィンドウの1つのグループを作成し、そのグループを最小化し、それによって以前に識別されたグループを取り出すことができるようとする新しい方法および装置を提供する。本方法により、ユーザは、それらのアプリケーションウィンドウからタスクへの関連付けを形式化し、例えば1つのグループを作成し、これらの関連付け、例えばアプリケーションウィンドウグループアイコン(application window group icon)を管理する機能を提供することができるようになる。 20

【0010】

本発明の前述の概要、ならびに例示の実施形態についての以降の詳細な説明は、一例として含められるが、請求されている本発明に関する限定としては含められていない添付図面と併せて読まれるときに、よりよく理解される。

【発明を実施するための最良の形態】

【0011】

様々な例示の実施形態の以降の説明においては、本明細書の一部分を形成し、本発明を実行することができる様々な実施形態が例証として示される添付図面に対する参照が行わ 30 れる。他の実施形態を利用することができます、本発明の範囲を逸脱することなく構成および機能の修正を行うことができることを理解すべきである。

【0012】

図1は、本発明を実装することができる適切なコンピューティングシステム環境100の一例を示すものである。コンピューティングシステム環境100は、適切なコンピューティング環境の一例にすぎず、本発明の用途または機能の範囲についてどのような限定も示唆することは意図されてはいない。コンピューティングシステム環境100は、例示のコンピューティングシステム環境100中に示されるコンポーネントの任意の1つまたは組合せに関連したどのような依存性も要件も有するものと解釈されるべきではない。

【0013】

本発明は、非常に多数の他の汎用または専用のコンピューティングシステムの環境またはコンフィギュレーションを用いて動作可能である。本発明と共に使用するのに適したものとすることができるよく知られているコンピューティングシステム、コンピューティング環境、および/またはコンピューティングコンフィギュレーションの例は、それだけには限定されないが、パーソナルコンピュータ、サーバコンピュータ、ハンドヘルドデバイス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、プログラマブルな大衆消費電子製品、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、以上のシステムまたはデバイスのどれかを含む分散コンピューティング環境などを含む。 40

【0014】

本発明は、コンピュータによって実行される、プログラムモジュールなどのコンピュー

10

20

30

40

50

タ実行可能命令の一般的な場合について説明することができる。一般に、プログラムモジュールは、特定のタスクを実行し、あるいは特定の抽象データ型を実装する、ルーチン、プログラム、オブジェクト、コンポーネント、データ構造などを含んでいる。本発明は、タスクが、通信ネットワークを介し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場合の分散コンピューティング環境において実行することもできる。分散コンピューティング環境においては、プログラムモジュールは、メモリストレージデバイスを含めて、ローカルコンピュータストレージ媒体にもリモートコンピュータストレージ媒体にも配置することができる。

#### 【0015】

図1Aを参照すると、本発明を実装するための例示のシステムは、コンピュータ110の形態の汎用コンピューティングデバイスを含んでいる。コンピュータ110のコンポーネントは、それだけには限定されないが、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めて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、様々なバスアーキテクチャのうちの任意のものを使用したメモリバスまたはメモリコントローラ、ペリフェラルバス、およびローカルバスを含めて、いくつかのタイプのバス構造のうちの任意のものであってもよい。一例として、限定するものではないが、そのようなアーキテクチャは、業界標準アーキテクチャ(Industry Standard Architecture)(ISA)バス、マイクロチャネルアーキテクチャ(Micro Channel Architecture)(MCA)バス、拡張ISA(Enhanced ISA)(EISA)バス、ビデオエレクトロニクス規格協会(Video Electronics Standard Association)(VESA)ローカルバス、およびメザニンバス(Mezzanine bus)としても知られているペリフェラルコンポーネント相互接続(Peripheral Component Interconnect)(PCI)バスを含んでいる。

#### 【0016】

コンピュータ110は、一般的に様々なコンピュータ読み取り可能媒体を含んでいる。コンピュータ読み取り可能媒体は、コンピュータ110によってアクセスすることができ、揮発性媒体も不揮発性媒体も、着脱可能媒体も着脱不能媒体も含む使用可能な任意の媒体とすることができます。一例として限定するものではないが、コンピュータ読み取り可能媒体は、コンピュータストレージ媒体および通信媒体を備えることができる。コンピュータストレージ媒体は、コンピュータ読み取り可能命令、データ構造、プログラムモジュール、他のデータなどの情報の記憶のための任意の方法または技術で実装された揮発性および不揮発性の着脱可能および着脱不能な媒体を含んでいる。コンピュータストレージ媒体は、それだけには限定されないが、望ましい情報を記憶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ランダムアクセスメモリ(random access memory)(RAM)、読み取り専用メモリ(read only memory)(ROM)、電子的消去可能プログラマブル読み取り専用メモリ(electrically erasable programmable read only memory)(EEPROM)、フラッシュメモリまたは他のメモリ技術、CD-ROM、デジタル多用途ディスク(digital versatile disk)(DVD)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、あるいは任意の他の媒体を含む。通信媒体は、一般的に搬送波や他の移送メカニズムなどの被変調データ信号の形のコンピュータ読み取り可能命令、データ構造、プログラムモジュールまたは他のデータを実装し、任意の情報配信媒体を含んでいる。用語「被変調データ信号」は、情報を信号の形で符号化するような方法で1つまたは複数のその特性が設定され、または変更された信号を意味する。一例として、限定するものではないが、通信媒体は、有線ネットワーク、直接配線接続などの有線媒体と、音響媒体、RF媒体、赤外線媒体、他の無線媒体などの無線媒体を含んでいる。以上のうちの任意の組合せもまた、コンピュータ読み取り可能媒体の範囲内に含めら

10

20

30

40

50

れるべきである。

【0017】

システムメモリ130は、ROM131やRAM132など、揮発性および／または不揮発性のメモリの形態のコンピュータストレージ媒体を含んでいる。起動中などにコンピュータ110内の要素間の情報を転送する助けを行う基本ルーチンを含む基本入出力システム(basic input/output system)(BIOS)133は、一般的にROM131中に記憶される。RAM132は、一般的に処理装置120によって直接にアクセス可能であり、かつ／または現在動作させられているデータおよび／またはプログラムモジュールを含んでいる。一例として、限定するものではないが、図1A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を示している。

10

【0018】

コンピュータ110は、他の着脱可能／着脱不能な揮発性／不揮発性のコンピュータストレージ媒体も含むこともできる。一例としてだけであるが、図1Aは、着脱不能な不揮発性の磁気媒体から読み取り、またはそれに書き込むハードディスクドライブ141と、着脱可能な不揮発性の磁気ディスク152から読み取り、それに書き込む磁気ディスクドライブ151と、CD-ROMや他の光媒体など着脱可能な不揮発性の光ディスク156から読み取り、またはそれに書き込む光ディスクドライブ155とを示している。例示の動作環境中において使用することができる他の着脱可能／着脱不能な揮発性／不揮発性のコンピュータストレージ媒体は、それだけには限定されないが、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、ソリッドステートRAM、ソリッドステートROMなどを含んでいる。ハードディスクドライブ141は、一般的にインターフェース140などの着脱不能メモリインターフェースを介してシステムバス121に接続され、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、一般的にインターフェース150などの着脱可能メモリインターフェースによってシステムバス121に接続される。

20

【0019】

以上で論じられ図1Aに示されたドライブとそれらの関連するコンピュータストレージ媒体は、コンピュータ110についてのコンピュータ読み取り可能命令、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を実現する。図1Aにおいては、例えば、ハードディスクドライブ141は、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を記憶するものとして示される。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と同じであっても、またはそれらとは異なっていてもどちらにもすることができます。これに注意すべきである。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は、少なくともそれらが異なるコピーであることを示すためにここでは異なる番号が付与される。ユーザは、デジタルカメラ163、キーボード162、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ポインティングデバイス161などの入力デバイスを介してコンピュータ110にコマンドおよび情報を入力することができる。(図示されていない)他の入力デバイスは、ペン、スタイラス(stylus)およびタブレット(tablet)、マイクロフォン、ジョイスティック、ゲームパッド、サテライトディッシュ(satellite dish)、スキャナなどを含むことができる。これらおよび他の入力デバイスは、多くの場合にシステムバス121に結合されたユーザ入力インターフェース160を介して処理装置120に接続されるが、パラレルポート、ゲームポート、ユニバーサルシリアルバス(universal serial bus)(USB)などの他のインターフェースおよびバス構造によって接続することもできる。モニタ191または他のタイプのディスプレイデバイスも、ビデオインターフェース190などのインターフェースを経由してシステムバス121に接続される。モニタに追加して、コ

30

40

50

ンピュータはまた、スピーカ 197 やプリンタ 196 などの他のペリフェラル出力デバイスを含むこともでき、これらは、出力ペリフェラルインターフェース 195 を介して接続することができる。

#### 【0020】

コンピュータ 110 は、リモートコンピュータ 180 などの 1 つまたは複数のリモートコンピュータに対する論理接続を使用してネットワーク化された環境中におい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のネットワークノードとすることができます、一般的にコンピュータ 110 に対して以上で説明された要素のうちの多くまたはすべてを含んでいるが、メモリストレージデバイス 181 だけが図 1A には示されている。図 1A 中に示された論理接続は、ローカルエリアネットワーク (local area network) (LAN) 171 と、ワイドエリアネットワーク (wide area network) (WAN) 173 を含んでいるが、他のネットワークを含むこともできる。そのようなネットワーキング環境は、オフィス、企業規模のコンピュータネットワーク、イントラネットおよびインターネットにおいて一般的になっている。

10

#### 【0021】

LAN ネットワーキング環境において使用されるときには、コンピュータ 110 は、ネットワークインターフェースまたはアダプタ 170 を介して LAN 171 に接続される。WAN ネットワーキング環境において使用されるときには、コンピュータ 110 は、一般的にインターネットなどの WAN 173 上で通信を確立するためのモデム 172 または他の手段を含んでいる。内部にあっても、または外部にあってもよいモデム 172 は、ユーザ入力インターフェース 160 または他の適切なメカニズムを経由してシステムバス 121 に接続することができる。ネットワーク化環境においては、コンピュータ 110 に対して示されるプログラムモジュール、またはその一部分は、リモートメモリストレージデバイスに記憶することができる。一例として限定するものではないが、図 1A は、リモートアプリケーションプログラム 185 をメモリデバイス 181 上に存在するものとして示している。図に示されるネットワーク接続は、例示的なものであり、コンピュータ間で通信リンクを確立する他の手段が使用されてもよいことが理解されよう。

20

#### 【0022】

図に示されるネットワーク接続は、例示的なものであり、コンピュータ間で通信リンクを確立する他の手段を使用することもできることが理解されよう。TCP/IP、イーサネット（登録商標）、FTP、HTTP など、様々なよく知られているプロトコルのうちの任意のものの存在が仮定され、システムは、クライアント - サーバのコンフィギュレーション中で動作させられて、それにより、ユーザは、ウェブベースのサーバからウェブページを取り出すことができるようになる。様々な従来のウェブブラウザのうちの任意のものを使用して、ウェブページ上でデータを表示し、操作することができる。

30

#### 【0023】

プログラミングインターフェース（またはより簡単に、インターフェース）は、コードの 1 つまたは複数の他のセグメントによって実現される機能と情報をやり取りし、あるいはその機能にアクセスするコードの 1 つまたは複数のセグメントを可能にするための任意のメカニズム、プロセス、プロトコルとみなすことができる。代わりに、プログラミングインターフェースは、他の 1 つ（または複数）のコンポーネントのうちの 1 つまたは複数のメカニズム、メソッド (method)、ファンクションコール (function call)、モジュールなどに結合して情報伝達することができるシステムのコンポーネントの 1 つまたは複数のメカニズム、メソッド、ファンクションコール、モジュール、オブジェクトなどとみなすことができる。先行する文章中の用語「コードのセグメント」は、コードの 1 つまたは複数の命令またはラインを含むことを意図しており、例えば、適用される専門用語にかかわらず、あるいはコードセグメントが別々にコンパイルされるかどうか、コードセグメントがソースコード、中間コード、またはオブジェクトコードとして提供されるかどうか、コードセグメントがランタイムのシステムまたはプロセス中で利用

40

50

されるかどうか、それらが同じマシンもしくは異なるマシン上に配置され、または複数のマシンにまたがって分散されるかどうか、またはコードのセグメントによって表される機能が、全体的にソフトウェアで、全体的にハードウェアで、またはハードウェアとソフトウェアの組合せの形で実装されるかどうかにかかわらず、コードモジュール、オブジェクト、サブルーチン、ファンクションなどを含んでいる。

【0024】

概念的に、プログラミングインターフェースは、図1Bまたは図1Cに示されるように、包括的に考えることができる。図1Bは、それを介して第1および第2のコードセグメントが情報をやり取りするコンジット(*conduit*)としてインターフェース、インターフェース1を示している。図1Cは、(第1および第2のコードセグメントの一部であっても、あるいはそうでなくてもよい)インターフェースオブジェクトI1およびI2を備えるものとしてインターフェースを示しており、これらのインターフェースオブジェクトにより、システムの第1および第2のコードセグメントは媒体Mを経由して情報をやり取りすることができるようになる。図1Cの図においては、人は、インターフェースオブジェクトI1とI2とを同じシステムの別のインターフェースとして考えることができ、人はまた、オブジェクトI1およびI2プラス媒体Mがインターフェースを構成するものと考えることもできる。図1Bおよび1Cは、双方向フローと、そのフローのおおのの側上のインターフェースを示すが、ある種の実装形態は、一方向の情報フローを有することができるだけであり(または以下で説明されるように情報フローがない)、あるいは一方の側上にインターフェースオブジェクトを有することができるだけである。一例として、限定するものではないが、アプリケーションプログラミングインターフェース(*application programming interface*)(*API*)、エントリポイント(*entry point*)、メソッド、ファンクション、サブルーチン、リモートプロシージャコール(*remote procedure call*)、およびコンポーネントオブジェクトモデル(*component object model*)(*COM*)インターフェースなどの用語は、プログラミングインターフェースの定義内に包含される。

10

20

30

40

【0025】

そのようなプログラミングインターフェースの態様は、第1のコードセグメントが情報(ここで、「情報」は、その最も広い意味で使用され、データ、コマンド、要求などを含んでいる)を第2のコードセグメントに送る方法と、第2のコードセグメントがその情報を受け取る方法と、その情報の構成、シーケンス、構文(*syntax*)、組織、スキーマ、タイミングおよびコンテンツを含むことができる。この点については、基礎になっている移送媒体それ自体は、情報がインターフェースによって定義されるように移送される限り、媒体が有線であれ、または無線であれ、あるいは両方の組合せであれ、インターフェースのオペレーションにとって重要でないこともある。ある種の状況においては、1つのコードセグメントが、単に第2のコードセグメントによって実施される機能にアクセスするときなど、情報転送は、別のメカニズムを経由している(例えば、バッファ、ファイルなどに配置された情報は、コードセグメントの間の情報フローから別になっている)か、または存在しないこともあるので、情報は、従来の意味で一方向または両方向に渡されないこともある。これらの態様のうちの任意のもの、または、すべては、例えばコードセグメントが疎結合または密結合のコンフィギュレーションにおけるシステムの一部分であるかどうかに応じて、与えられた状況において重要なこともあります、それ故に、このリストは、例示的であり非限定的であると考えられるべきである。

50

【0026】

プログラミングインターフェースのこの概念は、当業者には知られており、本発明の前述の詳細な説明から明らかである。しかし、プログラミングインターフェースを実装する他の方法が存在し、明示的に除外されない限り、これらはまた、本明細書に添付された特許請求の範囲によって包含されることが意図される。そのような他の方法は、図1Bおよび1Cの極度に単純化された図よりも高度または複雑であるように見えることもあるが、

それらは、それにもかかわらず、同じ全体的な結果を遂行する類似した機能を実行する。本発明者等は、次にプログラミングインターフェースの一部の例示的な代替実装形態について簡単に説明することにする。

### 【0027】

#### A. ファクタリング (FACTURING)

1つのコードセグメントから別のコードセグメントに対するコミュニケーションは、コミュニケーションを複数の離散的なコミュニケーションに分解することにより、間接的に遂行することができる。これは、図1Dおよび1Eに概略的に示される。図に示されるように、一部のインターフェースは、機能の分割可能な組の観点から説明することができる。したがって、図1Bおよび1Cのインターフェース機能は、ちょうど人が数学的に24、または $2 \times 2 \times 3 \times 2$ を提供することができるよう同じ結果を達成するためにファクタリングする (factor) ことができる。したがって、図1Dに示されるように、インターフェース、インターフェース1によって提供されるファンクションは、同じ結果を達成しながら、再分割してインターフェースのコミュニケーションを複数のインターフェース、インターフェース1A、インターフェース1B、インターフェース1Cなどへと変換することができる。図1Eに示されるように、インターフェースI1によって提供されるファンクションは、同じ結果を達成しながら、複数のインターフェースI1a、I1b、I1cなどに再分割することができる。同様に、第1のコードセグメントから情報を受け取る第2のコードセグメントのインターフェースI2は、複数のインターフェースI2a、I2b、I2cなどへとファクタリングすることができる。ファクタリングするときに、第1のコードセグメントと共に含められるインターフェースの数は、第2のコードセグメントと共に含められるインターフェースの数と一致する必要はない。図1Dおよび1Eの場合のいずれにおいても、インターフェース、インターフェース1およびI1の機能的な趣旨は、それぞれ図1Bおよび1Cと同じままに留まる。インターフェースのファクタリングはまた、ファクタリングを認識することが困難なこともあるように、連想的な性質、交換可能な性質、および他の数学的な性質に従うこともできる。例えば、オペレーションの順序は、重要でないこともあり、その結果、インターフェースによって実行されるファンクションは、インターフェースに到達するのに先立って、コードまたはインターフェースの別の部分によってうまく実行し、あるいはシステムの別個のコンポーネントによって実行することができる。さらに、プログラミング技術分野の当業者は、同じ結果を達成する異なるファンクションコールを行う様々な方法が存在することを認識することができる。

### 【0028】

#### B. 再定義

一部の場合においては、意図される結果を依然として遂行しながら、プログラミングインターフェースのある種の態様（例えば、パラメータ）を無視し、追加し、または再定義することが可能なこともある。これについては、図1Fおよび1Gに示される。例えば、図1Bのインターフェース、インターフェース1は、3つのパラメータ、入力、精度および出力を含み、第1のコードセグメントから第2のコードセグメントへと発行されるファンクションコール、スクエア (Square) (入力, 精度, 出力) を含んでいる。図1Fに示されるように、中間のパラメータ精度が、与えられたシナリオにおいて関係がない場合、それは、同様にただ無視し、あるいは（この状況では）意味のないパラメータと置き換えることさえできる。人はまた、関係のない追加パラメータを追加することもできる。いずれにしても、入力が第2のコードセグメントによって2乗された後に、出力が戻される限り、スクエアの機能は、達成することができる。精度は、確実にコンピューティングシステムのあるダウンストリームまたは他の部分に対する意味のあるパラメータとなり得る。しかし、ひとたび精度がスクエアを計算する狭い目的のためには必要でないことが認識されると、精度は、置き換え、または無視することができる。例えば、有効な精度値を渡す代わりに、誕生日など、意味のない値は、その結果に悪影響を及ぼすことなく渡すことができる。同様に、図1Gに示されるように、インターフェースI1は、インターフ

10

20

30

40

50

エース I 1' によって置き換えられ、インターフェースに対するパラメータを無視または追加するために再定義される。インターフェース I 2 も同様に、インターフェース I 2' として再定義され、不必要的パラメータ、あるいは他のどこかで処理することができるパラメータを無視するために再定義される。ここでのポイントは、一部の場合においては、プログラミングインターフェースは、ある目的のために必要とされない、パラメータなどの態様を含むことができ、それ故にそれらは、無視または再定義し、あるいは他の目的のために他のどこかで処理することができる。

#### 【 0 0 2 9 】

##### C. インラインコーディング ( I N L I N E C O D I N G )

それらのコードモジュールの間の「インターフェース」が形態を変更するように、2つの別個のコードモジュールの機能の一部またはすべてをマージすることも実現可能なこともある。例えば、図 1 B および 1 C の機能は、それぞれ図 1 H および 1 I の機能に変換することができる。図 1 Hにおいては、図 1 B の以前の第 1 のコードセグメントと第 2 のコードセグメントは、それらの両方を含むモジュールにマージされる。この場合に、それらのコードセグメントは、依然として互いに情報をやり取りしていることもあるが、インターフェースは、単一のモジュールにより適した形態に適応させることができる。したがって、例えば形式的なコールステートメント ( C a l l s t a t e m e n t ) およびリターンステートメント ( R e t u r n s t a t e m e n t ) が、もはや必要でないこともあり、インターフェース、インターフェース 1 に従った同様な処理または 1 つ ( または複数 ) の応答が、依然として効力があることもある。同様に、図 1 I に示されるように、図 1 C からのインターフェース I 2 の一部 ( または全部 ) は、インターフェース I 1 にインラインに書き込まれ、インターフェース I 1" を形成することができる。例示されるように、インターフェース I 2 は、I 2 a と I 2 b に分割され、インターフェース部分 I 2 a はインターフェース I 1 と共にインラインに符号化されて、インターフェース I 1" を形成している。具体的な例では、図 1 C からのインターフェース I 1 は、ファンクションコール、スクエア ( 入力 , 出力 ) を実行し、このスクエア ( 入力 , 出力 ) は、インターフェース I 2 によって受け取られ、このスクエア ( 入力 , 出力 ) は、第 2 のコードセグメントによって ( それを 2 乗する ) 入力と共に渡される値を処理した後に、2 乗された結果を出力と共に戻して渡す。そのような場合には、第 2 のコードセグメント ( 2 乗する入力 ) によって実行される処理は、そのインターフェースに対するコールなしに第 1 のコードセグメントによって実行することができる。

#### 【 0 0 3 0 】

##### D. 分離 ( D I V O R C E )

1 つのコードセグメントから別のコードセグメントへのコミュニケーションは、そのコミュニケーションを複数の離散的なコミュニケーションへと分解することにより、間接的に遂行することができる。これについては、図 1 J および 1 K に概略的に示される。図 1 J に示されるように、1 つまたは複数のミドルウェアの部分 ( それらが、元のインターフェースから機能および / またはインターフェースのファンクションを分離するので、1 つ ( または複数 ) の分離インターフェースという ) が、第 1 のインターフェース、インターフェース 1 上のコミュニケーションを変換して、異なるインターフェース、この場合には諸インターフェース、インターフェース 2 A 、インターフェース 2 B およびインターフェース 2 C に準拠させるように設けられる。これは、例えば、言わばインターフェース 1 プロトコルに従ってオペレーティングシステムと情報をやり取りするように設計されたインストールベースのアプリケーションが存在する場合に行われる可能性があるが、次いでそのオペレーティングシステムは、異なるインターフェース、この場合には諸インターフェース、インターフェース 2 A 、インターフェース 2 B およびインターフェース 2 C を使用するように変更される。要点は、第 2 のコードセグメントによって使用される元のインターフェースは、それが第 1 のコードセグメントによって使用されるインターフェースとはもはや互換性がないように変更され、それ故に仲介手段が、古いインターフェースと新しいインターフェースを互換にするために使用されることである。同様に、図 1 K に示され

10

20

30

40

50

るよう、第3のコードセグメントが、インターフェースI1からのコミュニケーションを受け取る分離インターフェースD1と、例えばD2と共に機能するように設計変更されたインターフェースI2aおよびI2bへとインターフェース機能を伝送するが、同じ機能結果を実現する分離インターフェースD2と共に導入することができる。同様に、D1およびD2は、同じまたは類似した機能結果を実現しながら、図1CのインターフェースI1およびI2の機能を新しいオペレーティングシステムに対して変換するように一緒に機能することができる。

【0031】

E. 再書き込み (REWRTING)

さらに別の可能性のある変形は、コードを動的に再書き込みして、インターフェース機能を何か他のもので置き換えることであるが、これは、同じ全体的な結果を達成する。例えば、中間言語（例えば、マイクロソフトのIL、Java（登録商標）バイトコード（Byte Code）など）で提示されるコードセグメントが、（.Netフレームワーク、Java（登録商標）ランタイム環境、または他の類似したランタイムタイプの環境によって提供される環境など）実行環境中におけるジャストインタイム（Just-in-Time）（JIT）のコンパイラまたはインターブリタに供給される。JITコンパイラは、第1のコードセグメントから第2のコードセグメントへとコミュニケーションを動的に変換するために、すなわち第2のコードセグメント（元のまたは異なる第2のコードセグメントのいずれか）によって必要とされるようにそれらを異なるインターフェースに準拠させるために書き込むことができる。これについては、図1Lおよび1Mに示される。図1Lから分かるように、このアプローチは、上記の分離シナリオに類似している。例えば、インストールベースのアプリケーションが、インターフェース1プロトコルに従ってオペレーティングシステムと情報をやり取りするように設計されるが、次いでそのオペレーティングシステムは、異なるインターフェースを使用するために変更される場合に、それは、行うことができる。JITコンパイラを使用して、インストールベースのアプリケーションからの実行中のコミュニケーションをオペレーティングシステムの新しいインターフェースに準拠させることができる。図1Mに示されるように、1つ（または複数）のインターフェースを動的に書き換えるこのアプローチは、同様に1つ（または複数）のインターフェースを同様に動的にファクタリングし、あるいはそうでなければ変更するように適用することができる。

【0032】

代替実施形態を経由してインターフェースと同じまたは類似した結果を達成するための上記のシナリオは、様々な方法で、逐次的および／または並列的に、あるいは他の介入するコードと共に、組み合わせることもできることにも注意すべきである。したがって、以上で提示された代替実施形態は、相互に排他的ではなく、混合し、マッチングさせ、組み合わせて、図1Bおよび1C中において提示される包括的なシナリオと同じまたは同等なシナリオを生成することができる。ほとんどのプログラミング構造と同様に、本明細書中で説明されないこともあるが、それにもかかわらず本発明の趣旨および範囲によって表されるインターフェースの同じまたは類似した機能を達成する他の類似した方法が存在することにも注意すべきであり、すなわち、それは、少なくとも部分的にはインターフェースの価値の基礎になるインターフェースによって表される機能であり、またそのインターフェースによって可能とされる有利な結果であることに注意すべきである。

【0033】

図2は、グラフィックユーザインターフェースシステムに共通の従来のウィンドウ管理シナリオ200を示している。図に示されるように、複数の開かれたアプリケーションウィンドウは、前面に位置付けられた他のより大きなウィンドウによって部分的に、または完全に見えなくされる。特に、図2は、アプリケーションウィンドウ203dが、ウィンドウ203a、203b、および203cよりもZ順（Z-order）が高くなっている、Z順のコンフィギュレーションの4枚のウィンドウ203a～203dのシナリオを示している。アプリケーションウィンドウ203a～203dは、デスクトップスペース

10

20

30

40

50

(desktop space) 201 内に示される。

【0034】

デスクトップスペース 201 は、アプリケーションプログラムに対応するウィンドウの表示を可能にするディスプレイの 1 つの区域である。底部にあるタスクバー 210 は、どのアプリケーションウィンドウが現在使用中であり、どれが見ることができ、または最小化されていることがあるかを示す。タスクバー 210 は、アクティブにすること、移動させること、隠すこと、最小化することなど、ウィンドウの操作をリストアップし、インペラブルにするために使用されるオンスクリーンウィンドウリモートコントロール (on-screen window remote control) の特定の実装形態である。ウィンドウ 203a は、タスクバーボタン (taskbar button) 213a によって表され、ウィンドウ 203b は、タスクバーボタン 213b によって表され、ウィンドウ 203c は、タスクバーボタン 213c によって表され、ウィンドウ 203d は、タスクバーボタン 213d によって表される。この例に示されるように、4 つのウィンドウ 203a ~ 203d が、開かれて示される。ウィンドウ 203a ~ 203d のコンテンツのファイル名は、それぞれのウィンドウのタイトルバー区域 223a ~ 223d に沿って示される。これらのファイル名は、ウィンドウを動作させるアプリケーションプログラムによって生成することができ、かつ / またはユーザによってカスタマイズ化可能とすることができる。各ウィンドウ 203a ~ 203d についての同じファイル名が、対応するタスクバーボタン 213a ~ 213d に示される。

10

【0035】

ウィンドウ 203a ~ 203d は、ユーザにコンテンツを表示するためにアプリケーションプログラムによって使用される。ウィンドウ 203a に関連するアプリケーションプログラムは、ウィンドウ 203b、203c、および / または 203d に対応する異なるアプリケーションプログラムまたは同じアプリケーションプログラムであってもよい。ウィンドウ 203a ~ 203d のおのののは、他のウィンドウ 203a ~ 203d に対して独立しており、また外部にある。本明細書中で使用されるように、第 1 のウィンドウは、第 1 のウィンドウが第 2 のウィンドウ内に含まれず、第 2 のウィンドウが第 1 のウィンドウ内に含まれないときに、第 2 のウィンドウに対して外部にある。2 つのウィンドウがただオーバーラップしているにすぎない場合に、第 1 のウィンドウは、第 2 のウィンドウ内に含まれないことを理解すべきである。

20

【0036】

図 2 に示されるように、ユーザは、タスクを完了するために、この組のアプリケーションウィンドウ 203a ~ 203d を明示的に開いている。タスクは、開かれたアプリケーションウィンドウ 203a ~ 203d のおののから情報またはコンテンツを取得することを必要とすることもある。例えば、このタスクは、インターネットブラウザウィンドウ (Internet browser window) とグラフィックス編集アプリケーションからの 2 つのウィンドウを必要とすることもある。ユーザが、この第 1 のタスクに関連していない追加のアプリケーションウィンドウを開く場合、これらの既存のアプリケーションウィンドウ 203a ~ 203d は、デスクトップスペース 201 中で変更されないまま留まることになる。既存のウィンドウ管理ソリューションを使用して、ユーザは、ビューからそれらのウィンドウを個別に取り除くために第 1 のタスクに関連するアプリケーションウィンドウ 203a ~ 203d のおののを最小化することができる。元のタスクに戻るためには、ユーザは、各最小化されたアプリケーションウィンドウ 203a ~ 203d を個別に取り出す必要があるはずである。

30

【0037】

図 3A に示されるように、本発明の態様によれば、ユーザインターフェースコントロール 330 が、1 組の現在開かれており、かつ / またはアクティブなアプリケーションウィンドウ 203a ~ 203d をウィンドウグループとして宣言するために設けられる。図 3A は、本発明の少なくとも一態様による、グループ化コントロール (grouping control) 330 を含むウィンドウ管理システムの一例を示している。本明細書中

40

50

で使用されるように、用語「宣言する」および「宣言すること」は、本明細書中ではアクションを開始して2つの独立な外部のアプリケーションウィンドウの間の関係を作成することを意味するように定義される。

#### 【0038】

シナリオ300aにおいては、ユーザは、アプリケーションウィンドウグループ化コントロール330に向かってポインタ331を移動させる。タスクバー区域210に示されているが、グループ化コントロール要素330は、デスクトップスペース201内など他のロケーションに配置することができることが、当業者によって理解されるべきである。さらに、作動可能ボタンとして図3Aには示されているが、グループ化コントロール330を開始して、他の手段によるアプリケーションウィンドウをグループ化することができる。例えば、一実施形態においては、ユーザは、单一および/または一連のキー/ボタンのストロークを実行して、アプリケーションウィンドウ203a～203dなど、複数のウィンドウをグループ化することができる。ユーザは、キーボード上であらかじめ定義された一連のキーストロークを押して、アプリケーションウィンドウをグループ化することができる。本発明の別の実施形態においては、タスクバーボタン213aをタスクバーボタン213bへとクリックしドラッグして、ウィンドウ203aおよびウィンドウ203bをグループ化するためのプロセスを開始することができる。さらに別の実施形態においては、ユーザは、電子的マウスなどの入力デバイス上の作動ボタンをクリックして、タスクバーボタン213aおよび213bを作動させることにより直接または間接にアプリケーションウィンドウ213aおよび213bをグループ化することができる。本発明は、本明細書中に含まれる例だけに限定されるものでないことが、当業者によって理解されるべきである。

10

20

30

30

#### 【0039】

図3Aに戻ると、グループ化コントロール330を押し下げるとすぐに、ウィンドウ203a～203dは、一緒にグループ化することができ、ここで次いでそのグループは、図3Bに示されるように、サムネイル画像340aのサイズまで一括して縮小することができる。グループ化コントロール330の押し下げの例示の例は、図3B中のシナリオ300bに示される。この例のシナリオ300bにおいては、ウィンドウ203a～203dのグループ化は、デスクトップスペース201中のサムネイル画像340aまで縮小されている。さらに、タスクバーボタン213a～213dが、タスクバー区域210から取り除かれている。この例には示されていないが、対応するタスクバーボタン213a～213dのうちの1つまたは複数が、タスクバー区域210内に留まることができるのを理解すべきである。

#### 【0040】

グループ化されたウィンドウは、グループの表現として一緒に管理することができ、例えば保存し、リフレッシュすることなどができる1組のウィンドウである。ウィンドウと一緒にグループ化することは、ユーザが、実行しなければならないウィンドウ管理の量を低減させ、同様にオーバーラップしたウィンドウから構成される作業表面のビジュアルクラッタ(visual clutter)も低減させる。さらに、ウィンドウのグループは、一緒に存続させ続けることができ、その結果、ウィンドウのグループは、後になってから取り出すことができる。任意のアプリケーションウィンドウは、アプリケーションウィンドウ203a～203dのサムネイル画像340aに追加し、あるいはサムネイル画像から後で取り除くことができる。

40

#### 【0041】

サムネイル画像340aから以前に最小化されたアプリケーションウィンドウを回復するために、ユーザは、ポインタ331を移動させ、アクティブにする、例えばサムネイル画像340aをクリックオン(click on)することができる。ユーザがデスクトップアイコン(desktop icon)340aをクリックオンするのに応じて、ウィンドウ203a～203dのグループ化は、グループ化コントロール330の実行に先立つそれらの状態へと回復させられる。この例においては、ウィンドウの外観は、図3A

50

において行ったとちょうど同じように見えるはずである。

【0042】

図3Cは、本発明の少なくとも一態様による、グループ化コントロール330の実行のすぐ後でのウィンドウ管理システムの他の例を示している。図3Bに示されるように、シナリオ300cは、アプリケーションウィンドウ203a～203dのサムネイル画像340aの代わりにあるタイプのアイコン340bとしてウィンドウ203a～203dのグループ化を示すことができる。代わりに、かつ／または追加して、タスクバーボタン340cは、ウィンドウ203a～203dのグループ化を表すために作成することができる。サムネイル画像340aと同様に、ウィンドウ203a～203dは、アイコン340bおよび／またはタスクバーボタン340cのアクティビ化により回復させることができる。そのような例においては、ウィンドウの外観は、この場合にもまた図3Aにおいて行ったとちょうど同じように見えるはずである。

10

【0043】

図4は、本発明の少なくとも一態様による、グループ化コントロール330の実行のすぐ後でのウィンドウ管理システムの別の例を示している。このシナリオ400においては、一部分、例えば4つの開かれたアプリケーションウィンドウ203a～203dのうちの3つは、1つのグループとして宣言されており、サムネイル画像440へと縮小されている。したがって、本発明の一実施形態においては、ユーザは、ウィンドウ203dをデスクトップスペース210中において開かれてアクティブのままにしながら、ウィンドウ203a～203cをグループ化するように選択することができる。多数の異なる方法を使用して、デスクトップスペース201中のすべての開かれたウィンドウよりも少ないウィンドウのグループ化を可能にすることができる。本発明は、1つの任意の方法だけに限定されるものではない。タスクバーに最小化された開かれたウィンドウは、210であり、それはグループコントロール330の作動によってウィンドウグループ内に含めることもできることが、当業者によって理解されるべきである。他の実施形態においては、システムは、たとえ別の開かれたウィンドウによってオーバーラップされていたとしても、最小化された任意のウィンドウをグループ化しないまま、開かれてアクティブなウィンドウをグループ化するだけの構成することができる。

20

【0044】

ひとたびウィンドウグループが宣言され、サムネイル画像340aへと最小化された後には、ユーザは、新しいアプリケーションウィンドウを開くこと、あるいは以前に最小化されたアプリケーションウィンドウを回復させることで再開することができる。ウィンドウグループが暗黙のうちに宣言された後に開かれたウィンドウは、第2のウィンドウグループのメンバと考えることができる。図5Aは、ウィンドウグループ340aが宣言され最小化されており、追加のアプリケーションウィンドウ503aおよび503bがその後に開かれているシナリオ500aを示している。これらのアプリケーションウィンドウは、第2のウィンドウグループのメンバである。ウィンドウ203a～203dと同様に、ウィンドウ503aおよび503bは、タスクバー区域210内に対応するタスクバーボタン513aおよび513bを有する。

30

【0045】

ひとたびウィンドウグループが宣言され、最小化された後には、ユーザは、サムネイル画像340a上でダブルクリックすることやコンテキストメニュー(contextual menu)から「開かれたグループ」を右クリックし選択することなど、標準の既存のオープン化メカニズムを介してグループを開くことにより、そのグループを取り出すことができる。図5Aに示されるように、ユーザは、ポインタ331をサムネイル画像グループ340aへと移動させる。グループを開くことは、フルスクリーンにアニメ化する(animating)、サムネイル画像グループ340aからのアプリケーションウィンドウ203a～203dをもたらすが、同時に、アプリケーションウィンドウ503a～503bは、図5Bに示されるサムネイル画像540へとアニメ化する、アプリケーションウィンドウの第2のグループ化としてシステムによって宣言される。第2のグループの

40

50

ウィンドウは、第1のグループのウィンドウを取り出したすぐ後に自動的に宣言することができることが、当業者によって理解されるべきである。代わりに、システムは、ユーザが第2のグループのウィンドウをアクティブに宣言する必要があるように構成することができる。

【0046】

図5Bは、サムネイル画像340aを経由して第1のウィンドウグループを開くシナリオ500bを示すものである。第2のグループは、その後にサムネイル画像540へと最小化されており、サムネイル画像340aとしての第1のグループの位置を置き換えていく。ひとたびウィンドウグループが取り出された後に、通常のウィンドウの対話が効力を有する。個別のアプリケーションウィンドウは、移動させ、アクティブにし、閉じることができる。ウィンドウ503aおよび503bは、別のグループのウィンドウがすでに存在していることにより、暗黙のうちにグループ化することができることを理解すべきである。したがって、ユーザは、グループコントロール330を2度目に作動してウィンドウ503aおよび503bをグループ化する必要はない。

10

【0047】

本発明の態様は、図に示されるウィンドウおよび／またはグループの数だけに限定されるものでないこと、ならびに任意数の異なるグループのウィンドウを作成することができるが、当業者によって理解されるべきである。さらに、本発明の様々な態様によれば、1つまたは複数のウィンドウを、ウィンドウグループに追加し、またはウィンドウグループから取り除くことができるることも理解すべきである。さらに、1つのウィンドウグループ中のアプリケーションウィンドウは、別のウィンドウグループに現れることもできる。さらにまた、本発明の態様により、ウィンドウグループはマシンの開始および／またはログアウト／ログインセッションにまたがって保存し、回復させることができるようになる。

20

【0048】

本発明の態様は、ウィンドウグループ上のウィンドウオペレーション (window operation) をサポートする。例えば、ウィンドウグループが、コンテンツを閉じ、保存し、またはリフレッシュするとすぐに作動されるコマンドは、そのグループ内のウィンドウのすべてがそのコンテンツを閉じ、保存し、またはリフレッシュするようにさせることができる。コマンドは、全体のグループに対してグローバルであっても、グループ内のウィンドウのサブセットに対して特有であっても、あるいは1つのアプリケーションウィンドウに対して特有であってもよい。ウィンドウアプリケーションが2つの別個のグループ内にあるときに、1つのグループ内のアプリケーションウィンドウ上で実行されるコマンドは、別のグループ内の同じアプリケーションウィンドウ上で同じオペレーションを実行し、あるいは実行しないように構成することができる。さらにまた、コマンドオプションは、新しいアプリケーションウィンドウが、グループに追加され、かつ／またはグループから取り除かれるときに、変化することができる。例えば、おののが、リフレッシュオペレーションを実行することができる2つのウィンドウを含むグループが、そのグループに追加されたそのコンテンツをリフレッシュする機能のない新しいウィンドウを有する場合、リフレッシュコマンドは、ウィンドウのグループを操作するときに、もはやユーザにとって使用可能なオプションではないこともある。グループとしてのウィンドウ上で使用するためのコマンドは、システムがどのように構成されているかに応じて、動的であっても、あるいは動的でなくてもよい。

30

【0049】

図6は、本発明の少なくとも一態様による、複数のアプリケーションウィンドウをグループ化するための方法の例示的な一実施例のフローチャートである。プロセスは、ステップ601から開始され、ここで複数のアプリケーションウィンドウが開かれる。ステップ603において、複数の開かれたアプリケーションウィンドウをグループへと宣言する入力が受け取られる。プロセスは、ステップ605へと進み、ここですべての開かれたウィンドウが、一緒にグループ化されるべきであるかどうかについての判定が行われる。すべ

40

50

ての開かれたウィンドウが一緒にグループ化されるべきである場合、プロセスは、ステップ607へと進み、ここで複数のウィンドウのすべてが、グループとして宣言され、次いでプロセスは、ステップ611へと進む。ステップ605の判定が、必ずしもすべての開かれたウィンドウが一緒にグループ化されるべきではないことを見出す場合には、プロセスは、ステップ609へと進み、ここで複数のアプリケーションウィンドウのうちの一部分が、グループとして宣言され、プロセスは、ステップ611へと進む。ステップ611において、アプリケーションウィンドウのグループの表現が表示される。ステップ611において表示されるウィンドウの表現と共に、プロセスは、終了することができる。ステップ611は、サムネイル画像340aなどのサムネイル画像、アイコン340bなどのアイコン、および/またはタスクバーボタン340cなどのタスクバーボタンであってもよいことが、理解されるべきである。

10

#### 【0050】

プロセスは、ステップ613へと続くこともでき、ここで、その表現中においてアプリケーションウィンドウのグループ上でオペレーションを実行する入力が受け取られているかどうかについての判定が行われる。1タイプのオペレーションは、アプリケーションウィンドウのコンテンツを表現内に保存するオペレーションを含むことができる。入力が受け取られていない場合、プロセスは終了する。ステップ613において入力が受け取られている場合には、プロセスは、ステップ615へと進み、ここでオペレーションは、グループ内のアプリケーションウィンドウ上で実行される。コンテンツを保存するそのようなケースでは、オペレーションがグローバルである場合、各アプリケーションウィンドウのコンテンツが保存される。代わりに、オペレーションが、アプリケーションウィンドウのサブセット、またはグループ内の単一のアプリケーションウィンドウを対象としている場合、オペレーションは、オペレーションが対象としていたサブセット上、または1つのウィンドウ上だけで実行される。

20

#### 【0051】

ステップ615から、プロセスは、ステップ617へと進み、ここでアプリケーションウィンドウをグループから取り出す入力が受け取られているかどうかについての判定が行われる。例えば、ユーザは、ユーザが、アプリケーションウィンドウをそれらの元の状態に回復させることを望むことを決定することができる。入力が受け取られていない場合、プロセスは終了する。ステップ617において入力が受け取られている場合には、プロセスは、ステップ619へと進み、ここでアプリケーションウィンドウは、デスクトップスペース中においてそれらの元の状態へと回復させられ、プロセスは終了する。

30

#### 【0052】

図7は、本発明の少なくとも一態様による、ウィンドウの複数のグループ化の間で切り換えるための方法の例示的な一実施例のフローチャートである。プロセスは、ステップ701から開始され、ここで複数のアプリケーションウィンドウが開かれる。ステップ703において、複数の開かれたアプリケーションウィンドウをグループへと宣言する入力が受け取られる。ステップ705において、アプリケーションウィンドウは、第1のグループとして宣言される。ステップ707へと進んで、第1のグループのアプリケーションウィンドウの表現が表示される。

40

#### 【0053】

ステップ709において、少なくとも1つの新しいアプリケーションウィンドウを開く入力が受け取られる。ステップ711において、少なくとも1つの新しいアプリケーションウィンドウが開かれる。少なくとも1つの新しいアプリケーションウィンドウは、第1のグループのアプリケーションウィンドウと同じアプリケーションウィンドウであってもよいことが理解されるべきである。ステップ711からは、ステップ713～721についての実線によって定義されるように、またステップ723～729についての破線によって定義されるように、プロセス中の2つの進行のうちの一方が行われることができる。第1の進行中に示されるように、ステップ711から、プロセスはステップ713へと進み、ここで少なくとも1つの新しいアプリケーションウィンドウをグループへと宣言する

50

入力が受け取られる。ステップ 715において、少なくとも1つの新しいアプリケーションウィンドウは、第2のグループとして宣言される。ステップ 717へと進んで、第2のグループの少なくとも1つのアプリケーションウィンドウの表現が表示される。ステップ 719において、選択された第1または第2のグループから1つ（または複数）のアプリケーションウィンドウを取り出す入力が受け取られているかどうかについての判定が行われる。受け取られていない場合、プロセスは、選択が行われ受け取られるまで待つ。ステップ 719において入力が受け取られている場合、プロセスは、ステップ 721へと進み、ここで選択されたグループの1つ（または複数）のアプリケーションウィンドウは、デスクトップスペース中においてそれらの元の状態へと回復させられ、プロセスは終了する。

10

#### 【0054】

諸ステップの間の破線によって定義される第2の進行中に示されるように、ステップ 711から、プロセスは、ステップ 723へと進み、ここで第1のグループからアプリケーションウィンドウを取り出す入力が受け取られているかどうかについての判定が行われる。受け取られていない場合、プロセスは、入力が受け取られるまで待つ。ステップ 723において入力が受け取られている場合には、プロセスは、ステップ 725へと進み、ここでシステムは、デスクトップスペース中の少なくとも1つの新しいアプリケーションウィンドウを第2のグループへと自動的に宣言する。ユーザが、第1のグループのアプリケーションウィンドウへと戻ることを決定しているので、システムは、暗黙のうちに新しい1つ（または複数）のアプリケーションウィンドウを新しいグループへとグループ化する。プロセスは、ステップ 727へと進み、ここで第2のグループのアプリケーションウィンドウの表現が表示される。ステップ 729において、第1のグループのアプリケーションウィンドウは、デスクトップスペース中においてそれらの元の状態へと回復させられ、プロセスは終了する。ユーザは、上記のステップのうちの1つまたは複数のステップに従うことにより、必要に応じてアプリケーションウィンドウのグループの間で切り換えることができる事が、当業者によって理解されるべきである。

20

#### 【0055】

別の実施形態は、直接にグループへの、ウィンドウの宣言およびグループ化をプログラムにより制御している。例えば、マイクロソフトオフィス（Microsoft Office）など1つまたは複数のアプリケーションは、いくつかのドキュメントから構成されるプロジェクトの概念を共有することができる。これらのドキュメントは、プロジェクト中の新しいドキュメントがユーザによって開かれるときに、同時に1つのグループ中へと開き、あるいはグループに自動的に追加することができる。

30

#### 【0056】

アプリケーションプログラミングインターフェース（API）に関しては、本発明の様々な態様をAPIによって実行することができる。例えば、公開APIは、オペレーティングシステムが本発明の様々な特徴を実現することができるようオペレーティングシステムとインターフェースすることができる。一実施形態においては、グループ化されたウィンドウのデータ表現を処理するためのソフトウェアアーキテクチャは、第1の開かれたウィンドウと第2の開かれたウィンドウとを1つのグループへとグループ化するように構成されたコンポーネントと、そのコンポーネントにアクセスするアプリケーションプログラムインターフェースとを含むことができる。APIは、2つのウィンドウをグループ化し、オペレーションを実行する必要な1つ（または複数）のファンクションにアクセスし、次いでそれらの結果をオペレーティングシステムに返送する要求を受け取ることができる。オペレーティングシステムは、APIから供給されるデータを使用して、本発明の様々な特徴を実施することができる。

40

#### 【0057】

アプリケーションプログラムにウィンドウグループをうまく利用させるAPIは、プログラムによりウィンドウをグループ化すること、および非グループ化すること、ならびにコマンドがグループ上で実装されるときに、アプリケーションプログラムに、保存する、

50

印刷する、コンテンツをリフレッシュする、バージョンを比較する、ソースコードをコンパイルするなど、どのようにしてウィンドウの状態が処理されるかを指定させることを含んでいる。この状態情報により、グループ中のアプリケーションプログラムのウィンドウは、グループが取り出され、対応するアプリケーションウィンドウが回復させられるときにデフォルトの代わりに、ある適切な状態に回復させることができるようになる。例えば、あるグループに宣言された最小化されたウィンドウは、元の状態に回復させることができる。

【0058】

本発明の様々な態様を実施する、本明細書中で説明される例示のシステムおよび方法が示されているが、本発明は、これらの実施形態だけに限定されるものではないことが当業者には理解されよう。当業者によって特に前述の教示の観点から修正を行うことができる。例えば、前述の実施形態の各要素は、単独で、または組み合わせて、あるいは他の実施形態の要素と部分的に組み合わせて利用することができる。本発明の趣旨および範囲を逸脱することなく、修正を行うことができるることも認識され理解されよう。したがって本明細書は、本発明に対して限定的である代わりに例示的なものとみなされるべきである。

10

【図面の簡単な説明】

【0059】

【図1A】本発明のある種の態様を実装することができる汎用デジタルコンピューティング環境の概略図である。

20

【図1B】本発明の1つまたは複数の態様をサポートする汎用コンピュータ環境を示す図である。

【図1C】本発明の1つまたは複数の態様をサポートする汎用コンピュータ環境を示す図である。

【図1D】本発明の1つまたは複数の態様をサポートする汎用コンピュータ環境を示す図である。

【図1E】本発明の1つまたは複数の態様をサポートする汎用コンピュータ環境を示す図である。

【図1F】本発明の1つまたは複数の態様をサポートする汎用コンピュータ環境を示す図である。

30

【図1G】本発明の1つまたは複数の態様をサポートする汎用コンピュータ環境を示す図である。

【図1H】本発明の1つまたは複数の態様をサポートする汎用コンピュータ環境を示す図である。

【図1I】本発明の1つまたは複数の態様をサポートする汎用コンピュータ環境を示す図である。

【図1J】本発明の1つまたは複数の態様をサポートする汎用コンピュータ環境を示す図である。

【図1K】本発明の1つまたは複数の態様をサポートする汎用コンピュータ環境を示す図である。

40

【図1L】本発明の1つまたは複数の態様をサポートする汎用コンピュータ環境を示す図である。

【図1M】本発明の1つまたは複数の態様をサポートする汎用コンピュータ環境を示す図である。

【図2】従来のウィンドウ管理シナリオを示す図である。

【図3A】本発明の少なくとも一態様による、グループ化コントロールを含むウィンドウ管理システムの一例を示す図である。

【図3B】本発明の少なくとも一態様による、グループ化コントロールの実行直後のウィンドウ管理システムの一例を示す図である。

【図3C】本発明の少なくとも一態様による、グループ化コントロールの実行直後のウィンドウ管理システムの他の例を示す図である。

50

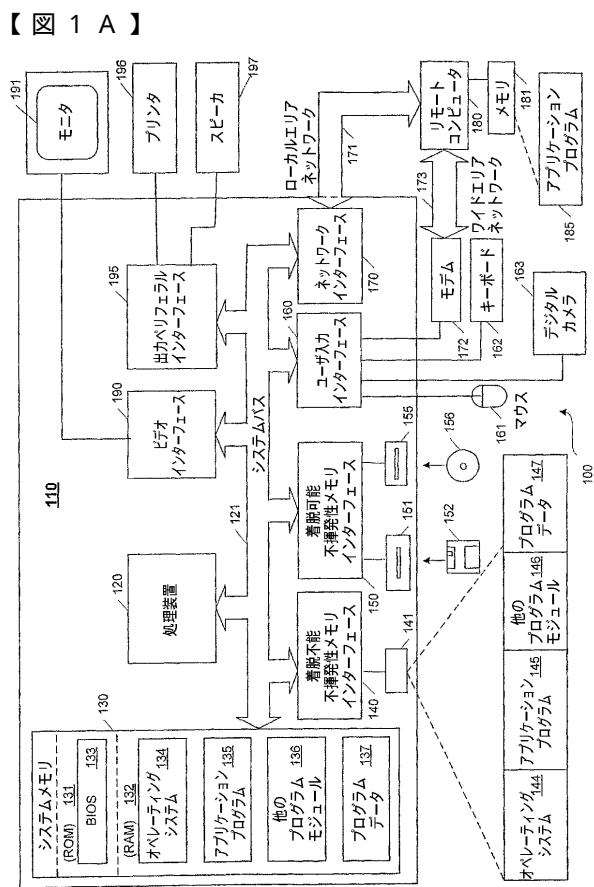
【図4】本発明の少なくとも一態様による、グループ化コントロールの実行直後のウィンドウ管理システムの別の例を示す図である。

【図5A】諸ウィンドウの複数のグループ化が、本発明の少なくとも一態様に従って示されるウィンドウ管理システムの別の例を示す図である。

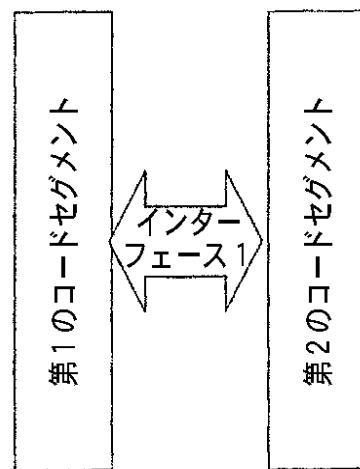
【図5B】本発明の少なくとも一態様による、ウィンドウの第1のグループ化の再オープン化直後の図5Aのウィンドウ管理システムの一例を示す図である。

【図6】本発明の少なくとも一態様による、2つのウィンドウをグループ化するための方法の例示的な一実施例のフローチャートである。

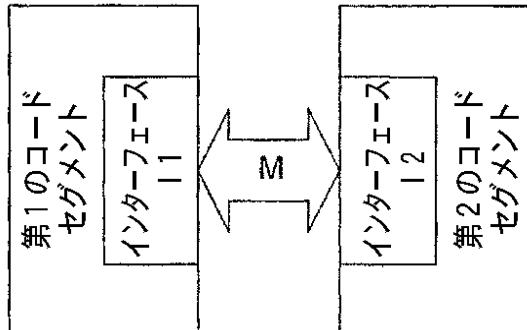
【図7】本発明の少なくとも一態様による、ウィンドウの複数のグループ化の間で切り換えるための方法の例示的な一実施例のフローチャートである。



【図1A】

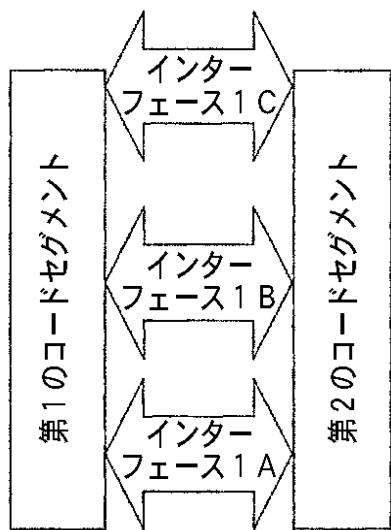


【図1B】

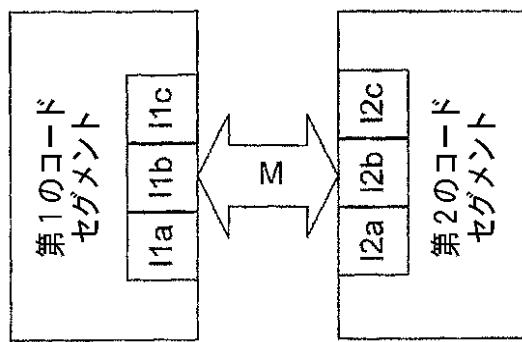


【図1C】

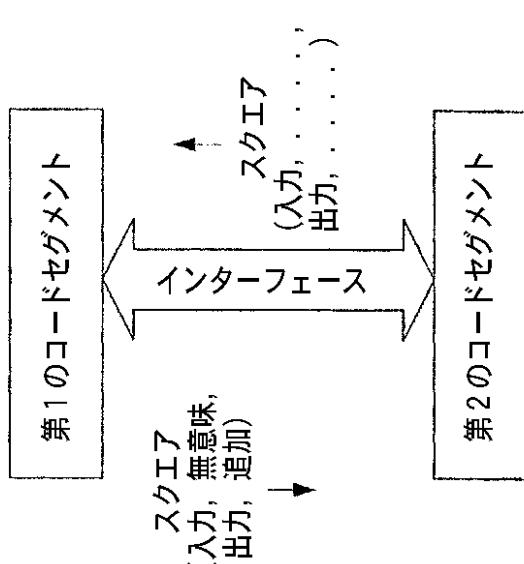
【図 1 D】



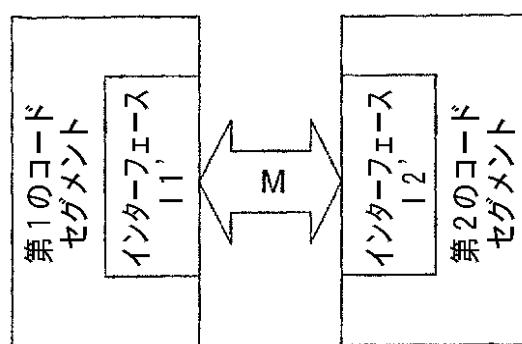
【図 1 E】



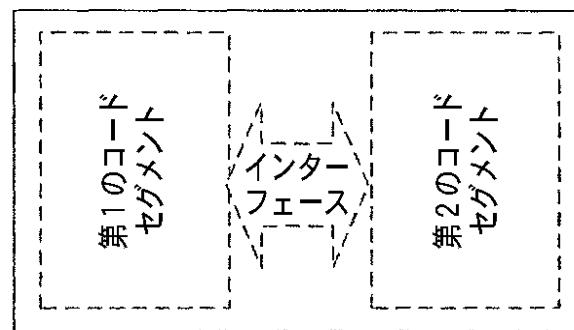
【図 1 F】



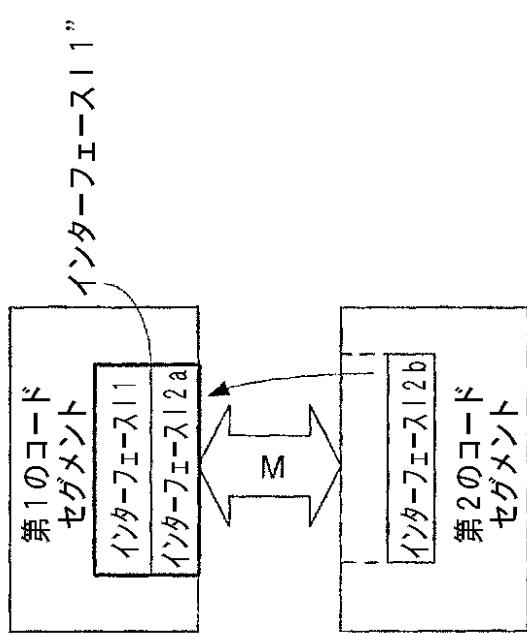
【図 1 G】



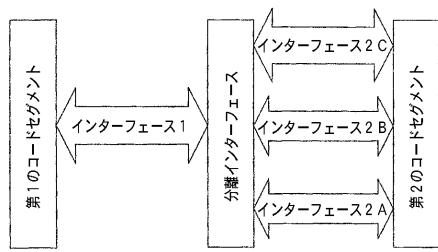
【図 1 H】



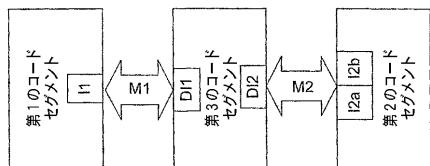
【図 1 I】



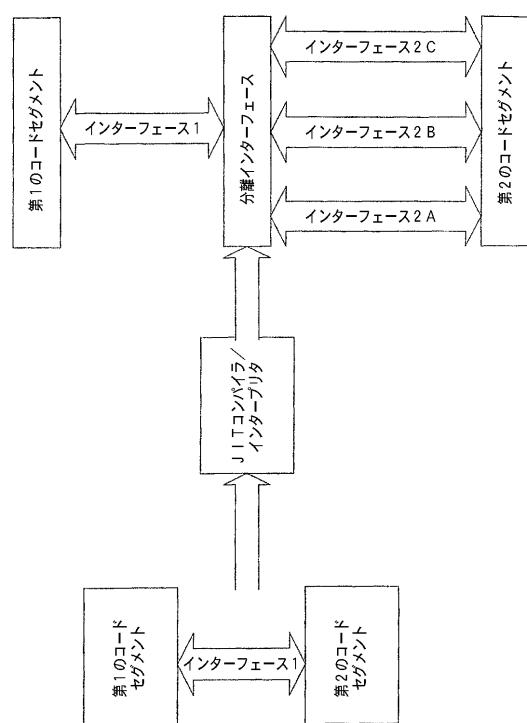
【図 1 J】



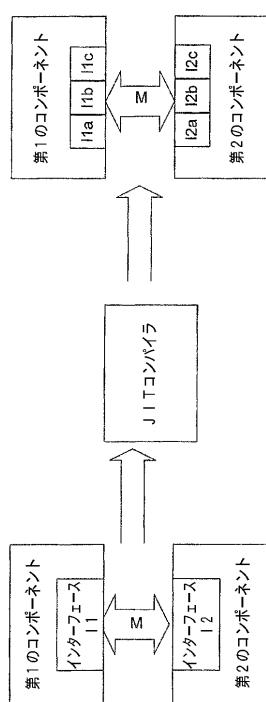
【図 1 K】



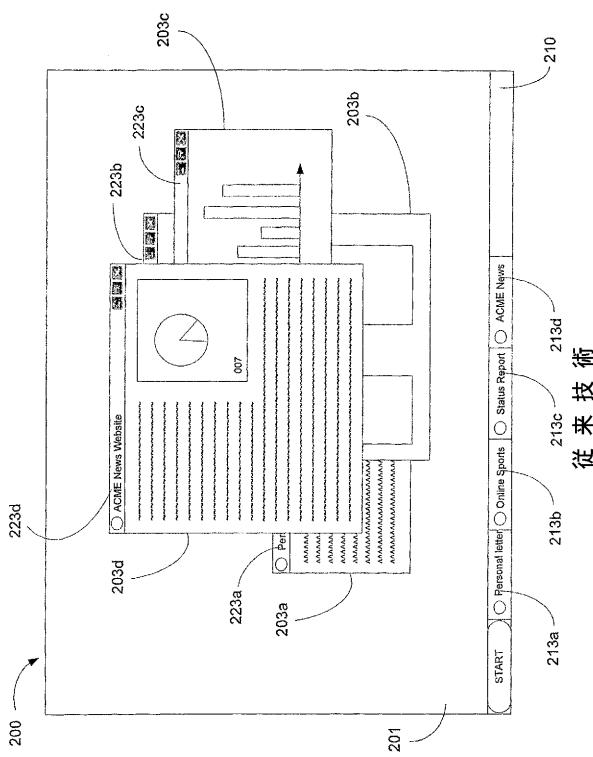
【図 1 L】



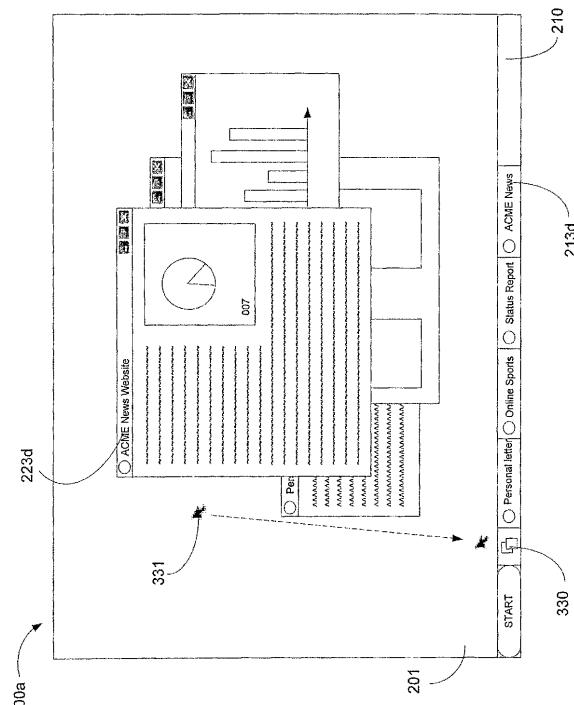
【図 1 M】



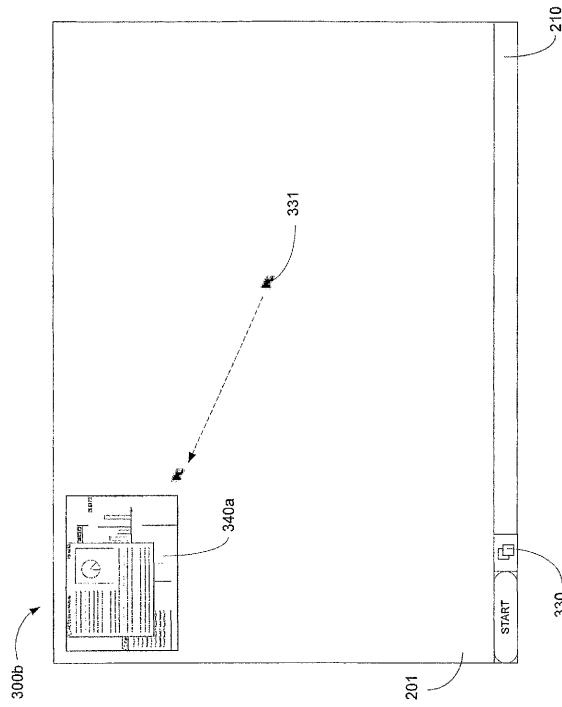
【 図 2 】



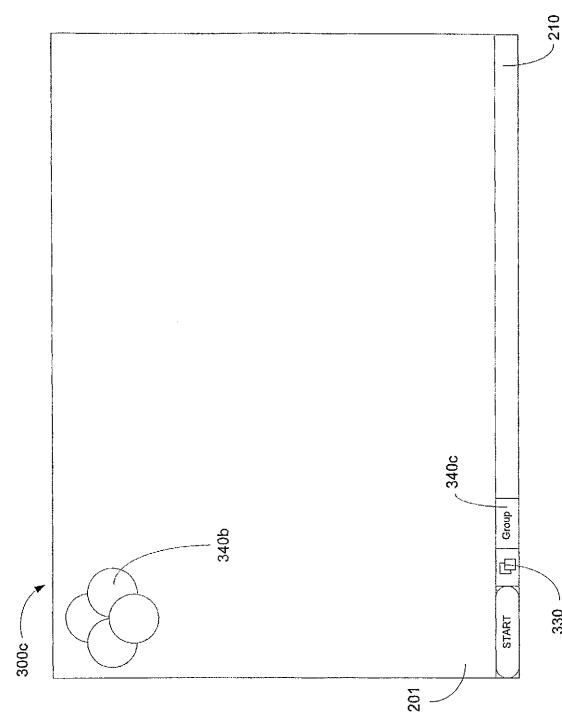
【 図 3 A 】



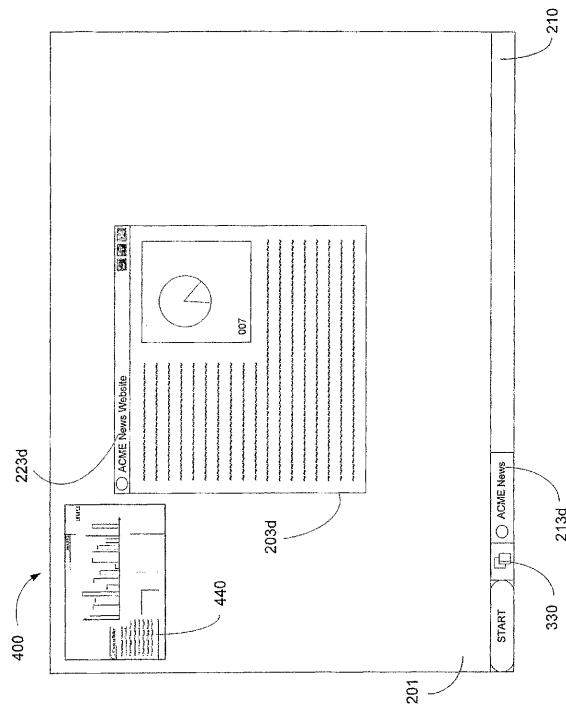
【 図 3 B 】



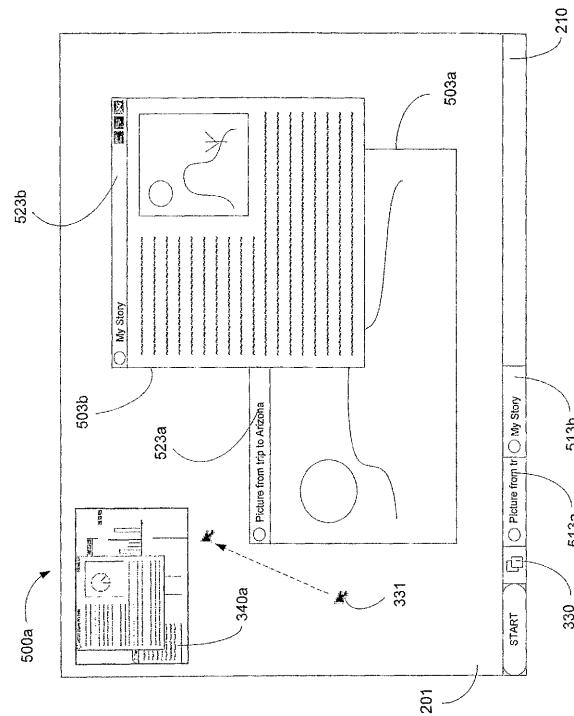
【図3C】



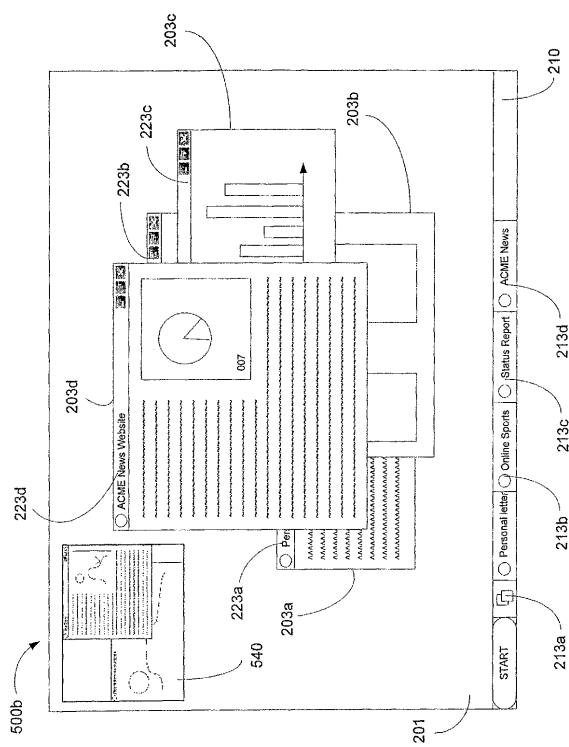
【図4】



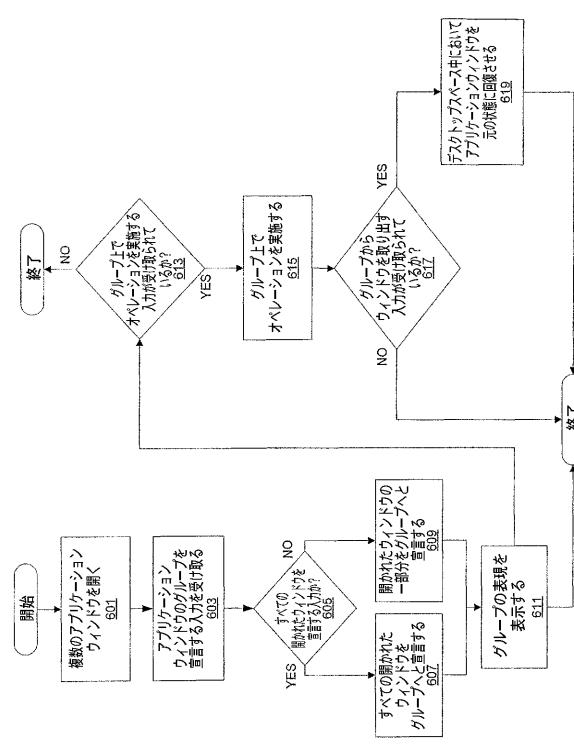
【図5 A】



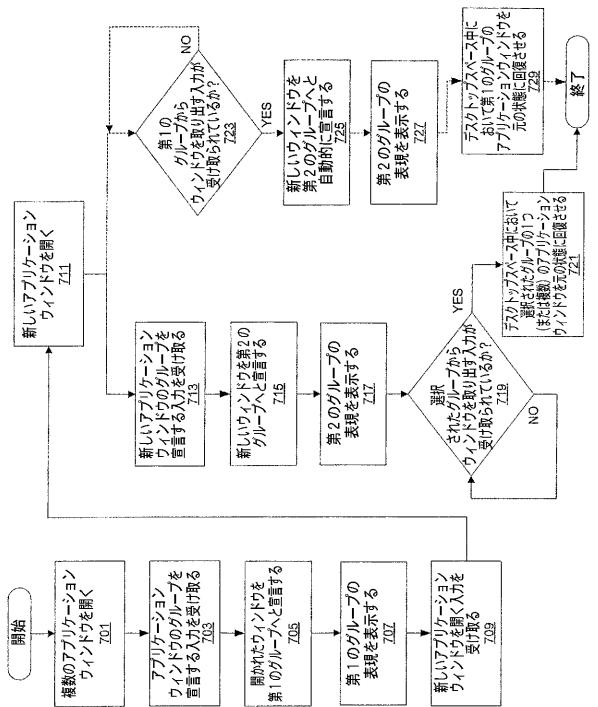
【図5 B】



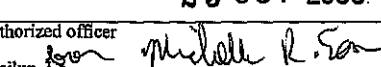
【図6】



【圖 7】



## 【国際調査報告】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US05/30312
<b>A. CLASSIFICATION OF SUBJECT MATTER</b> IPC: G06F 3/00( 2006.01)		
USPC: 715/781 According to International Patent Classification (IPC) or to both national classification and IPC		
<b>B. FIELDS SEARCHED</b> Minimum documentation searched (classification system followed by classification symbols) U.S. : 715/781		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched EAST		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X, E — Y, E Y	US 2005/0223334 A1 (GUIDO et al) 06 October 2005 (06.10.2006), see entire document.  US 2005/0210410 A1 (OHWA et al.) 22 September 2005 (22.09.2005), see entire documents.	1-8, 10-18 ————— 9, 19, and 20 9, 19 and 20
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/>		See patent family annex.
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed		
Date of the actual completion of the international search 18 August 2006 (18.08.2006)		Date of mailing of the international search report 23 OCT 2006
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US Commissioner for Patents P.O. Box 1450 Alexandria, Virginia 22313-1450 Facsimile No. (571) 273-3201		Authorized officer  Weilun LO Telephone No. 272-2100

---

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LS,MW,MZ,NA,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HU,IE,IS,IT,LT,LU,LV,MC,NL,PL,PT,RO,SE,SI,SK,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AT,AU,AZ,BA,BB,BG,BR,BW,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DE,DK,DM,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,HR,HU,ID,IL,IN,IS,JP,KE,KG,KM,KP,KR,KZ,LC,LK,LR,L,S,LT,LU,LV,MA,MD,MG,MK,MN,MW,MX,MZ,NA,NG,NI,NO,NZ,OM,PG,PH,PL,PT,RO,RU,SC,SD,SE,SG,SK,SL,SM,SY,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,YU,ZA,ZM,ZW

(72)発明者 ドナルド ジェイ・リンゼイ

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 コーポレーション内

(72)発明者 マーク アール・リガメリ

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 コーポレーション内

F ターム(参考) 5B069 BA00 CA13

5C082 AA01 AA22 AA24 CA56 CA62 CA81 DA86 MM09  
5E501 AA02 BA03 CA02 CA08 CB02 CB04 CB06 CB09 CB10 CB11  
CB12 CB13 CB15 EA05 EA06 EA07 EA10 FA06 FA13 FA14  
FA15 FA48 FB04 FB22