

(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2007-511851

(P2007-511851A)

(43) 公表日 平成19年5月10日(2007.5.10)

(51) Int. Cl.	F I	テーマコード (参考)
<b>G06F 3/048 (2006.01)</b>	G06F 3/048 655B	5B042
<b>G06F 11/32 (2006.01)</b>	G06F 11/32 B	5E501

審査請求 未請求 予備審査請求 未請求 (全 34 頁)

(21) 出願番号 特願2006-541338 (P2006-541338)  
 (86) (22) 出願日 平成16年11月17日 (2004.11.17)  
 (85) 翻訳文提出日 平成18年7月3日 (2006.7.3)  
 (86) 国際出願番号 PCT/US2004/038511  
 (87) 国際公開番号 W02005/050393  
 (87) 国際公開日 平成17年6月2日 (2005.6.2)  
 (31) 優先権主張番号 10/717, 221  
 (32) 優先日 平成15年11月18日 (2003.11.18)  
 (33) 優先権主張国 米国 (US)

(71) 出願人 505198994  
 ラズロ システムズ インコーポレイテッド  
 アメリカ合衆国 カリフォルニア州 94  
 403 サン マテオ キャンパス ドラ  
 イブ 2600 スウィート 200  
 (74) 代理人 110000110  
 特許業務法人快友国際特許事務所  
 (72) 発明者 アンドレア、ピーター  
 アメリカ合衆国、94062、カリフォル  
 ニア州、オークランド、イースト 31番  
 ストリート 1546

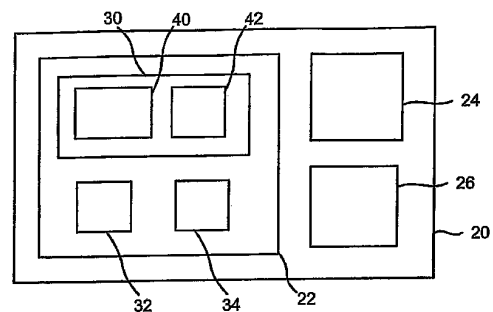
最終頁に続く

(54) 【発明の名称】 複数のアプリケーションを表示するためのユーザ・インターフェース

(57) 【要約】

複数のアプリケーションが表示される。第1のアプリケーションの表示の変更要求が受信される。第1のアプリケーションの表示との競合を避けるために、第1のアプリケーションの表示の変更に応じて、1つまたは複数の他のアプリケーションの表示の変更方法が決定される。第1のアプリケーションの表示が変更される。1つまたは複数の他のアプリケーションの1つまたは複数の表示は、第1のアプリケーションの表示との競合を避けるために、第1のアプリケーションの表示の変更に応じて自動的に変更される。

【選択図】 図8



**【特許請求の範囲】****【請求項 1】**

少なくとも第 1 のアプリケーションと第 2 のアプリケーションを含む 1 組のアプリケーション群を表示する工程と；

前記第 1 のアプリケーションの表示を変更する工程と；

前記第 1 のアプリケーションの前記表示の前記変更に応じて、前記第 2 のアプリケーションの表示を自動的に変更する工程を有するアプリケーションの管理方法。

**【請求項 2】**

前記第 1 のアプリケーションの前記表示を変更する前記工程は、前記第 1 のアプリケーションの前記表示のサイズを変更する工程を有し；

前記第 2 のアプリケーションの前記表示を変更する前記工程は、前記第 2 のアプリケーションの前記表示の位置またはサイズを変更する工程を有することを特徴とする請求項 1 に記載の方法。

10

**【請求項 3】**

前記第 1 のアプリケーションの前記表示を変更する前記工程は、前記第 1 のアプリケーションの前記表示の位置を変更する工程を有し；

前記第 2 のアプリケーションの前記表示を変更する前記工程は、前記第 2 のアプリケーションの前記表示の位置またはサイズを変更する工程を有することを特徴とする請求項 1 に記載の方法。

**【請求項 4】**

前記各アプリケーションは、サイズが固定された表示領域内に表示されることを特徴とする請求項 1 に記載の方法。

20

**【請求項 5】**

前記アプリケーション群は第 3 のアプリケーションを有し；

前記第 1 のアプリケーションおよび前記第 2 のアプリケーションは前記第 3 のアプリケーション内に表示され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の表示を変更する前記工程は、ユーザが対象の一部をドラッグすることに対応して実行されることを特徴とする請求項 1 に記載の方法。

**【請求項 7】**

前記第 2 のアプリケーションの前記表示を自動的に変更する前記工程は、

前記第 1 のアプリケーションの前記表示の変更による前記第 2 のアプリケーションへの影響を決定する工程と；

前記影響が許容されるか否かを判定する工程と；

前記影響が許容される場合に、前記第 2 のアプリケーションの前記表示を変更する工程を含むことを特徴とする請求項 1 に記載の方法。

30

**【請求項 8】**

前記第 1 のアプリケーションの表示を変更する前記工程は、前記表示の前記変更が許容されるかどうかを決定する工程を有することを特徴とする請求項 1 に記載の方法。

**【請求項 9】**

前記 1 組のアプリケーション群用のユーザ・インターフェース・アイテムを生成する工程と；

親アプリケーション用のレイアウト・オブジェクトを生成する工程と；

対応するアプリケーションが変更される場合に、1 つまたは複数の機能呼び出すためのポインタを各ユーザ・インターフェース・アイテムに与える工程と；

前記ユーザ・インターフェース・アイテム用の表示を提供する工程と；

前記対応するアプリケーションを実行する工程をさらに有する請求項 1 に記載の方法。

40

**【請求項 10】**

前記第 2 のアプリケーションの前記表示を自動的に変更する前記工程は、

前記第 1 のアプリケーションの前記表示の縦方向の配置が変化したことを判定する工程

50

と；

前記第 1 のアプリケーションとの縦方向の関係において、兄弟アプリケーションの縦方向の配置を前記第 1 のアプリケーションに合わせて調整する工程を有することを特徴とする請求項 1 に記載の方法。

【請求項 1 1】

前記第 2 のアプリケーションの前記表示を自動的に変更する前記工程は、

前記第 1 のアプリケーションの前記表示の横方向の配置が変化したことを判定する工程と；

前記第 1 のアプリケーションとの横方向の関係において、兄弟アプリケーションの横方向の配置を前記第 1 のアプリケーションに合わせて調整する工程を有することを特徴とする請求項 1 に記載の方法。

10

【請求項 1 2】

前記第 2 のアプリケーションの前記表示を自動的に変更する前記工程は、

前記第 1 のアプリケーションの前記表示の幅が変化したことを判定する工程と；

前記第 1 のアプリケーションの前記表示を含む第 1 の表示列の列幅を変更する工程と；

前記第 1 の表示列の前記列幅に基づいて前記第 1 の表示列内の表示の幅を変更する工程を有することを特徴とする請求項 1 に記載の方法。

【請求項 1 3】

前記第 2 のアプリケーションの前記表示を自動的に変更する前記工程は、

前記第 1 のアプリケーションの前記表示の幅が変化したことを判定する工程と；

前記第 1 のアプリケーションの前記表示を含む第 1 の表示列の列幅を変更する工程と；

前記第 1 の表示列の前記列幅に基づいて前記第 1 の表示列内の表示の幅を変更する工程と；

20

前記第 1 の表示列に隣接する第 2 の表示列の列幅を変更する工程と；

前記第 2 の表示列の前記列幅に基づいて前記第 2 の表示列内の表示の幅を変更する工程を有することを特徴とする請求項 1 に記載の方法。

【請求項 1 4】

前記第 2 のアプリケーションの前記表示を自動的に変更する前記工程は、

前記第 1 のアプリケーションの前記表示の縦方向のサイズが変化したことを判定する工程と；

30

残りの領域を兄弟アプリケーション間で分配する工程と；

前記分配工程に基づいて兄弟アプリケーションのサイズを調整する工程を有することを特徴とする請求項 1 に記載の方法。

【請求項 1 5】

前記第 1 のアプリケーションはグループのメンバであり、前記グループのメンバは前記グループ内で所定の関係を有し；

前記第 2 のアプリケーションの前記表示を自動的に変更する前記工程は、前記所定の関係を維持するように実行されることを特徴とする請求項 1 に記載の方法。

【請求項 1 6】

前記第 1 のアプリケーションの表示を変更するとともに前記第 2 のアプリケーションの前記表示を自動的に変更する前記工程は、

40

前記第 1 のアプリケーションの選択を受信する工程と；

前記第 1 のアプリケーションの前記表示を拡大し、前記第 1 のアプリケーションの前記表示を主要な場所内に配置する工程と；

所定の関係に基づいて既存の領域内に収まるように、前記第 2 のアプリケーションを含む前記第 1 のアプリケーションの兄弟アプリケーションの表示をサイズ変更するとともに位置あわせをすることを有することを特徴とする請求項 1 に記載の方法。

【請求項 1 7】

前記第 1 のアプリケーションの表示を変更し、前記第 2 のアプリケーションの前記表示を自動的に変更する前記工程は、

50

前記第 1 のアプリケーションの前記表示が 1 つまたは複数の兄弟アプリケーションと重なることを判定する工程と；

前記 1 つまたは複数の兄弟アプリケーションの表示がサイズ変更可能であることを判定する工程と；

最小限の間隔を維持するように前記 1 つまたは複数の兄弟アプリケーションの前記表示を自動的にサイズ変更する工程を有することを特徴とする請求項 1 に記載の方法。

【請求項 18】

前記第 1 のアプリケーションの表示を変更し、前記第 2 のアプリケーションの前記表示を自動的に変更する前記工程は、

前記第 1 のアプリケーションの前記表示が 1 つまたは複数の兄弟アプリケーションと重なることを判定する工程と； 10

前記 1 つまたは複数の兄弟アプリケーションのすべての表示をサイズ変更することはできないことを判定する工程と；

重なる兄弟アプリケーションのすべての表示をサイズ変更できるように、前記第 1 のアプリケーションの前記表示の前記サイズ変更を制限する工程と、を有することを特徴とする請求項 1 に記載の方法。

【請求項 19】

前記第 1 のアプリケーションの表示を変更し、前記第 2 のアプリケーションの前記表示を自動的に変更する前記工程は、

前記第 1 のアプリケーションの前記表示が 1 つまたは複数の兄弟アプリケーションと重なることを判定する工程と； 20

前記 1 つまたは複数の兄弟アプリケーションのすべての表示を十分にサイズ変更することはできないことを判定する工程と；

十分にサイズ変更することができない表示を限定的にサイズ変更する工程と；

十分にサイズ変更することができない表示に隣接するアプリケーションをサイズ変更する工程を有することを特徴とする請求項 1 に記載の方法。

【請求項 20】

前記第 1 のアプリケーションの表示を変更し、前記第 2 のアプリケーションの前記表示を自動的に変更する前記工程は、

前記第 1 のアプリケーションの前記表示を削除する工程と； 30

比例したサイズに基づいて、兄弟アプリケーションの残りの表示を前記第 1 のアプリケーションに適合させる工程を有することを特徴とする請求項 1 に記載の方法。

【請求項 21】

前記第 1 のアプリケーションの表示を変更し、前記第 2 のアプリケーションの前記表示を自動的に変更する前記工程は、

前記第 1 のアプリケーションの前記表示が既存の空き領域内に収まるか否かを判定する工程と；

前記第 1 のアプリケーションに対し重なる可能性のある隣接するアプリケーションを移動できるか否かを判定する工程と；

前記第 1 のアプリケーションに対し重なる可能性のある隣接するアプリケーションをサイズ変更することができるか否かを判定する工程を有することを特徴とする請求項 1 に記載の方法。 40

【請求項 22】

前記第 1 のアプリケーションの表示を変更し、前記第 2 のアプリケーションの前記表示を自動的に変更する前記工程は、

前記第 1 のアプリケーションの前記表示が既存の空き領域内に収まるか否かを判定する工程と；

前記第 1 のアプリケーションに対し重なる可能性のある隣接するアプリケーションを移動できるか否かを判定する工程と；

前記第 2 のアプリケーションのための前記表示を含む前記重なる可能性のある隣接する 50

アプリケーションを自動的に移動する工程と；

前記第1のアプリケーションのための前記表示を追加する工程を有することを特徴とする請求項1に記載の方法。

【請求項23】

前記第1のアプリケーションの表示を変更し、前記第2のアプリケーションの前記表示を自動的に変更する前記工程は、

前記第1のアプリケーションの前記表示が既存の空き領域内に収まるかどうかを判定する工程と；

前記第1のアプリケーションに対し重なる可能性のある隣接するアプリケーションをサイズ変更することができるかどうかを判定する工程と；

前記第2のアプリケーションのための前記表示を含む前記重なる可能性のある隣接するアプリケーションをサイズ変更する工程と；

前記第1のアプリケーションのための前記表示を追加する工程を有することを特徴とする請求項1に記載の方法。

【請求項24】

前記第2のアプリケーションの前記表示を自動的に変更する前記工程は、前記第1のアプリケーションの前記表示が前記第2のアプリケーションの前記表示と重なるのを避ける工程を有することを特徴とする請求項1に記載の方法。

【請求項25】

前記第2のアプリケーションの前記表示を自動的に変更する前記工程は、前記第2のアプリケーションの前記表示が所定のサイズ閾値未満であることを判定し、前記第2のアプリケーションの前記表示をアイコンに変換する工程を有することを特徴とする請求項1に記載の方法。

【請求項26】

プロセッサで読取り可能な記憶装置上で具現化されるプロセッサで読取り可能なコードを有する1つまたは複数のプロセッサで読取り可能な記憶装置であって、

プロセッサで読取り可能な前記コードは、

少なくとも第1のアプリケーションと第2のアプリケーションを含む1組のアプリケーション群を表示領域に表示する工程と；

前記第1のアプリケーションのサイズまたは位置を変更する工程と；

前記第1のアプリケーションの表示が前記第2のアプリケーションの表示と競合しないように、前記第1のアプリケーションのサイズまたは位置の変更に応じて、前記第2のアプリケーションのサイズまたは位置を自動的に変更する工程を有する方法を1つまたは複数のプロセッサが実行するようにプログラムされていることを特徴とする記憶装置。

【請求項27】

前記第2のアプリケーションのサイズまたは位置を自動的に変更する前記工程は、

前記第1のアプリケーションの幅が変化したことを判定する工程と；

前記第1のアプリケーションを含む第1の表示列の列幅を変更する工程と；

前記第1の表示列の前記列幅に基づいて、前記第1の表示列内の他のアプリケーションの幅を変更する工程を有することを特徴とする請求項26に記載の1つまたは複数のプロセッサで読取り可能な記憶装置。

【請求項28】

前記第1のアプリケーションと前記第2のアプリケーションは、サイズが固定された表示領域内で表示されることを特徴とする請求項26に記載の1つまたは複数のプロセッサで読取り可能な記憶装置。

【請求項29】

前記第2のアプリケーションのサイズまたは位置を自動的に変更する前記工程は、

前記第1のアプリケーションの幅が変化したことを判定する工程と；

前記第1のアプリケーションを含む第1の表示列の列幅を変更する工程と；

前記第1の表示列の前記列幅に基づいて、前記第1の表示列内の1つまたは複数の他の

10

20

30

40

50

アプリケーションの幅を変更する工程と；

前記第 1 の表示列に隣接する第 2 の表示列の列幅を変更する工程と；

前記第 2 の表示列の前記列幅に基づいて、前記第 2 の表示列内の 1 つまたは複数の他のアプリケーションの幅を変更する工程を有することを特徴とする請求項 2 6 に記載の 1 つまたは複数のプロセッサで読取り可能な記憶装置。

【請求項 3 0】

前記第 1 のアプリケーションのサイズまたは位置を変更し、前記第 2 のアプリケーションのサイズまたは位置を自動的に変更する前記工程は、

前記第 1 のアプリケーションが 1 つまたは複数の兄弟アプリケーションと重なることを判定する工程と；

前記 1 つまたは複数の兄弟アプリケーションの表示をサイズ変更することができることを判定する工程と；

最小限の間隔を維持するように前記 1 つまたは複数の兄弟アプリケーションを自動的にサイズ変更する工程を有することを特徴とする請求項 2 6 に記載の 1 つまたは複数のプロセッサで読取り可能な記憶装置。

【請求項 3 1】

前記第 1 のアプリケーションの表示が前記第 2 のアプリケーションの表示と重なると、前記第 1 のアプリケーションの表示が前記第 2 のアプリケーションの表示と競合することを特徴とする請求項 2 6 に記載の 1 つまたは複数のプロセッサで読取り可能な記憶装置。

【請求項 3 2】

前記第 1 のアプリケーションの表示が許容されない状態で前記第 2 のアプリケーションの表示と重なると、前記第 1 のアプリケーションの表示が前記第 2 のアプリケーションの表示と競合することを特徴とする請求項 2 6 に記載の 1 つまたは複数のプロセッサで読取り可能な記憶装置。

【請求項 3 3】

表示領域に表示された第 1 のアプリケーションの表示を変更する要求を受信する工程と；

前記第 1 のアプリケーションの前記表示との競合を避けるために、前記第 1 のアプリケーションの前記表示の変更に応じて、1 つまたは複数の他のアプリケーションのための 1 つまたは複数の表示がどのように変化すべきかを決定する工程と；

前記第 1 のアプリケーションの前記表示を変更する工程と；

前記第 1 のアプリケーションの前記表示との競合を避けるために、前記第 1 のアプリケーションの前記表示の変更に応じて、変化すべき 1 つまたは複数の他のアプリケーションのための 1 つまたは複数の表示を自動的に変更する工程を有するアプリケーション群の管理方法。

【請求項 3 4】

要求を受信する前記工程は、ユーザがウィンドウの一部をドラッグする工程を有することを特徴とする請求項 3 3 に記載の方法。

【請求項 3 5】

避けられる前記競合は、前記第 1 のアプリケーションの前記表示が、前記変化すべき 1 つまたは複数の他のアプリケーションのための前記 1 つまたは複数の表示と重なるのを避けることを有することを特徴とする請求項 3 3 に記載の方法。

【請求項 3 6】

避けられる前記競合は、前記第 1 のアプリケーションの前記表示が、前記変化すべき 1 つまたは複数の他のアプリケーションのための前記 1 つまたは複数の表示に、許容されない状態で重なるのを避けることを有することを特徴とする請求項 3 3 に記載の方法。

【請求項 3 7】

前記決定する工程は、前記第 1 のアプリケーションの前記表示が前記 1 つまたは複数の他のアプリケーションのための前記 1 つまたは複数の表示と重なることを避けるために、前記 1 つまたは複数の他のアプリケーションのための前記 1 つまたは複数の表示をどのよ

10

20

30

40

50

うに調整するかを決定する工程を有することを特徴とする請求項 33 に記載の方法。

【請求項 38】

前記第 1 のアプリケーションの前記表示に対する前記変更が許可されるか否かを判定する工程と；

前記第 1 のアプリケーションの前記表示に対する前記変更が許可されない場合に、前記第 1 のアプリケーションの前記表示に対する前記変更を拒否する工程をさらに有する請求項 33 に記載の方法。

【請求項 39】

前記 1 つまたは複数の他のアプリケーションは第 2 のアプリケーションを含み；

前記 1 つまたは複数の他のアプリケーションのための前記 1 つまたは複数の表示を変更する前記工程は、前記第 1 のアプリケーションの前記表示との重なりを避けるために前記第 2 のアプリケーションの表示をどのように変更するかを決定する工程と、前記第 2 のアプリケーションの前記表示に対する前記変更が許可されないことを判定する工程と、前記第 1 のアプリケーションの前記表示に対する前記変更を拒否する工程を有することを特徴とする請求項 33 に記載の方法。

【請求項 40】

前記 1 つまたは複数の他のアプリケーションは第 2 のアプリケーションを含み；

前記 1 つまたは複数の他のアプリケーションのための前記 1 つまたは複数の表示を変更する前記工程は、前記第 1 のアプリケーションの前記表示との重なりを避けるために前記第 2 のアプリケーションの表示をどのように変更するかを決定する工程と、前記第 2 のアプリケーションの前記表示に対する前記変更が前記第 2 のアプリケーションの前記表示を過度に縮小することを判定する工程と、前記第 2 のアプリケーションの前記表示に対する前記変更が前記第 2 のアプリケーションの前記表示を過度に縮小するという判定に応じて、前記第 2 のアプリケーションの前記表示をアイコンに変換する工程を有することを特徴とする請求項 33 に記載の方法。

【請求項 41】

前記 1 つまたは複数の他のアプリケーションは第 2 のアプリケーションを含み、

前記第 1 のアプリケーションと前記第 2 のアプリケーションは親アプリケーション内で実行され；

前記第 1 のアプリケーションは前記第 1 のアプリケーションの前記表示に対する前記要求された変更を前記親アプリケーションに報告し；

前記親アプリケーションは前記決定工程を実行することを特徴とする請求項 33 に記載の方法。

【請求項 42】

プロセッサで読取り可能な記憶装置上で具現化されるプロセッサで読取り可能なコードを有する 1 つまたは複数のプロセッサで読取り可能な記憶装置であって、

プロセッサで読取り可能な前記コードは、

第 1 のアプリケーションの表示を変更する要求を受信する工程と；

前記第 1 のアプリケーションの前記表示との競合を避けるために、前記第 1 のアプリケーションの前記表示の変更に応じて、1 つまたは複数の他のアプリケーションのための 1 つまたは複数の表示がどのように変化すべきかを決定する工程と；

前記第 1 のアプリケーションの前記表示を変更する工程と；

前記第 1 のアプリケーションの前記表示との競合を避けるために、前記第 1 のアプリケーションの前記表示の変更に応じて、変化すべき 1 つまたは複数の他のアプリケーションのための 1 つまたは複数の表示を自動的に変更する工程を有する方法を 1 つまたは複数のプロセッサが実行するようにプログラムされていることを特徴とする記憶装置。

【請求項 43】

前記決定工程は、前記第 1 のアプリケーションの前記表示が前記 1 つまたは複数の他のアプリケーションのための前記 1 つまたは複数の表示と重なることを避けるために、1 つまたは複数の他のアプリケーションのための前記 1 つまたは複数の表示をどのように調整

するかを決定する工程を有することを特徴とする請求項 4 2 に記載の 1 つまたは複数のプロセッサで読取り可能な記憶装置。

【請求項 4 4】

前記方法は

前記第 1 のアプリケーションの前記表示に対する前記変更が許可されるか否かを判定する工程と；

前記第 1 のアプリケーションの前記表示に対する前記変更が許可されない場合に、前記第 1 のアプリケーションの前記表示に対する前記変更を拒否する工程をさらに有することを特徴とする請求項 4 2 に記載の 1 つまたは複数のプロセッサで読取り可能な記憶装置。

【請求項 4 5】

前記 1 つまたは複数の他のアプリケーションは第 2 のアプリケーションを含み；

前記 1 つまたは複数の他のアプリケーションのための前記 1 つまたは複数の表示を変更する前記工程は、前記第 1 のアプリケーションの前記表示との重なりを避けるために前記第 2 のアプリケーションの表示をどのように変更するかを決定する工程と、前記第 2 のアプリケーションの前記表示に対する前記変更が許可されないことを決定する工程と、前記第 1 のアプリケーションの前記表示に対する前記変更を拒否する工程を有することを特徴とする請求項 4 2 に記載の 1 つまたは複数のプロセッサで読取り可能な記憶装置。

10

【請求項 4 6】

第 1 のアプリケーションの表示を変更する要求を受信する手段と；

前記第 1 のアプリケーションの前記表示との競合を避けるために、前記第 1 のアプリケーションの前記表示の変更に応じて、1 つまたは複数の他のアプリケーションのための 1 つまたは複数の表示がどのように変化すべきかを決定する手段と；

前記第 1 のアプリケーションの前記表示を変更する手段と；

前記第 1 のアプリケーションの前記表示との競合を避けるために、前記第 1 のアプリケーションの前記表示の変更に応じて、変化すべき 1 つまたは複数の他のアプリケーションのための 1 つまたは複数の表示を変更する手段を有するアプリケーションを実行可能な装置。

20

【請求項 4 7】

入力装置と；

表示装置と；

アプリケーション用のコードを記憶する記憶装置と；

前記入力装置、前記表示装置、および前記記憶装置と通信を行う処理装置を有し、

前記処理装置は、前記表示装置に前記アプリケーションを表示するために、前記コードにアクセスし、第 1 のアプリケーションの表示の変更する要求を受信し、前記第 1 のアプリケーションが前記第 2 のアプリケーションと競合しないように、前記第 1 のアプリケーションの前記表示を変更する前記要求に応じて、第 2 のアプリケーションの表示を自動的に変更することを特徴とするアプリケーションを実行可能な装置。

30

【請求項 4 8】

前記処理装置は、前記第 1 のアプリケーションの前記表示に対する前記変更が許可されるか否かを判定し、前記第 1 のアプリケーションの前記表示に対する前記変更が許可されない場合は、前記第 1 のアプリケーションの前記表示を変更しないことを特徴とする請求項 4 7 に記載の装置。

40

【請求項 4 9】

前記処理装置は、前記第 2 のアプリケーションの前記表示に対する前記変更が許可されるか否かを判定し、前記第 2 のアプリケーションの前記表示に対する前記変更が許可されない場合は、前記第 1 のアプリケーションの前記表示を変更しないことを特徴とする請求項 4 7 に記載の装置。

【請求項 5 0】

前記第 1 のアプリケーションの表示が前記第 2 のアプリケーションの表示と重ならない

50

場合は、前記第1のアプリケーションは前記第2のアプリケーションと競合しないことを特徴とする請求項47に記載の装置。

【発明の詳細な説明】

【技術分野】

【0001】

(技術分野)

本発明は複数のアプリケーションを表示するための技術に関する。

【背景技術】

【0002】

(関連技術の説明)

コンピュータの性能がより向上し、ソフトウェアがより高機能化するにつれ、コンピュータのユーザは、コンピュータの能力をより利用して複数のアプリケーションを同時に実行するようになってきている。例えば、ユーザはデスクトップ上に複数のウィンドウを開き、各ウィンドウで1つまたは複数のアプリケーションを実行する。あるいは、デスクトップ上に1つのウィンドウを開き、第1のアプリケーションを実行する。その後、第1のアプリケーションの中で他の複数のアプリケーションを実行することができる。後者の1つの例としては、ユーザがウェブ・ブラウザを実行すると、そのウェブ・ブラウザの中で複数のアプリケーションが実行される場合である。

10

【発明の開示】

【発明が解決しようとする課題】

20

【0003】

コンピュータ上でウィンドウ表示されるアプリケーションを複数実行することについての1つの問題は、複数のアプリケーションが表示される表示領域が過度に煩雑になることである。例えば、ウィンドウ表示されるアプリケーションが複数開かれると、1つまたは複数のアプリケーションによって他のアプリケーションが見えなくなることがある。ユーザがアプリケーションにアクセスしようとする時、まず、そのアプリケーションの表示を探し、サイズを変更し、手動で他のアプリケーションのサイズを変更するか、あるいは他のアプリケーションを移動させなければならない場合がある。通常、複数のアプリケーションを同時に見ることは難しい。

【0004】

30

したがって、表示領域を共有する複数のアプリケーション用のユーザ・インターフェースをより好適に管理する必要がある。

【課題を解決するための手段】

【0005】

本発明は、概して説明すると、複数のアプリケーションを1つの表示領域に表示する技術に関する。これら複数のアプリケーションは1つの表示領域を共有する。1つのアプリケーションが変更されると、それに応じて他の1つまたは複数のアプリケーションは自動的に調整される。例えば、1つの実施形態では、1つのアプリケーションが拡大されると、他のアプリケーションはそれに応じて縮小、および/または移動する。

【0006】

40

1つの実施形態では、本発明を実行するシステムは、少なくとも第1のアプリケーションと第2のアプリケーションとを含む1組のアプリケーションを表示する。第1のアプリケーションの表示が変更される。第2のアプリケーションの表示は、第1のアプリケーションの表示の変更に応じて自動的に変更される。異なる実施形態では、第2のアプリケーションの表示は、第1のアプリケーションの表示の変更前、変更後、または変更中に変更することができる。

【0007】

他の実施形態では、複数のアプリケーションが表示される。第1のアプリケーションの表示の変更の要求を受信する。第1のアプリケーションの表示との競合を避けるため、他の1つまたは複数のアプリケーションの表示は、第1のアプリケーションの表示の変更

50

応じてどのように変更するべきかが決定される。第1のアプリケーションの表示が変更される。他の1つまたは複数のアプリケーションの表示は、第1のアプリケーションの表示との競合を避けるために、第1のアプリケーションの表示の変更の要求に応じて自動的に変更される。

【0008】

1つの実施形態では、第1のアプリケーションの表示が第2のアプリケーションの表示と重なると、第1のアプリケーションは第2のアプリケーションと競合しているといえる。他の実施形態では、アプリケーションは、一部の重なりを許可するが他は許可しないように構成することができる。例えば、第2のアプリケーションは、下部の15ピクセルの重なりを許可するように構成できる。第1のアプリケーションの変更が第2のアプリケーションの下部の15ピクセルとのみ重なるものである場合、競合はない。第1のアプリケーションの変更が第2のアプリケーションの下部15ピクセルを越えて重なるものである場合、競合が発生している。他の実施形態では、アプリケーション間に競合があるか否かを判定するために様々な基準を用いることができる。

10

【0009】

本発明は、ハードウェア、ソフトウェア、またはハードウェアとソフトウェア両者の組合せを用いることで実現することができる。本発明のために使用されるソフトウェアは、ハード・ディスク・ドライブ、CD-ROM、光ディスク、フロッピ・ディスク、磁気テープ装置、RAM、ROM、フラッシュ・メモリ、または他の適切な記憶装置を含むプロセッサで読取り可能な1つまたは複数の記憶装置に保存される。別の実施形態では、ソフトウェアの一部またはすべては、カスタムIC、ゲート・アレイ、FPGA、PLD、および専用のプロセッサを含む専用ハードウェアで置き換えることができる。1つの実施形態では、1つまたは複数のプロセッサの動作を規定するために本発明を実施するソフトウェアが使用される。1つまたは複数のプロセッサは、1つまたは複数の記憶装置（ハード・ディスク・ドライブ、CD-ROM、DVD、光ディスク、フロッピ・ディスク、磁気テープ装置、RAM、ROM、フラッシュ・メモリ、または他の適切な記憶装置）、周辺機器（プリンタ、モニタ、キーボード、ポインティング・デバイス）、および/または通信インタフェース（例えば、ネットワーク・カード、無線送信器/受信機等）と通信を行うことができる。

20

【0010】

本発明のこれらおよび他の目的および利点は、本発明の好ましい実施形態を図面を参照しながら記載した以下の説明から明確になるであろう。

30

【発明を実施するための最良の形態】

【0011】

本発明は、複数のアプリケーションを1つの表示領域に表示する技術に関する。1つのアプリケーションが変更されると、他の1つまたは複数のアプリケーションは、それに応じて調整される。1つの実施形態では、システムは表示領域内のアプリケーション間で重ならない関係を維持するように動作する。例えば、表示領域10を示す図1について考える。1つの実施形態では、表示領域10はコンピュータ装置のモニタまたはスクリーンに描かれたウィンドウ、デスクトップまたは他の形式の表示領域である。表示領域10内では、アプリケーション12、14、16、18を含む4つのアプリケーションが実行されている。これらアプリケーションの使用中に、ユーザが1つのアプリケーションのサイズ、位置、またはローテーションの変更しようとすることがある。例えば、ユーザは、アプリケーションのサイズを変更するためにアプリケーションの角をドラッグしたり、アプリケーションのサイズを変更するためにアプリケーションの縁をドラッグしたり、アプリケーションを移動するためにアプリケーション全体をドラッグしたり、アプリケーションの位置合わせをするために新たなサイズをキー入力したり、新たな座標をキー入力したりすることなどができる。ユーザがアプリケーション12のサイズを変更しようとする場合を考える。本発明は、アプリケーション12が他のアプリケーションと重ならないようにアプリケーション12のサイズ変更を適切に行うために、他のアプリケーションの1つまた

40

50

は複数がそれに応じて自動的にサイズ変更され、あるいは移動されることを可能にする。単語「重なる」は、あるアプリケーションの表示（例えばウィンドウ）が他のアプリケーションの表示の上、またはその下に存在することを示すために使用される。図2は、アプリケーション12がサイズ変更されて大きくなった場合に、本発明の1つの実施形態において、行われることの1例を示している。この例では、アプリケーション14はその幅が縮小され、アプリケーション16はその高さが縮小され、アプリケーション18はその高さが縮小されている。アプリケーションがサイズを変更するにしたがって、そのコンテンツは調整される。例えば、画像がサイズ変更されおよび/またはその解像度が変更されたり、文字が見えなくなったり、あるいは表示内の項目の配置が変更される。

**【0012】**

10

1つの実施形態では、アプリケーションはある状況においては重なることができる。例えば、第1のアプリケーションは、ある領域（例えば、左端の10ピクセル、上部の6ピクセル等）が重なることは許可するが、他の領域は重ならない。第2のアプリケーションが、重なることを許可された第1のアプリケーションの領域と重なるように変更される場合は、競合はなく、第1のアプリケーションは調整されない。第2のアプリケーションが、重なることを許可されていない第1のアプリケーションの領域と重なるように変更される場合は、競合があり、第1のアプリケーションは、不適切な重なりを回避するように調整される。

**【0013】**

図3は、他のアプリケーション内に埋め込まれ、表示された1つまたは複数のアプリケーションを含む表示領域20を示している。例えば、表示領域20はアプリケーション22、24、26を含んでいる。アプリケーション22内には、アプリケーション30、32、34が存在する。アプリケーション30内には、アプリケーション40と42が存在する。アプリケーションを埋め込むことによりアプリケーションの階層が形成される。アプリケーションが変更されると、同じ階層レベルの他のアプリケーションはそれに応じて調整される。いくつかの実施形態では、アプリケーションが変更されると、他の階層レベルのアプリケーションもそれに応じて調整される場合がある。

20

**【0014】**

1つの例では、表示領域20は、アプリケーション（例えばウィンドウ、ブラウザ、ダッシュボード等）である。アプリケーション22のサイズ、位置、またはローテーションが変更されると、アプリケーション24と26はそれに応じて、可能な限り、変更される。同様に、アプリケーション42のサイズが変更されると（例えば、大きくなるか小さくなる、XまたはY方向へ移動する）、アプリケーション40もそれに応じて変更される。例えば、図4は、アプリケーション42の幅サイズが拡大された後の表示領域20を示している。それに応じて、アプリケーション40は、アプリケーション42のサイズの増加に合わせて、その幅サイズが自動的に縮小されている。図5は、アプリケーション30の高さサイズが拡大された後の表示領域20を示している。アプリケーション30のサイズの増加に応じて、アプリケーション32と34の高さサイズは自動的に縮小されている。図5では、アプリケーション30が拡大された場合に、アプリケーション40と42が拡大されていないことに留意されたい。他の実施形態では、アプリケーション40と42は、アプリケーション30内の空いている領域を活用して拡大することができる。

30

40

**【0015】**

図1～5には、本発明により提供される基本的な機能を示している。すなわち、1つのアプリケーションが変更される（または、変更するよう要求される）と、他のアプリケーションは自動的に調整される。これにより、ユーザは1つの表示領域で複数のアプリケーションを管理できるようになる。例えば、ユーザが特定のアプリケーションに関心があれば、そのアプリケーションを使用するために、ユーザはその特定のアプリケーションを拡大することができる。このアプリケーションが拡大されている間は、他のアプリケーションは縮小されるので表示領域は煩雑にならない。

**【0016】**

50

本明細書の目的では、アプリケーションは見たり、修正したり、または一連のデータの処理ができる任意の独立した機能として定義される。アプリケーションはサイズの変更を許可している。いくつかの実施形態では、アプリケーションは最小限のサイズを有していてもよい。アプリケーションは、グループ化されていても、埋め込まれていても、または他のアプリケーションを含んでもよい。アプリケーションの例としては、画像編集アプリケーション、カレンダー・アプリケーション、ニュース提供アプリケーション、電子メール・アプリケーション、文書処理アプリケーション、表計算アプリケーション、インターネット閲覧アプリケーション等が挙げられる。いくつかの実施形態では、表示領域の一部では本発明を利用できるが、表示領域の他の部分では本発明は使用されない。

【0017】

図6は、図1に対応するアプリケーションの階層を示している。すなわち、階層は2つのレベルを有している。トップ・レベルには、アプリケーション10が存在する。2番目のレベルには、アプリケーション12、14、16、18が存在する。従って、アプリケーション12～18はアプリケーション10内に埋め込まれている。

【0018】

図7は、図3～5に対応する階層の例を提供している。図7の階層は4つのレベルを含んでいる。トップ・レベルには、アプリケーション20が存在し、これは表示領域となっている。階層の2番目のレベルには、アプリケーション22、24、26が存在する。アプリケーション22～26はアプリケーション20内に存在する。階層の3番目のレベルには、アプリケーション30、32、34が存在し、それらのすべてはアプリケーション22内に表示される。階層の4番目のレベルには、アプリケーション40と42が存在し、両方ともアプリケーション30内に表示される。本発明の1つ実施形態では、階層内のどのレベルにおいても、アプリケーションはその兄弟アプリケーションと重ならない関係を維持するように動作する。兄弟アプリケーションとは、同じ階層レベルにある他のアプリケーションである。例えば、1つの実施形態では、システムは、アプリケーション30がアプリケーション32及び34と重ならないように動作する。同様に、システムは、アプリケーション22がアプリケーション24及び26と重ならないように動作する。

【0019】

アプリケーションのサイズ・パラメータ（幅、高さ、位置、ローテーション、その他）によるユーザの対話は、階層内で敏速な処理を開始させる。変更を要求された対象であるアプリケーションは「イニシエータ」となり、変更するという要求をその親アプリケーションに伝達する。親アプリケーションは、階層内のイニシエータの真上のアプリケーションである。例えば、アプリケーション40と42の親アプリケーションはアプリケーション30であり、アプリケーション30、32、34の親アプリケーションはアプリケーション22であり、アプリケーション22～26の親アプリケーションはアプリケーション20である。親アプリケーションは、イニシエータが変更の要求をすることを検知すると、仲介プログラムとなる。親アプリケーションは、イニシエータの兄弟アプリケーションと関連付けてイニシエータの要求を評価し、またいくつかの実施形態では親プログラム自身を評価する。その変更が許可されると、イニシエータはサイズ・パラメータを新しい値に設定する。親アプリケーションは、これに回答して、兄弟アプリケーションのサイズ・パラメータの調整を行う。親アプリケーションが、自身のサイズ・パラメータを調整してもよい。これは実時間処理であり、イニシエータに対する処理が行われるのにしたがって、イニシエータに対する動作として親アプリケーションとのネゴシエーションが行われ、（アニメーションの有無にかかわらず）修正されるアプリケーションと隣接するアプリケーションとの間でユーザのために連続的な対話が行われる。

【0020】

図8は、本発明にしたがってアプリケーションがどのように表示されるかを管理する処理の1つの実施形態を説明するフローチャートである。工程102では、アプリケーションのサイズ、位置、またはローテーションを変更する試みがなされる。上述のように、これは、オペレータがマウスを使用してウィンドウまたは他のインタフェース・アイテムの

10

20

30

40

50

角、縁、または他の部分をドラッグすることによりアプリケーションのサイズを変更することができる。他の実施形態では、オペレータは新しいサイズまたは新しい位置をキー入力することができる。サイズまたは位置の変更を試みることにより、工程104でその要求が行われる。1つの実施形態では、この要求は以下の様々なイベントのうちのいずれか1つまたは複数を含むことができる：onX（X位置の変更）、onY（Y位置の変更）、onWidth（幅の変更）、onHeight（高さの変更）、onDelete、onInsert、onClick（マウスボタンのクリック）、特定の実施形態に特有な他のイベント。イベントは、イベントを生成しているアプリケーション、イベント内で要求される変更（古い値、新しい値、および/または差分値）、およびイベントの形式に関する情報を含む。この要求は、工程106において親アプリケーションにより受信される。工程108では、その変更が許可されるかどうか判定される。1つの実施形態では、アプリケーションに対して何を変更することができるかを定義するため、所定の1連の規則を使用できる。親アプリケーションは、これらの定義された規則の範囲内で変更を許可することができるか否かを判定することができる。他の実施形態では、変更を要求するアプリケーションは、それに応じて変更できるかどうかをアプリケーション自身で判定できる。この規則の例としては、アプリケーションは最小限のサイズまたは最大限のサイズを有していなければならないという規則を挙げることができる。さらに、規則により、アプリケーションの位置を表示領域のうちの特定の位置に限定することができる。変更が許可されないと変更は拒否され、図8の手順は、変更を行うことなく終了される（工程110）。

10

20

#### 【0021】

変更が許可されると、工程112によって、同じ階層レベルの他のアプリケーションへの変更の影響が決定される。ネゴシエーションが行われ、要求の詳細、要求を受信するアプリケーションの許容範囲、および親アプリケーションに基づいて、受諾、拒否、または調整という結果となる。これらの各要素は、要求の結果としてアプリケーションのサイズ/位置/ローテーションを変更することができる。この手順は、親アプリケーションへの鎖を（階層の上のレベルへ）上ることも鎖を下ることもできる。いくつかの実施形態では、変更は、異なる階層レベルのアプリケーションに影響することができる。工程114では、同じ階層レベルの他のアプリケーションへの変更の影響が許可されるか否かが判定される。変更による他のアプリケーションへの変更の影響が許可されないと、その変更は工程116において拒否される。例えば、拡大中のアプリケーション12（図1参照）が、アプリケーション16を過度に縮小させると（すなわち、アプリケーション16が最小限のサイズを有するので）、アプリケーション12に対する変更は拒否される。他のアプリケーションへの変更の影響を許可できれば、変更を求めた最初のアプリケーションは工程118において変更され、そして同じ階層レベルで適切に影響を受けるアプリケーションは、工程120において変更される。なお、工程120は、工程118の前後、またはその間に実行することができる。他の実施形態では、工程118は、図8の処理の初期において実行することができる。

30

#### 【0022】

図9は、1つまたは複数のアプリケーションを開始する処理の1つの実施形態を説明するフローチャートを示している。工程180では、各アプリケーションを表示するために、アプリケーションUI（ユーザ・インターフェース）オブジェクトが生成される。他の実施形態では、様々な要素（オブジェクト以外）が各アプリケーションに対して生成される。工程182では、レイアウト・オブジェクトが生成される。レイアウト・オブジェクトはアプリケーションに関連するオブジェクトであり、アプリケーション内にすべての情報を表示する役割をする。したがって、埋込まれた他のアプリケーションを有するすべてのアプリケーションに対して、レイアウト・オブジェクトは、埋込まれたアプリケーションが親アプリケーションの中でどのように表示されるか記述する。例えば、図7の階層においては、アプリケーション22は、アプリケーション30、32、34がアプリケーション22の中でどのように表示されるかを記述するレイアウト・オブジェクトを有してい

40

50

る。本発明の他の実施形態ではレイアウト・オブジェクトを使用しない。工程 184 では、特定のアプリケーションに対する UI オブジェクトがアクセスされる。工程 186 では、工程 184 において UI オブジェクトがアクセスされたアプリケーションの親アプリケーションが、アプリケーション UI オブジェクトにポインタを提供する。このポインタは、親アプリケーションに対するレイアウト・オブジェクトの機能を参照する。ポインタを受信したアプリケーションはこのポインタを保存する。アプリケーションがサイズまたは位置（または、他の表示特性）を変更されたときは、アプリケーションは受信したポインタにより参照される機能呼び出す。ポインタにより参照された機能は、レイアウト・オブジェクト内にある。呼び出された機能は、生成されたイベントに関する情報を受信する。ポインタを受信したアプリケーションは、工程 188 において、親アプリケーションの 10 ためのユーザ・インターフェース内に置かれる。例えば、工程 188 は、アプリケーション 42 をアプリケーション 30 内に挿入する工程を含むことができる（図 3 参照）。工程 190 では、処理すべきアプリケーションが他に存在するか否かが判定される。1 つの実施形態では、工程 184 ~ 188 は、表示される各アプリケーションに対して実行される。処理すべきアプリケーションがさらにあれば、この方法が工程 184 において継続して行われる。処理すべきアプリケーションが存在しなければ、工程 192 においてアプリケーションはコンピュータ装置により実行される。

#### 【0023】

図 10A ~ 10J は、アプリケーションの変更による他のアプリケーションへの影響、他のアプリケーションの応答的な変更の方法を決定する処理の様々な実施形態を説明する 20 フローチャートを示している。図 10A ~ 10J では特定の例が提供されるが、ここで説明される概念は他のレイアウトに適用することができる。

#### 【0024】

例えば、図 10A は、単純なレイアウトに対する処理の 1 つの実施形態を説明するフローチャートを示している。単純なレイアウトは、横方向または縦方向に制約がない 1 組のアプリケーションを表示する表示領域を有している。いくつかの実施形態では、単純なレイアウトは、アプリケーション間に最小限の間隔の要求を有することができる。他の実施形態もまた、アプリケーションの最小限のサイズおよび / または最大限のサイズを含むことができる。図 11A は、単純なレイアウトの例を示している。図 11A は、6 つのアプリケーション 702、704、706、708、710、712 を有する表示領域 700 30 を示している。6 つのアプリケーションのそれぞれは、縦方向または横方向にサイズを変更することができる。6 つのアプリケーションのそれぞれは、横方向または縦方向へ移動することもできる。

#### 【0025】

図 10A の工程 200 において、イニシエータ・アプリケーションの縦方向の配置（サイズ、または位置）が変更されたか否か（または、変更されることが要求されたか否か）が判定される。変更されている（または、変更が要求されている）アプリケーションに 40 応答して、そのアプリケーションと縦方向の位置関係にある同じ階層レベルの兄弟アプリケーションは、工程 202 においてその縦方向の配置が自動的に調整される。縦方向の位置関係とは、互いに上下に隣接しているか、あるいは 1 つが縦方向のサイズを変更されると互いに影響を及ぼし合うアプリケーションを指す。例えば、図 11C は、アプリケーション 704 の縦方向のサイズが拡大されると、工程 202 においてアプリケーション 708 と 712 が影響を受けることを示している。アプリケーション 708 は、縦方向においてアプリケーション 704 と隣接している。いくつかの実施形態では、アプリケーション 704 の拡大はアプリケーション 708 にのみ影響を与える。他の実施形態では、アプリケーション 704 の拡大はアプリケーション 708 と 712 に影響を与える。同じ階層レベルの他の兄弟アプリケーションに影響を与える様々な方法がある。1 つの実施形態では、拡大されるアプリケーションに対応するために隣接するアプリケーションのみが縮小される。他の実施形態では、縦方向の位置関係にあるすべてのアプリケーションは、均等に比例配分的に縮小される。図 11C の実施形態では、兄弟アプリケーション 708 と 712 50

は、サイズが変更されるのではなく、移動される。工程 202 の後か、あるいは工程 200 の後で縦方向の配置が変更されなかった場合、システムは、イニシエータ・アプリケーションの横方向の配置（サイズまたは位置）が変更されているかどうかを判定する。変更されていないければ、図 10A の処理は完了する。変更されていれば、工程 206 において、イニシエータと横方向の位置関係にある兄弟アプリケーションは横方向の配置（サイズまたは位置）を変更する。イニシエータと横方向の位置関係を有する兄弟アプリケーションとは、横方向において隣接しているか、若しくはイニシエータの横方向のサイズの変更により影響を受けるかのいずれかの兄弟アプリケーションをいう。例えば、図 11B は、アプリケーション 702 の横方向のサイズが拡大されると、その隣接するアプリケーション 704 が影響を受けることを示す。工程 206 は、そのサイズを変更せずに、アプリケーション 704 の位置を調整することを含んでいる。 10

#### 【0026】

図 10B は、アプリケーションが複数の列で配列されるとともにその表示領域の高さ方向のサイズが固定されていないシステムを説明するフローチャートを示している。例えば、図 11D は、2 列に表示されたアプリケーション 732、734、736、738、740、742 を含む表示領域 730 を示している。1 番目の列はアプリケーション 732、734、736 を含んでいる。2 番目の列はアプリケーション 738、740、742 を含んでいる。特定の列のアプリケーションが高さ方向において拡大されると、図 11C に示すように上下のアプリケーションは移動される。アプリケーションがその幅を増加、縮小させると、列全体の幅が増加、縮小され、それによって図 11E に示すように、その列内のすべてのアプリケーションの幅が縮小、増加される。すなわち、図 11E は、アプリケーション 732 が横方向に拡大されたことを示す。アプリケーション 734 と 736 も、アプリケーション 732 と同じ列にあるために、これに回答して拡大されている。1 番目の列の拡大に応じて、2 番目の列（アプリケーション 738、740、742 を含む）は幅が減少され、その列内のすべてのアプリケーションも幅が減少されている。 20

#### 【0027】

図 10B の工程 240 では、イニシエータの高さが変更されたか否かが判定される。イニシエータは、図 8 の工程 102 において変更の要求を受信したアプリケーションである。イニシエータの高さを変更することが要求されると、工程 242 において、図 11C に示すように、同じ列内のすべての兄弟アプリケーションの縦方向の位置が変更される。イニシエータの高さが変更されなかった場合は（すなわち、工程 242 後では）、工程 244 において、システムはイニシエータの幅が変更されたかどうかを判定する。変更されていないければ、図 10B の処理は終了する。 30

#### 【0028】

イニシエータの幅が変更された場合、工程 246 において、アプリケーション間の最小限の間隔を保持するために、列の幅をどの程度にすべきかが決定される。工程 248 では、列の幅が許容範囲内にあるかどうかを判定される。すなわちいくつかの実施形態では、許容される最小限および最大限の列の幅がある。新たな列の幅が許容範囲内でない場合、イニシエータまたはその兄弟アプリケーションに対する変更の実行は許可されない（工程 250）。新たな列の幅が許容範囲内にある場合、工程 252 において、システムは、隣接する列の幅に対してどのような変更をすべきかを決定する。すなわち、1 番目の列の幅が拡大されると、隣接する列は縮小しなければならない。いくつかの実施形態では、表示領域（例えば表示領域 730）はその幅が固定された表示領域である。したがって、1 つの列が拡大されると、他の列は縮小されなければならない。同様に、いくつかの列が縮小されると、他の列は拡大されなければならない。工程 254 では、システムは、新たな隣接する列が許容範囲内（例えば、最小限の幅と最大限の幅の間）にあるかどうかを判定する。隣接する列の新たな幅が許容範囲内に無い場合、いずれの変更も許可されない（工程 256）。隣接する列の新たな幅が許容範囲内にある場合、工程 258 において、隣接する列の幅は調整され、隣接する列内のすべてのアプリケーションはその幅を新たな列の幅に調整される。工程 260 では、イニシエータ・アプリケーションを含む列も調整される 40 50

。兄弟アプリケーションだけでなくイニシエータ・アプリケーションも、工程 260 において、その幅を新たな列の幅に調整される。例えば、図 11E は、アプリケーション 738、740、742 を有する 2 番目の列は幅が狭くなるが、アプリケーション 732、734、736 を有する 1 番目の列は幅が広くなることを示している。

#### 【0029】

図 10C は、単一の列を有する表示領域の処理の 1 つの実施形態を説明するフローチャートであり、ここでは、表示領域のサイズは固定されている、または少なくとも縦方向のサイズ（高さ）は固定されている。例えば、図 11F は、単一の列で配置された 3 つのアプリケーション 762、764、766 を有する表示領域 760 を示している。表示領域 760 のサイズは変更することができない。表示領域 760 も表示領域の端との間、および各アプリケーション間の最小限の間隔を有している。

10

#### 【0030】

工程 300 では、イニシエータの幅が変更されたか否かが判定される。変更されていれば、工程 302 において、列の幅に対する変更が決定される。いくつかの実施形態では、列の幅は、列の中で最も幅の広いアプリケーションの幅に等しく（または、わずかに大きく）なければならない。工程 304 では、新たな列の幅が許容範囲内にあるか否かが判定される。そうでなければ、この変更は許可されない（工程 306）。新たな列の幅が許容範囲内にあると、工程 308 において、列の幅は調整され、イニシエータ・アプリケーションの幅は調整され、その列のアプリケーションの幅は同様に調整される。例えば、図 11G は、アプリケーション 764 の幅の拡大による影響を示している。アプリケーション 764 の拡大（または、拡大の要求）に応じて、列全体が拡大され、アプリケーション 762 と 766 も同様にその幅を拡大される。

20

#### 【0031】

工程 308 の後、あるいは工程 300 において幅が変更されなかった場合、引き続き図 10C の工程 310 の処理が行われる。工程 310 では、イニシエータの縦方向のサイズが変更されたか否かが判定される。変更されていなければ、図 10C の処理は終了する。変更されていれば、システムは、その列における残りの領域を算出する。すなわち、システムは、列のどれだけがイニシエータ・アプリケーションにより占有されていないかを決定する。次に、工程 314 において、その残りの領域は兄弟アプリケーション間で均等に分配される。他の実施形態では、兄弟アプリケーションは、現在のサイズに比例させてその領域を分配することができる。工程 316 では、システムは、兄弟アプリケーションを、工程 314 で決定されたように調整できるか否かを判定する。調整できなければ、いずれの変更も許可されない（工程 322）。兄弟アプリケーションが工程 314 の算出結果に従って調整することができれば、工程 318 においてイニシエータの縦方向の位置 / サイズが調整され、工程 320 において兄弟アプリケーションの縦方向の位置 / サイズが調整される。例えば、図 11H は、縦方向に拡大されたアプリケーション 762 を示している。拡大するアプリケーション 762 に応じて、アプリケーション 764 と 766 は、自動的に（人間の介入なしで）その高さを縮小される。

30

#### 【0032】

図 10D は、表示領域の高さが固定されており、複数の列を有する表示領域に対して実行される処理を説明するフローチャートである。すなわち、表示領域は縦方向に拡大できない。いくつかの実施形態では、表示領域は、横方向にも拡大できない。図 10D の工程 360 では、イニシエータ・アプリケーションの幅が変更されたか否かが判定される。変更されなければ、引き続き工程 378 の処理が行われる。イニシエータの幅が変更された場合、工程 362 において、システムは列の幅がイニシエータの新たな幅と一致するように新たな列の幅を算出する。工程 364 では、新たな列の幅が許容範囲内にあるか否かが判定される。そうでなければ、変更は許可されない（工程 366）。新たな列の幅が許容範囲内にあると、工程 368 において、システムは隣接する列の幅に対して行うべき変更を決定する。隣接するアプリケーションの新たな列の幅が許容範囲内でない場合（工程 370）は、いかなる変更も許可されない（工程 372）。隣接する列の新たな列の幅が許

40

50

容範囲内にあると、工程 374 において、隣接する列の幅は調整され、隣接する列のすべてのアプリケーションは隣接する列の幅と一致するようにその幅を調整される。工程 376 では、イニシエータの列の幅は調整され、イニシエータのサイズは要求通りに調整され、そしてイニシエータの列のすべての兄弟アプリケーションは、その幅をその列の幅と同じ幅（または、所定の一連の規則により定義された列の幅より若干小さい、または若干大きい適切な幅）に調整される。

#### 【0033】

工程 378 では、イニシエータの縦方向のサイズが変更されたか否かが判定される。変更されていない場合は、図 10B の処理は終了する。イニシエータの縦方向のサイズが変更された場合、工程 380 において、システムは列における残りの領域を決定する。上述したのと同様に、工程 382 においては、残りの領域は兄弟アプリケーション間で分配される。工程 384 では、工程 382 において決定されたように兄弟アプリケーションを調整することができるかどうか判定される。そうでなければ、縦方向のサイズの変更は許可されない（工程 386）。兄弟アプリケーションが調整できれば（工程 384）、イニシエータの縦方向のサイズが工程 388 において調整され、兄弟アプリケーションの縦方向のサイズ（または位置）が工程 390 において調整される。

10

#### 【0034】

いくつかの実施形態では、2つ以上のアプリケーションをグループ化できる。グループとは互いに所定の関係を有するアプリケーションの組である。この所定の関係は、グループ内のアプリケーションのサイズ変更、またはグループ内のアプリケーションの位置合わせに関する規則を有することができる。例えば、すべてのアプリケーションは同じサイズを有するという規則にしてもよい。したがって、1つのアプリケーションのサイズの変更は、すべてのアプリケーションのサイズを変更する。あるいは、グループの規則は、すべてのアプリケーションが一直線上に、曲線上に、対角線上に、または他の形状/関係上になければならないこと示していてもよい。サイズ変更、間隔設定、または他の表示パラメータに関するあらゆる規則を取り決めることができる。

20

#### 【0035】

図 10E は、グループ・メンバーの間で、サイズを調整、および/または位置を変更する処理を説明するフローチャートである。工程 400 では、グループのあるメンバーが変更される。工程 402 では、グループの他のメンバーもまた、1つまたは複数のグループの規則により定義される関係を維持するように変更される。あるいは、変更を要求されたイニシエータ・アプリケーションはグループの一部でなくてもよい。しかしながら、イニシエータを変更することにより、イニシエータをグループの1つまたは複数のメンバーと重なるように位置させることができる。したがって、システムは、イニシエータとの重なりを避けるために、グループのメンバーを変更しようとする。工程 410 でグループのそのメンバーを変更することにより、工程 402 では、システムはグループの規則に従ってグループの他のメンバーを自動的に変更させる。例えば、図 11I は、アプリケーション 782、784、786 のグループを示している。図 11I に示されたこの特定のグループは、アプリケーションが同一のサイズであり、かつ対角線上に位置しなければならないという規則を有している。図 11J は、アプリケーション 784 の移動、及びサイズの変更による影響を示している。これによって、システムに自動的に、グループの3つのメンバー全てが同じサイズを有し、対角線に沿って位置するように、アプリケーション 782 とアプリケーション 786 を移動させ、サイズの変更をさせる。

30

40

#### 【0036】

他の実施形態は、1組のアプリケーションをルール・レイアウトで表示する。例えば、図 11K は、アプリケーション 800 が他のアプリケーションより大きく表示され、アプリケーション 800 が主要位置に表示されたルール・レイアウトを示している。他のアプリケーション 802、804、806、808、810 はアプリケーション 800 の下部に横方向に直線状に並んで配置されており、各アプリケーションは同じサイズであり、アプリケーション 800 より小さい。アプリケーション 802 ~ 810 のいずれかを選択す

50

ること、選択されたアプリケーションは、より大きなサイズに変更されて、主要位置のアプリケーション 800 と置き換えられる。次に、アプリケーション 800 は他のアプリケーションと並んだ位置に移動され、他のアプリケーションと同じサイズに変更される。例えば、図 11L は、アプリケーション 806 が選択されてアプリケーション 800 と置き換えられたことを示している。したがって、アプリケーション 806 はサイズが変更されて主要位置に配置されている。アプリケーション 800 はより小さいサイズに変更されて、アプリケーション 806 下部のアプリケーションの並びに配置されている。図 10F には、図 11K と 11L との間の変換を実行するためのフローチャートを示している。工程 430 では、選択されたアプリケーションが拡大され、主要位置に配置される。工程 432 では、未選択の兄弟アプリケーションが既存の領域内で等しいサイズとなるようにサイズ変更される。工程 434 では、兄弟アプリケーションは所定の関係に基づいて配置される。図 11K と 11L は、その所定の関係を主要なアプリケーションの下部の直線として示している。他の実施形態では、兄弟アプリケーションは、(例えば、円または他の形状を囲むような) 曲線上に、様々な形状、様々なサイズ等で、そして横方向、縦方向、対角線上等に位置することができる。

10

#### 【0037】

上述の実施形態では、様々なアプリケーションの間における 1 つまたは複数の規則または相互関係について説明している。他の実施形態として、アプリケーションをどこにでも配置することが可能であり、任意の方法でサイズを変更することが可能であり、どこへでも移動可能であり、列またはグループを持たない表示領域を説明する。この自由型の実施形態について、図 10G のフローチャートにより説明する。図 10G の工程 460 では、イニシエータが要求通りに変更されたときに、どの兄弟アプリケーションがイニシエータと重なるかが割り出される。1 つの実施形態では、システムは、何らかの重なりがあるかを確認する。別の実施形態では、システムは許可されない重なりがあるかを確認する。上述したように、いくつかの実施形態では、アプリケーションが所定の重なりを許可し、その他の重なりは許可しない。工程 462 では、重なりを避けるために重なる兄弟アプリケーションのサイズの変更が可能か否かが判定される。いくつかの兄弟アプリケーションは最小限のサイズを有する場合があります、これらは最小限のサイズより小さいサイズに変更することができない。すべての兄弟アプリケーションを、重なることなくサイズを変更できる場合、工程 464 において、アプリケーション間で必要な最小限の間隔を維持するように兄弟アプリケーションのサイズが変更される。例えば、図 11M は、アプリケーション 832、834、836、840、842 を含む表示領域 830 を示している。図 11N は、アプリケーション 836 が横方向(幅)に拡大されたときの状態を示している。工程 460 では、アプリケーション 842 がアプリケーション 836 と重なるということが確認される。したがって、工程 464 では、アプリケーション 842 はその幅を縮小される。

20

30

#### 【0038】

工程 462 において、重なりを避けるためにすべての兄弟アプリケーションのサイズを変更することができないと判定されると、工程 466 において、十分にサイズを変更できる兄弟アプリケーションのサイズが変更される。工程 468 では、十分にサイズを変更できない兄弟アプリケーションが、許容される限度内でサイズを変更される。工程 470 では、十分にサイズが変更されない兄弟アプリケーションの 1 つがアクセスされる。工程 472 では、イニシエータとの重なりを避けるために、アクセスされた兄弟アプリケーションを移動させることができるか否かが判定される。表示領域の境界がこれを許可しないために兄弟アプリケーションを移動させることができないときは、変更は工程 474 において拒否される。兄弟アプリケーションが他の兄弟アプリケーションのいずれとも重なることなく移動できるときは、その兄弟アプリケーションは工程 476 において移動され、引き続き工程 482 において処理が行われる。この移動により兄弟アプリケーションが他の兄弟アプリケーションと重なると、図 10G の処理は、移動される兄弟アプリケーションをイニシエータとして、工程 478 において帰納的に呼び出される。この後退的機能は、

40

50

兄弟アプリケーションの移動を適切に行うため、移動させるかサイズを変更させるかを判定している間に、兄弟アプリケーションに隣接するアプリケーションに与えられなければならない。工程 480 では、兄弟アプリケーションはそれに応じて移動及びサイズ変更される。工程 482 では、本処理は、考慮すべき兄弟アプリケーションがさらに存在するかどうかを判定する。考慮すべき兄弟アプリケーションがさらに存在すれば、引き続き工程 470 の処理が行われるが、そうでなければ、イニシエータは工程 484 においてサイズを変更される。

**【0039】**

図 110 は、工程 466 ~ 484 で実行される処理の例を示している。図 110 の例では、アプリケーション 832 は、拡大するように要求されている。アプリケーション 834 は、アプリケーション 832 が適切にサイズ変更できる程度に十分にはサイズを変更することができない。したがって、アプリケーション 834 が限定的にサイズ変更及び移動できるように、アプリケーション 836 のサイズが変更される。

10

**【0040】**

図 10G が代替の実施形態を示していることに留意されたい（工程 466 A への破線を参照）。この代替の実施形態では、すべての兄弟アプリケーションのサイズを変更できない場合（図 10G の工程 462 参照）、イニシエータのサイズの変更は、重なる兄弟アプリケーションのすべてが適切にサイズ変更されるようにある程度制限される。したがって、イニシエータに許容されるサイズ変更量は、要求されたものより小さくなる。

**【0041】**

別の代替の実施形態では、兄弟アプリケーションがその最小限のサイズより小さいサイズにサイズ変更されようとしていると、その兄弟アプリケーションはアイコンに変換される。例えば、図 11P は、兄弟アプリケーション 834 がその縦方向の最小限のサイズ以下に縦方向にサイズ変更するよう要求される程度にまでアプリケーション 832 が拡大された状態を示している。このとき、アプリケーション 834 のサイズを変更するのではなく、アプリケーション 834 の表示はアイコンに変換され、表示領域 830 の隅（または他の場所）に配置される。

20

**【0042】**

図 10H は、表示領域からアプリケーションを削除する 1 つの実施形態の処理を説明するフローチャートである。例えば、図 11Q は、アプリケーション 862、864、866、868 を有する表示領域 860 を示している。これらのアプリケーションの 1 つが削除されると、他のアプリケーションは、表示内の利用可能な空き領域を利用するために自身のサイズを調整することができる。他の実施形態では、アプリケーションは、1 つのアプリケーションが削除された後にサイズを変更しない。図 10H の工程 500 では、1 つのアプリケーションが削除される。工程 502 では、システムは、残りのアプリケーションがグループの関係を有しているかどうかを判定する。有していれば、残りのアプリケーションは、工程 504 において、そのグループの基づいてそのサイズおよび/または位置を調整する。グループの関係がない場合、工程 506 において、削除されたアプリケーションに隣接するアプリケーションのサイズが、それらの隣接するアプリケーションのサイズに比例して調整される。すなわち、残りの隣接するアプリケーションは、当初のサイズに比例して拡大され、削除されたアプリケーションにより後に残された使用可能な領域を縮小する。例えば、図 11R は、表示領域 860 からアプリケーション 866 を削除した影響を示している（図 11Q 参照）。この例では、アプリケーション 866 により後に残された空き領域を利用するため、アプリケーション 862 と 864 は均等に拡大される。

30

40

**【0043】**

図 10I は、アプリケーションが追加される場合に実行される処理の 1 つの実施形態を説明するフローチャートである。工程 540 では、新たなアプリケーションが、他のアプリケーションに影響することなく要求された配置に収まるかどうか判定される。そうならば、そのアプリケーションは工程 542 において要求された配置で追加される。そうで

50

なければ、工程 5 4 4 において、重なっている隣接するアプリケーションのいずれかを移動することができるか否かが判定される。重なっている隣接するアプリケーションを移動できる場合、工程 5 4 6 において重なっている隣接するアプリケーションがそれに応じて移動され、工程 5 4 8 において新たなアプリケーションが要求された位置に追加される。重なっている隣接するアプリケーションを移動できない場合、工程 5 6 0 において重なっている隣接するアプリケーションのサイズを変更することができるか否かが判定される。そうならば、工程 5 6 2 において重なっている隣接するアプリケーションは適切にサイズ変更され、工程 5 6 4 において新たなアプリケーションは要求された位置に追加される。重なっている隣接するアプリケーションのサイズを変更できない場合、工程 5 6 6 において重なっている隣接するアプリケーションと新たなアプリケーションのサイズを変更できるか否かが判定される。そうでなければ、変更（新たなアプリケーションの追加）は拒否される（工程 5 7 4）。そうならば、工程 5 6 8 において重なっている隣接するアプリケーションのサイズが変更され、工程 5 7 0 において新たなアプリケーションのサイズが変更され、工程 5 7 2 において新たなアプリケーションが要求された配置で追加される。他の実施形態では、新たなアプリケーションの追加は、一部の重なっている隣接するアプリケーションのサイズ変更と、その他の重なっている隣接するアプリケーションの移動を組合せたものを含むことができる。図 1 1 S と図 1 1 T は、アプリケーションの追加の 1 例を提供している。図 1 1 S は、アプリケーション 9 0 2、9 0 4、9 0 6、9 0 8、9 1 0 を含む表示領域 9 0 0 を示している。新しいアプリケーション 9 2 0 を図 1 1 S の表示領域 9 0 0 の位置 X に追加することを考える。位置 X へのアプリケーション 9 2 0 の追加により、アプリケーション 9 2 0 はアプリケーション 9 0 6 と重なる。図 1 1 T は、アプリケーション 9 2 0 を追加するためにアプリケーション 9 0 6 のサイズを変更する例を示している。

10

20

#### 【0044】

図 1 0 J は、表示領域内のある場所から表示領域内の他の場所にアプリケーションを移動する処理の 1 つの実施形態を説明するフローチャートの例を提供している。工程 6 0 0 では、アプリケーションを削除し、兄弟アプリケーションを調整する処理（図 1 0 H）が実行される。工程 6 0 2 では、アプリケーションを追加し、兄弟アプリケーションを調整する処理（図 1 0 I）が実行される。工程 6 0 4 では、両方の処理が正常に実行されたか否かが判定される。そうならば、その変更は工程 6 0 6 において許可される。そうでなければ、その変更は工程 6 0 8 において元に戻される。

30

#### 【0045】

図 1 0 A ~ J はいくつかのレイアウトの例を提供しているが、本発明により、他のレイアウトを使用することもできる。ここで説明した、アプリケーションの表示を適合させる種々の技術と概念は、いかなるタイプのレイアウト、グループ、またはユーザ・インターフェースの構成にも制限されない。

#### 【0046】

いくつかの実施形態では、アプリケーションは、アニメーションを使用して移動またはサイズ変更することができる。アニメーションが使用される場合、影響を受ける兄弟アプリケーションの移動および/またはサイズ変更は、イニシエータのアニメーションが終了した後に、実行することができる。あるいは、影響を受ける兄弟アプリケーションの移動および/またはサイズ変更は、イニシエータの各ピクセル変更の後に、またはイニシエータのアニメーションの各ステップの後に、またはイニシエータのアニメーションのステップのサブセットの後に実行することができる。

40

#### 【0047】

第 1 のアプリケーションのサイズの変更または移動が要求され、第 1 のアプリケーションの変更が完了する前に第 2 のアプリケーションの変更または移動が要求されることもある。1 つの実施形態では、第 2 のアプリケーションの変更または移動が要求された場合、第 1 のアプリケーションに対する変更は中止される。他の実施形態では、第 1 のアプリケーションの変更と兄弟アプリケーションへの影響が完了し、続いて第 2 のアプリケーシ

50

ンの変更と兄弟アプリケーションへの影響が行われる。他の実施形態では、システムは、第1のアプリケーションの変更と兄弟アプリケーションへの影響を実行し、次にアニメーションによって第2のアプリケーションを変更する要求に対してスムーズに変形させる。

【0048】

本発明を実施することができるシステムの一例を図12に示す。図12のシステムは、ネットワークを介して（例えば、インターネットを介して）クライアントにコンテンツを提供するために使用できるプレゼンテーション・サーバ1450を有している。プレゼンテーション・サーバ1450は、ユーザ・インターフェースのマーク・アップ言語記述を受信し、そのマーク・アップ言語記述を実行可能なコードへ動的にコンパイルするように設計されている。ある環境においては、マーク・アップ言語記述は、ユーザ・インターフェースの種々のデータ・ソースおよび/またはウェブ・サービスへの接続と同様に、アプリケーションのユーザ・インターフェースを記述するように設計されたXMLベースの言語である。それは、ユーザ・インターフェース・アイテムをバック・エンド・データ・ソースとサービスに自動的に接続するための構文法だけでなく、「ウィンドウ」、「ボタン」、「テキスト」、「スクロールバー」などの標準的なユーザ・インターフェースの基本要素を含んでいる。マーク・アップ言語は、Java（登録商標）scriptのようなアプリケーションの動作手順を規格するためのスクリプト言語を含むこともできる。

10

【0049】

1つの実施形態では、プレゼンテーション・サーバ1450は、所定のプレゼンテーション・レンダラのための高度に最適化/圧縮されたオブジェクト・コードを生成する。プレゼンテーション・レンダラは、画像を表示し音を再生することができるソフトウェア環境、ハードウェア、1つまたは複数のソフトウェア・プログラムのセットなどである。1つの実施形態では、プレゼンテーション・サーバ1450は、サーバに置かれたマーク・アップ言語を記述するファイルおよびデータをオブジェクト・コードにコンパイルするJava（登録商標）Servletとして実行される。1つの実施形態では、プレゼンテーション・サーバ1450は、Macromedia Flash Playerのためのオブジェクト・コードを生成する。プレゼンテーション・サーバ1450は、Jakarta Tomcatと、BEA WebLogicおよびIBM WebsphereのようなJ2EEサーバを含む任意の標準的なJava（登録商標）Servletの実行によりホストされることがある。図12は、アプリケーション・サーバ1452でホストされるプレゼンテーション・サーバ1450を示している。アプリケーション・サーバ1452はデータベース管理サービス（DB Mgmt）1454を含んでおり、これはリレーショナル・データベース1456と通信を行う。リレーショナル・データベース以外の形式のデータも使用できる。プレゼンテーション・サーバ1450は、ネットワークを介してクライアントと通信を行うウェブ・サーバ1458を介し、クライアントからコンテンツの要求を受信し、応答を送信する。ネットワークは、インターネット、LAN、WANなどを含む、周知の技術の標準的ないずれのネットワークであってもよい。例えば、図12には、ウェブ・サーバ1458を介してプレゼンテーション・サーバ1450と通信するプラグ・イン1462（例えば、Flash Player）を備えたHTTPクライアント1460（例えば、ブラウザ）を示している。1つの実施形態では、クライアントはプラグ・インとしてウェブ・クライアント内に埋込まれたMacromedia Flash Playerである。Flash Playerはプレゼンテーション・サーバ1450の適切な伝達手段であるが、他の多くのプレゼンテーション・レンダラも利用することができる。他の実施形態では、レンダラはプラグ・インである必要はない。

20

30

40

【0050】

図13は、図12のシステムの動作の1つの典型的な実施形態について説明するフローチャートである。工程1502では、プレゼンテーション・サーバは、コンテンツに対する要求を受信する。コンテンツは、ウェブ・ページ、メディア、データ、アプリケーション、またはネットワークを介してアクセス可能な他のいかなるものであってもよい。工程

50

1504では、プレゼンテーション・サーバは、受信した要求に応じて、コンテンツのマーク・アップ言語記述にアクセスする。いくつかの実施形態では、他の言語が使用される。工程1506では、プレゼンテーション・サーバは、要求に応じて、またはマーク・アップ言語記述に応じて、いずれかのデータおよび/またはメディアにアクセスする。場合によっては、この要求がいかなるデータまたはメディアも必要しないため、工程1506が省略されてもよい。工程1508では、マーク・アップ言語記述、データ、および/またはメディアはコンパイルされる。工程1510では、実行可能なコードがクライアントに送信される。工程1512では、実行可能なコードがクライアントで実行される。実行可能なコードは、上述したように表示領域のアプリケーションを管理する機能を含む。プレゼンテーション・サーバを使用するシステムに関するさらなる情報は、2002年3月5日出願の米国特許出願第10/092,010号「プレゼンテーション・サーバ」に記載されており、その全体を参照することによりここに援用する。

10

#### 【0051】

プレゼンテーション・サーバ1450内のコンパイラはさまざまなプログラミング言語によるコードを受信することができる。1つの実施形態では、コンパイラは例えばXMLベースの言語記述のようなマーク・アップ言語記述を受信する。他のマーク・アップ言語も使用できる。さらに、マーク・アップ言語以外のプログラミング言語を使用できる。

#### 【0052】

1つの実施例では、ユーザ・インターフェースは一連のビューから構成されている。ビューは、表示可能なオブジェクト（例えば、ウィンドウ、ボタン、スライダ等）の表示/記述である。ビューはアトリビュート（attribute）およびアトリビュート修飾子を有している。アトリビュートは、例えばビューの外観、音、動作のようなビューの特性を定義する。アトリビュート修飾子は、そのビューまたは他のビューの他のアトリビュートに対するユーザの入力、または変更のようなイベントに応じてアトリビュートを変更する。アトリビュート修飾子の例としては、レイアウト、アニメータ、コンストレイントが挙げられる。1つの実施形態では、レイアウトはそのビューの下位のビューを変更するために使用される。アニメータはビューの外観をアニメ化するために使用される。コンストレイントは、値が変わった場合に、パウンドする特性が表現式の初期値ではなく現在値で維持されるように、繰り返し評価される表現式である。インタフェースの1つの例は、2002年3月5日出願の米国特許出願第10/092,360号「連続型ユーザ・インターフェースを提供するインタフェース・エンジン」に記載されており、その全体を参照することでここに援用する。

20

30

#### 【0053】

アトリビュートの書式は：`(attribute name) = $time { expression }`、であり、ここで、“expression”はアトリビュートの値を設定するために使用される所定の形式の表現式であり、“\$time”は、表現式をいつ評価しなければならないかを示すマーカである。“time”に対して使用できる様々な値が存在する。1つの実施形態では、“time”は、“immediately”、“once”、“always”を示すことができる。

#### 【0054】

XMLベースのソース言語の1つの実施形態では、アプリケーションのエンクロージング・タグは`canvas`要素である。`canvas`要素はビューである。例えば、

40

```
<canvas>
[application]
</canvas>
```

#### 【0055】

以下は、本発明の1つの実施形態と共に使用できる、XMLベースの言語の第1の例のソース・コードである。

50

```

<canvas>
  <view bgcolor= " red " width= " 20 " height= " 20 " id= " refview "
    onclick= " this.animate ( ' width ' , 10, 1000, true) "
  />
  <view bgcolor= " blue " width= " 10 " height= " 20 "
    x= " $immediately{refview.width} " />
  <view bgcolor= " yellow " width= " 10 " height= " 20 "
    x= " $once{refview.width} " />
  <view bgcolor= " grey " width= " 10 " height= " $immediately{20} "
    x= " $always{refview.width} " />
</canvas>

```

10

## 【 0 0 5 6 】

上記のコードは、ユーザ・インターフェースのための4つのオブジェクトを生成する。各オブジェクトはビューである。したがって、コードは4つのビューを示す。第1のビューは、初期においては、幅が20ピクセル、高さが20ピクセルであり、“refview”のIDを有する赤い長方形である。長方形がマウスまたは他のポインティング・デバイスによりクリックされると、幅がアニメ化されて拡大される。第2のビューは、幅が10ピクセル、高さが20ピクセルの青い長方形である。長方形のx座標（アトリビュートx）は、“immediately”に等しいマーカを有する。したがって、エンクロージング要素が具体化されると、表現式“refview.width”が評価される。その表現式は、xを、refviewとして認識されたオブジェクトの幅に等しくなるように定義している。第3のビューは、幅が10ピクセル、高さが20ピクセルの黄色の長方形である。長方形のx位置は、“once”を示すマーカを伴うアトリビュートを有している。したがって、エンクロージング構文が初期化されると、アトリビュートは表現式の値に初期化される。その表現式は、xを、refviewとして認識されたビューの幅に等しくなるように定義されている。第4のオブジェクトは、幅が10ピクセル、高さが20ピクセルの灰色の長方形である。xアトリビュートにより定義されたx座標は“always”に等しいマーカを有している。したがって、refview（赤い長方形）の幅が変化する場合は常に、x座標も変化する。

20

30

## 【 0 0 5 7 】

上記のソース・コードが実行されると、4つのオブジェクトが最初に画面に表示される。赤い長方形が実行される前に表現式が評価されたので、青い長方形のx座標は0となる。黄色の長方形は、x値が赤いウィンドウの初期の幅に等しくなり、その値は20である。灰色の長方形は、初期のx位置が赤いウィンドウの初期の幅に等しくなり、その値は20である。赤いウィンドウがクリックされると、赤いウィンドウはその幅を拡大し、これによって灰色の長方形のx位置が変更される。したがって、赤い長方形の幅が拡大するに当たって、灰色の長方形のx位置は赤い長方形の幅に応じて継続的に変更される。

## 【 0 0 5 8 】

上記の典型的なXMLベースの言語についてのより詳細な情報は、2003年8月15日出願の米国特許出願第10/642,360号“ソフトウェア環境における表現式の評価”に記載されており、その全体を参照することによりここに援用する。上記ソース・コードの例は、本発明を実施するために使用できる多くのタイプのソース・コードとソフトウェア環境のうちの一つにすぎない。他の多くのタイプのソース・コードとソフトウェア環境も使用できる。例えば、本発明は、Java（登録商標）、C++、その他の多くのプログラミング言語を使用することにより実施することができる。本発明は、プレゼンテーション・サーバを使用することにより実施する必要はなく、ネットワーク環境において実施される必要もない。例えば、ネットワークに接続されていないコンピュータ装置を使用して本発明を実施することができる。あるいは、本発明は、ラップトップ型コンピュー

40

50

タ、電話、携帯型コンピュータ装置等のようなモバイル・コンピュータ装置で実施することができる。

【0059】

上述した本発明の詳細な説明は、図解、説明の目的のために提示されたものである。包括的であること、あるいは本発明を開示したとおりの形式に限定することを意図していない。上記の説明を参照して多くの変形及び変更を行うことが可能である。上述した実施形態は、本発明の原理とその実用的なアプリケーションについて最良に説明するために選択されており、これによって当業者は、意図した所定の用途に適するように、種々の実施形態及び種々の変形によって、本発明を最適に利用することができる。本発明の範囲は明細書に添付した特許請求の範囲により規定される。

10

【図面の簡単な説明】

【0060】

【図1】表示された1組のアプリケーションを示している。

【図2】変更後の、図1に示された1組のアプリケーションを示している。

【図3】表示された1組のアプリケーションを示している。

【図4】変更後の、図3に示された1組のアプリケーションを示している。

【図5】変更後の、図3に示された1組のアプリケーションを示している。

【図6】アプリケーションの階層を示している。

【図7】アプリケーションの階層を示している。

【図8】アプリケーションの表示を変更する処理の1つの実施形態を説明するフローチャートを示している。

20

【図9】アプリケーションを開始する処理の1つの実施形態を説明するフローチャートを示している。

【図10A】アプリケーションの変更が他アプリケーションにどのような影響を与えるかを決定する処理の種々の実施形態を説明するフローチャートを示している。

【図10B】アプリケーションの変更が他アプリケーションにどのような影響を与えるかを決定する処理の種々の実施形態を説明するフローチャートを示している。

【図10C】アプリケーションの変更が他アプリケーションにどのような影響を与えるかを決定する処理の種々の実施形態を説明するフローチャートを示している。

【図10D-1】アプリケーションの変更が他アプリケーションにどのような影響を与えるかを決定する処理の種々の実施形態を説明するフローチャートを示している。

30

【図10D-2】アプリケーションの変更が他アプリケーションにどのような影響を与えるかを決定する処理の種々の実施形態を説明するフローチャートを示している。

【図10E】アプリケーションの変更が他アプリケーションにどのような影響を与えるかを決定する処理の種々の実施形態を説明するフローチャートを示している。

【図10F】アプリケーションの変更が他アプリケーションにどのような影響を与えるかを決定する処理の種々の実施形態を説明するフローチャートを示している。

【図10G】アプリケーションの変更が他アプリケーションにどのような影響を与えるかを決定する処理の種々の実施形態を説明するフローチャートを示している。

【図10H】アプリケーションの変更が他アプリケーションにどのような影響を与えるかを決定する処理の種々の実施形態を説明するフローチャートを示している。

40

【図10I】アプリケーションの変更が他アプリケーションにどのような影響を与えるかを決定する処理の種々の実施形態を説明するフローチャートを示している。

【図10J】アプリケーションの変更が他アプリケーションにどのような影響を与えるかを決定する処理の種々の実施形態を説明するフローチャートを示している。

【図11A】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。

【図11B】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。

【図11C】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示

50

している。

【図11D】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。

【図11E】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。

【図11F】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。

【図11G】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。

【図11H】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。 10

【図11I】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。

【図11J】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。

【図11K】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。

【図11L】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。

【図11M】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。 20

【図11N】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。

【図11O】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。

【図11P】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。

【図11Q】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。

【図11R】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。 30

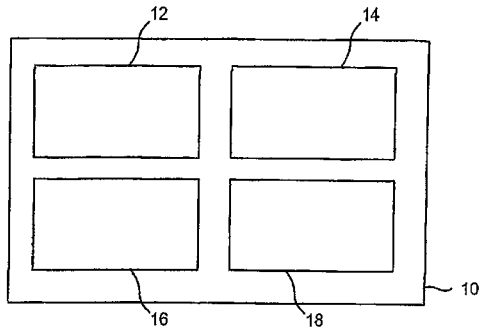
【図11S】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。

【図11T】アプリケーションの変更の前後におけるアプリケーションの種々の表示を示している。

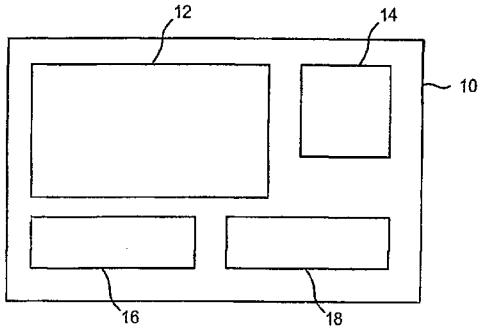
【図12】本発明を実施するために使用可能なシステムの1つの実施形態のブロック図である。

【図13】図12のシステムを動作させる処理の1つの実施形態を示すフローチャートである。

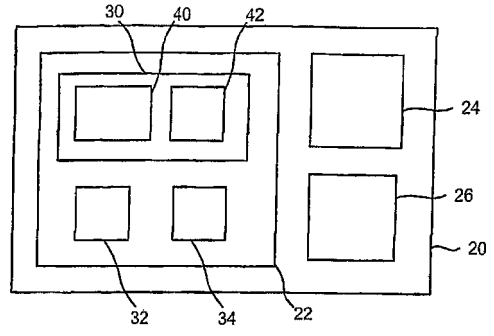
【 図 1 】



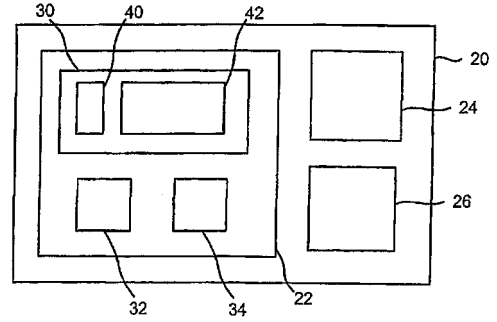
【 図 2 】



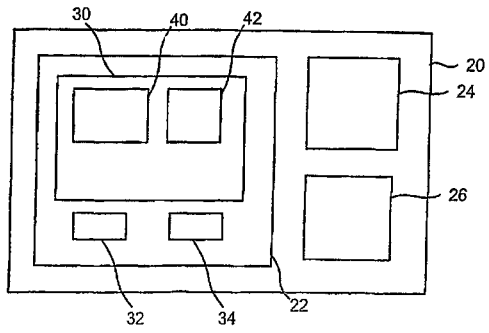
【 図 3 】



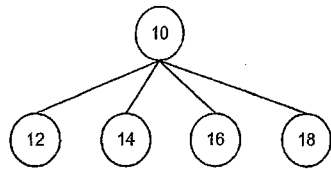
【 図 4 】



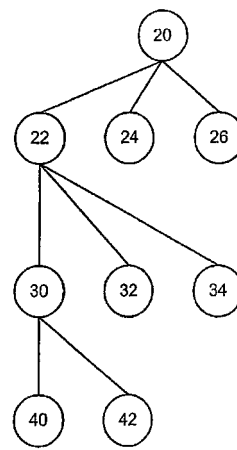
【 図 5 】



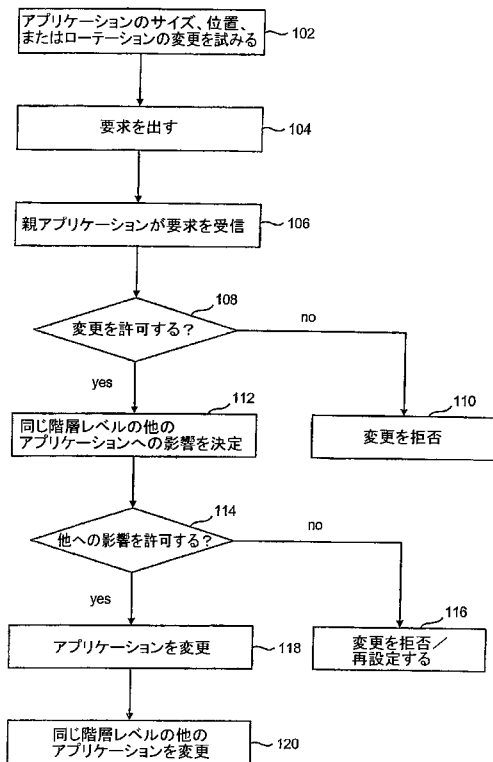
【 図 6 】



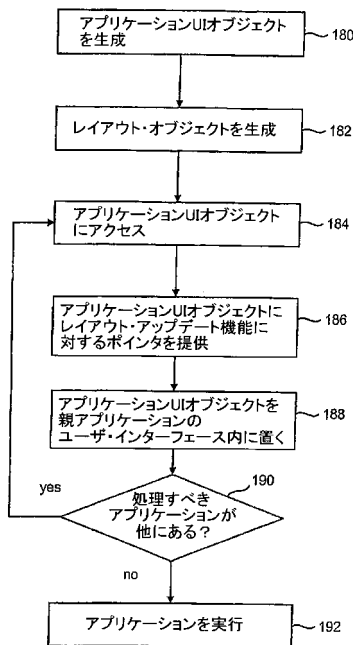
【 図 7 】



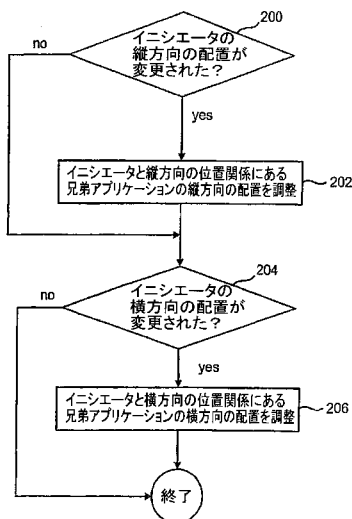
【 図 8 】



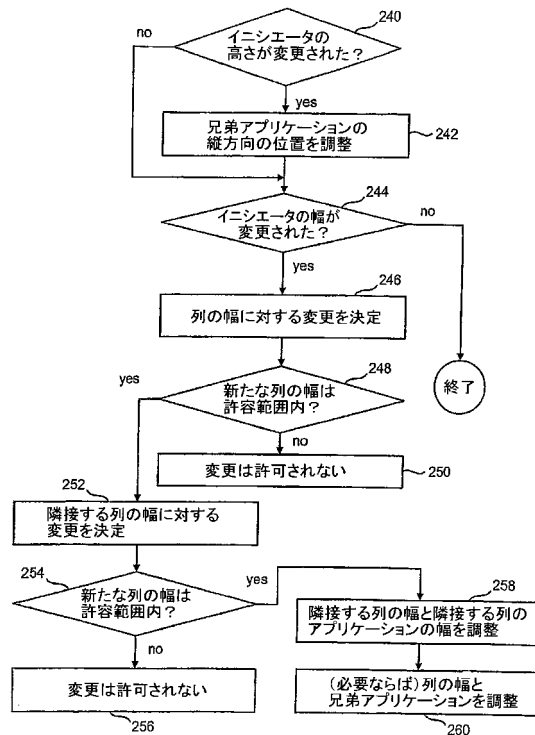
【 図 9 】



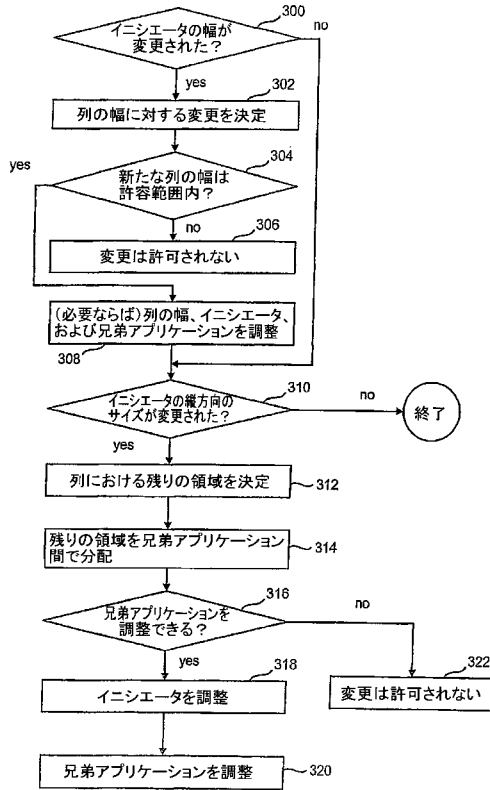
【 図 10 A 】



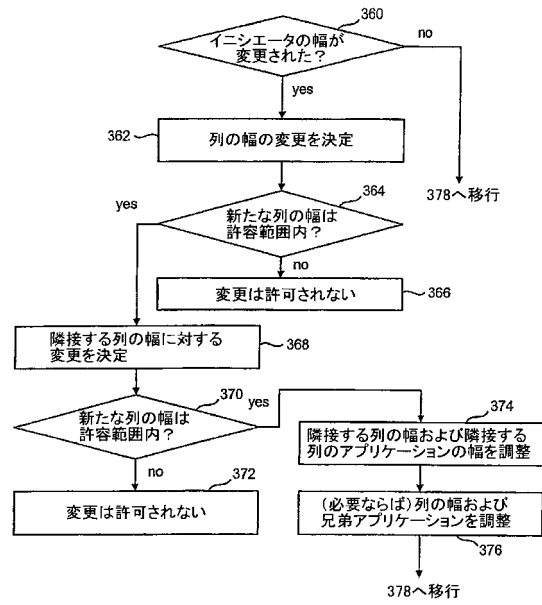
【 図 10 B 】



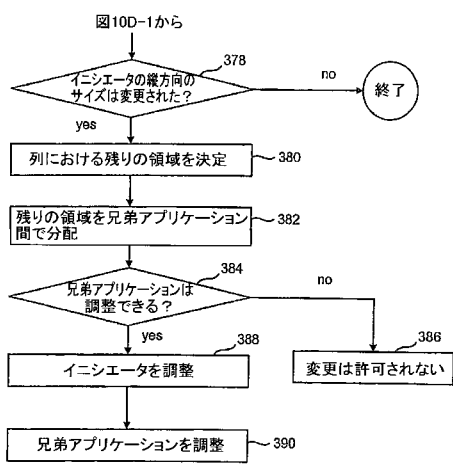
【図10C】



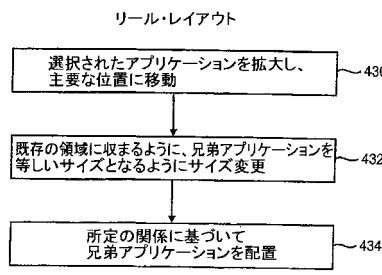
【図10D-1】



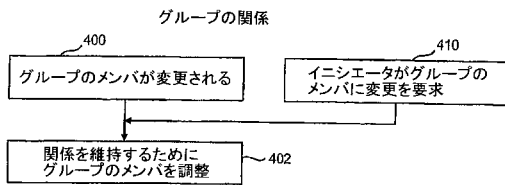
【図10D-2】



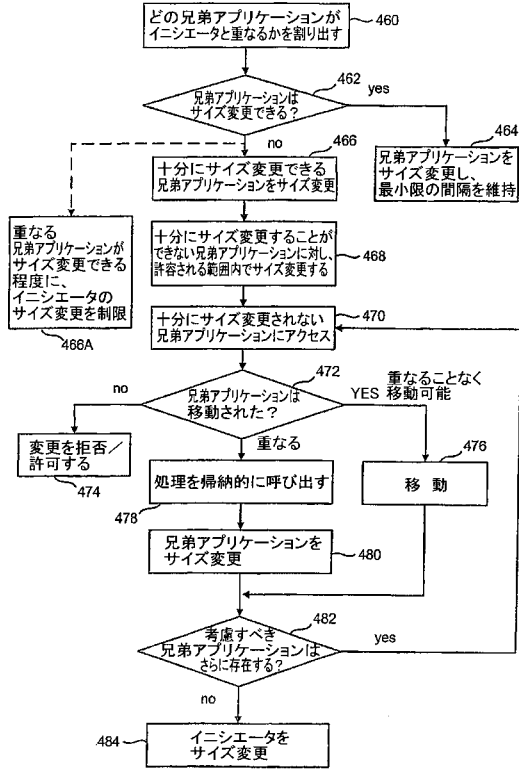
【図10F】



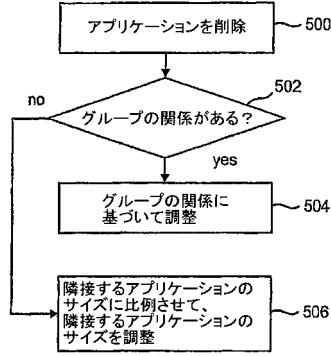
【図10E】



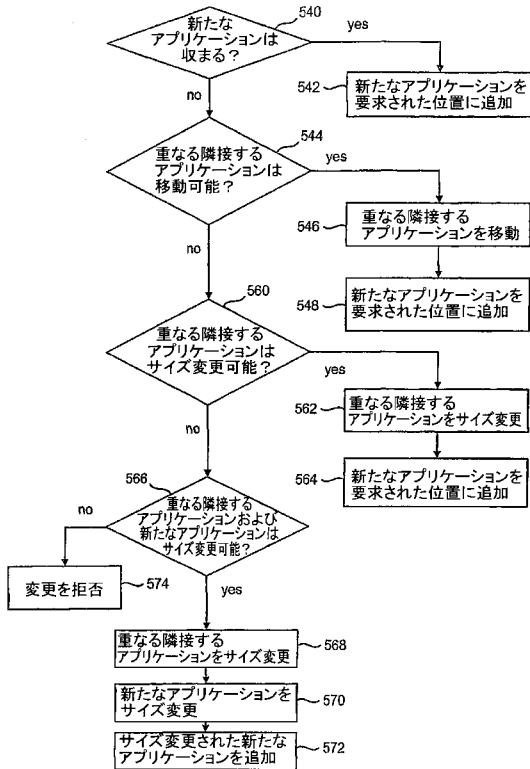
【図10G】



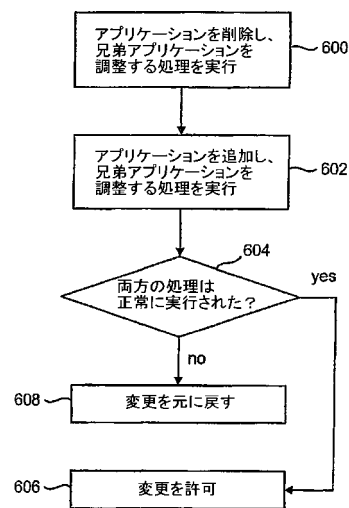
【図10H】



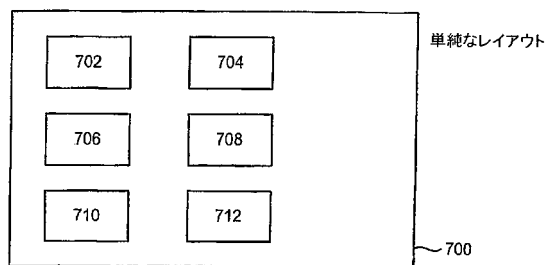
【図10I】



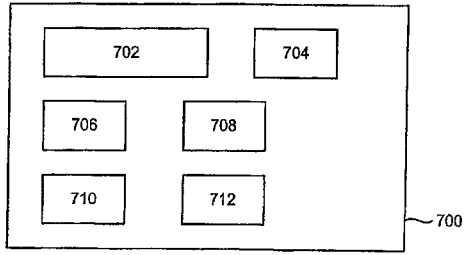
【図10J】



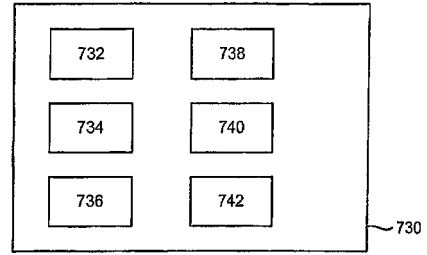
【図11A】



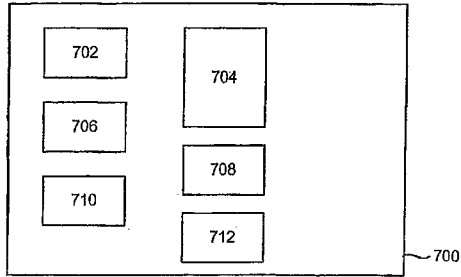
【図 1 1 B】



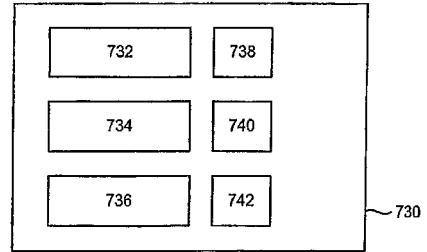
【図 1 1 D】



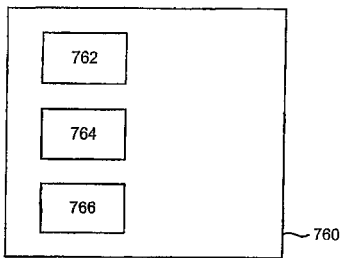
【図 1 1 C】



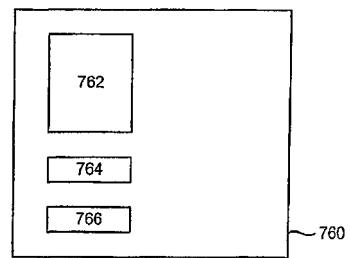
【図 1 1 E】



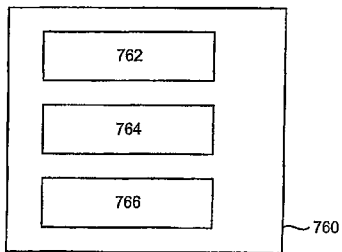
【図 1 1 F】



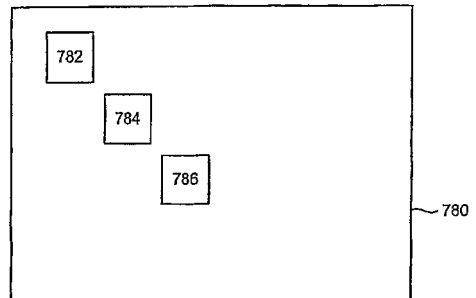
【図 1 1 H】



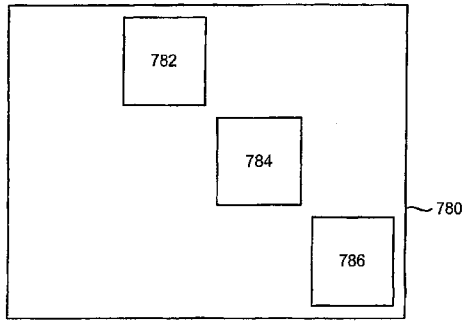
【図 1 1 G】



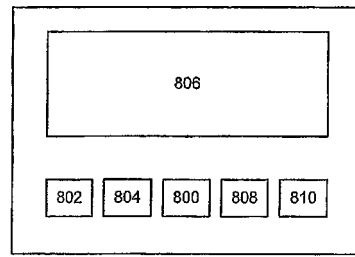
【図 1 1 I】



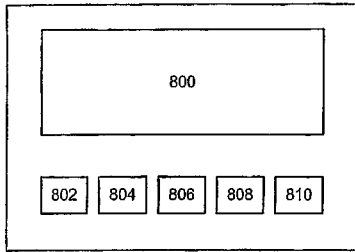
【図 1 1 J】



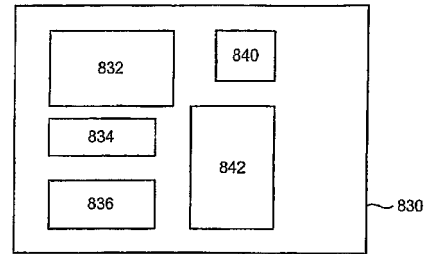
【図 1 1 L】



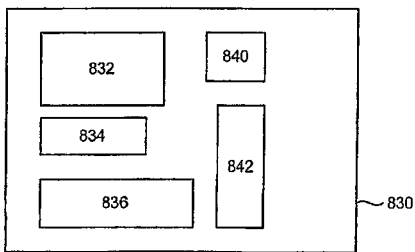
【図 1 1 K】



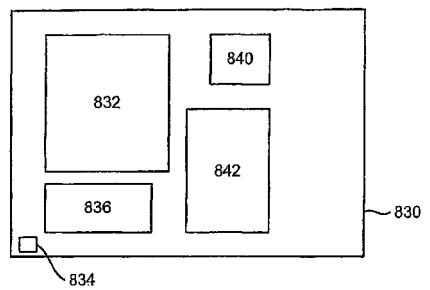
【図 1 1 M】



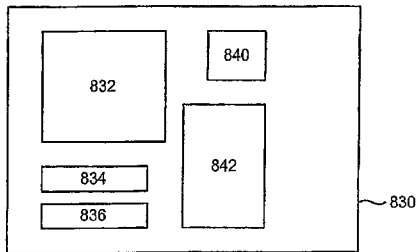
【図 1 1 N】



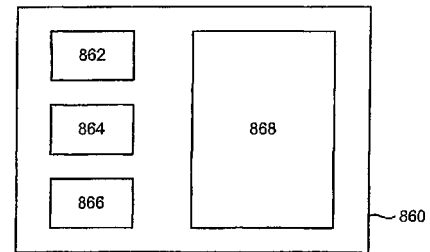
【図 1 1 P】



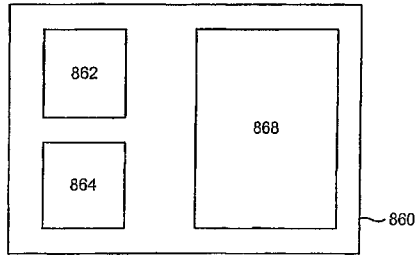
【図 1 1 O】



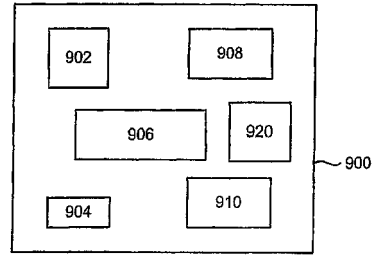
【図 1 1 Q】



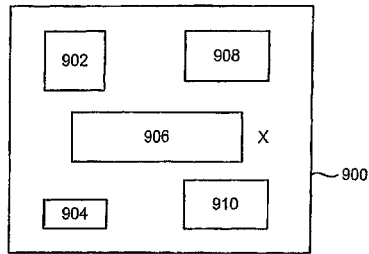
【図11R】



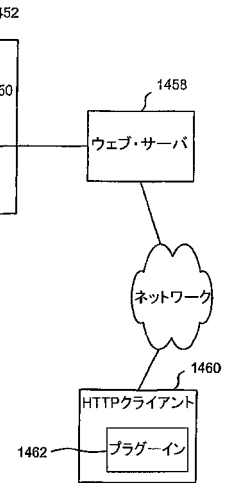
【図11T】



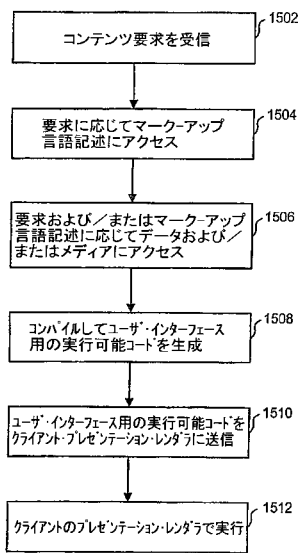
【図11S】



【図12】



【図13】



## 【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US04/38511
<b>A. CLASSIFICATION OF SUBJECT MATTER</b> IPC: G06F 15/00( 2006.01)  USPC: 715/788,781,798,840 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. : 715/788, 781, 798, 840  Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched  Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) position, size, changing display window, application, shrink		
<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	US 6,473,102 B1 (RODDEN et al) 29 October 2002, column 5, lines 3-22.	1-50
A	US, 5,513,342 A (LEONG et al) 30 April 1996.	1-50
<input checked="" 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	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family	
Date of the actual completion of the international search 20 February 2006 (20.02.2006)		Date of mailing of the international search report <b>23 MAR 2006</b>
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 <i>James R. Matthews</i> Cao (Kevin) Nguyen Telephone No. (571)272-4053

Form PCT/ISA/210 (second sheet) (April 2005)

## 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, LU, 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, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW

(72) 発明者 シミスター、ブレット、ジェイ。

アメリカ合衆国、94121、カリフォルニア州、サンフランシスコ、21番 アベニュー 768

(72) 発明者 テムキン、デイビッド、ティー。

アメリカ合衆国、94115、カリフォルニア州、サンフランシスコ、サッター ストリート 2053 アpartment 201

Fターム(参考) 5B042 MC08 NN04 NN11

5E501 AA02 BA03 CA02 FA06 FA47 FB25