

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号
特許第4629792号
(P4629792)

(45) 発行日 平成23年2月9日(2011.2.9)

(24) 登録日 平成22年11月19日(2010.11.19)

(51) Int.Cl.

F I

G O 6 T 11/80 (2006.01)

G O 6 F 3/048 (2006.01)

G O 6 T 11/80 F

G O 6 F 3/048 6 5 4 A

請求項の数 14 (全 25 頁)

(21) 出願番号	特願2009-502785 (P2009-502785)	(73) 特許権者	500046438
(86) (22) 出願日	平成19年2月21日 (2007.2.21)		マイクロソフト コーポレーション
(65) 公表番号	特表2009-532755 (P2009-532755A)		アメリカ合衆国 ワシントン州 9805
(43) 公表日	平成21年9月10日 (2009.9.10)		2-6399 レッドモンド ワン マイ
(86) 国際出願番号	PCT/US2007/004455		クロソフト ウェイ
(87) 国際公開番号	W02007/120388	(74) 代理人	100077481
(87) 国際公開日	平成19年10月25日 (2007.10.25)		弁理士 谷 義一
審査請求日	平成22年1月12日 (2010.1.12)	(74) 代理人	100088915
(31) 優先権主張番号	11/395,624		弁理士 阿部 和夫
(32) 優先日	平成18年3月31日 (2006.3.31)	(72) 発明者	ヨッヒェン ゼーマン
(33) 優先権主張国	米国 (US)		アメリカ合衆国 98052 ワシントン
早期審査対象出願			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 インターナショナル パテンツ内
			最終頁に続く

(54) 【発明の名称】 グラフライクダイアグラムを編集するための2次元ツリー

(57) 【特許請求の範囲】

【請求項 1】

グラフライクビジュアル言語を使用してシステム、サービス、またはプロセスのグラフィカルダイアグラムをモデリングするコンピューティングシステムにおいて、前記グラフィカルダイアグラムを2次元のグラフィカルツリーライク構造の表示を動的にレイアウトすることにより、前記2次元のグラフィカルツリーライク構造を介して前記グラフィカルダイアグラムとのユーザ対話を容易にし、および前記2次元のグラフィカルツリーライク構造の最適な表示を可能にするための方法であって、

前記グラフィカルダイアグラムを、少なくとも1つの分岐を含む2次元のグラフィカルツリーライク構造として前記コンピューティングシステム内のディスプレイデバイスまたはウィンドウに表示することであって、該2次元のグラフィカルツリーライク構造の前記少なくとも1つの分岐は、該2次元のグラフィカルツリーライク構造のルートから始まり、前記グラフライクビジュアル言語を使用してモデリングされた前記グラフィカルダイアグラムにおける前記システム、サービス、またはプロセスのオブジェクト、属性、動作、およびアソシエーションの機能的関係および内部挙動を表す複数のサブ部分を含み、前記複数のサブ部分は、垂直方向および水平方向に変更されるように構成され、各サブ部分は、線で接続されているユーザ対話要素と形状とを含み、各サブ部分は、あるサブ部分の前記ユーザ対話要素が別のサブ部分の前記形状と隣接して表示されるように、別のサブ部分と接続されており、各ユーザ対話要素は、対応するサブ部分を拡張または縮小するために選択可能である、表示することと、

前記 2 次元のグラフィカルツリーライク構造の前記複数のサブ部分のうち、第 2 のサブ部分に接続されている第 1 のサブ部分に対応する第 1 のユーザ対話要素を選択する第 1 のユーザ対話を前記コンピューティングシステムのユーザインタフェースを介して受け取ることであって、前記第 1 のサブ部分は、該選択の前に拡張されていることと、

前記第 1 のユーザ対話に基づいて、前記第 2 のサブ部分を変更することなく、前記第 1 のユーザ対話要素が前記第 2 のサブ部分に対応する第 2 のユーザ対話要素と接続され隣接して表示されるように、前記第 1 のサブ部分の代わりに前記第 1 のユーザ対話要素のみを表示して前記第 1 のサブ部分を動的に縮小することにより、前記 2 次元のグラフィカルツリーライク構造の表示を動的にレイアウトすることと

を含むことを特徴とする方法。

10

【請求項 2】

前記第 2 のサブ部分に対応する前記第 2 のユーザ対話要素を選択する第 2 のユーザ対話を、前記ユーザインタフェースを介して受け取ることと、

前記第 2 のユーザ対話に基づいて、前記第 2 のサブ部分を動的に縮小することと、

前記第 2 のサブ部分を縮小した後に、前記第 2 のサブ部分が隠れるように前記第 1 のサブ部分を動的に拡張することにより、前記 2 次元のグラフィカルツリーライク構造の表示を動的にレイアウトすることと

をさらに含み、前記第 1 のサブ部分に隠された前記第 2 のサブ部分は、該第 2 のサブ部分に対応する前記第 2 のユーザ対話要素を選択する追加のユーザ対話を受け取ることにより再び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 3】

新しいサブ部分を前記 2 次元のグラフィカルツリーライク構造の所望の位置に位置決めすることを示すユーザ対話を、前記ユーザインタフェースを介して受け取ると、前記新しいサブ部分を、前記 2 次元のグラフィカルツリーライク構造に自動的に追加することとを特徴とする請求項 1 に記載の方法。

【請求項 4】

前記複数のサブ部分のうちの 1 つまたは複数のサブ部分を水平方位から垂直方位または垂直方位から水平方位に切り替えることによって前記 2 次元のグラフィカルツリーライク構造の表示を動的にレイアウトすることをさらに含み、前記 2 次元のグラフィカルツリーライク構造は、該 1 つまたは複数のサブ部分を水平方位から垂直方位または垂直方位から水平方位に切り替えによって前記ディスプレイデバイスまたは前記ウィンドウのアスペクト比に関して最適に表示されること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 5】

前記複数のサブ部分のうち、各々同じサブ部分または前記ルートに接続されているサブ部分は、マルチコネクタサブ部分にマージされ、前記マルチコネクタサブ部分は、

各サブ部分が接続されている前記同じサブ部分または前記ルートに隣接して表示される共通のユーザ対話要素であって、前記マルチコネクタサブ部分全体を拡張または縮小するのに選択可能な共通のユーザ対話要素と、

前記共通のユーザ対話要素から伸びるマルチコネクタ線と、

前記同じサブ部分または前記ルートに接続されている各サブ部分に対応する特定のユーザ対話要素であって、それぞれ、対応する各サブ部分の前記形状に隣接して表示され、特定の線で前記マルチコネクタ線に接続され、対応する各サブ部分を拡張または縮小するために選択可能な特定のユーザ対話要素と

40

を含むことを特徴とする請求項 1 に記載の方法。

【請求項 6】

請求項 1 に記載の方法をコンピュータに実行させることを特徴とするプログラム。

【請求項 7】

グラフィカルビジュアル言語を使用してシステム、サービス、またはプロセスのグラフィカルダイアグラムをモデリングするコンピューティングシステムにおいて、前記グラフィカルダイアグラムの 2 次元のグラフィカルツリーライク構造の表示を動的にレイアウト

50

することにより、前記 2 次元のグラフィカルツリーライク構造を介して前記グラフィカルダイアグラムとのユーザ対話を容易にし、および前記 2 次元のグラフィカルツリーライク構造の最適な表示を可能にするための方法であって、

前記グラフィカルダイアグラムを、少なくとも 1 つの分岐を含む 2 次元のグラフィカルツリーライク構造として前記コンピューティングシステムのディスプレイデバイスまたはウィンドウに表示することであって、前記 2 次元のグラフィカルツリーライク構造の前記少なくとも 1 つの分岐は、該 2 次元のグラフィカルツリーライク構造のルートから始まり、前記グラフィカルダイアグラムにおいてモデリングされた前記システム、サービス、またはプロセスの複数のオブジェクト、属性、動作、およびアソシエーションの機能的関係および内部挙動を表す複数のサブ部分を含み、前記複数のサブ部分は、少なくとも 1 つの端部分を含み、各サブ部分は、線で接続されているユーザ対話要素と形状とを含み、各サブ部分は、あるサブ部分の前記ユーザ対話要素が別のサブ部分の前記形状と隣接して表示されるように、別のサブ部分と接続されており、各ユーザ対話要素は、対応するサブ部分を拡張または縮小するために選択可能である、表示することと、

前記複数のサブ部分のうち前記少なくとも 1 つの端部分以外の第 1 のサブ部分に対応する第 1 のユーザ対話要素を選択するユーザ対話を受け取ることであって、前記第 1 のサブ部分は、該選択の前に拡張されており、前記第 1 のサブ部分は、第 2 のサブ部分に接続されている、受け取ることと、

前記ユーザ対話に基づいて、前記少なくとも 1 つの端部分を変更せずに、前記第 1 のユーザ対話要素が前記第 2 のサブ部分に対応する第 2 のユーザ対話要素と接続され隣接して表示されるように、前記第 1 のサブ部分の代わりに前記第 1 のユーザ対話要素のみを表示することによって前記第 1 のサブ部分を動的に縮小することにより、前記 2 次元のグラフィカルツリーライク構造の表示を動的にレイアウトすることと

を含むことを特徴とする方法。

【請求項 8】

前記第 2 のサブ部分は、端部分であることを特徴とする請求項 7 に記載の方法。

【請求項 9】

前記 2 次元のグラフィカルツリーライク構造をどのように変更して表示すべきかを指定する、予め定義された 1 つまたは複数のルールを受け取ることと、

前記 1 つまたは複数のルールおよび前記受け取られたユーザ対話に基づいて、前記第 1 のサブ部分を動的に縮小することにより、前記 2 次元のグラフィカルツリーライク構造の表示を動的にレイアウトすることと

をさらに含むことを特徴とする請求項 7 に記載の方法。

【請求項 10】

前記第 1 のサブ部分が縮小されている間に前記第 1 のサブ部分の前記第 1 のユーザ対話要素を選択するユーザ対話を受け取ることと、

前記第 1 のサブ部分を拡張することにより、前記 2 次元のグラフィカルツリーライク構造の表示を動的にレイアウトすることと

をさらに含むことを特徴とする請求項 7 に記載の方法。

【請求項 11】

請求項 7 に記載の方法をコンピュータに実行させることを特徴とするプログラム。

【請求項 12】

システム、サービス、またはプロセスを表すグラフィカルダイアグラムを、グラフィカルビジュアル言語を使用してモデリングし、2 次元のビジュアルツリーライク構造として表示するコンピューティングシステム内において、前記 2 次元のビジュアルツリーライク構造の表示および変更を使用されるアプリケーションが終了された場合であっても前記 2 次元のビジュアルツリーライク構造の表示に対して行われた変更が維持されるように、前記ビジュアルツリーライク構造の 1 つまたは複数のサブ分岐に対する変更を前記コンピューティングシステムの永続メモリ内で永続的に保持するための方法であって、

前記アプリケーションをアクチベートして、少なくとも 1 つの分岐を含む前記 2 次元の

10

20

30

40

50

グラフィカルツリーライク構造を前記コンピューティングシステムのディスプレイデバイスまたはウィンドウに表示することであって、前記少なくとも1つの分岐は、前記2次元のグラフィカルツリーライク構造のルートから始まり、前記グラフィカルダイアグラムにおいてモデリングされた前記システム、サービス、もしくはプロセスの複数オブジェクト、属性、動作、およびアソシエーションの機能的関係および内部挙動に関して全体的なオブジェクトモデルを表す複数のサブ分岐を含み、各サブ分岐は、線で接続されているユーザ対話要素と形状とを含み、各サブ分岐は、あるサブ分岐の前記ユーザ対話要素が別のサブ分岐の前記形状と隣接して表示されるように、別のサブ分岐と接続されており、各ユーザ対話要素は、対応するサブ分岐を拡張または縮小するために選択可能である、アクチベートすることと、

10

前記複数のサブ分岐のうち、第2のサブ分岐と接続されている第1のサブ分岐に対応する第1のユーザ対話要素を選択する第1のユーザ入力を、前記コンピュータシステムのユーザインタフェースを介して受け取ることと、

前記第1のユーザ入力に基づいて前記第1のサブ分岐を変更することであって、前記第1のユーザ入力が前記第1のサブ部分の縮小を選択しているとき、前記第1のサブ分岐は、前記第2のサブ分岐に対応する第2のユーザ対話要素に接続され隣接して表示されるように、前記第1のサブ分岐の代わりに前記第1のサブ分岐に対応する第1のユーザ対話要素のみを表示して前記第1のサブ分岐を動的に縮小することによって変更されることと、

前記ユーザインタフェースを介して第2のユーザ入力を受け取ることと、

前記第2のユーザ入力に基づいて、前記アプリケーションが終了された後、再アクチベートされるときに前記2次元のグラフィカルツリーライク構造が前記アプリケーションの終了前と同じように表示されるように、前記第1のユーザ入力に基づいて縮小された状態の前記第1のサブ分岐を前記永続メモリ内で維持することと

20

を含むことを特徴とする方法。

【請求項13】

前記ユーザインタフェースを介して前記第2のサブ分岐の縮小を選択するユーザ入力を受け取ることと、

前記第2のサブ分岐が縮小されると前記第1のサブ分岐が隠されるように、前記第1のサブ分岐を拡張することに先立って前記第2のサブ分岐を縮小することと、

後で前記アプリケーションが再アクチベートされて追加のユーザ入力の受け取ったときに該追加のユーザ入力に基づいて前記第2のサブ分岐を拡張することができるように、前記アプリケーションを終了する際に前記第1のサブ分岐を隠している縮小された状態の前記第2のサブ分岐を前記永続メモリに保存することと

30

をさらに含むことを特徴とする請求項12に記載の方法。

【請求項14】

請求項12に記載の方法をコンピュータに実行させることを特徴とするプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、システム、サービス、もしくはプロセスなどに関するまたは全体的なオブジェクトシステムを表す、通常は手動の変更および再スケーリングを必要とする、グラフィカルダイアグラムをモデリングするのにグラフィックビジュアル言語を使用するコンピューティングシステムに関する。

40

【背景技術】

【0002】

コンピューティングシステムは、我々が働き、遊ぶ形を変革してきた。コンピューティングシステムは、ラップトップコンピュータ、デスクトップコンピュータ、携帯情報端末、電話機、ならびにたとえば冷蔵庫および自動車など、従来はコンピューティングシステムに関連しなかったデバイスさえ含むさまざまな形で現れる。コンピューティングシステムは、ネットワークを介して相互接続された複数の構成要素コンピューティングシステム

50

さえ含むものとしてすることができる。したがって、コンピューティングシステムの中には、掌におさまるほどに小さい物と、地球の大半に広がる物とがある。

【 0 0 0 3 】

ますます人気が高まってきたコンピューティングテクノロジーの一領域が、グラフィカルモデリングであり、これは、ユーザが、多数のシステム、サービス、またはプロセスのフローダイアグラムなどのグラフィックダイアグラムをビジュアルに表示することを可能にする。ダイアグラムの例には、ワークフローダイアグラム、データフローダイアグラム、E - Rダイアグラム、ビジネスプロセスダイアグラムなどが含まれる。より具体的には、そのようなグラフィック（図形状、グラフ様、graph - like）ダイアグラムは、ユーザが、さまざまなコンピューティングシステムもしくは他のシステム、ビジネスモデルもしくはプロジェクトモデルなどの機能性、挙動の諸態様、およびさまざまなコンポーネントまたは要素の関係をビジュアルに通信することを可能にし、したがって、しばしば複雑なシステムまたはプロセスの効率的な通信を可能にする。

10

【 0 0 0 4 】

たとえば、グラフィックダイアグラムは、販売トランザクションのフローをビジュアライズすることができる。フローダイアグラムの要素およびコンポーネントは、所望の製品の選択、発注、配送の手配および支払いなどのイベントを表すことができる。このダイアグラムは、上でリストしたタスクを実行するエンティティをビジュアライズすることもできる。多数のエンティティがかかわる非常に複雑な販売トランザクションについて、トランザクションをグラフィカルに表示することは、他の形より理解するのが簡単である場合がある。

20

【 発明の開示 】

【 発明が解決しようとする課題 】

【 0 0 0 5 】

グラフィックダイアグラムは、システムおよびプロセスをビジュアライズするのに有用であるが、その有用性に対する制限がある。たとえば、ダイアグラムが大きく複雑になるにつれて、ディスプレイデバイスのサイズ、解像度、および他の能力に起因して、コンポーネントおよび要素のすべてを表示するのに十分な余地が単純になる場合がある。言い換えると、使用可能なディスプレイスペースおよび機能が、一時に表示できるコンポーネントおよびオブジェクトの個数を制限する。これは、ユーザがプロセスの始めと終りとの両方を表示することを望むダイアグラムについて特に問題である。

30

【 0 0 0 6 】

1つの解決策は、より大きいディスプレイデバイスを使用することである。しかし、これは、これに伴うコストのゆえに、ほとんどのユーザにとって非実用的である。さらに、ダイアグラムが、どの実用的なディスプレイデバイスでもそのダイアグラムの要素のすべてを表示できないほどに大きくなる場合がある。

【 0 0 0 7 】

一般的に使用されている、このディスプレイ制限に対するもう1つの解決策は、ユーザに手動でダイアグラムを再スケーリングさせることである。たとえば、ユーザは、しばしば、新しい要素がダイアグラムに追加されるどの時にも、「ズームアウト」する。この解決策は、少数の要素だけを伴うダイアグラムについては有効である可能性があるが、より大きいダイアグラムについてはうまく働かない。ダイアグラムは、しばしば、複数回ズームアウトされた後に判読しにくくなる。

40

【 0 0 0 8 】

もう1つの一般的な解決策は、一緒に、判読できるサイズの必要なダイアグラム要素のすべてを含む複数のダイアグラムを作ることである。しかし、これは、しばしば、時間がかかる物になる可能性がある。さらに、ダイアグラムのフローまたは「メンタルマップ」は、しばしば、複数のダイアグラムを使用することによって失われ、これは、ユーザをいらいらさせる可能性がある。

【 0 0 0 9 】

50

もう1つの解決策は、判読できる形でダイアグラムをスケーリングすることを試みてダイアグラムの要素を自動的にレイアウトするツールを作成することであった。しかし、そのようなツールは、複雑なアルゴリズムを実施し、その計算的複雑さは、これらを、処理するのが非常に遅い物にする。新しい要素がダイアグラムに追加されるどの時にも、これらのツールは、そのアルゴリズムを再処理し、新しいダイアグラムを作成しなければならず、そのようなツールが、対話型エディタに非実用的な物になる。さらに、これらのアルゴリズムのほとんどは、新しい要素が追加される場合に、ダイアグラムのすべての要素を再配置する。その結果、ユーザは、しばしば、そのダイアグラムの「メンタルマップ」を失い、これは、迷惑である可能性がある。

【課題を解決するための手段】

10

【0010】

上で識別された現在のダイアグラム生成方法およびダイアグラム表示方法の欠陥および短所は、本明細書で開示される例の実施形態を介して克服される。ここで述べる概要が、以降の「発明を実施するための最良の形態」でさらに説明される概念にしたがって選択される物を単純化した形で紹介するために提供されることに留意されたい。この概要は、特許請求される主題の主要な特徴または本質的特徴を識別することを意図された物ではなく、特許請求される主題の範囲を判定する際の助けとして使用されることを意図された物でもない。

【0011】

1つの例の実施形態は、ユーザ対話の容易さおよび最適ディスプレイを可能にするために、グラフィックビジュアル言語に基づくグラフィカルダイアグラムを2次元ツリーライク(ツリー状、tree-like)構造に動的にレイアウトすることを提供する。この実施形態では、グラフィカルツリーライク構造が表示される。グラフィカルツリーライク構造は、グラフィカルツリーライク構造は少なくとも1つの分岐を含む。この分岐はグラフィカルツリーライク構造をルートからスタートさせ、分岐はまた、垂直方向および水平方向で変更されるように構成される1つまたは複数のサブ部分を含む。サブ部分は、1つまたは複数のオブジェクト、属性、動作、およびアソシエーションの機能的関係および内部挙動に関して全体的なオブジェクトモデルを表す。次に、サブ部分のうちの1つまたは複数を変更するユーザ入力を受け取る。ユーザ入力に基づいて、1つまたは複数のサブ部分が、ユーザ対話の容易さ、グラフィカルツリーライク構造のディスプレイの最適化、またはその両方のために、垂直に、水平に、またはその両方で動的に変更される。

20

30

【0012】

もう1つの例の実施形態も、ユーザ対話の容易さおよび最適ディスプレイを可能にするために、グラフィックビジュアル言語に基づくフローダイアグラムを2次元ツリーライク(2次元ツリー状の)構造に動的にレイアウトすることを提供する。この実施形態では、グラフィカルツリーライク(図形形状のツリー)構造が表示される。グラフィカルツリーライク構造は、ツリーライク構造のルートから始まり、複数のサブ部分を含む、少なくとも1つの分岐を含む。サブ部分のうちの少なくとも1つは、端部分でもある。サブ部分は、1つまたは複数のオブジェクト、属性、動作、およびアソシエーションの機能的関係および内部挙動に関して全体的なオブジェクトモデルを表す。次に、端部分以外のサブ部分のうちの1つまたは複数を変更するユーザ入力を受け取る。ユーザ入力に基づいて、端部分以外の1つまたは複数のサブ部分が、端部分をも変更することなくスケーリングされ、拡張され、または縮小される。この変更は、ユーザ対話の容易さ、グラフィカルツリーライク構造のディスプレイの最適化、またはその両方を可能にする。

40

【0013】

さらなる例の実施形態は、グラフィカルツリーライク構造を表示するアプリケーションが終了される場合にグラフィカルツリーライク構造の1つまたは複数のサブ分岐に対して行われた変更をメモリ内に永続的に保持することを提供する。この実施形態では、グラフィカルツリーライク構造を表示するアプリケーションがアクチベートされる。グラフィカルツリーライク構造は、ツリーライク構造のルートから始まり、1つまたは複数のオブジ

50

ェクト、属性、動作、およびアソシエーションの機能的関係および内部挙動に関してシステム、サービス、またはプロセスをモデリングする1つまたは複数のサブ分岐を含む、少なくとも1つの分岐を含む。少なくとも1つのサブ分岐を変更するユーザ入力を受け取られる。このユーザ入力に基づいて、変更が永続メモリ内で維持される。ユーザがアプリケーションを終了した後で後ほどアプリケーションを再アクチベートする際に、グラフィカルツリーライク構造が、そのアプリケーションが終了される前に見えていたとおりに表示される。

【0014】

本発明のさらなる特徴および利点については、以降の説明で示され、部分的にはその説明から明白になり、あるいは、本発明を实践することによって知ることができる。本発明の特徴および利点は、添付の特許請求の範囲で具体的に指摘される手段および組合せによって実現し、入手することができる。本発明の上記および他の特徴は、以降の説明および添付の特許請求の範囲からさらに十分に明白になり、あるいは、以降で示されるように本発明を实践することによって知ることができる。

10

【0015】

本発明の上で列举された有利な特徴および他の有利な特徴を入手できる形を説明するために、短く上述した本発明のより具体的な説明を、添付図面に示された本発明の特定の実施形態に言及することによって与える。これらの図面が、本発明の通常の実施形態を示すのみであり、したがって本発明の範囲の限定と考えられてはならないことを理解して、本発明を、添付図面の使用を介してさらに具体的および詳細に記述し、説明する。

20

【発明を実施するための最良の形態】

【0016】

本発明は、ユーザ対話の容易さおよび最適ディスプレイを可能にするためにグラフィックビジュアル言語を2次元ツリーライク構造に動的にレイアウトする方法、システム、およびコンピュータプログラム製品にわたる。本発明の実施形態は、下で詳細に説明するように、さまざまなコンピュータハードウェアまたはモジュールを含む、特殊目的コンピュータまたは汎用コンピュータを含むことができる。

【0017】

例の実施形態は、動的な対話および最適なディスプレイを可能にする形で、グラフィックビジュアル言語をレイアウトする現在のツールのさまざまな欠点を克服するように構成された複数のメカニズムを提供する。たとえば、本明細書の一実施形態は、2次元ツリーライク構造内でフローダイアグラムを動的にレイアウトするメカニズムを提供する。グラフィックビジュアル言語は、2次元グラフィカルツリーライク構造として表示される。グラフィカルツリーライク構造は、ツリーライク構造のルートから始まり、サブ部分を含む、少なくとも1つの分岐を含む。サブ部分は、ツリーライク構造の1つまたは複数のオブジェクト、属性、動作、またはアソシエーションの機能的関係および内部挙動に関して全体的なオブジェクトモデルを表すのに使用される。サブ部分は、少なくとも1つの端部分をも含むことができるが、これは、すべての実施形態に必要なわけではない。

30

【0018】

ツリーライク構造は、ユーザ対話を実施する形で表示される。たとえば、サブ部分の変更を制御するユーザ対話(UI)要素を表示することができる。端部分ではない1つまたは複数のサブ部分を変更する、UI要素の選択などのユーザ対話を受け取られる。この変更は、サブ部分のスケールリング、縮小、および拡張を含む。

40

【0019】

ユーザ対話に基づいて、2次元ツリーライク構造が動的に変更される。たとえば、いくつかの実施形態で、端部分以外のサブ部分をスケールリングして、使用可能ディスプレイスペース内でのツリーライク構造のディスプレイを最適化することができ、あるいは、サブ部分を縮小または拡張して、ツリーライク構造の大きさに関わりなくツリーライク構造の所望の部分のビューイングを可能にすることができる。他の実施形態では、サブ部分を水平に、垂直に、またはその両方で変更することができる。

50

【 0 0 2 0 】

他の実施形態は、ツリーライク構造を実行するアプリケーションが終了される場合であってもツリーライク構造に対して行われた変更の永続化を可能にする。たとえば、グラフィカルツリーライク構造を表示し、前の例の形でグラフィカルツリーライク構造を変更するユーザ入力を受け取るアプリケーションが、アクチベートされる。しかし、ユーザ入力に基づいて、ツリーライク構造に対する変更が、永続メモリ内で維持される。さらに、他のサブ部分に対する後続の変更によって隠されるようになる、あるサブ部分に対する変更をも、永続メモリ内で維持することができる。ユーザがそのアプリケーションを終了した後のある時の、そのアプリケーションの再アクチベートの際に、グラフィカルツリーライク構造が、そのアプリケーションが終了される前に見えていたとおりに表示される。可視と隠れとの両方の変更の維持は、アプリケーションが終了され、再始動される場合に常に変更を元に戻すことを必要とせずに、長期にわたってグラフィカルツリーライク構造を表示することを可能にする。さらに、隠れ状態の維持は、アプリケーションが終了される場合にどのサブ部分も失われないことを保証する。

10

【 0 0 2 1 】

さらなる実施形態は、変更できないグラフィックビジュアル言語およびツリーライク構造の、動的に変更できる２次元ツリーライク構造への変換を可能にする。これらの実施形態は、ツリーライク構造のルートから始まり、システム、サービス、およびプロセスをモデリングする複数のサブ部分を含む１つの分岐を含み、端部分を含む、１次元ツリーライク構造の変換を可能にする。この１次元ツリーライク構造のサブ部分は、端部分をも変更しなければ変更することができず、これは、端部分をも変更することなくサブ部分を変更できる２次元ツリーライク構造と対照的である。これらの構造を２次元ツリーライク構造に変換することによって、ユーザに、２次元ツリーライク構造の動的な変更およびディスプレイ最適化を利用する能力が与えられる。さらに、これは、サブ部分の方位を必要に応じて垂直と水平との間で自動的に切り替えることによる、使用可能なディスプレイスクリーンまたはウィンドウのアスペクト比への自動最適化を可能にする。

20

【 0 0 2 2 】

有利な特徴へのより具体的な参照を、下で図面に関してより詳細に記述するが、本発明の範囲内の実施形態は、コンピュータ実行可能命令またはデータ構造を担持するかその上に格納されたコンピュータ可読媒体をも含む。そのようなコンピュータ可読媒体は、汎用コンピュータまたは特殊目的コンピュータによってアクセスできるすべての使用可能な媒体とすることができる。限定ではなく例として、そのようなコンピュータ可読媒体には、RAM、ROM、EEPROM、CD-ROMもしくは他の光ディスクストレージ、磁気ディスクストレージもしくは他の磁気ストレージデバイス、またはコンピュータ実行可能命令もしくはデータ構造の形で所望のプログラムコード手段を担持するか格納するのに使用でき、汎用コンピュータまたは特殊目的コンピュータによってアクセスできるすべての他の媒体を含めることができる。情報がネットワークまたは別の通信接続（ハードワイヤド、無線、またはハードワイヤドもしくは無線の組合せのいずれか）を介してコンピュータに転送され、または供給される時に、そのコンピュータは、その接続をコンピュータ可読媒体として正しくみなす。したがって、すべてのそのような接続が、正しくコンピュータ可読媒体と呼ばれる。上記の組合せも、コンピュータ可読媒体の範囲に含まれない。

30

40

【 0 0 2 3 】

コンピュータ実行可能命令は、たとえば、汎用コンピュータ、特殊目的コンピュータ、または特殊目的処理デバイスにある種の機能または機能のグループを実行させる命令およびデータを含む。本主題を、構造的特徴および／または方法論的行為に固有の言葉で説明してきたが、添付の特許請求の範囲で定義される本主題が、必ずしも上で説明した特定の特徴または行為に限定されないことを理解されたい。そうではなく、上述の特定の特徴または行為は、特許請求の範囲を実施する例の形態として開示された物である。

【 0 0 2 4 】

50

本明細書で使用される際に、用語「モジュール」または「コンポーネント」は、コンピューティングシステム上で実行されるソフトウェアオブジェクトまたはルーチンを指す。本明細書で説明されるさまざまなコンポーネント、モジュール、エンジン、およびサービスは、コンピューティングシステム上で実行されるオブジェクトまたはプロセスとして（たとえば、別々のスレッドとして）実施することができる。本明細書で説明されるシステムおよび方法は、好ましくはソフトウェアで実施されるが、ハードウェアまたはソフトウェアおよびハードウェアの組合せでの実施も、可能であり、企図されている。この説明では、「コンピューティングエンティティ」は、本明細書で前に定義した任意のコンピューティングシステムまたはコンピューティングシステム上で走行する任意のモジュールもしくは変調の組合せとすることができる。

10

【0025】

ここで図面を参照すると、図1は、本明細書で開示されるさまざまな実施形態を実施するのに使用できるコンピューティングシステム100を示す。コンピューティングシステム100は、本実施形態を実施する場合に使用できるさまざまなモジュールおよびコンポーネントを示す。コンピューティングシステム100が、例としてのみ図示されているのであって、これを添付の特許請求の範囲または本明細書で開示されるすべての実施形態の範囲を限定するのに使用してはならないことに留意されたい。言い換えると、本明細書で説明される実施形態を実施するのに使用できる多数の他のコンピューティングシステムおよびコンポーネント構成がある。

【0026】

20

コンピューティングシステム100は、グラフィックビジュアル言語110を含み、グラフィックビジュアル言語110は、システム、サービス、およびプロセスのフローダイアグラムをモデリングするのに使用することができる。グラフィックビジュアル言語の一般的な例は、モデリングされるシステム、サービス、またはプロセス内に通常はどのような物があるのかを強調する構造ダイアグラム、モデリングされるシステム、サービス、またはプロセス内で一般に発生する物を強調する挙動ダイアグラム、およびモデリングされるシステム、サービス、またはプロセス内の物の間での制御およびデータのフローを強調する対話(interaction)ダイアグラムを含む。他の一般的なグラフィックビジュアル言語は、Unified Modeling Language (UML)のさまざまなダイアグラムタイプ、データベースの世界のエンティティリレーションシップダイアグラム、およびビジネスプロセスデザイン用のビジネスプロセスダイアグラムを含む。一般に、本明細書で説明される実施形態は、任意の汎用グラフィックビジュアル言語とすることができるグラフィックビジュアル言語110に適用される。したがって、本明細書での、特定のグラフィックビジュアル言語へのすべての言及は、例示のみのためであり、これを本明細書で開示される実施形態の範囲を限定するのに使用してはならない。

30

【0027】

コンピューティングシステム100は、2次元ツリーライク構造190としてのグラフィックビジュアル言語110の表示を実行するデザインツール/アプリケーション120をも含む。デザインツール/アプリケーション120は、表示される2次元ツリーライク構造190の動的な変更をも実行する。デザインツール/アプリケーション120とそのさまざまなモジュールおよび/またはコンポーネントのいずれかを、ソフトウェア、ハードウェア、またはこの2つの任意の組合せとして実施することができ、単一のローカルコンピューティングデバイス上または単一のリモートコンピューティングデバイス上あるいは分散コンピューティング環境内でホスティングすることができる。

40

【0028】

デザインツール120は、グラフィックビジュアル言語110を2次元ツリーライク構造190として表示するレイアウト/プレゼンテーションモジュール130を含む。いくつかの実施形態で、レイアウト/プレゼンテーションモジュール130は、ユーザによってデザインツール/アプリケーション120に入力される要素を表示する。デザインツール/アプリケーション120は、その後、実施されるグラフィックビジュアル言語110

50

によって定義される関係に従って、相互接続する線を要素の間に自動的に描く。

【0029】

他の実施形態では、レイアウトモジュール130は、グラフィックビジュアル言語110を受け取るか、グラフィックビジュアル言語110が受け取られる時に先立って存在するグラフィックビジュアル言語110に他の形でアクセスする。既存のグラフィックビジュアル言語110は、デザインツール120をホスティングするコンピュータ(1つまたは複数)から、または任意のタイプのネットワークを介してコンピューティングシステム100に接続された、ある他のコンピュータから受け取ることができる。そのような実施形態では、次でより詳細に説明するように、グラフィックビジュアル言語110を、変更できないサブ部分を有するツリーライク構造として受け取る場合があり、あるいは、グラフィックビジュアル言語110を、1次元でのみ変更できるサブ部分を有する1次元ツリーライク構造として受け取る場合がある。その後、既存のグラフィックビジュアル言語を、やはり次でより詳細に説明するように、2次元ツリーライク構造への変換のために変換モジュール150に供給することができる。

10

【0030】

グラフィック言語110が先在するのかユーザによって作成されるのかに関わりなく、レイアウト/プレゼンテーションモジュール130は、そのグラフィック言語を自動的に2次元ツリーライク構造190として表示する。ツリーライク構造190は、ツリーライク構造のルートから始まる少なくとも1つの分岐を含み、複数のサブ部分またはサブ分岐を含む。サブ部分の1つを、端部分とすることもできる。サブ部分は、グラフィック言語110によって表示されるシステム、サービス、またはプロセスを、モデリングされるシステム、サービス、またはプロセスのオブジェクト、属性、動作、およびアソシエーションの機能的関係および内部挙動に関してモデリングする。

20

【0031】

さらに、2次元ツリーライク構造190は、そのツリーライク構造のサブ部分をデザインツール/アプリケーション120によって垂直に、水平に、斜めに、またはこの3つのすべてで動的に自動的に変更することを可能にする形で、レイアウト/プレゼンテーションモジュール130によって構築され、または変換される。いくつかの実施形態で、端部分でもあるサブ部分は、変更されることから除外される。2次元ツリーライク構造190の特定の例を、下で与える。

30

【0032】

デザインツール/アプリケーション120は、さらに、変更モジュール140を含む。変更モジュール140は、ツリーライク構造190のサブ部分を変更するユーザ入力180をユーザから受け取る。たとえば、ユーザ入力は、縮小、拡張、またはスケーリングのために、変更可能なサブ部分(1つまたは複数)を選択する。ユーザ入力にตอบสนองして、変更モジュール140は、レイアウトモジュール130によって表示されつつあるツリーライク構造のサブ部分のうちの1つまたは複数を動的に自動的に変更する。

【0033】

たとえば、サブ部分(1つまたは複数)をスケーリングして、使用可能なディスプレイスペースを最適化することができ、あるいは、ツリーライク構造190の選択されたサブ部分のフォーカスされたディスプレイを可能にするためにサブ部分(1つまたは複数)を縮小または拡張する。有利なことに、ツリーライク構造190の構成は、所与のサブ部分から分岐するいずれかのまたはすべてのサブ部分をも変更することを必要とせず、その所与のサブ部分を変更することを可能にし、したがって、ユーザが所望のサブ部分をオンザフライで動的に表示することを可能にする。サブ部分は、水平に、垂直に、斜めに(垂直または水平の組合せと考えることができる)、あるいはこの3つの任意の組合せで変更する。

40

【0034】

さらに、変更されたサブ部分は、動的に位置決めされ、その結果、表示されるサブ部分は、使用可能なディスプレイスペースを最適化するようになり、これは、有利なことに、

50

フローダイアグラムが最適の外見を維持することを可能にする。いくつかの状況で、水平に表示されていたサブ部分を、ディスプレイのバランスをとるために垂直にまたは斜めに再位置決めする。垂直におよび斜めに表示されるサブ部分に対する変更をも、変更することができる。上の利益は、次の２次元ツリーライク構造の特定の例の議論でより明白になる。

【 0 0 3 5 】

いくつかの実施形態で、変更モジュール 1 4 0 は、ツリーライク構造 1 9 0 をどのようにに変更し、および / または表示しなければならないのかを指定する事前定義のルール 1 6 0 をも受け取る。そのような実施形態で、変更モジュール 1 6 0 は、既に説明したユーザ対話および受け取ったルール 1 6 0 に基づいて、少なくとも 1 つの端部分以外のサブ部分を動的にスケーリングし、拡張し、または縮小する。1 つの例のルール 1 6 0 は、レイアウトモジュール 1 3 0 に、第 2 グラフィカル要素をフローダイアグラム内で第 1 グラフィカル要素の右に配置し、すべての追加オブジェクトを第 2 オブジェクトの下に配置するように指示する。

【 0 0 3 6 】

さらなる実施形態で、デザイナツール / アプリケーション 1 2 0 は、永続 (p e r s i s t e n t) メモリ 1 7 0 へのアクセスを有する。永続メモリ 1 7 0 は、任意のタイプの穏当な永続メモリとすることができ、デザイナツール / アプリケーション 1 2 0 をホスティングするコンピュータ上に存在することができ、あるいは、任意のタイプのネットワーク接続を介してこれにアクセスすることができる。そのような実施形態で、永続メモリ 1 7 0 は、デザイナツール / アプリケーション 1 2 0 がユーザによって終了される場合にツリーライク構造 1 9 0 のサブ部分に対して行われたすべての変更を維持するのに使用される。ユーザがデザイナツール / アプリケーション 1 2 0 を再アクチベートする場合に、ツリーライク構造 1 9 0 は、次でより詳細に説明するように、デザイナツール / アプリケーション 1 2 0 が終了される前に見えていたとおりに表示される。

【 0 0 3 7 】

図 2 A を参照すると、特定のグラフィックビジュアル言語フローダイアグラム 2 0 0 が示されている。フローダイアグラム 2 0 0 は、アクタが注文を行うためにソフトウェアをどのようにに使用できるかを示すユースケース図 (U M L の一部) である。このダイアグラムに示されているのは、さまざまなオブジェクトおよびコネクタタイプである。たとえば、形状 2 0 1 は、アクタタイプの形状である。他の形状は、アクタが行うことまたはプロセスによって行われることを表す。さまざまな形状の間の線は、形状の間の関係を表すコネクタである。図 2 A では、コネクタは、たとえば、 < u s e s > タイプまたは < e x t e n d > タイプならびに他のコネクタタイプである。

【 0 0 3 8 】

たとえば、ユーザ (アクタ) 2 0 1 は、注文 2 0 8 を行うことができる。発注は、コネクタ 2 1 1 によって示されるように、カタログ 2 0 4 を要求することに拡張することができる。発注は、コネクタ 2 1 2 ~ 2 1 4 によって示されるように、カスタマデータの供給 2 0 2、製品の注文 2 0 3、および支払いの手配 2 0 5 というアクションを使用する。コネクタ 2 1 5 および 2 1 6 によって示されるように、支払いのタイプを現金とすること 2 0 6 ができ、あるいは、信用販売の手配 2 0 7 をすることができる。

【 0 0 3 9 】

前に述べたように、デザイナツール / アプリケーション 1 2 0 は、フローダイアグラム 2 0 0 を受け取り、フローダイアグラム 2 0 0 を、サブ部分 2 3 1 ~ 2 3 7 を含む 2 次元ツリーライク構造に変換し、サブ部分 2 3 1 ~ 2 3 7 は、前に説明したように、垂直方向と水平方向との両方でおおよび潜在的に斜めの方向でスケーリングし、縮小し、拡張することによって変更する。ここで図 2 B を参照すると、フローダイアグラム 2 0 0 は、変換および表示プロセスを受けた後に、デザイナツール / アプリケーション 1 2 0 で見える可能性がある形で示されている。ダイアグラム 2 0 0 は、ツリーライク構造 1 9 0 に対応するものとして示され (これは必要ではない)、それでも、前に説明したオブジェクト 2

01～208を含むことに留意されたい。また、ユーザ対話（UI）要素221～229も、デザインツール/アプリケーション120によってこのツリーライク構造に追加されていることに留意されたい。UI要素は、ユーザがツリーライク構造200のサブ部分を動的に変更することを可能にするのに使用される。

【0040】

図2Bでは、コネクタ212、213、および214は、これらが<uses>タイプのコネクタであるという共通性に基づいて単一の接続分岐に変換されている。同様の形で、コネクタ215および216は、これらの共通のコネクタタイプに基づいて単一の接続分岐に変換されている。これらの両方が、マルチコネクタの例であり、ここでは、2次元ツリーライク構造への変換中に、サブ部分の間の複数の接続が、組み合わせられたコネクタを表す1つのコネクタに組み合わせられる。もちろん、状況がそれを正当化する場合には、異なるタイプのコネクタを単一のコネクタに組み合わせられることも可能である。たとえば、図2では、一般にサブ部分237に対応する垂直線は、マルチコネクタであるが、サブ部分233および234に対応する水平線は、特定のコネクタである。そのような実施形態では、異なるタイプのコネクタが単一コネクタに組み合わせられたことの表示を、サブ部分233および234に対応する水平コネクタのとなりに配置する。

【0041】

このダイアグラムでは、ユーザ201は、ツリーライク構造のルートとして働く。複数のルートがある場合があることに留意されたい。たとえば、「注文を行う」208は、それから分岐する2つの分岐のルートとすることができ、「支払いを手配する」205は、やはりそれから分岐する分岐のルートとして働く。ツリーライク構造200は、さらに、分岐の端として働く複数の端部分を含む。たとえば、「カタログを要求する」204は、ある分岐の端であるため、端部分とする。類似する形で、「カスタマデータを供給する」202、「製品を注文する」203、「支払いを手配する」207、および「現金を支払う」206も、分岐の端であるため、端部分と考える。いくつかの実施形態では、変更を、端部分以外のサブ部分に対して行うことができる。他の実施形態では、変更を、すべてのサブ部分に対して行うことができる。

【0042】

前に述べたように、デザインツール/アプリケーション120は、UI要素221～229を2次元ツリーライク構造200に追加する。これらのUI要素は、ユーザがさまざまなサブ部分231～237を水平と垂直との両方で動的に変更することを可能にする。これらの変更は、通常は、さまざまなサブ部分のスケールリング、縮小、または拡張のうちの1つである。複数の例を、これから説明する。これらの例は、通常は、示される変更として縮小を使用するが、明示的に言及されない場合であっても、拡張および/またはスケールリングをも実行できることに留意されたい。

【0043】

たとえば、UI 222を選択することができ、これは、オブジェクト204を含むサブ部分232の垂直方向での縮小をもたらす。それと同時に、UI 223を選択することができ、これは、対応するオブジェクトと共にサブ部分233～237の縮小をもたらす。したがって、2次元ツリーライク構造200は、水平にまたは垂直のいずれかで変更される能力を有する。図2Bには示されていないが、2次元ツリーライク構造は、下で図3に関して説明するように、斜め方向で変更することもできる。

【0044】

さらに、サブ部分を、水平方向と垂直方向との両方で変更することもできる。たとえば、UI 226を選択することができ、これは、オブジェクト206および207の垂直方向での縮小をもたらす。逆に、UI 227および/または228を選択することができ、これは、オブジェクト206および/または207の水平縮小をもたらす。類似する形で、サブ部分233、234、および237を、UI 223が選択される場合に垂直に変更することができ、それぞれUI 224、225、および229が選択される場合に水平に変更することもできる。

【 0 0 4 5 】

いくつかの実施形態で、2次元ツリーライク構造200は、変更されるサブ部分から分岐するサブ部分に対する変更をも自動的に行うことなく、そのツリーライク構造200のサブ部分に対して変更を行う能力をも有する。たとえば、UI 225を縮小することができる。これは、オブジェクト205を含むサブ部分234を縮小するはずである。しかし、サブ部分235および236は、自動的に縮小はされない。これらは、単純に、使用可能なビューイングスペースを最適化するために再スケーリングされるものとすることができる。もちろん、ユーザがオブジェクト206を表示することを望むのみである場合があり、その場合には、この目的のユーザ対話を受け取った際に、オブジェクト207をも水平に縮小することができる。

10

【 0 0 4 6 】

さらなる例として、オブジェクト208を含むサブ部分231を、UI 221を選択することによって縮小することができる。多数の従来の1次元ツリーライク構造では、これは、ツリーライク構造の残りを自動的に縮小するはずである。2次元ツリーでは、他のサブ部分は、ユーザ対話がそれらに対する変更を指定するまで表示されるが、いくつかの実施形態では、ディスプレイの最適化を可能にするためにこれらのサブ部分を自動的にスケーリングすることができる。したがって、ユーザは、ユーザに望ましいサブ部分だけを表示するために、ツリーライク構造200を動的に変更することができる。

【 0 0 4 7 】

図2Bならびに続く図3および4では、1つのUI要素だけが、オブジェクトごとまたはサブ部分ごとに図示されていることに留意されたい。図示のUI要素は、サブ部分の縮小および拡張を可能にする。これよりさらに多数のUI要素を使用することもできることを了解されたい。そのような追加のUI要素は、プロパティフラグ、ユーザコメント、オブジェクトおよび/またはサブ部分の間の追加の関係などを拡張し、縮小するのに使用することができる。したがって、任意の個数のUI要素を実装する2次元ツリーライク構造が、本明細書で開示される実施形態によって企図されている。

20

【 0 0 4 8 】

デザインツール/アプリケーション120によって提供される動的変更能力をさらに示すために、図1のツリーライク構造190に対応するものとするのできる一般化された2次元ツリーライク構造300を、図3Aに示す。図3Aは、さまざまな形状301~313を含む。これらの形状が、さまざまなオブジェクト、属性、動作、アソシエーションなどを含むか説明することに留意されたい。形状301~313の正確な性質は、本明細書で説明される実施形態に重要ではなく、これを反映するためにさまざまな形状として図示されている。ツリーライク構造300は、さらに、拡張/縮小メカニズムおよびスケーリングメカニズムを介するツリーライク構造のサブ部分の動的変更を可能にするためにツリーライク構造内に配置されるUI要素321~333を含む。ツリーライク構造300の全体的レイアウトが、例示のためのみであり、本明細書で開示される実施形態を限定するのにこれを使用してはならないことに留意されたい。

30

【 0 0 4 9 】

形状301は、ツリーライク構造のルートとして働くが、他の要素もこのツリーライク構造のルートとすることができる。ルートから分岐しているのが、さまざまなサブ部分350~361である。サブ部分のうちのいくつかは、水平方向に配位され、いくつかは垂直方向に配位され、いくつかは斜め方向に配位されていることに留意されたい。ツリーライク構造300は、2次元であるため、状況に応じてこれらの方向のいずれにおいても変更することができる。前に述べたように、変更は、通常、スケーリング、縮小、または拡張である。

40

【 0 0 5 0 】

図3Bを参照すると、サブ分岐のうちのいくつかは変更された後のツリーライク構造300が示されている。たとえば、ユーザが、斜めのサブ部分350を変更することを望む場合がある。ユーザは、UI要素321を選択する入力を供給することができ、したがっ

50

て、形状 3 0 2 を含むサブ部分 3 5 0 を斜め方向で縮小する。いくつかの実施形態では、UI 3 2 1 が選択される場合に、サブ部分 3 5 1 および 3 5 2 をも斜めに縮小することができる。他の実施形態では、2 次元ツリーライク構造 3 0 0 は、まだサブ部分 3 5 1 および 3 5 2 を表示している間に、斜めのサブ部分 3 5 0 だけを縮小する。1 次元ツリーライク構造では、サブ部分 3 5 1 および 3 5 2 は、通常、必ず縮小される。後で、希望に応じて UI 3 2 1 を選択して、斜めのサブ部分 3 5 0 を動的に拡張することができる。

【 0 0 5 1 】

類似する形で、ユーザが、垂直のサブ部分 3 6 0 を変更することを望む場合もある。ユーザは、UI 要素 3 3 1 を選択する入力を供給し、したがって、形状 3 1 2 を含むサブ部分 3 6 0 を垂直方向で縮小する。1 次元ツリーライク構造では、サブ部分 3 6 0 も、通常、必ず縮小される。しかし、いくつかの実施形態では、2 次元ツリーライク構造 3 0 0 は、サブ部分 3 6 1 がまだ表示されている間に、垂直のサブ部分 3 6 0 だけを縮小する。後で、希望に応じて UI 3 3 1 を選択して、垂直のサブ部分 3 6 0 を動的に拡張することができる。

【 0 0 5 2 】

同様に、ユーザが、水平のサブ部分 3 5 4 を変更することを望む場合もある。ユーザは、UI 要素 3 2 6 を選択する入力を供給し、したがって、形状 3 0 6 を含むサブ部分 3 5 4 を水平方向で縮小する。前に説明したように、1 次元ツリーライク構造では、サブ部分 3 5 5 および 3 5 6 も、通常は縮小される。いくつかの実施形態で、2 次元ツリーライク構造 3 0 0 は、サブ部分 3 0 7 および 3 0 8 がまだ表示されている間に、水平のサブ部分 3 0 6 だけを縮小する。斜めおよび垂直の変更と同様に、後で、希望に応じて UI 3 2 6 を選択して、水平のサブ部分 3 5 4 を動的に拡張することができる。

【 0 0 5 3 】

さらに、図 3 B は、形状 3 1 4 および UI 要素 3 3 4 を含む新しいサブ部分 3 6 2 が 2 次元ツリーライク構造 3 0 0 に追加されていることをも示す。デザイナーツール / アプリケーション 1 2 0 は、ユーザが、ツリーライク構造 3 0 0 のうちで新しいサブ部分を追加することが望ましい部分を選択することを可能にする。たとえば、ユーザは、ユーザが新しいグラフィカルサブ部分またはオブジェクトを選択し、ドラッグし、ツリーライク構造上の望ましい位置に配置することを可能にするユーザインターフェースを使用することができる。デザイナーツール / アプリケーション 1 2 0 は、その新しいサブ部分またはオブジェクトを自動的に追加することができ、ツリーライク構造 3 0 0 は、使用可能なディスプレイスペースを最適化するためにスケールされる。したがって、ユーザは、新しいサブ部分またはオブジェクトをツリーライク構造 3 0 0 に動的に追加することができる。

【 0 0 5 4 】

図 3 C を参照すると、2 次元ツリーライク構造 3 0 0 に対するさらなる変更が示されている。この図では、ルート形状 3 0 1 および形状 3 1 1 を含む端部分 3 5 9 が示されている。図 3 A のサブ部分 3 5 0 およびサブ部分 3 5 0 の先のすべてのサブ部分を表す箱 3 8 0 も示されている。同様に、箱 3 9 0 は、図 3 A のサブ部分 3 5 3 およびサブ部分 3 5 3 の先のすべてのサブ部分を表す。箱 3 8 0 および 3 9 0 が、便宜のためのみに使用されており、デザイナーツール 1 2 0 による実際のディスプレイを暗示することを意図されていないことに留意されたい。

【 0 0 5 5 】

図 3 C は、変更される部分から分岐するサブ部分をも変更する必要なしに、任意のサブ部分を変更した、2 次元ツリーライク構造 3 0 0 の能力を示す。3 つの例の変更だけがツリーライク構造 3 0 0 に示されているが、図 3 A に示された他のサブ部分のどれであっても、望みどおりに水平に、垂直に、および / または斜めにスケールし、縮小し、または拡張することができる。したがって、ユーザは、ユーザによって望ましいと考えられる部分を表示するために、箱 3 8 0 および 3 9 0 の諸部分を動的に拡張する能力を有する。これは、ユーザが最大の関心を有するサブ部分に焦点を合わせることが有利に可能にする。

【 0 0 5 6 】

さらに、図 3 C は、ツリーライク構造 3 0 0 が、望まれる場合にルート形状および端部分を除くすべての要素を縮小する能力を有することを示す。これは、たとえばプロセスの始めおよび終わりだけを表示することが望ましい、数百個のサブ部分を有するツリーライク構造に有利である可能性がある。デザイナーツール 1 2 0 が、変更が図 3 B および 3 C で行われる場合に使用可能なディスプレイスペースに最適におさまるようにツリーライク構造 3 0 0 を動的にスケーリングもするが、これが図示されていないことに留意されたい。

【 0 0 5 7 】

本明細書で開示されるいくつかの実施形態が、端部分または変更されるサブ部分から分岐するサブ部分をも変更する必要なしに変更できるサブ部分を有する 2 次元ツリーライク
10
に關係することを、上で複数回述べた。これは、1 次元ツリーライク構造が通常変更される形とは異なる。そのような 1 次元ツリーライク構造は、一般に、ルートと、主分岐および / または他のサブ部分から分岐するさまざまなサブ部分を含む 1 つまたは複数の分岐とを有するものとして表示される。1 次元ツリーライク構造の例は、コンピュータオペレーティングシステムの周知のファイルおよびフォルダディレクトリを含む。1 次元ツリーライク構造は、ユーザがさまざまなサブ部分を縮小し、拡張することを可能にする。しかし、上述の 2 次元ツリーライク構造のいくつかの実施形態とは非常に対照的に、端部分として働くサブ部分も、通常、その上の別のサブ部分のいずれかが変更される時に必ず変更される。

【 0 0 5 8 】

たとえば、図 5 に、通常の 1 次元ツリーライク構造 5 0 0 を示す。ツリーライク構造 5 0 0 は、任意のグラフィカルオブジェクトとすることができる形状 5 0 1 ~ 5 0 4 を含む。第 2 サブ部分 5 2 2 は、第 1 サブ部分 5 2 1 および第 3 サブ部分 5 2 3 から分岐し、第 3 サブ部分 5 2 3 は、このツリーライク構造では端部分としても働き、第 2 サブ部分 5 2 2 から分岐する。ユーザが、第 1 サブ部分 5 2 1 を縮小する場合に、形状 5 0 2、5 0 3、および 5 0 4 を含む第 2 サブ部分 5 2 2 および第 3 サブ部分 5 2 3 も、自動的に縮小される。類似する形で、第 2 サブ部分 5 2 2 を縮小すると、形状 5 0 3 および 5 0 4 を含む第 3 サブ部分 5 2 3 が、自動的に縮小される。ユーザは、単純に、第 3 サブ部分 5 2 3 をまだ表示している間に第 2 サブ部分 5 2 2 を縮小することができず、第 2 サブ部分 5 2 2 および / または第 3 サブ部分 5 2 3 をまだ表示している間に第 1 サブ部分 5 2 1 を縮小
20
30
することができない。類似する形で、サブ部分 5 2 1 および / または 5 2 2 を拡張すると、サブ部分 5 2 3 も自動的に拡張される。上のシナリオのすべてにおいて、端部分 5 2 3 の上のサブ部分に対する変更のすべてが、端部分 5 2 3 も変更されることをもたらす。

【 0 0 5 9 】

ここで図 4 A ~ 4 C を参照すると、2 次元ツリーライク構造のいくつかの実施形態の追加の変更態様のさらなる例が示されている。図 4 A では、図 1 のツリーライク構造 1 9 0 に対応するものとしてすることができる 2 次元ツリーライク構造 4 0 0 が、形状 4 0 1 ~ 4 0 5 と共に示されている。前の例と同様に、形状 4 0 1 ~ 4 0 5 の正確な性質は、重要ではない。この例では、形状 4 0 1 は、ツリーライク構造 4 0 0 のルートを表し、ツリーライク構造 4 0 0 は、サブ部分 4 1 0 ~ 4 1 3 をも含む。上述のとおり、サブ部分 4 1 1 を変更するユーザ入力を受け取ることができる。
40

【 0 0 6 0 】

図 4 B を参照すると、サブ部分 4 1 1 の変更は、UI 要素 4 6 0 をもたらし、UI 要素 4 6 0 は、縮小された形状 4 0 2 および 4 0 3 を表す。ツリーライク構造 4 0 0 は、形状 4 0 1、4 0 4、および 4 0 5 を含む。さらに、サブ部分 4 1 0、4 1 2、および UI 要素 4 6 0 を含むサブ部分 4 3 0 が、その後、作成される。

【 0 0 6 1 】

ここで図 4 C を参照すると、ユーザがサブ部分 4 3 0 を変更したことが示されており、この変更は、縮小された形状 4 0 1 および 4 0 4 と以前に縮小された形状を表す UI 要素 4 6 0 とを表す UI 要素 4 6 1 をもたらす。UI 要素 4 6 0 が、図 4 C では UI 要素 4 6
50

1の内部に示されているが、これが、必要な特徴ではない可能性があることに留意されたい。たとえば、いくつかの実施形態で、単一のUI要素が、ユーザによって選択される場合に、UI要素460および461のすべての縮小されたサブ部分を拡張させることができる。他の実施形態では、UI要素460を、UI要素461内で隠すことができ、UI要素461が拡張されるまで不可視とすることができる。

【0062】

もう1つの実施形態で、図4Cは、デザイナツール120が、隠されたサブ部分の状態を記憶できることを示す。たとえば、UI要素461が、その後ユーザによって拡張される場合に、形状401および404は表示されるが、形状402および403は、UI要素460がその後拡張されるまでは、まだ縮小されており、UI要素460によって表される。

10

【0063】

他の実施形態では、前に述べたように、コンピューティングシステム100は、2次元ツリーライク構造に対して行われる変更を維持するのに使用できる永続メモリ170を含む。たとえば、戻って図4A~4Cを参照すると、サブ部分411がUI要素460に縮小される場合に、この変更を、永続メモリ内で維持することができる。ユーザが、デザイナツール/アプリケーション120を終了する場合に、その変更は、失われない。後に、ユーザがデザイナツール/アプリケーション120を再アクチベートする場合に、ツリーライクダイアグラム400が、図4Bに示されているように表示される。

【0064】

20

同様に、ユーザが、デザイナツール/アプリケーション120の終了に先立ってサブ部分430をUI要素461内に縮小する場合に、サブ部分430に対して行われた変更と隠されている可能性があるサブ部分411に対して行われた変更との両方が、永続メモリ170内で維持される。後に、ユーザがデザイナツール/アプリケーション120を再アクチベートする場合に、ツリーライクダイアグラム400が、図4Cに示されているように表示される。有利なことに、UI要素460によって表される変更は、隠されていなくても、やはり保持され、ユーザがUI要素461を拡張する場合に、UI要素460が図4Bのように表示されるようになる。その後UI要素460が拡張される場合に、ツリーライク構造400は、図4Aに示されているように表示される。したがって、ツリーライク構造を実行するアプリケーションを終了する場合であっても隠れ変更および非隠れ変更を永続させる能力は、ユーザに、アプリケーションを終了する際に、行われた変更のすべてを記憶する必要なしに変更を行う能力を与える。これは、数百個のネストしたサブ部分を含むツリーライク構造に特に有用である。変更が永続化されない場合には、ユーザは、アプリケーションが起動されるたびに変更をやり直さなければならない。

30

【0065】

本発明を、機能的ステップおよび/または非機能的行為を含む方法に関して説明することもできる。次は、本発明を実践する際に実行できるステップおよび/または行為の説明である。通常、機能的ステップは、達成される結果に関して本発明を説明するが、非機能的行為は、特定の結果を達成するためのより特定のアクションを説明する。機能的ステップおよび/または非機能的行為を、特定の順序で説明し、または主張する場合があるが、本発明は、ステップおよび/または行為の特定の順序付けまたは組合せに必ずしも限定されない。さらに、特許請求の範囲での列挙および図6~9の流れ図の次の説明におけるステップおよび/または行為の使用は、そのような言葉の所望の特定の使用を示すために使用される。

40

【0066】

前に述べたように、図6~9は、本発明のさまざまな例示的实施形態の流れ図を示す。図6~9の次の説明は、時々、図1~4からの対応する要素に言及する。これらの図からの特定の要素に言及する場合があるが、そのような言及は、例示のみのために使用され、明示的に主張されない限り、説明される実施形態の範囲を限定しまたは他の形で狭めることは意味しない。

50

【 0 0 6 7 】

まず図 6 に移ると、グラフィックビジュアル言語のフローダイアグラムを、ユーザ対話の容易さおよび最適ディスプレイを可能にする 2 次元ツリーライク構造に動的にレイアウトする方法 6 0 0 の流れ図が示されている。方法 6 0 0 は、ツリーライク構造のルートから始まり、複数のサブ部分を含む、少なくとも 1 つの分岐を含むグラフィカルツリーライク構造を表示すること 6 0 1 を含み、サブ部分は、少なくとも 1 つの端部分を含む。サブ部分は、1 つまたは複数のオブジェクト、属性、動作、またはアソシエーションの機能的関係および内部挙動に関してシステム、サービス、またはプロセスをモデリングする。

【 0 0 6 8 】

たとえば、デザイナツール / アプリケーション 1 2 0 のレイアウト / プレゼンテーションモジュール 1 3 0 は、2 次元ツリーライク構造 1 9 0 を表示することができる。ツリーライク構造 2 0 0、3 0 0、および 4 0 0 が、上の定義を満足する 2 次元ツリーライク構造の例である。

【 0 0 6 9 】

方法 6 0 0 は、少なくとも 1 つの端部分以外のグラフィカルツリーライク構造の 1 つまたは複数のサブ部分を変更するユーザ対話を受け取ること 6 0 2 をも含む。たとえば、デザイナツール / アプリケーション 1 2 0 の変更モジュール 1 4 0 が、ツリーライク構造 1 9 0 のサブ部分を変更するユーザ入力 1 8 0 を受け取ることができる。いくつかの実施形態で、ユーザ対話は、スケーリング、縮小、または拡張のために 1 つまたは複数のサブ部分を選択することができる。これは、変更されるサブ部分に関連する UI 要素をアクチベートすることによって行うことができる。

【 0 0 7 0 】

いくつかの実施形態で、デザイナツール / アプリケーション 1 2 0 の変更モジュール 1 4 0 は、ツリーライク構造 1 9 0 をどのように変更し、表示しなければならないのかを指定する 1 つまたは複数の事前定義のルール 1 6 0 をも受け取る。ルールおよび受け取られたユーザ対話に基づいて、ツリーライク構造 1 9 0 が、動的に変更される。

【 0 0 7 1 】

方法 6 0 0 は、さらに、端部分を変更することなく、ユーザ対話の容易さ、ディスプレイの最適化、またはこの両方を可能にするために、ユーザ対話に基づいて、端部分以外の 1 つまたは複数のサブ部分のスケーリング、拡張、または縮小のうちの 1 つによって動的に変更すること 6 0 3 を含む。たとえば、デザイナツール / アプリケーション 1 2 0 の変更モジュール 1 4 0 は、ツリーライク構造 1 9 0 を動的に変更でき、このツリーライク構造 1 9 0 は、レイアウトモジュール 1 3 0 によって、表示のために最適化することができる。サブ部分は、垂直に、水平に、斜めに、またはこれらの任意の組合せで変更することができる。

【 0 0 7 2 】

たとえば、いくつかの実施形態で、図 3 B に示されているように、1 つまたは複数のサブ部分を、端部分を縮小せずに、垂直に、水平に、斜めに、またはこれらの任意の組合せで縮小することができる。そのような実施形態では、縮小は、サブ部分のうちの少なくとも 1 つを上述の UI 要素などの単一のグラフィカルオブジェクトに自動的にフォーマットすることを含むことができる。この単一のグラフィカルオブジェクトは、一般に、追加のユーザ対話の受け取りの際に少なくとも 1 つのサブ部分に戻って拡張されるように構成される。追加のユーザ対話は、ユーザが UI 要素を拡張すること程度の単純なものとすることができる。

【 0 0 7 3 】

さらなる実施形態では、第 1 サブ部分に結合された少なくとも 1 つのノードを有する第 2 サブ部分を、動的にスケーリングし、縮小し、または拡張することができる。その後、第 1 サブ部分を、端部分を変更せずに第 2 サブ部分の後に動的に変更することができる。これは、図 4 A ~ 4 C に示されたツリーライクデータ構造での隠れ変更を可能にする。

【 0 0 7 4 】

他の実施形態では、サブ部分のうちの1つまたは複数を、端部分を拡張せずに垂直に、水平に、斜めに、またはこれらの任意の組合せで拡張することもできる。そのような実施形態では、拡張は、UI要素などの拡張可能グラフィカルオブジェクトを少なくとも1つのサブ部分に自動的にフォーマットすることを含むことができる。

【0075】

さらなる他の実施形態では、動的変更が、ツリーライク構造への新しいサブ部分の追加を含むことができる。たとえば、形状314を含むサブ部分362が、2次元ツリーライク構造300に追加される。新しいサブ部分は、ツリーライク構造内の所望の位置を選択することと、デザイナツール/アプリケーション120に、ツリーライク構造のバランスを自動的にとることとしながら、新しいサブ部分を自動的に位置決めさせることとによって追加することができる。

10

【0076】

ここで図7に移ると、ツリーライク構造のディスプレイに対して行われた変更が、そのツリーライク構造を変更するのに使用されたアプリケーションがユーザによって終了される場合であっても維持されるようにする、ビジュアルツリーライク構造の1つまたは複数のサブ分岐に対する変更をメモリ内で永続的に保持する方法700の流れ図が示されている。方法700は、ツリーライク構造のルートから始まり複数のサブ分岐を含む少なくとも1つの分岐を含むツリーライク構造を表示するアプリケーションをアクチベートすること701を含む。サブ分岐は、1つまたは複数のオブジェクト、属性、動作、およびアソシエーションの機能的関係および内部挙動に関してシステム、サービス、またはプロセス

20

【0077】

たとえば、デザイナツール/アプリケーション120をアクチベートして、ツリーライク構造190を表示することができ、その例が、図2～4に示されている。いくつかの実施形態で、ツリーライク構造は、1次元ツリーライク構造とすることもできる。

【0078】

方法700は、複数のサブ分岐のうちの少なくとも1つを変更するユーザ入力を受け取ること702をも含む。たとえば、変更モジュール140が、ツリーライク構造190の1つまたは複数のサブ分岐を変更するユーザ入力180を受け取る。いくつかの実施形態で、ユーザ入力は、サブ分岐を拡張または縮小するUI要素を選択することができる。変更は、変更されるサブ分岐をスケーリングすること、縮小すること、および拡張することを含むことができる。

30

【0079】

方法700は、さらに、ユーザがアプリケーションを終了した後で後ほど再アクチベートされた際に、ツリーライク構造を、アプリケーションが終了されたときに見えていたとおりに正確に表示できるように、ユーザ入力に基づいて、サブ分岐に対して行われた変更を永続メモリ内で維持すること703を含む。たとえば、ツリーライク構造190に対して行われた変更を、永続メモリ170内で維持することができる。デザイナツール/アプリケーション120が終了され、後に再アクチベートされる場合に、デザイナツール/アプリケーション120が終了された場合のツリーライク構造190のディスプレイが、もう一度表示される。

40

【0080】

いくつかの実施形態で、第1サブ分岐に結合されたノードを有する第2サブ分岐を変更するユーザ入力を受け取られる。第2サブ分岐が変更される。この変更の後に、第1サブ分岐が、第2サブ分岐の変更が隠されるように変更される。その後、第2サブ分岐の隠された変更された状態を、永続メモリ内で保存することができる。アプリケーションが、後で実行される場合に、UI要素を選択することなどのユーザ入力の受け取りは、隠された第2サブ分岐を表示することを可能にする。

【0081】

たとえば、図4A～4Cを参照すると、サブ部分411をUI要素460に縮小するこ

50

とができる。この後に、サブ部分 4 3 0 を UI 要素 4 6 1 に縮小することができ、この UI 要素 4 6 1 は、上で説明したように UI 要素 4 6 0 を隠す。UI 要素 4 6 1 を含むツリーライク構造 4 0 0 を永続メモリ 1 7 0 に保存することによって、前に説明したように、アプリケーション 1 2 0 が再アクチベートされる場合のサブ部分 4 1 1 の回復が可能になる。

【 0 0 8 2 】

図 8 を参照すると、ビジュアルツリーライク構造を 2 次元ツリーライク構造に変換する方法 8 0 0 が示されている。方法 8 0 0 は、変更できない複数のサブ部分を含む第 1 ツリーライク構造または第 1 グラフライクビジュアル言語を受け取ること 8 0 1 を含む。代替案では、ツリーライク構造のルートから始まり複数のサブ部分を含む少なくとも 1 つの分岐を含む第 2 ツリーライク構造。サブ部分は、1 つまたは複数のオブジェクト、属性、動作、またはアソシエーションの機能的関係および内部挙動に関してシステム、サービス、またはプロセスをモデリングする。サブ部分は、1 つまたは複数のサブ部分が変更される場合に必ず変更される少なくとも 1 つの端部分を含む。

【 0 0 8 3 】

たとえば、デザイナツール / アプリケーション 1 2 0 のレイアウトモジュール 1 3 0 が、グラフライクビジュアル言語 1 1 0 を受け取ることができ、このグラフライクビジュアル言語 1 1 0 は、変更できない先在するツリーライク構造、端部分が変更される場合に限りて変更できる先在する 1 次元ツリー構造、または先在する任意のグラフライクビジュアル言語とすることができる。グラフライクビジュアル言語 1 1 0 は、上述したようにユーザによって作成されたこの 3 つのうちのいずれかとすることもできる。任意のグラフライクビジュアル言語は、モデリングされるシステム、サービス、またはプロセス内に存在しなければならない物を強調する構造ダイアグラム、モデリングされるシステム、サービス、またはプロセス内で発生しなければならない物を強調する挙動ダイアグラム、およびモデリングされるシステム、サービス、またはプロセス内の物の間での制御およびデータのフローを強調する対話 (i n t e r a c t i o n) ダイアグラムのうちの 1 つを含むことができる。任意のグラフライク言語は、Unified Modeling Language (UML) のさまざまなダイアグラムタイプ、データベースの世界のエンティティリレーションシップダイアグラム、およびビジネスプロセスデザイン用のビジネスプロセスダイアグラムをも含むことができる。

【 0 0 8 4 】

方法 8 0 0 は、受け取られた第 1 のまたは第 2 のツリーライク構造または任意のグラフライクビジュアル言語を 2 次元ツリーライク構造に変換すること 8 0 2 をも含む。2 次元ツリーライク構造は、2 次元ツリーライク構造のルートから始まり、端部分を含む複数のサブ部分を含む、少なくとも 1 つの分岐を含み、端部分以外のサブ部分は、端部分をも変更することなくスケーリング、縮小、または拡張によって変更することができる。

【 0 0 8 5 】

たとえば、変換モジュール 1 5 0 は、受け取られたツリーライク構造またはグラフライクビジュアル言語を 2 次元ツリーライク構造 1 9 0 に変換することができる。2 次元ツリーライク構造は、前に説明したように、垂直に、水平に、斜めに、またはこの 3 つの任意の組合せで変更することができる。変換された 2 次元ツリーライク構造は、本明細書で説明する他の方法および実施形態に関する 2 次元ツリーライク構造に関して前に説明した機能性のすべてを可能にもする。

【 0 0 8 6 】

ここで図 9 を参照すると、グラフライクビジュアル言語のダイアグラムをユーザ対話の容易さおよび最適ディスプレイを可能にする 2 次元ツリーライク構造に動的にレイアウトする方法 9 0 0 の流れ図が示されている。方法 9 0 0 は、ツリーライク構造のルートから始まり、垂直方向および水平方向で変更されるように構成された複数のサブ部分を含む、少なくとも 1 つの分岐を含むグラフィカルツリーライク構造を表示すること 9 0 1 を含む。サブ部分は、1 つまたは複数のオブジェクト、属性、動作、またはアソシエーションの

機能的関係および内部挙動に関してシステム、サービス、またはプロセスをモデリングする。サブ部分は、斜めにも変更されるように構成することもできる。

【 0 0 8 7 】

たとえば、デザイナツール / アプリケーション 1 2 0 のレイアウト / プレゼンテーションモジュール 1 3 0 が、2次元ツリーライク構造 1 9 0 を表示することができる。ツリーライク構造 2 0 0、3 0 0、および 4 0 0 が、上の定義を満足する2次元ツリーライク構造の例である。

【 0 0 8 8 】

方法 9 0 0 は、少なくとも1つ端部分以外のグラフィカルツリーライク構造の1つまたは複数のサブ部分を変更するユーザ対話を受け取ること 9 0 2 をも含む。たとえば、デザイナツール / アプリケーション 1 2 0 の変更モジュール 1 4 0 が、ツリーライク構造 1 9 0 のサブ部分を変更するユーザ入力 1 8 0 を受け取ることができる。いくつかの実施形態で、ユーザ対話は、垂直に、水平に、および潜在的に斜めにスケーリングし、縮小し、または拡張するために1つまたは複数のサブ部分を選択することができる。これは、変更されるサブ部分に関連するUI要素をアクチベートすることによって行うことができる。

【 0 0 8 9 】

方法 9 0 0 は、さらに、ユーザ対話に基づいて、ユーザ対話の容易さ、ディスプレイの最適化、またはこの両方を可能にするために水平に、垂直に、またはその両方で動的に変更すること 9 0 3 を含む。たとえば、デザイナツール / アプリケーション 1 2 0 の変更モジュール 1 4 0 が、ツリーライク構造 1 9 0 を動的に変更することができ、その後、ツリーライク構造 1 9 0 は、レイアウトモジュール 1 3 0 によってディスプレイのために最適化することができる。サブ部分を、スケーリングし、拡張し、または縮小することができる。

【 0 0 9 0 】

さらなる実施形態では、第1サブ部分に結合された少なくとも1つのノードを有する第2サブ部分を、動的に変更することができる。その後、第1サブ部分を、第2サブ部分の後に、第2サブ部分に対する変更を隠す形で動的に変更することができる。これは、図 4 A ~ 4 C に示された、ツリーライクデータ構造での隠れ変更を可能にする。隠れ変更は、その後、追加のユーザ入力を受け取る際に表示することができる。

【 0 0 9 1 】

さらに別の実施形態では、動的変更が、ツリーライク構造への新しいサブ部分の追加を含むことができる。たとえば、形状 3 1 4 を含むサブ部分 3 6 2 が、2次元ツリーライク構造 3 0 0 に追加される。新しいサブ部分は、ツリーライク構造内の所望の位置を選択することと、デザイナツール / アプリケーション 1 2 0 に、ツリーライク構造のバランスを自動的にとることとしながら、新しいサブ部分を自動的に位置決めさせることとによって追加することができる。複数のコネクタを1つのコネクタに組み合わせることなど、他の変更技法を実装することもできる。

【 0 0 9 2 】

追加の実施形態では、グラフィカルツリーライク構造のディスプレイの最適化は、ディスプレイデバイスまたはウィンドウのアスペクト比に関して最適化するために、1つまたは複数のサブ部分を水平方位から垂直方位に自動的に切り替えること、あるいは1つまたは複数のサブ部分を垂直方位から水平方位に切り替えることを含む。たとえば、いくつかの状況で、グラフィカルツリーライク構造の1つまたは複数の分岐が、グラフィカルツリーライク構造のディスプレイをディスプレイデバイスのアスペクト比に関して最適化されていない物にする場合がある。たとえば、グラフィカルツリーライク構造が、垂直に長すぎる場合があり、その一方で、ディスプレイデバイスまたはウィンドウは、水平に未使用のディスプレイスペースを有する。その場合に、デザイナツール / アプリケーション 1 2 0 は、1つまたは複数の垂直のサブ部分を、水平に表示されるように自動的に切り替えることができる。これらのサブ部分は、ディスプレイを最適化された状態に保つために切り替えを必要とする新しい要素がグラフィカルツリーライク構造に追加される場合など、情

10

20

30

40

50

況がそれを正当化する場合に、後で垂直表示に戻って切り返ることができる。

【0093】

たとえば、図3Aを参照すると、いくつかの実施形態で、デザイナツール/アプリケーション120は、ディスプレイデバイスのアスペクト比に関して最適化するために、垂直のサブ部分360および361を要素310から水平に分岐するものとして表示するように、サブ部分360および361のディスプレイを自動的に切り替えることができる。類似する形で、デザイナツール/アプリケーション120は、状況がそれを正当化する場合には、水平のサブ部分354~356（またはこれらのサブ部分の任意のサブセット）を要素305から垂直に分岐するものとして表示するように、サブ部分354~356の表示を自動的に切り替えることができる。デザイナツール/アプリケーション120が、ディスプレイデバイスまたはウィンドウのアスペクト比に関して最適化する必要に応じて、図3Aおよび3Bの要素のいずれのディスプレイ方位をも自動的に切り替えることができることに留意されたい。

10

【0094】

本発明は、その趣旨または本質的特性から逸脱せずに他の特定の形態で実施することができる。説明された実施形態は、すべての面において、例示的であり、制限的ではないものと考えられなければならない。したがって、本発明の範囲は、前述の説明ではなく、添付の特許請求の範囲によって示される。特許請求の範囲の意味および同等性の範囲に含まれるすべての変更は、特許請求の範囲の範囲に含まれなければならない。

【図面の簡単な説明】

20

【0095】

【図1】グラフィカル2次元ツリーライク構造としてグラフィックビジュアル言語を動的に表示するコンピューティングシステムを示す図である。

【図2A】グラフィックビジュアル言語の例を示す図である。

【図2B】例の実施形態によるグラフィカル2次元ツリーライク構造として表示された図2Aのグラフィックビジュアル言語を示す図である。

【図3A】例の実施形態による例のグラフィカル2次元ツリーライク構造を示す図である。

【図3B】図3Aのグラフィカル2次元ツリーライク構造に対して行われる変更を示す図である。

30

【図3C】図3Aのグラフィカル2次元ツリーライク構造に対するさらなる変更を示す図である。

【図4A】例の実施形態によるもう1つの例のグラフィカル2次元ツリーライク構造を示す図である。

【図4B】図4Aのグラフィカル2次元ツリーライク構造に対して行われる変更を示す図である。

【図4C】図4Aのグラフィカル2次元ツリーライク構造に対するさらなる変更を示す図である。

【図5】グラフィカル1次元ツリーライク構造を示す図である。

【図6】例の実施形態による、グラフィックビジュアル言語のフローダイアグラムをグラフィカル2次元ツリーライク構造に動的にレイアウトする方法を示す流れ図である。

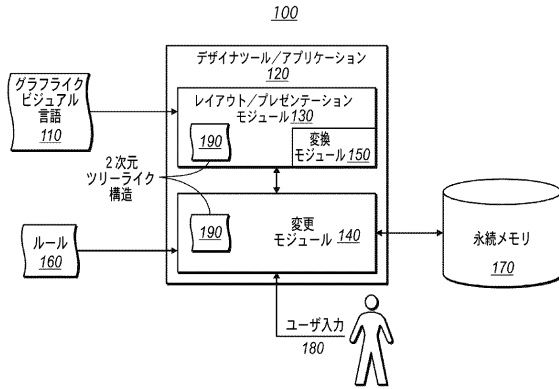
40

【図7】例の実施形態による、グラフィカルツリーライク構造の1つまたは複数のサブ分岐に対する変更をメモリ内で永続的に保持する方法を示す流れ図である。

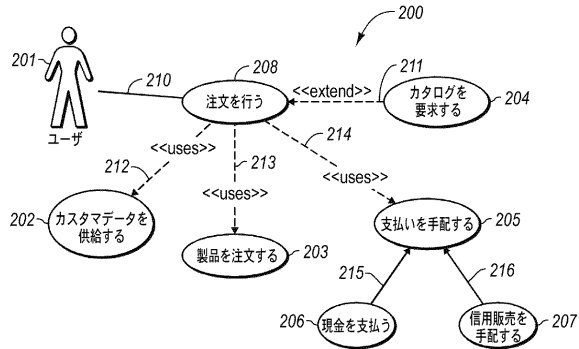
【図8】例の実施形態による、グラフィカルツリーライク構造をグラフィカル2次元ツリーライク構造に変換する方法を示す流れ図である。

【図9】例の実施形態による、グラフィックビジュアル言語のフローダイアグラムをグラフィカル2次元ツリーライク構造に動的にレイアウトする方法を示す流れ図である。

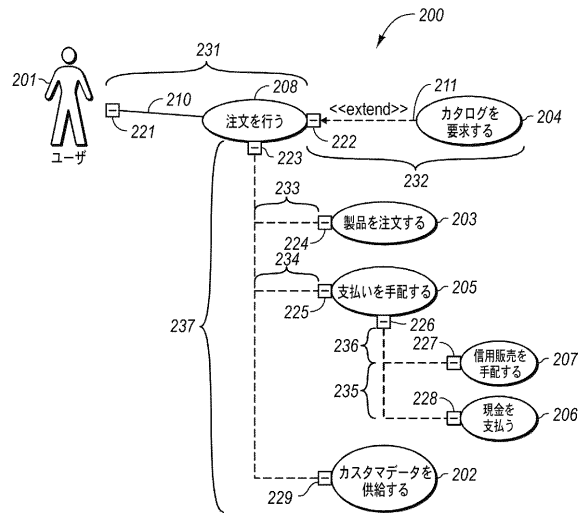
【図 1】



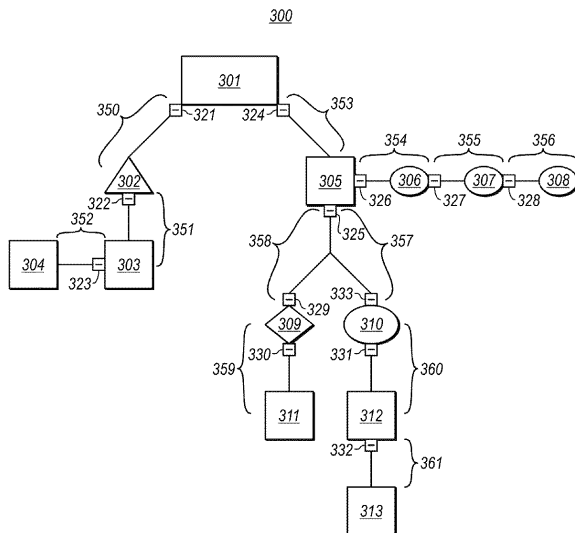
【図 2 A】



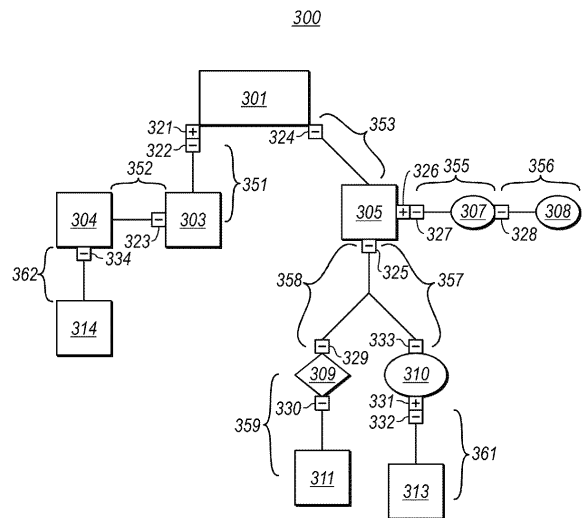
【図 2 B】



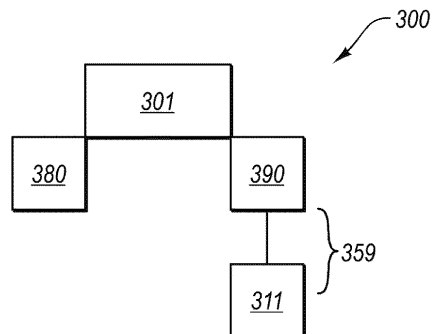
【図 3 A】



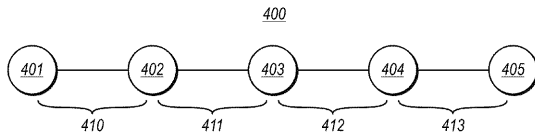
【図 3 B】



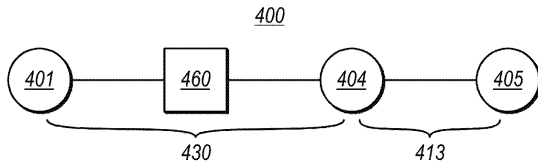
【図 3 C】



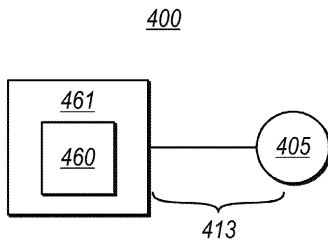
【図 4 A】



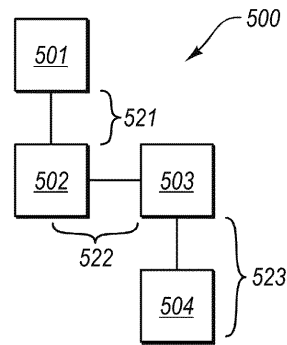
【図 4 B】



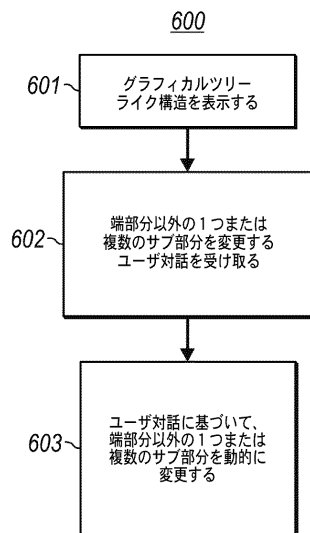
【図 4 C】



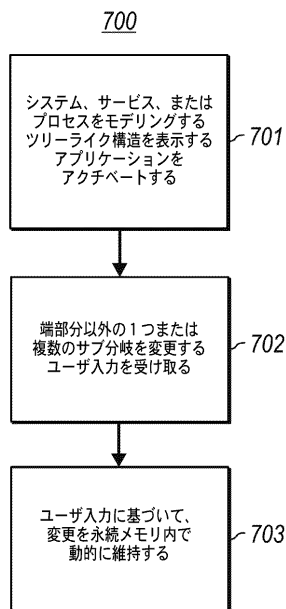
【図 5】



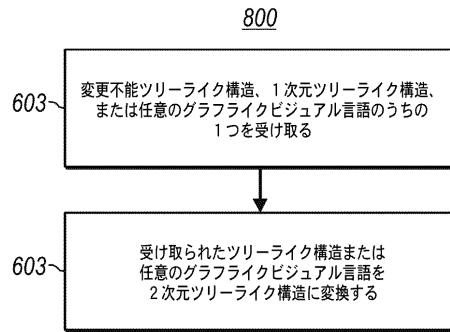
【図 6】



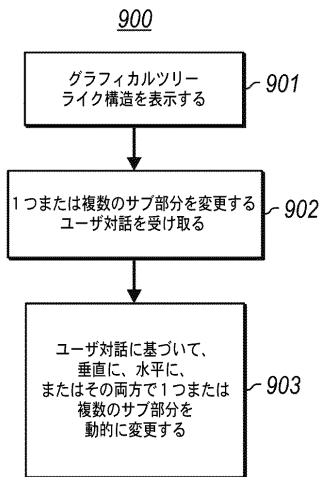
【図 7】



【図 8】



【図 9】



フロントページの続き

審査官 田中 幸雄

(56)参考文献 特開 2 0 0 1 - 1 2 5 9 2 5 (J P , A)
特開 2 0 0 4 - 2 5 9 2 8 6 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)
G06T 11/80
G06F 3/048