

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4381708号  
(P4381708)

(45) 発行日 平成21年12月9日 (2009. 12. 9)

(24) 登録日 平成21年10月2日 (2009. 10. 2)

(51) Int. Cl.	F I
<b>G 0 6 F 9/44 (2006. 01)</b>	G O 6 F 9/06 6 2 O C
<b>G 0 6 F 15/00 (2006. 01)</b>	G O 6 F 15/00 3 1 O R
<b>G 0 6 F 3/048 (2006. 01)</b>	G O 6 F 15/00 3 9 O
	G O 6 F 3/048 6 5 1 C

請求項の数 4 (全 47 頁)

(21) 出願番号	特願2003-92812 (P2003-92812)	(73) 特許権者	392026693
(22) 出願日	平成15年3月28日 (2003. 3. 28)		株式会社エヌ・ティ・ティ・ドコモ
(65) 公開番号	特開2003-330712 (P2003-330712A)		東京都千代田区永田町二丁目11番1号
(43) 公開日	平成15年11月21日 (2003. 11. 21)	(74) 代理人	100098084
審査請求日	平成18年3月28日 (2006. 3. 28)		弁理士 川▲崎▼ 研二
(31) 優先権主張番号	60/378713	(72) 発明者	ウオング, ホイ リー キャンディー
(32) 優先日	平成14年5月8日 (2002. 5. 8)		アメリカ合衆国, カリフォルニア州951
(33) 優先権主張国	米国 (US)		10 サンノゼ, スイート300, メトロ
(31) 優先権主張番号	10/271226		ドライブ181
(32) 優先日	平成14年10月15日 (2002. 10. 15)	(72) 発明者	チュー, ハオーファ
(33) 優先権主張国	米国 (US)		アメリカ合衆国, カリフォルニア州951
			10 サンノゼ, スイート300, メトロ
			ドライブ181

最終頁に続く

(54) 【発明の名称】 グラフィカルユーザインターフェイスシステム

(57) 【特許請求の範囲】

【請求項 1】

第1の装置、第2の装置およびサーバを有し、前記第1の装置から前記第2の装置へアプリケーション・グラフィカルユーザインターフェイスの実行状態をマイグレーションするグラフィカルユーザインターフェイスシステムにおいて、

前記第1の装置は、

アプリケーションを実行するための第1のプラットフォームを生成するためのソフトウェアプログラムが記憶された第1の記憶手段から、該ソフトウェアプログラムを読み出して実行することによって、前記第1のプラットフォームを生成するとともに、

前記第1のプラットフォーム上で、第1プラットフォーム特定型のグラフィカルユーザインターフェイスの表示形式を示す第1のプレゼンテーションであって該第1のプレゼンテーションのページを形成する1又は複数のウィジェットを含む第1のプレゼンテーションを実行し、該実行結果に従って画面表示を行う第1の制御手段

を有し、

前記第2の装置は、

アプリケーションを実行するためのプラットフォームであって前記第1のプラットフォームとは異なる第2のプラットフォームを生成するためのソフトウェアプログラムが記憶された第2の記憶手段から、該ソフトウェアプログラムを読み出して実行することによって、前記第2のプラットフォームを生成するとともに、

前記第2のプラットフォーム上で、第2プラットフォーム特定型のグラフィカルユーザ

10

20

インターフェイスの表示形式を示す第2のプレゼンテーションであって該第2のプレゼンテーションのページを形成する1又は複数のウィジェットを含む第2のプレゼンテーションを実行し、該実行結果に従って画面表示を行う第2の制御手段

を有し、

前記サーバは、

プラットフォーム独立型の表示形式を示すプレゼンテーションモデルであって該プレゼンテーションモデルのページを形成する1又は複数のウィジェットとその実行状態を示す実行状態データであって予め定められた値が設定された実行状態データとを含むプラットフォーム独立型のプレゼンテーションモデルを生成するためのソフトウェアプログラムが記憶された第3の記憶手段から、該ソフトウェアプログラムを読み出して実行することによって、前記プレゼンテーションモデルを生成し、

10

前記第1の装置から、前記第1プラットフォーム特定型のグラフィカルユーザインターフェイスのウィジェットの実行状態を表す実行状態データを受信し、

前記プラットフォーム独立型のグラフィカルユーザインターフェイスのウィジェットの集合であるライブラリと前記第1プラットフォーム特定型のグラフィカルユーザインターフェイスのウィジェットの集合であるライブラリとの間で予めマッピングされた第1の対応関係に基づいて、前記第1プラットフォーム特定型のグラフィカルユーザインターフェイスのウィジェットの実行状態データによって、前記プラットフォーム独立型のプレゼンテーションモデル内にある、前記第1プラットフォーム特定型のグラフィカルユーザインターフェイスのウィジェットに対応するプラットフォーム独立型のグラフィカルユーザインターフェイスのウィジェットの実行状態データを更新し、

20

前記プラットフォーム独立型のグラフィカルユーザインターフェイスのウィジェットの集合であるライブラリと前記第2プラットフォーム特定型のグラフィカルユーザインターフェイスのウィジェットの集合であるライブラリとの間で予めマッピングされた第2の対応関係に基づいて、前記プラットフォーム独立型のプレゼンテーションモデルのコピーを前記第2のプラットフォームの機能に基づいてカスタマイズすることによって前記第2のプラットフォーム上で動作可能な前記第2プラットフォーム特定型の第2のプレゼンテーションを生成し、前記プラットフォーム独立型のグラフィカルユーザインターフェイスのウィジェットの実行状態に従って、前記生成した第2プラットフォーム特定型の第2のプレゼンテーション内にある、対応する第2プラットフォーム特定型のグラフィカルユーザインターフェイスのウィジェットの実行状態データの値を設定し、実行状態データの値を設定した前記第2のプレゼンテーションを前記第2の装置へ送信する第3の制御手段

30

を有することを特徴とするグラフィカルユーザインターフェイスシステム。

#### 【請求項2】

前記実行状態は、前記プラットフォーム特定型のグラフィカルユーザインターフェイスのウィジェットの状態変化、前記第1プラットフォーム特定型のグラフィカルユーザインターフェイスのウィジェットの外観、及び前記第1プラットフォーム特定型のグラフィカルユーザインターフェイスのウィジェット内にあるデータを選択的に有していることを特徴とする請求項1に記載のグラフィカルユーザインターフェイスシステム。

#### 【請求項3】

40

前記第3の制御手段は、カスタマイズの一環として実行状態を保存しながら、前記プラットフォーム独立型のグラフィカルユーザインターフェイスのウィジェットのレイアウトを、前記第2の対応関係に基づいて、対応する第2プラットフォーム特定型のグラフィカルユーザインターフェイスのウィジェットに応じたレイアウトに変更することを特徴とする請求項1に記載のグラフィカルユーザインターフェイスシステム。

#### 【請求項4】

前記第3の制御手段は、実行状態をネットワークを通じて、送受信することを特徴とする請求項1に記載のグラフィカルユーザインターフェイスシステム。

#### 【発明の詳細な説明】

【0001】

50

**【発明の属する技術分野】**

本発明は、グラフィカルユーザインターフェイス(GUI)に関し、特に、異種装置プラットフォーム間を移動するプラットフォーム特定型のGUIウィジェットのマイグレーションに関する。

**【0002】****【従来の技術】**

パソコン、個人情報端末(PDA)、無線電話、そしてページャなどの個人向けの電子装置(携帯装置)が、ここ近年流行している。これらの携帯装置は、無線や有線ネットワークを利用した通信を行なえるので、モバイルコンピューティングを可能にしている。ネットワークは、これらの携帯装置に、情報ソースおよび他の類似の装置への接続を提供している。一般的に、ネットワークはインターネット上での通信も含んでいる。

10

**【0003】**

モバイルコンピューティング装置は、ある型のユーザインターフェイス(UI)を持っている。ある装置のユーザインターフェイスの特性は、その装置の処理能力とハードウェアによる。例えば無線電話には、限られた処理能力、及び限られたグラフィックとユーザインターフェイス能力しか有していない小さなグラフィカルユーザインターフェイス(GUI)しか与えられていない。逆に、ノートパソコンには、相対的に拡張性のあるグラフィック能力とユーザインターフェイス能力、そして処理能力が与えられている。携帯装置の人気の高まるに連れて、アプリケーションのメインプラットフォームは、デスクトップ型パソコンから携帯装置へとシフトすると思われる。

20

**【0004】****【発明が解決しようとする課題】**

携帯装置のアプリケーションの開発が困難な1つの理由は、携帯装置の異質性である。多種多様なディスプレイのサイズ、解像度、コマンドの入力方法、そしてGUIライブラリがあるため、アプリケーションの開発者は、各装置プラットフォームのグラフィカルユーザインターフェイス(GUI)用に、アプリケーションを再設計及び再実装する必要がある。たくさんの異なる携帯装置が利用され、または市場に参入しているので、再設計と再実装に、たくさんの労働力と生産コストを要する努力が現在もなされている。

**【0005】**

1つの解決策は、モデルベース技術による開発である。1つの例が、ユーザインターフェイスモデリングである。一般的に、ユーザインターフェイスモデリングは、プラットフォームモデル、プレゼンテーションモデル、そしてタスクモデルを含む。プラットフォームモデルでは、サポートされる装置用のユーザインターフェイスを形成する操作機能が示される。プレゼンテーションモデルでは、サポートされる装置に関連したユーザインターフェイスの外観に関する階層、及び様式の選択と配置が示される。タスクモデルでは、サポートされる装置のユーザが実行するタスクが示される。この技術を使用することにより、特定の装置用のインターフェイスの生成のため、異なるモデル間におけるマッピングが開発されると思われる。

30

**【0006】**

モデルベース型アプローチの実装は、モデルを実装するためのハイレベル言語の開発を含んでいる。さらに開発者は、装置をサポートするためにモデルの重要な部分を設定し、かつ特定する。ハイレベル言語は非常に複雑なので、モデルベース型アプローチを実行する前に、開発者は、実行メカニズムの知識とハイレベル言語を共に学ぶ必要がある。さらにモデルベース型アプローチは、開発者が作った異なるモデルに基づくコードを作成する。装置のユーザインターフェイス要求においては小さな差異でも、2つの似たタイプの装置用に生成されたコードにおいては大きな差異になることもある。従って開発者に求められるプログラミング技能のレベルは、とても重要になってくる。

40

**【0007】****【課題を解決するための手段】**

本発明では、ノンモデルベース型の拡張性のあるグラフィカルユーザインターフェイス(

50

SGUI) システムを開示している。SGUIシステムにより、拡張性のあるアプリケーションは、複数の異種装置プラットフォーム上において動作することが可能となる。拡張性のあるアプリケーションは、アプリケーション・グラフィカルユーザインターフェイス(GUI)を含んでもよい。アプリケーションGUIは、拡張性のあるアプリケーションが動作している異種装置プラットフォームの中の1つのユーザインターフェイスで動くSGUIシステムに適合していてもよい。アプリケーション開発者は、モデルベース型のアプローチの複雑さを伴わない異種装置プラットフォーム内において、ユーザインターフェイス機能から独立している複数のアプリケーションGUIを設定してもよい。さらに異種装置プラットフォームに関連しているユーザインターフェイスとGUIライブラリの複雑さも、SGUIシステムを利用しているアプリケーション開発者により解消されることが可能になる。

10

#### 【0008】

SGUIシステムは、拡張性のあるGUIライブラリ、カスタマイズモジュール、そしてレンダーマネージャモジュールを有している。カスタマイズモジュールは、タスクマネージャモジュールと変換マネージャモジュールを有している。拡張性のあるアプリケーションが起動、または異種装置プラットフォーム(ここでは、ターゲット装置プラットフォーム)へ移ると、アプリケーションGUIの中間表現が、拡張性のあるアプリケーションによりインスタンス化される。その中間表現は、階層的な構造において、少なくとも1つの論理パネルと、少なくとも1つの拡張性のあるグラフィカルユーザインターフェイス(SGUI)コンポーネントの表示を含むことが可能な装置プラットフォーム独立型の論理構造形式におけるプラットフォーム独立型のプレゼンテーションモデルである。中間表現内におけるノードは、論理パネルを表すコンテナノードと、SGUIコンポーネントを表すコンポーネントノードを含んでもよい。

20

#### 【0009】

中間表現は、アプリケーションGUIのアプリケーション開発者により特定されたプロパティ、及びターゲット装置プラットフォームの機能に基づいたカスタマイズモジュールによりカスタマイズされたものであってもよい。中間表現のカスタマイズにより、プラットフォーム独立型のプレゼンテーションモデルは、プラットフォームの特定表示を表している特定のプラットフォーム依存型の論理構造へと変換する。

#### 【0010】

拡張性のあるアプリケーションが、第1異種装置プラットフォームから第2異種装置プラットフォームへマイグレーションしたとき、拡張性のあるアプリケーション内にあるアプリケーション・グラフィカルユーザインターフェイスの実行状態も、またマイグレーションする。現在の実行状態を第1異種装置プラットフォームから第2異種装置プラットフォーム上で実行することにより、実行状態のマイグレーションは、ユーザに対して継続性を提供する。プラットフォーム独立型のプレゼンテーションモデル、及びプラットフォーム特定型のプレゼンテーションにおいて、現在の実行状態のマイグレーションはカスタマイズモジュールにより実行されてもよい。

30

#### 【0011】

第1異種装置プラットフォーム用の第1プラットフォーム特定型のプレゼンテーションは、少なくとも1つのプラットフォーム特定型のGUIウィジェットを有してもよい。プラットフォーム独立型のプレゼンテーションモデルのコピーに内にある、対応するプラットフォーム独立型のGUIウィジェットの実行状態を更新するため、プラットフォーム特定型のGUIウィジェットの現在の実行状態を確認し、それを用いてもよい。第2異種装置プラットフォーム用のプラットフォーム特定型のプレゼンテーション開発のため、プラットフォーム独立型のプレゼンテーションモデルのコピーをカスタマイズしてもよい。プラットフォーム独立型のGUIウィジェットの実行状態が、第2プラットフォーム特定型のプレゼンテーション内の対応するプラットフォーム特定型のGUIウィジェットに適用されてもよい。このように拡張性のあるアプリケーションがマイグレーションしたとき、第1異種装置プラットフォームからの現在の実行状態が、第2異種装置プラットフォームの実行状態で実行されてもよい。

40

50

## 【 0 0 1 2 】

SGUIシステムの魅力的な特徴は、プラットフォーム独立型のプレゼンテーションモデルのカスタマイズ中において、第1異種装置プラットフォームからの現在の実行状態を保存することである。カスタマイズにおいては、動的にプラットフォーム独立型のGUIウィジェットのレイアウトを再配置し、そしてプラットフォーム独立型のGUIウィジェットを、第2異種装置プラットフォームのGUIと適合性のある他のウィジェットへの変換を含んでもよい。独立型のGUIウィジェットの変換、及びそれに関連する実行状態の変換は、カスタマイズモジュールにより実行されてもよい。

## 【 0 0 1 3 】

## 【 発明の実施の形態 】

本発明における他の特徴と利点は、図面と以下の詳細な説明で明らかにされる。好ましい実施形態の以上の説明は、ただの一例に過ぎず、本発明の範囲を定義する特許請求の範囲を制限するものではない。

## 【 0 0 1 4 】

本発明の好ましい実施形態において、異なる複数の装置プラットフォーム上で動作することのできる拡張性のあるアプリケーションと共に使う拡張性のあるグラフィカルユーザインターフェイス(SGUI)アーキテクチャを説明する。拡張性のあるアプリケーションは、異種装置プラットフォームに対応付けられた多種のユーザインターフェイスと共に動作するように、SGUIシステムにより動的に適合されてもよい。SGUIシステムを使用することにより、拡張性のあるアプリケーションを、その拡張性のあるアプリケーションが動作することになる異種装置プラットフォームから独立したユーザインターフェイス機能を有するように設計できる。拡張性のあるアプリケーションの開発者は、SGUIシステムの使用をしなれば、種々の異種装置プラットフォームへのユーザインターフェイスを作成する時に、多くの複雑なプログラム作成が必要になるが、SGUIシステムを使用すると、これらを行なうことができる。さらに、SGUIシステムは、モデルベース型の技術のようなモデル構造がなくても、異種装置プラットフォームで動作する。このようなわけで、アプリケーション開発者によるプログラミングは比較的簡単であり、モデル構築、及び/またはモデル操作技術は必要でない。

## 【 0 0 1 5 】

拡張性のあるアプリケーションには、異なるプラットフォーム装置上で動作することのできるアプリケーションが含まれる。拡張性のあるアプリケーションのほとんどは、異種装置プラットフォームの表示画面用のプレゼンテーションを生成する機能を有している。プレゼンテーションは、視覚的な素材からなる1ページまたはそれ以上のページ(またはプレゼンテーション単位)を含んでもよい。視覚的な素材は、異種装置プラットフォームのユーザのために、表示画面上に配置されてもよい。拡張性のあるアプリケーションは、アプリケーションGUIでプレゼンテーションを生成してもよい。アプリケーションGUIは、拡張性のあるグラフィカル・ユーザインターフェイス・アプリケーションのコンポーネントである。

## 【 0 0 1 6 】

異種装置プラットフォームは、表示機能と拡張性のあるアプリケーションを動作させる機能を兼ね備えた装置であってもよい。典型的な異種装置プラットフォームは、無線電話、ポケットPC、個人情報端末(PDA)、ページャー、デスクトップコンピュータ、ノートブックコンピュータ、車載コンピュータ、または拡張性のあるアプリケーションを実行する機能のある表示画面を兼ね備えた有線/無線装置を含んでもよい。また異種装置プラットフォームは、例えばWindows NT(商標)、Windows 2000(商標)、Linux(商標)、Solaris(商標)、MacOS(商標)、Palm(商標)、Windows CE(商標)、独自のオペレーティングシステム、またはその他の装置に機能性を提供するオペレーティングシステムを含んでもよい。オペレーティングシステムは、拡張性のあるアプリケーションを起動させたりサポートするための機能を提供してもよい。

## 【 0 0 1 7 】

異種装置プラットフォームは、拡張性のあるアプリケーションを動作させたり、機能を提供するハードウェアまたはソフトウェアを含んでいてもよい。例えば、メモリのような記憶手段を含んでいてもよい。記憶手段は、例えば、オペレーティングシステム、拡張性のあるアプリケーション、および装置プラットフォームに関する他の情報を記憶してもよい。さらに、異種装置プラットフォームは、ユーザインターフェイスを含んでもよい。ユーザインターフェイスは、例えば、表示装置、キーボード、マウスに似たポインティングデバイス、タッチスクリーン、キーパッド、オーディオ機能、または異種装置プラットフォームのユーザ用のインターフェイスを提供する手段を含んでもよい。

#### 【0018】

異種装置プラットフォームの機能は、多岐に異なってもよい。ここで言う機能とは、異種装置プラットフォームと拡張性のあるアプリケーションの動作に影響を与えるハードウェア的な特徴と装置機能のことである。つまり、例えば、装置プラットフォームのタイプ（例：無線電話、PDA、ラップトップPC等）、利用可能なユーザインターフェイスライブラリ、表示画面のサイズと解像度である。さらに、代表的な機能として、表示画面に関するユーザインターフェイス、計算処理容量、メモリのサイズ、プラットフォームの使用、または装置プラットフォームのユーザインターフェイスと動作に影響するその他のパラメータが挙げられる。

#### 【0019】

本発明の好ましい実施形態において、異種装置プラットフォームは、Java（登録商標）の技術で作られた拡張性のあるアプリケーションを実行する移動装置である。Java（登録商標）の技術は、Java（登録商標）VM（Virtual Machine; パーチャルマシン）上で動くJava（登録商標）プログラミング言語を有している。Java（商標）VMは、多種に渡る装置プラットフォーム上に移植されてもよい。典型的なJava（商標）技術の実装は、ノートPCの装置プラットフォーム上で動くJava（商標）VM、ポケットPCの装置プラットフォーム上にあるパーソナルJava（商標）VM、無線電話の装置プラットフォーム上で動くJava（商標）KVMである。他の実施形態においては、他のプログラミング言語/技術、および他の装置プラットフォームが、SGUIシステムで使用される。他のプログラミング言語/技術とは、C言語、C++言語、マイクロソフトのCシャープ言語、及び/または他の技術である。

#### 【0020】

SGUIシステムは、拡張性のあるアプリケーションに対応付けられたアプリケーションGUIの中間表現をインスタンス化させる。中間表現と同様にアプリケーションGUIは、非装置に特定したもの（例えば、装置プラットフォームから独立しているもの）であるが、装置プラットフォームと適合性がある方が好ましい。プラットフォーム独立型のプレゼンテーションモデルにおける中間表現のインスタンスの作成は、拡張性のあるアプリケーションのアプリケーションGUIに基づいてもよい。中間表現は、拡張性のあるアプリケーションにより生成されるプレゼンテーション用の論理構造を提供してもよい。この論理構造を、異種装置プラットフォーム用にカスタマイズして、SGUIシステムによるプラットフォーム特定型のプレゼンテーションを作成してもよい。カスタマイズは、プレゼンテーションを表示させる異種装置プラットフォームの機能と、アプリケーションGUI内で特定されたプロパティに基づいてもよい。SGUIシステムによるカスタマイズは、装置プラットフォーム用のグラフィカルユーザインターフェイスの開発において共通な3つの問題を提起している。即ちそれらは、表示画面サイズ、入力方法、及びユーザインターフェイスライブラリである。

#### 【0021】

##### ・表示画面サイズ

拡張性のあるアプリケーションは、複数の異種装置プラットフォーム上で動作することができる。よって、拡張性のあるアプリケーションを開発しているアプリケーションプログラマは、表示画面用に特定なサイズを採り上げる必要はない。例えば、ドコモ 503i 無線電話は120×130画素の表示画面を兼ね備え、コンパックのIPAQポケットPCは320×240画素の表示画面を兼ね備え、典型的なノートPCは、1024×768画素の表示画面を兼ね備えてい

10

20

30

40

50

る。表示画面のサイズは、表示画面上におけるプレゼンテーションのページを形成しているコンポーネントのレイアウトや、視覚的なプレゼンテーションの品質に影響を及ぼすこともある。

#### 【0022】

ページを形成するコンポーネントを、グラフィカルユーザインターフェイス（GUI）コンポーネントと呼んでもよい。GUIコンポーネントは、表示画面の状況に応じて、固定およびアニメ化が可能であり、及び/またはインタラクティブであるプレゼンテーションの表示機能である。GUIコンポーネントの典型例は、グラフィカルラベル、グラフィカルボタン、グラフィカルチェックボックス、グラフィカルテキストフィールド、グラフィカルテキストエリア、グラフィカルリスト/テーブル、グラフィカルバックグラウンド、データ入力フィールド、プルダウンメニュー、及び/または表示画面上に表示する機能を兼ね備えているものである。GUIコンポーネントは、固有の識別子と一緒にGUIライブラリに保存されてもよい。この識別子により、例えば、既に述べられているアプリケーションGUIは、GUIコンポーネントでプレゼンテーションを作る。

10

#### 【0023】

プレゼンテーションは、表示画面のサイズの関数として、複数ページに細かく分割されてもよい。例えば、ノートブックPCのような相対的に大きな表示画面は、大きなサイズ、及び/またはたくさんの量のGUIコンポーネントを含んだ大きなページを表示することができる。一方、例えば、無線電話のような小さな画面は、かなりコンパクトなGUIコンポーネントを持った小さなページのみ表示することができる。さらに、小さなページは、表示画面上において、一度にほんの少しのGUIコンポーネントしか表示できない。

20

#### 【0024】

SGUIシステムは、異種装置プラットフォームの表示画面を最大限に埋めるために用いることもできる。中間表現を用いることにより、SGUIシステムは、ダイナミックなレイアウトとアプリケーションGUIのグラフィカルユーザインターフェイス（GUI）変換を提供できる。このようにアプリケーションGUIは、拡張性のあるアプリケーションが動いている異種装置プラットフォームの表示画面に拡大縮小されてもよい。

#### 【0025】

##### ・入力方法

異種装置プラットフォームは、また拡張性のあるアプリケーションとアプリケーションGUIに対応付けられている異なる入力方法を兼ね備えていてもよい。例えば入力方法として、無線電話はキーパッド、ポケットPCは手書き入力用のペン、ノートブックPCはキーボード、及び/またはマウスのようなポインティングデバイスを利用する。従って、拡張性のあるアプリケーションのアプリケーション開発者は、対応しているアプリケーションGUI用特定の入力方法を採用しなくてもよい。

30

#### 【0026】

SGUIシステムは、異種装置プラットフォームに対応できる装置プラットフォーム独立型のGUIイベントを提供する。GUIイベントは、ページ上の視覚的な素材内における状態の変化である。ユーザが、異種装置プラットフォームのユーザインターフェイスと対話することにより、GUIイベントを引き起こされることもある。SGUIシステムにより、装置プラットフォーム特定のGUIイベントを、装置プラットフォーム独立のGUIイベントに、動的に変換される。このように、SGUIシステムを利用している拡張性のあるアプリケーションは、異種装置プラットフォームのユーザインターフェイスと適合性があり、かつ異種装置プラットフォームのユーザインターフェイスから独立しているGUIイベント機能を持つように設計される。

40

#### 【0027】

##### ・ユーザインターフェイスライブラリ

異種装置プラットフォームは、また異なるアプリケーションプログラムインターフェイス（API）をサポートしてもよい。APIは、広範囲にわたる機能を提供する出来合いのソフトウェアコンポーネントの集まりの1つである。例えば、Java（登録商標）技術は、Java（

50

登録商標) プリケーションプログラムインターフェイス (API) を持っている。Java (登録商標) (API) は、Java (登録商標) 型プログラム内で実装される標準的な機能を提供するソフトウェアコンポーネントの集まりである。

【 0 0 2 8 】

Java (登録商標) プラットフォームは、「一度書いたらどこでも動く (Write Once, Run Anywhere; WORA)」を主張しているが、移動装置環境で開発しているJava (登録商標) アプリケーションにとって、これは当てはまらないこともある。例えば無線電話で、Java (登録商標) プログラミング言語は、Java (登録商標) 2 Micro Edition (J2ME) であってもよい。基本的にJ2MEのデザインは、柔軟性と分断に基づいている。柔軟性を成し遂げるため、J2MEは装置プラットフォーム特定型のAPIを設定している。装置プラットフォーム特定型のAPIは、プロファイルとして言及される。J2MEアプリケーションは、特定の装置プラットフォームで動くように書かれているので、異なるプロファイルは分断してしまう。例えば、ドコモ503iの無線電話はJ2ME DoJa APIとして言及されるJava (登録商標) プロファイルをサポートしてもよい。同様に、ポケットPCはPersonal Java (登録商標) AWT APIを、ノートブックパソコンはJava (登録商標) 2 Standard Edition (J2SE) SWING APIをサポートしてもよい。

10

【 0 0 2 9 】

APIは、異種装置プラットフォームによって、異なるGUIライブラリをサポートしてもよい。例えば、PDAは、対応しているGUIライブラリ内において、より多くのタッチスクリーン関連のGUIコンポーネントを兼ね備えていてもよい。一方、例えば無線電話は、対応しているGUIライブラリ内において、指示関連のGUIコンポーネントしか兼ね備えていてなくてもよい。既に知られている通り、GUIライブラリは、プレゼンテーションを作成するためにアクセス可能な既成のGUIコンポーネントを編集したものである。

20

【 0 0 3 0 】

1つの実施形態のSGUIシステムは、異種装置プラットフォームと適合性があり、かつその上でサポートされる装置プラットフォーム独立型APIを提供する。装置プラットフォーム独立型のAPIは、異種装置プラットフォームに対応付けられている異なるGUIライブラリへ、ブリッジング (橋渡し) 機能を提供する。したがって、SGUIシステムを利用しているアプリケーション開発者は、異種装置プラットフォーム用のプラットフォーム特定型のGUIプレゼンテーションを開発するため、異なるGUIライブラリに精通している必要はない。

30

【 0 0 3 1 】

SGUIシステムの典型的な動作例は、拡張性のあるアプリケーションがある装置プラットフォームから別の装置プラットフォームへマイグレーションすることである。拡張性のあるアプリケーションのマイグレーションは、ソース (供給元) 装置プラットフォームからターゲット装置プラットフォームへ、動作中の拡張性のあるアプリケーションのインスタンスのマイグレーションを含んでいる。例えば、会社でマウスやキーボードを使い、ラップトップPCで、Eメールに関連している拡張性のあるアプリケーションを操作しているユーザを考えてください。ここで仕事の完了前に、ユーザは会社を出る必要がある。この時にユーザは、まだアクティブな拡張性のあるアプリケーションを、ラップトップPC (ソース (供給元) 装置プラットフォーム) からPDA (ターゲット (目的) 装置プラットフォーム) へと動かすことにより、Eメール作業機能を続行できる。ラップトップPCの表示画面はPDAより大きいので、アプリケーションのプレゼンテーションは、SGUIシステムによりPDAの表示画面に適合される。これに伴い、ラップトップPCのマウスやキーボードで動いていたユーザインターフェイス機能もまた、SGUIシステムにより、PDA上にあるタッチスクリーンのユーザインターフェイス機能に適合されてもよい。SGUIシステムは、ソースおよびターゲット装置プラットフォームから独立しているユーザインターフェイスの拡張性のあるアプリケーションの移動性を提供する。このように拡張性のあるアプリケーションは、SGUIシステムを使いながら動作する時にだけ必要であり、特に2つの装置プラットフォーム上で動作する時に必要なわけではない。

40

【 0 0 3 2 】

50



図 1 は、拡張性のあるグラフィカルユーザインターフェイス(SGUI)システム 10 のある実施形態のブロック図である。SGUIシステム 10 は、拡張性のあるGUIライブラリモジュール 12、カスタマイズモジュール 14、そしてレンダーマネージャモジュール 16 を持っている。図に描かれた実施形態は、本発明の理解を助けるためのものであり、これらの装置は個々に独立したものではなく、同図より少ないまたは多い装置が、SGUIシステム 10 を説明している様々な実施形態で利用されてもよい。

【0033】

拡張性のあるGUIライブラリモジュール 12 は、拡張性のあるアプリケーションの開発中において、アプリケーションの開発者が使用するツールであってよい。さらに、拡張性のあるGUIライブラリモジュール 12 は、SGUIシステム 10 による中間表現と共に使用するため、GUIコンポーネントのライブラリを提供してもよい。ある実施形態において、装置プラットフォーム独立型APIは、拡張性のあるアプリケーションの開発中において使用されてもよい。また中間表現を作るため、装置プラットフォーム独立型APIは、装置プラットフォーム独立型のアプリケーションGUI内に、アプリケーション開発者により実装されてもよい。したがって、装置プラットフォーム独立型のアプリケーションGUIを使用して中間表現のインスタンス化には、拡張性のあるGUIライブラリモジュール 12 内において、装置プラットフォーム独立型のAPIを実行することが含まれる。中間表現のインスタンス化は、中間表現は装置プラットフォーム独立型のプレゼンテーションモデルということを表している図 1 において、「装置独立型中間表現」となっている。

【0034】

一般的に、拡張性のあるGUIライブラリモジュール 12 は、Java(登録商標)アブストラクト ウィンドウ ツールキット(AWT)のようなGUIライブラリの他のフォームに似ている。本発明の好ましい実施形態において、拡張性のあるGUIライブラリモジュール 12 は、Java(登録商標)のアプリケーションGUIに似たスタイルで作られている。従って、Java(登録商標)技術に詳しいアプリケーションプログラマーが、装置プラットフォーム特定型APIを利用する現存の装置プラットフォーム特定型のアプリケーションGUIを、SGUIシステム 10 内において、装置プラットフォーム独立型のAPIを利用する装置プラットフォーム独立型のアプリケーションGUIに変換する要求は少なくなる。

【0035】

カスタマイズモジュール 14 は、ターゲット装置プラットフォームの装置独立の中間表現をカスタマイズするために動作する。「ターゲット装置プラットフォーム」という語は、拡張性のあるアプリケーションがディスプレイ用のプレゼンテーションを生成する、異種装置プラットフォームを特定する。ターゲット装置プラットフォームとして、拡張性のあるアプリケーションが特定された装置プラットフォーム上を動作している間に、プレゼンテーションは生成されてもよい。さらに拡張性のあるアプリケーションが、ソースからマイグレーションの目的である装置プラットフォーム(ターゲット装置プラットフォーム)へマイグレーションする時、プレゼンテーションが生成されてもよい。ある実施形態におけるカスタマイズモジュール 14 は、タスクマネージャモジュール 18 と変換マネージャモジュール 20 を持っている。

【0036】

ある実施形態におけるタスクマネージャモジュール 18 は、ターゲット装置プラットフォームの機能に基づいて装置独立型の中間表現の機能を取り除いてもよい。その中間表現は、タスクマネージャモジュール 18 によりタスクが取り除かれたことを表している「取り除かれた中間表現」と図 1 に示されている。ある実施形態において、拡張性のあるアプリケーションは、ターゲット装置プラットフォームの機能を特定し、そしてその機能をタスクマネージャモジュール 18 へ提供する。またある実施形態において、タスクマネージャモジュール 18 は、ターゲット装置プラットフォームの機能を特定することができる。

【0037】

中間表現から取り除かれたタスクは、ターゲット装置プラットフォームに適合しない拡張性のあるアプリケーションの代表的な機能/能力を持っていてもよい。例えば、ラップト

10

20

30

40

50

ップパソコンは、電話のテンキー機能に関連しているタスクは使わない。同様に、利用できる能力に制約があるため、ターゲット装置プラットフォーム上に全てのタスクを表示することは望ましくないとはいえる。例えば、多量のテキストデータを入力するのに、無線電話のテンキーを使うのは面倒くさい上、時間もかかる。だからタスクマネージャモジュール 18 は、無線電話にとっては面倒である編集機能に関連したタスクを取り除いてもよい。ある実施形態において、タスクマネージャは、ターゲット装置プラットフォームの機能、及び/または中間表現内におけるアプリケーション開発者により特定された他のプロパティに応じて、タスクを取り除いてもよい。

#### 【0038】

変換マネージャモジュール 20 は、中間表現を装置プラットフォーム依存型のプレゼンテーションに変換するように動く。装置プラットフォーム依存型のプレゼンテーションは、ターゲット装置プラットフォーム内で、特定のユーザインターフェイスに適合してもよい。中間表現の変換は、タスクマネージャモジュール 18 が不要なタスクを取り除くことにより始まる。中間表現の変換には、ターゲット装置プラットフォームの機能ベースの中間表現、中間表現の論理構造、及び/またはアプリケーションGUIが特定したプロパティを動的に変更することが含まれる。以下における変換マネージャモジュール 20 による変換において、中間表現は装置プラットフォーム依存型の中間表現であり、この装置プラットフォーム依存型の中間表現は、プラットフォーム特定のプレゼンテーション方式における特定のある異種装置プラットフォーム(ターゲット装置プラットフォーム)に対するカスタマイズを示すために「装置依存型中間表現」として図 1 に表されている。

#### 【0039】

レンダーマネージャモジュール 16 は、ターゲット装置プラットフォームの表示画面上にプレゼンテーションを表示するようにしてもよい。レンダーマネージャモジュール 16 は、カスタマイズモジュール 14 による中間表現のカスタマイズを行う。レンダーマネージャモジュール 16 は、装置プラットフォーム依存型の中間表現から、プレゼンテーションを取り出してもよい。さらにレンダーマネージャモジュール 16 は、ターゲット装置プラットフォームの表示画面上にプレゼンテーションを表示するために、拡張性のあるGUIライブラリモジュール 12 を利用してもよい。

#### 【0040】

レンダーマネージャモジュール 16 の動作は、プレゼンテーションを取り出すため、装置プラットフォーム依存型の中間表現をトラバースすることを含む。装置プラットフォーム依存型の中間表現は、ターゲット装置プラットフォームに依存している装置プラットフォーム用にカスタマイズされたアプリケーションGUIの符号化表示である。このように、レンダーマネージャモジュール 16 により取り出されたプレゼンテーションは、ターゲット装置プラットフォームの表示画面上に写っている装置依存型のアプリケーションGUIを表示する。

#### 【0041】

SGUIシステム 10 の好ましい実施形態において、SGUIシステム 10 は、ターゲット装置プラットフォームに対しアプリケーションGUIの継ぎ目のないスケーリングを提供するため、拡張性のあるアプリケーションと連動して動いてもよい。アプリケーション開発者は、アプリケーションGUIから装置プラットフォーム独立型の中間表現を作るためのツールとして、拡張性のあるGUIライブラリモジュール 12 を使うことができる。図 1 に示すように、SGUIシステム 10 の稼動中、アプリケーションGUIは、拡張性のあるGUIライブラリモジュール 12 を使う装置プラットフォーム独立型の中間表現をインスタンス化してもよい。カスタマイズモジュール 14 は、拡張性のあるアプリケーションが動いているターゲット装置プラットフォームに基づく装置プラットフォーム依存型の中間表現に対し、中間表現をカスタマイズしてもよい。レンダーマネージャモジュール 16 は、ターゲット装置プラットフォームのユーザインターフェイス用にカスタマイズされたプレゼンテーションを取り出すため、装置プラットフォーム依存型の中間表現を使用してもよい。そしてプレゼンテーションは、レンダーマネージャモジュール 16 により、ターゲット装置プラット

ームの表示画面上に表示されてもよい。

【 0 0 4 2 】

図 1 に表されているモジュールのより細かな詳細について説明する。

【 0 0 4 3 】

図 2 は、拡張性のあるGUIライブラリモジュール 1 2 の 1 実施形態のより詳細なブロック図である。ある実施形態の拡張性のあるGUIライブラリモジュール 1 2 は、拡張性のあるGUIコンポーネントライブラリモジュール 2 2、中間表現モジュール 2 4 (IR)、そして拡張性のあるGUIイベント変換モジュール 2 6 を有している。他の実施形態において、これより少ないまたは多いモジュールにより、拡張性のあるGUIライブラリモジュール 1 2 の機能を表してもよい。拡張性のあるGUIコンポーネントライブラリモジュール 2 2 は、有名なGUIコンポーネントライブラリに共通している点がいくつかある。GUIコンポーネントライブラリモジュール 2 2 のある実施形態において、GUIコンポーネントライブラリモジュール 2 2 は、複数の拡張性のあるグラフィックユーザインターフェイス(SGUI)コンポーネントと装置プラットフォーム独立型のAPIを有している。既に述べたように、装置プラットフォーム独立型のAPIは、異種装置プラットフォームと適合性があり、そして装置独立型のアプリケーションGUIを作るために利用できる。

【 0 0 4 4 】

SGUIコンポーネントは、ライブラリ内におけるグラフィカルユーザインターフェイスコンポーネントである。SGUIコンポーネントは、SGUIシステムで動作可能な異種装置プラットフォームによりサポートされる装置独立型のグラフィカルユーザインターフェイスコンポーネントとして描かれている。さらにSGUIコンポーネントは、グラフィカルユーザインターフェイスコンポーネントが、異種装置プラットフォーム用にカスタマイズ可能なことを表すために用いられてもよい。またSGUIコンポーネントは、SGUIコンポーネント複合として言及されてもよい。ここでSGUIコンポーネント複合は、関連のある機能を提供するためにグループ化された複数のSGUIコンポーネント単体である。

【 0 0 4 5 】

装置独立型のアプリケーションGUIは、SGUIコンポーネントの表示を含む装置独立型の中間表現のインスタンスを作るのに使われてもよい。しかしながらディスプレイ用のプレゼンテーションを作る前に、中間表現は、装置プラットフォーム依存型の中間表現が表示可能になるように、変換(またはカスタマイズ)されてもよい。

【 0 0 4 6 】

好ましい実施形態において、GUIコンポーネントライブラリモジュール 2 2 は、Java(登録商標) SWINGに似ている。これにより、Java(登録商標)スイングに明るいアプリケーション開発者は、拡張性のあるGUIコンポーネントライブラリモジュール 2 2 を容易に学ぶことができる。ある実施形態において、Java(登録商標)スイングとSGUIコンポーネントの類似点は、共に接頭辞がSだということである。つまり、この実施形態でJava(登録商標)SWINGの接頭辞は、Java(登録商標)SWINGのJではなく、Sなのである。

【 0 0 4 7 】

一般的に、クロスプラットフォームGUIライブラリを作るために、4つの有名な技術がある。それら4つの技術とは「Least Common Denominator (LCD)」、 「Superset Emulation」、 「Direct API Emulation」 and 「Layering/Wrapper」である。LCD技術は、全てのGUIライブラリに共通の特徴を持っている。Superset Emulation技術は、全てのGUIライブラリ用のGUIコンポーネントのスーパーセットを提供できる。ダイレクトAPIエミュレーションの技術は、直接的に、あるGUIライブラリの装置特定型APIを別のGUIライブラリの装置特定型APIへマッピングできる。レイヤリング/ラッパー技術は、GUIライブラリへマッピングする、アブストラクトなGUIライブラリを提供する。

【 0 0 4 8 】

SGUIシステムのある実施形態において、ダイレクトAPIエミュレーションとラッパー技術のコンビネーションが実装される。他の実施形態において、他の技術との組み合わせが使われてもよい。しかしながら、LCD技術の利用は、SGUIコンポーネントの利用を制限する

こともある。さらに、SGUIシステムがサポートするGUIライブラリの数が増えた時、スーパーセットエミュレーションの利用は、拡張性を制限することもある。

【 0 0 4 9 】

ダイレクトAPIエミュレーションとラッパー技術の組み合わせが、拡張性のあるGUIコンポーネントライブラリモジュール 2 2 により実装されてもよい。拡張性のあるGUIコンポーネントライブラリモジュール 2 2 は、他のGUIライブラリへマッピングされてもよい。ある実施形態において、拡張性のあるGUIコンポーネントライブラリモジュール 2 2 は、装置特定のGUIライブラリへマッピングされてもよい。装置特定のGUIライブラリは、例えば、Java（登録商標）スイング、Java（登録商標）AWT、及び/またはJava（登録商標）Dojaを有していてもよい。他の実施形態において、拡張性のあるGUIコンポーネントライブラリモジュール 2 2 は、非装置特定のライブラリを有している増設された/異なるライブラリへマッピングされてもよい。

10

【 0 0 5 0 】

拡張性のあるGUIコンポーネントライブラリモジュール 2 2 内でのマッピングにおいて、SGUIコンポーネントを、他のライブラリ内にある装置特定のGUIコンポーネントへマッピングする。このマッピングは、対応する装置特定のGUIコンポーネントが装置特定のGUIライブラリ内において利用可能な場所で実行されてもよい。もし装置特定のGUIライブラリが利用できないとき、SGUIコンポーネントは、装置特定のGUIライブラリ内において、装置特定のGUIコンポーネント複合へマッピングされてもよい。ある実施形態において、マッピングは定着してない。したがって、この実施形態において、アプリケーション開発者は、拡張性のあるGUIコンポーネントライブラリモジュール 2 2 内においてSGUIコンポーネントのマッピングを変えたり/加えたりすることにより、デフォルトマッピングを無視してもよい。

20

【 0 0 5 1 】

再び図 2 に戻るが、中間表現(IR)モジュール 2 4 は、中間表現を作るためにアプリケーションGUIが使用する装置独立型APIを有している、いかなるアプリケーションでもよい。中間表現は、拡張性のあるGUIコンポーネントライブラリモジュール 2 2 を利用する装置独立型アプリケーションGUIを表している。好ましい実施形態において、中間表現は木構造のような構造を有しているので、だから中間表現は中間表現(IR)ツリー(木)と呼ばれてもよい。他の実施形態において、例えば、アウトライン構造、マトリックス構造、または他の論理ベースの構造を、中間表現を表すために用いてもよい。

30

【 0 0 5 2 】

IRツリーは、SGUIコンポーネントが配置されているかもしれない論理構造を提供する。SGUIコンポーネントは、IRツリー内の簡単な中間表現内に装置プラットフォーム独立型のアプリケーションGUIを入れることにより、配置されてもよい。アプリケーションGUIは装置プラットフォーム独立型なので、IRツリーは拡張性のあるアプリケーションによりサポートされる、すべての可能なSGUIコンポーネントを含んでいてもよい。SGUIコンポーネントは、装置プラットフォーム独立型のプレゼンテーションへ簡単な変換を提供するように配置されてもよい。さらに詳細は後で述べるが、レイアウト構造、レイアウトの制約、そして制約を示すその他のプロパティは、IRツリーの論理構造内において特定されてもよい。さらに、IRツリーは、Java(登録商標)のアプリケーションGUI開発との類似を永続させるため、Java(登録商標)技術の階層的なコンテナ概念を取り入れることが望ましい。

40

【 0 0 5 3 】

図 3 は、IRツリー 4 0 の構成例を示すブロック図である。IRツリー 4 0 は、階層型構造内にある相互に連結した複数のノードを持っている。図の階層構造は、数ある構造の中における 1 つの例に過ぎない。好ましい実施形態において、IRツリー 4 0 は、少なくとも 1 つのコンポーネントノード 4 2 と 1 つのコンテナノード 4 4 を持っている。他の実施形態において、その他のカテゴリーを持つノードが、アプリケーションGUIを表すように特定されてもよい。

【 0 0 5 4 】

50

ある実施形態のコンポーネントノード42は、個々のSGUIコンポーネントを表している。コンポーネントノード42より下の位置には他のノードがこれ以上無いので、コンポーネントノード42は、IRツリー40の「葉」と見なしてもよい。ある実施形態のコンテナノード44は、IRツリー40内における論理パネルを表している。論理パネルは、装置プラットフォームの表示画面に表示するためIRツリー40から作られたプレゼンテーションのページ内で、異なるセクション、または異なるパネルを表している。例えば、Eメールに関するアプリケーション内において、コンテナノード44は、ユーザが受け取った全てのEメールを表示するウィンドウを表すことができる。

#### 【0055】

図に示す通り、コンテナノード44は、下位の位置に自身と連結しているコンポーネントノード42、及び/または他のコンテナノード44を持っていてもよい。IRツリー40の下位に位置しているノードは、上位の関連論理パネルにより表されるセクションの一部またはパネルである、関連サブ論理パネル及び関連SGUIコンポーネントを表している。例えば、Eメールを表示するウィンドウにおいて、下位にあるコンテナノード44は、ウィンドウ内で記述的なヘッダーを提供してもよいし、そして接続先のコンポーネントノード42は、受信したメールを分類する機能を提供してもよい。

#### 【0056】

図3のIRツリー40における例で、最も高い位置にあるルートコンテナノード44aは、全体的なプレゼンテーションを表してもよい。その他のコンテナノード44は、プレゼンテーション内で異なる論理パネルを表しているので、ルートコンテナノード44aの階層的な子供と考えてよい。コンテナノード44c、44d、44e、44g、そして44hは、ルートコンテナノード44aの一番下位のコンテナノードであると言える。逆に言えば、コンテナノード44b、44c、44dは、ルートコンテナノード44aの直系の階層的な子供である。さらに図にあるように、いくつかのコンテナノード44の階層的な子供達はコンポーネントノード42である。例えば、コンテナノード44c、44dの子供は、それぞれコンポーネントノード42g、42hとコンポーネントノード42j、42kである。

#### 【0057】

またIRツリー40は、拡張性のあるアプリケーションのタスク構造でもよい。タスク構造は、拡張性のあるアプリケーションが実行するタスクに関連するSGUIコンポーネントをグループ化することにより、表示されてもよい。タスクという言葉は、装置プラットフォームのユーザが、拡張性のあるアプリケーションで実行できる機能を特定している。機能は、拡張性のあるアプリケーションと装置プラットフォームに依存している。例えば、無線電話機(装置プラットフォーム)におけるEメール機能(拡張性のあるアプリケーション)は、タッチスクリーンを使いながらEメールを見ることが、削除すること、そして作成することができる機能を有している。したがって、ユーザの指示を示し、かつ受け入れるSGUIコンポーネントは、機能別にグループ化される。

#### 【0058】

タスク構造は、特定のタスク(機能)に関するSGUIコンポーネント(コンポーネントノード42)を、論理パネル(コンテナノード44)内にグループ化する。コンテナノード44内によるグルーピングにより、装置プラットフォームのディスプレイ上におけるSGUIコンポーネントの対応するグループが表すタスクを表示することができる。例えば、図3において、タスク46はコンテナノード44e、そしてその階層的な子供であるコンポーネントノード42aと42bである。一方、タスクは、複数のコンテナノード44と、それに対応するコンポーネントノード42である。

#### 【0059】

ある実施形態において、SGUIコンポーネントのグルーピングにより、アプリケーションGUIがタスクの優先度を示している制約を提供するためのプロパティを特定する。プロパティは、対応するコンテナノード44に対し、タスクの優先順位を添付してもよい。例えば、タスクの優先順位は、特定の装置プラットフォームに適しているタスクを特定しても良

10

20

30

40

50

いし、異種装置プラットフォームにおける別のタスクを特定しても良いし、及び/または、タスク及び/または拡張性のあるアプリケーションが動いている装置プラットフォームに関連する情報を特定しても良い。

#### 【0060】

既に述べたように、アプリケーション開発者は、その開発中に、拡張性のあるアプリケーション内で、IRツリー40を作ってもよい。コンテナノードとコンポーネントノードの階層構造を特定する他に、アプリケーション開発者は、IRツリー40の論理構造、及び/またはIRツリー40のレイアウトパラメータに影響する制約を提供するプロパティを特定してもよい。プロパティは、コンポーネントノード42(SGUIコンポーネント)、コンテナノード44(論理パネル)、及び/またはIRツリー40に対し、特定されてもよい。IRツリー40のインスタンスは、特定のプロパティを含んでもよいし、また特定のプロパティベースで作られてもよい。好ましい実施形態において、プロパティの特定化は、アプリケーションGUI内において起こってもよい。また他の実施形態において、プロパティの特定化は、ターゲット装置プラットフォーム内、または他の関連ある装置またはアプリケーション内の拡張性のあるアプリケーションにおいて起こり得る。

10

#### 【0061】

IRツリー40は、プレゼンテーションのページ内(プレゼンテーション部)において、アプリケーションGUIのレイアウトを表してもよい。ある実施形態において、IRツリー40内の相互関連のパターンによりレイアウトストラクチャは、直系の階層的子供ノードにより、高い水準の論理パネル用に特定されてもよい。アプリケーション開発者は、アプリケーションGUIで、コンポーネントノード42とコンテナノード44用のレイアウト構造を示しているプロパティを特定してもよい。プロパティは、論理パネル(コンテナノード44)とコンテナノード44の階層的な子供であるSGUIコンポーネント(コンポーネントノード42)の位置を特定してもよい。例えば、コンポーネントノード42aにおけるレイアウト構造は、コンテナノード44eが表す論理パネルの右上において、SGUIコンポーネントの位置を特定してもよい。同様に、コンポーネントノード42bは、コンテナノード44eが表す論理パネルの左上において、対応するSGUIコンポーネントの位置を示しているレイアウト構造を含んでいてもよい。コンテナノード44eも、コンテナノード44bにより表される論理パネルのセンターにおいて、対応する論理パネルの位置を特定するレイアウト構造を含んでいてもよい。

20

30

#### 【0062】

さらに、図3のIRツリー40の例において、プレゼンテーションのページ48は、コンテナノード44とコンポーネントノード42の階層的に関連のあるグループから形成されていてもよい。図の例において、ページ48はコンテナノード44bで形成されている。ページ48は、コンテナノード44b以下の関連のある階層レベルが課したレイアウト構造に起因してもよい。具体的に説明すると、44f内におけるコンテナノード44g、44h、そして44b内における44e、44fの特定化と同様に、対応するコンテナノード44e、44g、44h内におけるコンポーネントノード42a、42b、42c、42d、42e、42fのレイアウト構造を特定するプロパティが、ページ48を作ることにもできる。他の実施形態において、他のレイアウト構造の特定化に関連するIRツリー40における他のノードの階層構造は、ページのフォーメーション(形成)に起因してもよい。

40

#### 【0063】

ターゲット装置プラットフォーム用のプレゼンテーションのページの生成時、SGUIシステムは、提案されているレイアウト構造に従ってもよい。しかしながら、ある実施形態において、例えば、ターゲット装置プラットフォームの表示画面を最大限に利用することに高い優先順位が設定されている場合において、提案されたレイアウト構造に従うことを無視してもよい。

#### 【0064】

IRツリー40の装置独立型プラットフォームが有する性質のせいで、アプリケーションGUIが生成されたときに、異種装置プラットフォーム内における多様な機能は、レイアウト

50

構造の特定化を難しくする。例えば、許容されるページの最大のサイズは、ターゲット装置プラットフォームの表示画面のサイズに依存しており、開発時間を特定するのは難しい。さらに、開発時間においては、SGUIコンポーネントの実際サイズはわからない。SGUIコンポーネントは、ターゲット装置プラットフォームの機能に基づき、多種あるサイズの中から1つのSGUIコンポーネントに変換するので、SGUIコンポーネントの実際サイズはわからない。

#### 【0065】

ある実施形態において、アプリケーション開発者は、プレゼンテーション内において配置されたSGUIコンポーネントに関するSGUIコンポーネントのサイズを予測することは可能である。仮に、その予測が正確(SGUIコンポーネントの変換後のサイズに近い)なら、ターゲット装置プラットフォームの表示画面内のレイアウトにおけるSGUIコンポーネント間の大きな「穴」は、避けることができる。また別の実施形態において、SGUIシステムは、いわゆる「穴」を避けるために、同じコンテナノード44(論理パネル)内で、SGUIコンポーネントに似たサイズを保存するように努める。また他の実施形態において、自動レイアウト生成のためのフォーム、ターゲット装置プラットフォーム用レイアウトのアプリケーション開発者によるカスタマイズ、及び/またはSGUIコンポーネントのサイズの多様性を指定するための技術により、多様なレイアウト構造は指定される。

#### 【0066】

タスクの優先順位とレイアウト構造を示すプロパティの特定化に加えて、アプリケーション開発者は、IRツリー40の論理構造、及び/またはIRツリー40のレイアウトパラメータに影響する制約を提供するプロパティを特定してもよい。ある実施形態において、コンテナノード44のレイアウト優先順位に関連しているプロパティが特定される。本実施形態のプロパティは、コンテナノード44がページ上に配置されるオーダを特定する。最も高いレイアウト優先順位のコンテナノード44は、表示画面内で最初のページに位置し、かつ対応するコンポーネントノード42により特定されるSGUIコンポーネントを保持してもよい。例えば、装置プラットフォームのユーザのために、あるタスクを易しくしたいというアプリケーション開発者は、対応するパネル(コンテナノード44)に、高いレイアウト優先順位を設定する。

#### 【0067】

他の実施形態において、レイアウトの制約を特定するプロパティは、コンポーネントノード42に含まれてもよい。レイアウトの制約は、同ページに互いに隣接して配置される2つまたはそれ以上のSGUIコンポーネントを条件として指定してもよい。例えば、イエスを示している選択可能なボタンとノーを示している選択可能なボタンは、ボタンが隣接していることを条件として指定するレイアウトの制約を含んでもよい。

#### 【0068】

同様に他の実施形態で、プロパティは、コンテナノード44内のレイアウトグループを特定してもよい。レイアウトグループは、コンテナノード44内のノードが、レイアウト構造内で別ページに配置されるかどうかを条件として指定することもできる。ある実施形態において、直系の階層的な子供としてコンポーネントノード42を持っている全てのコンテナノード44は、レイアウトグループに特定されない限り分割できない。同様に、直系の階層的な子供としてコンポーネントノード42を持っていない本実施形態の全てのコンテナノード44は分割可能である。

#### 【0069】

別の実施形態において、プロパティはコンテナノード44のラベルを特定できる機能を持っている。ラベルは、論理パネルに対応付けられてもよいし、ページ間を移動するプレゼンテーション内において、ナビゲーションバー/またはメニューを作成してもよい。他の実施形態においてプロパティは、他の仕様や、論理コンディション、及び/またはコンポーネントノード42、コンテナノード44及び/またはIRツリー40に関連する付加機能を特定するのに利用されてもよい。

#### 【0070】

10

20

30

40

50

再び図2に戻り、拡張性のあるGUIイベント変換モジュール26は、ターゲット装置プラットフォームのユーザインターフェイスにより生成されるGUIイベントを変換する技術(方法)であってもよい。ある実施形態において、拡張性のあるGUIイベント変換モジュール26は、装置独立型のGUIイベントAPIを持っている。装置独立型のGUIイベントAPIは、拡張性のあるアプリケーションに対するイベントを処理する。前にも述べたがGUIイベントは、表示画面上のSGUIコンポーネント内の状態の変化を表しているが、これはユーザが対応する装置プラットフォームに関連しているユーザインターフェイスと対話するためである。

#### 【0071】

先行技術では、GUIイベントは、装置特定型アプリケーションGUIアプリケーションプログラムインターフェイス(API)により定義されている。装置特定型GUI APIは、GUIイベントを生成する、特定装置の入出力方法に接続されているGUIイベントを定義する。拡張性のあるGUIイベント変換モジュール26は、異種装置プラットフォーム用のGUIイベントを変換するのに使われる。また拡張性のあるGUIイベント変換モジュール26は、異なる複数の入力方法で動作が可能な装置独立型のGUIイベント変換機を有している。

#### 【0072】

ある実施形態において、GUIイベント変換機は、アクションと呼ばれる抽象的なイベントを定義する。アクションは入力方法から独立している。例えば、ボタンとして動作するGUIコンポーネントに対応付けられたアクションは、装置プラットフォームに依存しているラップトップコンピュータのマウスクリック、ポケットコンピュータのスタイラス・タップ、または携帯電話のソフトキーから生成されてもよい。拡張性のあるアプリケーションは、GUIイベント変換機のGUIイベントを表すアクションにより提供されてもよい。このように、GUIイベントを生成する入力方法は、拡張性のあるアプリケーションに対し、無関係なものであってもよい。

#### 【0073】

拡張性のあるGUIイベント変換モジュール26は、プラットフォーム独立型のアクションの拡張性のあるユーザインターフェイスライブラリを持っている。拡張性のあるユーザインターフェイスライブラリは、拡張性のあるGUIコンポーネントライブラリ22、または異なるライブラリに含まれていてもよい。ある実施形態において、拡張性のあるGUIイベント変換モジュール26は、装置プラットフォーム独立型のGUIイベントAPIに基づいた論理マッピングを含んでいてもよい。拡張性のあるアプリケーションの開発者は、装置プラットフォーム特定型のGUIイベントと同等のプラットフォーム独立型のアクションの間で、論理マッピング用の装置プラットフォーム独立型のAPIを利用してもよい。このようにプラットフォーム独立型のアクションは、対応する装置プラットフォーム特定型のGUIイベントの結果として、拡張性のあるアプリケーションへと配信されてもよい。

#### 【0074】

いくつかの拡張性のあるアプリケーションにおいて、装置プラットフォームの制約のために、プラットフォーム特定型のGUIイベントとプラットフォーム独立型のアクション間におけるマッピングは不可能かもしれない。例えば、ユーザがマップ・イメージにおける特定のポイントをクリックするとき、拡張性のあるアプリケーションは、拡大または縮小できるインターラクティブマップを有していてもよい。パソコンサイドでは、装置プラットフォーム特定型のマウスイベントは、マウスがクリックされた特定の位置を把握する。しかしながら、無線電話において、ユーザインターフェイスとして位置決め入力装置は含まなくてもよい。このように、無線電話における同等のプラットフォーム独立型のアクションは、マップされることはない。この場合、拡張性のあるGUIイベント変換モジュール26のライブラリは、別段の定めがなければプラットフォーム独立型アクションが表す、プラットフォーム特定入力方法用の装置プラットフォーム依存型アクションを有していてもよい。これに伴い、アプリケーションGUIは、制約を伴うターゲット装置プラットフォーム用のアプリケーションGUIをカスタマイズするプロパティを持っていてもよい。ある実施形態において、一般的に装置プラットフォーム特定型GUIイベントは拡張性がないた

10

20

30

40

50



め、拡張性のあるアプリケーションの主要な機能を実装するのに装置プラットフォーム独立型のアクションを用いなくてもよい。本実施形態において、装置プラットフォーム依存型のアクションは、主要な機能に対する付加拡張機能として用いられてもよい。

【0075】

図4は、変換マネージャモジュール20のより詳細なブロック図における1つの実施形態である。変換マネージャモジュール20は、変換ルールモジュール50、サイズ変換コンポーネントモジュール52、代替コンポーネントモジュール54、GUI変換ルールモジュール単体56、GUI変換ルールモジュール複合58、ダイナミックレイアウトモジュール60、スタイルガイドモジュール62とSGUIコンポーネントリストモジュール64から成っている。他の実施形態において、変換マネージャモジュール20内での機能は、本実施形態で使用されているモジュールより少ない、または多い数のモジュールで表される。

10

【0076】

変換ルールモジュール50は、IRツリーを動的に作るために、拡張性のあるGUIコンポーネントライブラリモジュール22(図2)を利用してもよい。IRツリーの構造は、対応するプレゼンテーションのページにあるIRツリー内に表されるSGUIコンポーネントと、変換ルールで選択的に変換するSGUIコンポーネントから成っている。

【0077】

変換ルールモジュール50は、IRツリー40(図3)におけるSGUIコンポーネントを変換するのに使われる複数の変換ルールから成っている。ある実施形態において、変換ルールモジュール50は、変換ルールの2つのカテゴリを表しているサイズ変換コンポーネントモジュール52と、代替コンポーネントモジュール54から成っている。サイズ変換コンポーネントモジュール52は、ターゲット装置プラットフォームの表示画面のサイズをベースにSGUIコンポーネントのサイズを調整するルールを提供できる。代替コンポーネントモジュール54は、IRツリー内におけるアプリケーションGUIを表すSGUIコンポーネントと置き換えに、代替のSGUIコンポーネントを特定できる。

20

【0078】

ある実施形態の代替コンポーネントモジュール54は、代替のSGUIコンポーネントに対する2つのカテゴリのルールを表しているGUI変換ルールモジュール単体56と、GUI変換ルールモジュール複合58から成っている。GUI変換ルールモジュール単体56は、SGUIコンポーネントをターゲット装置プラットフォームに特定するSGUIコンポーネントに変換するルールを提供する。

30

【0079】

図5は、第1SGUIコンポーネント70がターゲット装置プラットフォームに特定する第2SGUIコンポーネント72に、変換ルールで変換する例を示している。ターゲット装置プラットフォームの機能、ターゲット装置プラットフォームのタイプ、ターゲット装置プラットフォームのユーザインターフェイス、拡張性のあるアプリケーションの機能、及び/またはSGUIコンポーネントのディスプレイに関するパラメータをベースに、SGUIコンポーネントはターゲット装置プラットフォームに特定すると考えてよい。

【0080】

図の例では、第1SGUIコンポーネント70は、ユーザにより選択される「高」、「中」、「低」、いわゆる優先度の選択リストを表している。同様に、第2SGUIコンポーネント72は、優先度の選択ボタンを表している。仮に、第1SGUIコンポーネント70とターゲット装置プラットフォーム用のコンポーネントノード42(図3)を含んでいるIRツリー40(図3)はPDAだとした場合、GUI変換ルールモジュール単体56(図4)は、第1SGUIコンポーネント70を第2SGUIコンポーネント72に変換する変換ルールを提供してもよい(矢印74)。

40

【0081】

PDAの表示画面におけるサイズの限定、PDAで利用可能なユーザインターフェイス、またはPDAや拡張性のあるアプリケーションに関連性のある理由により、本実施例における第2SGUIコンポーネント72は特定のものと考えられてもよい。この図における例は、SGUIコ

50

ンポーネントに適用可能な数ある変換ルールの一例にすぎない。

【 0 0 8 2 】

図 4 に再び戻り、GUI変換ルールモジュール複合 5 8 は、SGUIコンポーネント複合が、ターゲット装置プラットフォームに特定するSGUIコンポーネントに変換する変換ルールを提供する。前にも述べた通り、SGUIコンポーネント複合は、IRツリー(図 3)におけるコンテナノード 4 4 (図 3)および対応するコンポーネントノード 4 2 (図 3)により表されてもよい。

【 0 0 8 3 】

図 6 は、第 1 SGUI コンポーネント複合 7 6 が、ターゲット装置プラットフォームに特定している第 2 SGUI コンポーネント複合 7 8 へ変換する変換例を示している。

10

【 0 0 8 4 】

図の例によると、第 1 SGUI コンポーネント複合 7 6 は、複数のデータエントリ 8 0、優先度セクション 8 2、そして好みセクション 8 4 から成っている。第 2 SGUI コンポーネント複合 7 8 は、ピックリスト 8 6 と単一のデータエントリ 8 8 から成っている。図 5 における例と同様、矢印 9 0 で示されているように、第 1 SGUI コンポーネント複合 7 6 を第 2 SGUI コンポーネント複合 7 8 に変換するのに、GUI変換ルールモジュール複合 5 8 から利用可能な変換ルールを用いてもよい。

【 0 0 8 5 】

図 4 に戻るが、ダイナミックレイアウトモジュール 6 0 は、装置プラットフォーム依存型のプレゼンテーションを表すIRツリーを構成してもよい。ダイナミックな構成とは、プレゼンテーションのページのIRツリーが表すSGUIコンポーネントの配置と、選択的にSGUIコンポーネントが変換することを含んでいる。ある実施形態において、レイアウト構造やアプリケーションGUIが特定する他の制約に従い、IRツリーにより表されるSGUIコンポーネントがページ上にあってもよい。さらに、ダイナミックレイアウトモジュール 6 0 は、適合性を確認するために、プレゼンテーションとターゲット装置プラットフォームの機能とを比べてもよい。例えば、制約に関する機能は、プレゼンテーションのページサイズでもよい。ダイナミックレイアウトモジュール 6 0 が構成するページのサイズは、プレゼンテーションを表示するターゲット装置プラットフォームの表示画面サイズを超えてはいけない。制約に関する他の機能例は、SGUIコンポーネントの最小の視覚サイズ、またはターゲット装置プラットフォームに関する他の機能である。

20

30

【 0 0 8 6 】

ダイナミックレイアウトモジュール 6 0 は、変換ルールを選択 & 適用する。そしてSGUIコンポーネントを機能あるいは制約に従う形に変換するため、変換ルールモジュール 5 0 から利用可能な変換ルールを適用する。例えば表示画面のサイズの制約を満たすため、ダイナミックレイアウトモジュール 6 0 は、もっとコンパクトなプレゼンテーションを作るために、変換ルールをIRツリー内のSGUIコンポーネントに適用してもよい。変換ルールの選択は、拡張性のあるアプリケーション、ターゲット装置プラットフォームの表示画面サイズ、装置プラットフォームのユーザインターフェイス機能、及び/またはディスプレイ、及びターゲット装置プラットフォームとの相互作用に関するパラメータにより、決められてもよい。

40

【 0 0 8 7 】

ある実施形態において、ダイナミックレイアウトモジュール 6 0 は、その機能や制約をベースにプレゼンテーション用のページを生成してもよい。また別の実施形態において、変換ルールの適用により、ターゲット装置プラットフォームに特定する様々な構成のページセットが生成される。ページセットとは、予想される装置プラットフォーム依存型プレゼンテーションの一部分である。予想されるプレゼンテーション内で開発されたページをベースに、ダイナミックレイアウトモジュール 6 0 は、最も望ましい装置プラットフォーム依存型のプレゼンテーションを得るために、ページセット内から選択的にページを利用してもよい。ダイナミックレイアウトモジュール 6 0 の望ましい機能とは、ページ数の最小化、アプリケーションGUIが特定するプロパティに従うこと、それぞれのページ用の表示

50

画面を最大に利用すること、及び/またはプレゼンテーション、ターゲット装置プラットフォーム及び/または拡張性のあるアプリケーションに関連しているその他の特徴である。

【0088】

さらに図4に示されているように、本実施形態のダイナミックレイアウトモジュール60は、スタイルガイドモジュール62とSGUIコンポーネントリストモジュール64から成り立っている。また他の実施形態において、少ないまたは多数のモジュールが、機能を表すため、ダイナミックレイアウトモジュール60内で特定されてもよい。

【0089】

スタイルガイドモジュール62は、少なくとも1つのアプリケーション用のスタイルガイドパラメータセットをSGUIコンポーネントに提供してもよい。スタイルガイドパラメータは、プレゼンテーション内において変わらない外観を提供するため、SGUIコンポーネントの視覚的な外観を標準化してもよい。典型的なスタイルガイドパラメータは、フォントサイズ、コンポーネント間におけるスペース、色彩などから成っている。ある実施形態において、スタイルガイドパラメータの1セットは、全ての装置プラットフォームに使われている。また別の実施形態において、スタイルガイドパラメータのセットは、1つまたはそれ以上のターゲット装置プラットフォームに対応付けられていてもよい。

【0090】

SGUIコンポーネントリストモジュール64は、予想されるSGUIコンポーネントのリストを編集する。予想されるSGUIコンポーネントのリストは、IRツリー内にあるSGUIコンポーネントの変換から生じるSGUIコンポーネントを表してもよい。ダイナミックレイアウトモジュール60は、リスト作成のため、変換ルールをIRツリー内にあるSGUIコンポーネントに適用してもよい。ある実施形態において、SGUIコンポーネントリストモジュール64が、リスト内のSGUIコンポーネントのサイズに応じて、SGUIコンポーネントのリストを編集してもよい。

【0091】

ダイナミックレイアウトモジュール60は、SGUIコンポーネントのリスト作成のため、変換ルールモジュール50から選択的に適切な変換ルールを適用してもよい。ある実施形態において、適切な変換ルールの選択のために、ダイナミックレイアウトモジュール60はルールに基づいた技術(方法)を用いてもよい。その技術(方法)においては、2つの前提条件に基づいた選択的な変換ルールが適用可能である。最初の前提条件とは、変換するSGUIコンポーネントの特徴を考慮することである。第2の前提条件とは、プレゼンテーション用のターゲット装置プラットフォームの機能を考慮することである。

【0092】

前提条件に加えて、ある実施形態では、ダイナミックレイアウトモジュール60のランキングから、適切な変換ルールを選択してもよい。変換ルールのランキングは、変換ルールを適用した結果に基づいてもよい。例えばどのくらいの確率で変換ルールが、異なる拡張性のあるアプリケーション用に適正な変換を実現するかということに基づき、ランキングが決まるということである。よって、かつ高い確率で変換が成功する変換ルールはランキングが高く、逆にある特定の拡張性のあるアプリケーション用にしか特定されていない変換ルールはランキングが低い。例えば、(少ない数の選択可能なアイテムにおける)選択ボタン(図5参照)に対する選択リストの変換は、世界的な規模で拡張性のあるアプリケーションにおいて適用可能なので、高いランキングが指定されてもよい。

【0093】

SGUIコンポーネントリストの生成に続いて、ダイナミックレイアウトモジュール60は、プレゼンテーションにおける使用目的でリストからSGUIコンポーネントを選択してもよい。ある実施形態において、最大サイズの表示画面に基づき、SGUIコンポーネントが選択される。また他の実施形態において、ターゲット装置プラットフォームの機能や拡張性のあるアプリケーションの機能、及び/またはその他条件が、選択過程において用いられてもよい。SGUIコンポーネントはリストから選択され、プレゼンテーションを表すページ上に

10

20

30

40

50

配置されてもよい。リストから選択されたSGUIコンポーネントは、リストを作るのに用いられたSGUIコンポーネントを置き換えてもよい。SGUIコンポーネントの置き換えにより、装置プラットフォーム独立型のSGUIコンポーネントは、効果的にターゲット装置プラットフォームに特定するSGUIコンポーネントに変換される。

#### 【0094】

ある実施形態において、各々のページは、表示画面の最大のサイズと比較されてもよい。この実施形態において、ページを構成するSGUIコンポーネントを縮小するか、または拡大する必要があるかどうかにより、ダイナミックレイアウトモジュール60は、リストから大きな、または小さなサイズのSGUIコンポーネントを選択してもよい。ある実施形態において、SGUIコンポーネントに対応するリスト内に、似たサイズのSGUIコンポーネントがある場合におけるSGUIコンポーネントの選択は、リスト内でSGUIコンポーネントを作るのに用いられる変換ルール(既に紹介済み)のランキングに基づいてもよい。

10

#### 【0095】

本発明の好ましい実施形態において、ダイナミックレイアウトモジュール60は、プレゼンテーションに対応しているページを作るために、IRツリーを通して繰り返し処理を行ってもよい。ある実施形態において、ダイナミックレイアウトモジュール60によるプロセスは、IRツリーの最も低い階層レベルから高いレベルに向けて処理を行う。ページを満たすために、ダイナミックレイアウトモジュール60は、論理パネルをプレゼンテーションのページに付け加えてもよい。ページが満たされていないときは、兄弟ノードあるいは高いレベルにあるノードからの階層的に関連のある論理パネルが、ページに付け加えられてもよい。例えば、図3によれば、コンテナノード44h、44f、そして44eが、ページ48を作るために、コンテナノード44gに付け加えられてもよい。これとは逆にページがはみ出したときは、表示画面のサイズに合わせるために、ページ内においてSGUIコンポーネントの変換を実行してもよい。

20

#### 【0096】

ページがおおむね満たされたとき、プレゼンテーション用の新しいページが作られる。おおむね満たされたページとは、利用可能なSGUIコンポーネントで、ターゲット装置プラットフォームの表示画面を最大化することである。ある実施形態において、おおむね満たされたページとは、およそ95%から100%の割合で、ターゲット装置プラットフォームの表示画面を満たしていることである。レイアウト構造、他の条件として指定されてる制約、及び/またはSGUIコンポーネントは表示画面が完全に満たされることを防ぐが、おおむね満たされたページ内では、現条件のもとで満たされている部分を最大限に活用する。

30

#### 【0097】

図7は、図3のIRツリー40を参照しながら、図4の変換マネージャモジュール20のある実施形態の動作を示しているブロック図である。変換マネージャモジュール20の動作がブロック100で始まる。変換マネージャモジュール20は、好ましくは、以下においてIRツリー40をインスタンス化し、そして取り除くことにより発生する。IRツリー40において、最も高いレイアウト優先度を持ち、かつ最も低いレベルにあるコンテナノード44が特定される。(ブロック102)低いレベルにあるコンテナノードは、SGUIコンポーネント(コンポーネントノード42)を表す階層的な子供ノードである。図3におけるIRツリー40の例では、コンテナノード44c、44d、44e、44g、そして44hが、最も低いレベルにあるコンテナノードである。レイアウトの優先度は、アプリケーションGUIのプロパティにより特定される。ここでは動作を説明する目的上、図3のコンテナノード44gが、最も高いレイアウト優先度を持っているとする。

40

#### 【0098】

スタイルガイドモジュール62は、特定のコンテナノード44(コンテナノード44g)に関連しているSGUIコンポーネント(コンポーネントノード42cと42d)に対し、ターゲット装置プラットフォームに対応するスタイルガイドパラメータセットを適用する。(ブロック104)SGUIコンポーネントは、IRツリー40のレイアウト構造とアプリケーションGUIに条件として指定される制約に従い、プレゼンテーションのページ上に配置される

50

。(ブロック 1 0 6)そして、ページのサイズが決定する。(ブロック 1 0 8)その後、表示画面からページのサイズがはみ出したかどうかが決まる。(ブロック 1 1 0)ページのサイズが、ターゲット装置プラットフォームの表示画面のサイズより大きければ、ページはターゲット装置プラットフォームの表示画面からはみ出していることになる。

【 0 0 9 9 】

図 8 のブロック 1 1 4 において、ページがはみ出していたら、適切な変換ルールが選択され、ダイナミックレイアウトモジュール 6 0 により、そのルールがSGUIコンポーネントに適用される。予想されるSGUIコンポーネントのリストが作成され、サイズに従いながらSGUIコンポーネントリストモジュール 6 4 と共に編集される。(ブロック 1 1 6)例えば、サイズ縮小の必要がある等の条件をベースにしたリストから、SGUIコンポーネントは特定される。(ブロック 1 1 8)そして、リスト内にある 1 つ以上のSGUIコンポーネントが特定されたかどうかを決定する。(ブロック 1 2 0)もしノーなら、特定されたSGUIコンポーネントが選択される。(ブロック 1 2 2)その後、選択されたSGUIコンポーネントが、ページ(ブロック 1 0 6)内で以前に表示された対応するSGUIコンポーネントと交換され(ブロック 1 2 4)、動作は図 7 のブロック 1 0 4 に戻る。もし複数のSGUIコンポーネントが特定されたら、最も高いランクの変換ルールから作られたSGUIコンポーネントが選択され(ブロック 1 2 5)、動作はブロック 1 2 4 に進む。

【 0 1 0 0 】

図 9 に戻り、ブロック 1 1 0 (図 7)においてページがはみ出していない場合、ページが満たされていないことが決定される。(図 9: ブロック 1 2 6)ページのサイズがターゲット装置プラットフォームの表示画面のサイズより小さい時、ページは満たされていないという事になる。もしページが満たされないと決定した時、まだページ上に配置されていない兄弟ノード(図 3 のコンテナノード 4 4 h)の特定作業が行われる。(ブロック 1 2 8)もし兄弟ノードが配置されていない場合、最も高い優先度を持つ特定された兄弟ノードが選択され(ブロック 1 4 4)、そして選択されたノードの処理を始めるため、動作は図 7 のブロック 1 0 4 に戻る。もし兄弟ノードが既に配置されているなら(または存在しないなら)、次に高い階層レベル(図 3 のコンテナノード 4 4 f)からの関連ノードが特定される。(ブロック 1 3 0)

【 0 1 0 1 】

ルートコンテナノード(コンテナノード 4 4 a)かどうかを判断するのに、次に高い階層レベルからのノードが確認される。(ブロック 1 3 2)もしイエスなら、まだページ上に置かれていない付加的で最も低いレベルのコンテナノードの有無を調べるためにIRツリーを確認する。(ブロック 1 3 4)もしそのような最も低いレベルのコンテナノードが存在するなら、最も低いレベルにありかつ次に高い優先度を持っている別のコンテナノード 4 4 (図 3)を特定するため、動作はブロック 1 0 2 (図 7)へ戻る。もしそのような最も低いレベルのコンテナノードが存在しないなら動作は終わる。(ブロック 1 3 6)

【 0 1 0 2 】

ブロック 1 3 2 に戻り、次に高い階層レベルにおけるノード(図 3 のコンテナノード 4 4 f)がルートコンテナノード(コンテナノード 4 4 a)でない場合、ノードがページ内に以前に配置されたかどうかを決める。(ブロック 1 3 8)ノードが以前に配置されていないなら、ノード処理を始めるために、動作はブロック 1 0 4 (図 7)へ戻る。ノードが以前に配置されているなら、次の高い階層レベルにあるノードの兄弟ノード(図 3 のコンテナノード 4 4 e)が配置されていないかどうかを判定する。(ブロック 1 4 0)もし兄弟ノードがページ内で既に配置されているなら(または存在しないなら)、動作はブロック 1 3 4 へ戻る。兄弟ノードが配置されていないなら、最も高いレイアウト優先度を持っている兄弟ノードが選択され(ブロック 1 4 2)、選択されたノードの処理を始めるため、動作はブロック 1 0 4 (図 7)へ戻る。

【 0 1 0 3 】

ブロック 1 2 6 へ戻り、ページが満たされていないとは言えない場合、ページはおおむね満たされていると見なされる。(ブロック 1 4 6)残りのSGUIコンポーネントを配置するた

10

20

30

40

50

めに、新しいページが割り当てられ(ブロック 1 4 8)、動作はブロック 1 3 4 へ戻る。

【 0 1 0 4 】

別の実施形態において、ページが満たされたと判断されると、リストからの代替SGUIコンポーネントを用いることにより、同ページにおける付加的な変化の生成のための追加的な動作が実行されてもよい。追加的な動作に続いて、ダイナミックレイアウトモジュール 6 0 は、既に前述した通り、最も望ましい装置プラットフォーム依存型のプレゼンテーションを選択してもよい。

【 0 1 0 5 】

通信システム内におけるSGUIシステムの利用例が説明される。図 1 0 は、装置プラットフォーム 2 0 2、変換サーバ 2 0 4、そしてプラットフォーム 2 0 2、変換サーバ 2 0 4 とネットワーク 2 0 8 を介して通信しているライブラリサーバ 2 0 6 を有している、通信システム 2 0 0 の実施例を示している。装置プラットフォーム 2 0 2 は、既に述べられている異種装置プラットフォームでもよい。変換サーバ 2 0 4 およびライブラリサーバ 2 0 6 は、ネットワーク 2 0 8 内において、サーバ機能を実行するいかなる装置であってもよい。ある実施形態において、動作システムを含む 1 つまたはそれ以上のサーバコンピュータは、変換サーバ 2 0 4 およびライブラリサーバ 2 0 6 として動作する。動作システムの例として、Microsoft (登録商標)、Windows NT(登録商標)、UNIX(登録商標)、またはサーバ機能をサポートできる他の動作システムが挙げられる。サーバ 2 0 4、2 0 6 は、データを保存し、データにアクセスする機能や、SGUIシステムの一部などのアプリケーションを実行する機能を有していてもよい。ネットワーク 2 0 8 は、インターネット、公衆または専用イントラネット、イクストラネット、及び/または無線、光ファイバ、及び/または有線通信を介し、データ送信や指令が可能なネットワーク構成の形式などであってもよい。

【 0 1 0 6 】

通信システム 2 0 0 内において、SGUIシステムは動作可能である。ある実施形態において、拡張性のあるアプリケーション 2 1 0 は、装置プラットフォーム 2 0 2 上において動作してもよい。さらに、拡張性のあるGUIライブラリモジュール 1 2 とレンダーマネージャモジュール 1 6 は、装置プラットフォーム 2 0 2 上において動作してもよい。さらに、この実施形態では、タスクマネージャモジュール 1 8 と変換マネージャモジュール 2 0 は、変換サーバ 2 0 4 上のカスタマイズモジュール 1 4 内で動作してもよい。既に前実施形態で説明されている拡張性のあるアプリケーション 2 1 0、拡張性のあるGUIライブラリモジュール 1 2、カスタマイズモジュール 1 4、そしてレンダーマネージャモジュール 1 6 は、本実施形態においても同様の働きをする。

【 0 1 0 7 】

他の実施形態において、SGUIシステムにおける異なる機能、及び拡張性のあるアプリケーションは、異種装置プラットフォームとサーバ間において配信されてもよい。例えば、各動作をサポートするのに十分な能力があるところで、SGUIシステムと拡張性のあるアプリケーション 2 1 0 は、装置プラットフォーム 2 0 2 上を動作可能となる。また別の例において、SGUIシステムと拡張性のあるアプリケーション 2 1 0 の一部分は装置プラットフォーム 2 0 2 上において動作可能であり、またSGUIシステムと拡張性のあるアプリケーション 2 1 0 のその他の部分は、その他の装置プラットフォーム上で動作可能である。また別の例においてSGUIシステムの大部分は、サーバ上またはその他類似の装置上で動作可能であり、そしてSGUIシステムの不可欠な部分だけが、装置プラットフォーム 2 0 2 上において動作可能であってもよい。

【 0 1 0 8 】

ある実施形態において、拡張性のあるGUIライブラリモジュール 1 2 における拡張性のあるGUIコンポーネントライブラリ 2 2 全体(図 2)は、装置プラットフォーム 2 0 2 に保存されてもよい。この実施形態において、変換サーバ 2 0 4 は、ネットワーク 2 0 8 経由で、拡張性のあるGUIコンポーネントライブラリへアクセスしてもよい。既に説明されているが、変換マネージャモジュール 2 0 は、GUIコンポーネントの変換中に、拡張性のあるG

10

20

30

40

50

UIコンポーネントライブラリへ、アクセスしてもよい。

【0109】

別の実施形態において、拡張性のあるGUIコンポーネントライブラリ22のサブセットと共に、拡張性のあるGUIライブラリモジュール12の一部分は、装置プラットフォーム202に保存されてもよい。拡張性のあるGUIコンポーネントライブラリ22の保存については、そのサブセットに限られているが、その理由は、例えば、装置プラットフォーム202の限られた保存機能、情報検索の増加、及び/または装置プラットフォーム202及び/または拡張性のあるアプリケーション210の動作に関する理由である。拡張性のあるGUIコンポーネントライブラリ22のサブセットは、装置プラットフォーム202により頻繁に用いられるSGUIコンポーネントを含んでいてもよい。この実施形態では、拡張性のあるGUIコンポーネントライブラリ22全体(図2)と共に、拡張性のあるGUIライブラリモジュール12の一部分がライブラリサーバ206に保存され、そして必要があれば、装置プラットフォーム202によりネットワーク208を介してアクセスされる。またカスタマイズモジュール14は、装置プラットフォーム202内の拡張性のあるGUIコンポーネントライブラリ22のサブセット、及び/またはライブラリサーバ206内の拡張性のあるGUIコンポーネントライブラリ22全体へアクセスしてもよい。別の実施形態において、変換サーバ204は、装置プラットフォーム202からの、または変換サーバ204自身からのアクセスに備え、拡張性のあるGUIコンポーネントライブラリ22を保存してもよい。

10

【0110】

また別の実施形態において、変換サーバ204は、GUIライブラリモジュール12の一部分を持っていてもよい。例えば装置プラットフォーム202の制限された処理能力、装置プラットフォーム202のリソースを保存する要求、そしてSGUIシステムの動作に関するその他の理由により、GUIライブラリモジュール12の一部分は、変換サーバ204内に保存されてもよい。ある実施形態において、中間表現モジュール24(図2)の一部分は、変換サーバ204に含まれていてもよい。変換サーバ204に含まれている中間表現モジュールの一部分は、モバイルコードをベースにしたIRツリーのインスタンスを作るために動作してもよい。この実施形態においてモバイルコードは、拡張性のあるアプリケーション210のアプリケーションGUIから、ネットワーク208を介して、変換サーバ204へ送信されてもよい。中間表現モジュールの一部分は、アプリケーションGUIを表すIRツリーのインスタンスを作るために、モバイルコードを実行してもよい。

20

30

【0111】

図11は、図10を参照しながら説明されたSGUIシステムのある実施形態における動作を説明したブロック図である。拡張性のあるアプリケーション210がターゲット装置プラットフォーム(装置プラットフォーム202)へマイグレーション、またはターゲット装置プラットフォームから起動したとき、動作は始まる(ブロック250)。拡張性のあるアプリケーション210は、IRツリーのインスタンスがターゲット装置プラットフォーム上において作成されるかどうかを判断する。(ブロック252)もしイエスなら、IRツリーのインスタンスを作るために、拡張性のあるアプリケーション210内のアプリケーションGUIは、GUIライブラリモジュール12に関連して利用される。(ブロック254)アプリケーションGUIを表しているIRツリーは、変換サーバ204に送信されてもよい。(ブロック256)もしIRツリーのインスタンスがターゲット装置プラットフォーム上において作成されないなら、モバイルコードが変換サーバ204に送信される。(ブロック258)アプリケーションGUIを表しているIRツリーのインスタンスを作るために、変換サーバ204はモバイルコードを実行する。(ブロック260)既に説明済みだが、IRツリーは装置プラットフォーム独立型で、そしてアプリケーションGUIに特定されているプロパティを含んでいる。

40

【0112】

ブロック264において、ターゲット装置プラットフォームの機能が、変換サーバ204に送信される。ターゲット装置プラットフォームの機能として、タスクマネージャモジュ

50

ール 18 は、IR ツリーから不必要な論理パネル及び/またはSGUI コンポーネントを削除する。(ブロック 266) 変換マネージャモジュール 20 は、タスクマネージャモジュール 18 から IR ツリーを受け取り、IR ツリーの変換を始める。(ブロック 268) IR ツリーの変換は、IR ツリーにおいて表されている SGUI コンポーネントの構成を含む。装置プラットフォーム依存型のプレゼンテーションのレイアウトを作るため、IR ツリーは変換される。

#### 【0113】

再び図 12 に戻るが、変換の後、装置依存型 IR ツリーはターゲット装置プラットフォームに送信される。(ブロック 270) レンダーマネージャモジュール 16 は装置依存型 IR ツリーを通過して、SGUI コンポーネントのインスタンス化を開始する。(ブロック 272) レンダーマネージャモジュール 16 は、SGUI コンポーネントがターゲット装置プラットフォーム内において拡張性のある GUI コンポーネントライブラリ 22 に保存されているかどうかを判断する。(ブロック 274) もしイエスなら、レンダーマネージャモジュール 16 は、装置プラットフォーム 202 の表示画面上に、カスタマイズされたアプリケーション GUI を表示することを指示する。(ブロック 276) SGUI コンポーネントが、ターゲット装置プラットフォーム 202 内における拡張性のある GUI コンポーネントライブラリ 22 内に保存されていないなら、レンダーマネージャモジュール 16 は、ライブラリサーバ 206 から装置プラットフォーム 202 へ SGUI コンポーネントを動的にダウンロードする。(ブロック 278) そして、レンダーマネージャモジュール 16 は、装置プラットフォーム 202 の表示画面上に、装置依存型アプリケーション GUI の表示を指示する。(ブロック 276)

#### 【0114】

GUI のマイグレーションのイベント前に、ユーザに対して利用可能な技術的特徴を提供する他、SGUI システム 10 は、第 1 異種装置プラットフォームから第 2 異種装置プラットフォームへマイグレーションする GUI アプリケーションの実行状態を保存することができる。ここでいう実行状態とは、SGUI コンポーネントに影響を与えている拡張性のあるアプリケーションのエンドユーザがもたらした変化を指している。この変化は SGUI コンポーネントの変数(外観の変化、データエントリ、状態変化)を含んでもよい。図 13 は、第 1 異種装置プラットフォーム 302、第 2 異種装置プラットフォーム 304、そしてサーバコンピュータ 306 を有している SGUI システム 10 の実施形態により動作している通信システムのブロック図である。他の例において、本実施形態とは異なる数のプラットフォーム、及びサーバが利用可能なのはいうまでもない。

#### 【0115】

第 1 異種装置プラットフォーム 302、第 2 異種装置プラットフォーム 304 は、前に説明された、いかなる異種装置プラットフォームであってもよいし、かつ各々はレンダーマネージャモジュール 16 を有してもよい。サーバコンピュータ 306 はコンピュータ装置に類似のものであってよいし、また拡張性のある GUI ライブラリ 12 やカスタマイズモジュール 14 内にあるタスクマネージャモジュール 18 及び変換マネージャモジュール 20 を有していてもよい。他の例においてモジュールは、コンピュータ装置の処理能力、ネットワーク 208 の構成、プロバイダーの優先度などの様々な変化に応じながら、通信システム内にある装置へと別々に配信されてもよい。第 1 異種装置プラットフォーム 302 & 第 2 異種装置プラットフォーム 304 とサーバ 306 間における通信は、ネットワーク 208 を介してもよい。

#### 【0116】

前にも説明した通り、装置プラットフォーム独立型(PI)の中間表現は、拡張性のあるアプリケーションを開発の一部として開発される。プラットフォーム独立型の中間表現は、プラットフォーム独立型の中間モデル形式で保存されてもよい。プラットフォーム独立型の中間モデルは、サーバ 306、第 1 異種装置プラットフォーム 302、第 2 異種装置プラットフォーム 304 あるいはネットワーク 208 と通信しているその他の装置内で保存されてもよい。

#### 【0117】



本実施形態において、装置プラットフォーム独立型の中間表現は、最大の表示画面と最も大規模なユーザインターフェイスメカニズムを有している装置プラットフォームに基づいて特定されてもよい。ここでアプリケーション開発者は、拡張性のあるアプリケーションは最大の表示画面と最も大規模なユーザインターフェイスメカニズムで動作していることが予測できる。例えば、装置プラットフォーム独立型の中間表現は、大きな表示画面と複数のユーザインターフェイス（例：タッチスクリーン、マウス、キーボードなど）を持つラップトップコンピュータの動作に基づいて特定されてもよい。従って、プラットフォーム独立型のプレゼンテーションモデルから生成されたプラットフォーム特定型のプレゼンテーションは、プラットフォーム独立型のプレゼンテーションモデルと同サイズ、あるいはそれより小さいものであってもよい。

10

#### 【0118】

図14は、IRツリー40及び対応するGUIプレゼンテーション404形式における拡張性のあるサーチアイテムアプリケーション用のプラットフォーム独立型のプレゼンテーションモデルである。前に説明済みだが、ルートノード44aはIRツリー40上に位置し、かつGUIプレゼンテーションを表してもよい。ルートノード44aの子供ノードは、コンテナノードあるいは拡張性のあるアプリケーションのエンドユーザに対して利用可能な異なるタスクを表しているタスクノードである。コンポーネントノード、個々のSGUIコンポーネント及びSGUIコンポーネント複合を表している「葉ノード」が広がるまで、コンテナノード44は、さらにサブタスクノード、サブのサブタスクノード等に分割可能な論理パネルであってよい。

20

#### 【0119】

図14の例において、「アイテムタスクのサーチ」として特定されているコンテナノード44mは、「タスクのサーチ」として特定されている第1タスクノード44n、「結果タスクの分類」として特定されている第2タスクノード44p、および「結果タスクの表示」として特定されている第3タスクノード44qを有している。タスクノード44n、44p、44qの各々は論理パネルを表しており、かつコンポーネントノードを有している。図の例においてコンポーネントノードは、IRツリー40のプラットフォーム独立型のプレゼンテーションモデルから生成されたJ2SE Swing GUIのようなGUIプレゼンテーション404内に描かれている。従ってGUIプレゼンテーション404を形成するために、IRツリー40のSGUIコンポーネントは、J2SE Swingにおけるプラットフォーム特定型のGUIコンポーネントへマッピングする。

30

#### 【0120】

「タスクのサーチ」として特定されたコンテナノード44nは、3つのサブタスクノード（コンポーネントノード）に分割される論理パネルを表してもよい。ここで3つのサブタスクノードとは、SGUIコンポーネントあるいはSGUIコンポーネント複合である。3つのサブタスクノードは次のように表されている。

1. 「アイテムの検索」(バーチャルラベル408)として表されている第1コンポーネントノード
2. 「サーチ」(バーチャルボタン410)として表されている第2コンポーネントノード
3. バーチャルテキスト領域412として表されている第3コンポーネントノード。

40

拡張性のあるアプリケーションのエンドユーザが選択可能な分類優先タスクと表示選択タスクを提供するために、第2コンテナノード44p & 第3コンテナノード44qと関連しているSGUIコンポーネント及び/またはSGUIコンポーネント複合も同様に、GUIプレゼンテーション404内にあるプラットフォーム特定のGUIコンポーネントにより表される。

#### 【0121】

プラットフォーム独立型のプレゼンテーションモデル内において、SGUIコンポーネント及びSGUIコンポーネント複合は、プラットフォーム独立(PI)ウィジェット、あるいはプラットフォーム独立型のGUIウィジェットと見なしてもよい。SGUIコンポーネント及びSGUIコンポーネント複合は、プラットフォーム特定のプレゼンテーションにおけるプラットフォーム特定(PS)のウィジェットにより表される。プラットフォーム特定型のウィジェットを

50

有するSGUIコンポーネントおよびSGUIコンポーネント複合は、事前に決められたマッピングに基づいている。ここでのマッピングは、拡張性のあるGUIコンポーネントライブラリモジュール22と他のライブラリにおける装置特定のGUIコンポーネント間のことである。事前に決められたマッピングは、プラットフォーム特定のプレゼンテーション内にあるプラットフォーム特定型のウィジェットを、プラットフォーム独立型のプレゼンテーションモデル内の(プラットフォーム独立型のウィジェットに対応している)SGUIコンポーネント、およびSGUIコンポーネント複合に対応づける。

#### 【0122】

前にも説明した通り、プロパティはIRツリー40のタスクノード内に含まれてもよい。ここでプロパティは、少なくとも1つのタスクノード用の詳細なレイアウトプロパティ、少なくとも1つのタスク優先プロパティ、少なくとも1つの優先プロパティ、そして1つ以上の対応するタスクノードに関連する少なくとも1つの分割可能なプロパティのことである。タスクノード用の詳細なレイアウトプロパティは、例えばページ内でプラットフォーム独立型のウィジェットの位置を提供する、レイアウトを制御するJava(登録商標)Gridbagに似ている。タスク優先プロパティはブール(Boolean)機能の配列として実装され、その配列は特定の異種装置プラットフォームを表し、及び/または、タスクが特定のプラットフォームに適しているかどうかを示す。優先プロパティは、プラットフォーム独立型のウィジェット所望のレイアウト結果を表すための整数(インテジャ)として実装される。分割可能なプロパティは、プラットフォーム独立型のウィジェットが複数ページのプレゼンテーションに拡散可能なことを表すブール機能として実装される。

#### 【0123】

ある実施形態において、各々のタスクノードは、対応するタスクノードの重要性を示している、少なくとも1つの重要プロパティを含んでもよい。重要プロパティは、タスクノードにより表されているプラットフォーム独立型のウィジェットが、コアあるいはオプションかということを示すブール機能として実装される。

#### 【0124】

コアプラットフォーム独立型のウィジェットは、よく使われるプラットフォーム独立型のウィジェット、あるいは拡張性のあるアプリケーション内において主要なタスクを実行するプラットフォーム独立型のウィジェットとして定義されてもよい。残りのプラットフォーム独立型のウィジェットは、オプションのプラットフォーム独立型のウィジェットとして定義されてもよい。例えば、図14における「サーチアイテムタスク」コンテナノード44mは、前に説明した通り、3つの階層的なサブタスク(コンテナノード44n, 44p, 44q)を持っている。サーチアイテム実行のため、拡張性のあるアプリケーションのエンドユーザはバーチャルテキスト領域412にアイテム名を入力し、そしてサーチ処理を始める為にバーチャルボタン410を起動してもよい。しかしながら本例において、分類優先タスクと表示選択タスクはサーチ結果を表示するための付加的な機能なので、エンドユーザは両タスクを特定させる必要はない。このように、サーチタスク(コンテナノード44n)と関連のあるプラットフォーム独立型のウィジェットはコアとして特定され、そして結果分類タスク(コンテナノード44p)、および結果表示タスク(コンテナノード44q)と関連のあるプラットフォーム独立型のウィジェットはアプリケーション開発者によるオプションと特定されてもよい。

#### 【0125】

再び図13に戻り、中間表現におけるアプリケーション開発者により特定されたプロパティは、タスクマネージャ18と変換マネージャモジュール20により利用されてもよい。GUIのマイグレーション中において、変換マネージャモジュール20はタスク優先プロパティを利用し、タスクマネージャ18は残りのプロパティを利用してよい。

#### 【0126】

図15は、第1異種装置プラットフォーム302から第2異種装置プラットフォーム304へ、GUIアプリケーションの実行状態がマイグレーションするプロセスを表しているフロー図である。同図のGUIアプリケーションのマイグレーションにおいて、プラットフォ

10

20

30

40

50

ーム特定型のプレゼンテーションは、以前カスタマイズモジュール 14 におけるプラットフォーム独立型のプレゼンテーションモデルから生成され、そして第 1 異種装置プラットフォーム 302 へ提供されていた。第 1 異種装置プラットフォーム 302 は、プラットフォーム特定型のプレゼンテーションに基づき、対応する表示画面上に対話式的表示を提供してもよい。

#### 【0127】

第 1 プラットフォーム 302 上に表示される、プラットフォーム特定型のプレゼンテーション内のプラットフォーム特定型のウィジェットの実行状態がシリアル化された時、動作は始まる。(ブロック 502) プラットフォーム特定型のウィジェットのシリアル化された実行状態と共に、マイグレーションのターゲット装置として第 2 異種装置プラットフォーム 304 を特定するマイグレーション要求は、ネットワーク 208 経由でサーバ 306 へ伝達される。(ブロック 504) プラットフォーム独立型のプレゼンテーションモデルに含まれているプラットフォーム独立型のウィジェットは、第 1 プラットフォーム 302 が提供するプラットフォーム特定型のウィジェットの実行状態により更新される。(ブロック 506) プラットフォーム独立型のプレゼンテーションモデルのコピーが作られる。(ブロック 508)

#### 【0128】

サーバ 306 は、第 2 異種装置プラットフォーム 304 の GUI 能力を調査し、そして表示画面サイズ、ユーザ入力メカニズム等のユーザインターフェイス関連の能力を確認する。(ブロック 510) タスクマネージャ 18 はタスク優先プロパティを利用し、プラットフォーム独立型のプレゼンテーションモデルのコピーから不必要なタスクノードを除去し、そして第 2 異種装置プラットフォーム 304 用にプラットフォーム特定型のプレゼンテーションの生成を始める。(ブロック 512)

#### 【0129】

変換マネージャモジュール 20 はプラットフォーム独立型のプレゼンテーションモデルのコピーからノードを選択し、ノードに関連するプラットフォーム独立型のウィジェットを、第 2 異種装置プラットフォーム 304 用に、プラットフォーム特定型のウィジェットへマップする。(ブロック 514) プラットフォーム特定型のウィジェットは、プラットフォーム特定型のプレゼンテーションのページ上に配置される。(ブロック 516) ここでページサイズが、第 2 異種装置プラットフォーム 304 の表示画面より大きいかどうかを判定する。(ブロック 518) もしページサイズが大きすぎて表示画面に適さない場合、変換マネージャモジュール 20 は、プラットフォーム独立型のプレゼンテーションモデルのコピーからプラットフォーム独立型のウィジェットをカスタマイズし、1 あるいはそれ以上のページを表示画面 520 へと適合させる。(ブロック 520) カスタマイズされたプラットフォーム独立型のウィジェットは、プラットフォーム特定型のプレゼンテーション用のプラットフォーム特定型のウィジェットへマップされる。(ブロック 522) プラットフォーム独立型のプレゼンテーションモデルにおけるプラットフォーム独立型のウィジェットの実行状態が、プラットフォーム特定型のプレゼンテーション用のプラットフォーム特定型のウィジェットの実行状態へ適用される。(ブロック 523)

#### 【0130】

プラットフォーム独立型のプレゼンテーションモデル内において、未処理のノードがないかどうかを判定する。(ブロック 524) もしイエスなら動作はブロック 514 へ戻り、別ノードを選択し、対応するプラットフォーム特定型のウィジェットをページ等へ追加し、そしてプラットフォーム特定型のプレゼンテーションの生成を続ける。もしノーなら、プラットフォーム特定型のプレゼンテーションにより利用されるプラットフォーム特定型のプレゼンテーションの生成されたページ、プラットフォーム独立型のプレゼンテーションモデルの最終コピー、および拡張性のある GUI ライブラリ 12 の一部分を第 2 異種装置プラットフォーム 304 へ送信する。(ブロック 526) もしページが表示画面に適していたら(ブロック 518)、動作はそのままブロック 524 へと進み、未処理のノードがないかどうかを判定する。第 2 異種装置プラットフォーム 304 のレンダーマネージャモ

10

20

30

40

50

ジュール5 1 6は、送信されたプラットフォーム特定型のプレゼンテーション上、プラットフォーム独立型のプレゼンテーションモデルの最終コピー上、および拡張性のあるGUIライブラリ1 2の一部分を使っている表示画面上にプレゼンテーションを表示する。

【0 1 3 1】

ページに適したプラットフォーム特定型のプレゼンテーションの生成中におけるカスタマイズは、ウィジェットのレイアウトカスタマイズとウィジェットの変換カスタマイズを含んでいてもよい。このようなカスタマイズは、次における要件に基づいている。(1)アプリケーション開発者が特定した最低限の指示により、比較的高品質のプレゼンテーションを生成する。(2)プレゼンテーション生成の遅れを最小限にするため、カスタマイズを比較的シンプルにする。(3)プラットフォーム特定型のプレゼンテーション全体のページを見るため、表示画面上においてスクロールを避ける。

10

【0 1 3 2】

前にも説明した通り、レイアウトカスタマイズは、IRツリーにおける各々のノード処理、およびページ内において関連のあるプラットフォーム特定型のウィジェットを配置することを含んでもよい。またレイアウトカスタマイズ中におけるプラットフォーム特定型のウィジェットの配置において、ページを最適化して、ターゲット装置異種プラットフォームの表示画面に適合させる変換を実行してもよい。プラットフォーム特定型のウィジェットの最初のレイアウトは、アプリケーション開発者が特定するノードに関連したプラットフォーム特定型のウィジェットのレイアウトプロパティに基づいてもよい。レイアウトプロパティは、レイアウトを制御するJava(登録商標)Gridbagに類似のものであってよいし、かつページ上で対応するプラットフォーム特定型のウィジェットの相対的な位置を示すものであってもよい。もしページサイズがターゲット異種装置プラットフォームの表示画面に適合していないなら、プラットフォーム独立型のウィジェットをカスタマイズすることにより、プラットフォーム特定型のウィジェットのレイアウトは更にカスタマイズされる。

20

【0 1 3 3】

更にレイアウトカスタマイズは、フローレイアウトマネージャを使いながら、ページ内にあるプラットフォーム独立型のウィジェットを再配置してもよい。フローレイアウトマネージャを使うプラットフォーム独立型のウィジェットの再配置においては、例えば、プラットフォーム独立型のウィジェットを一行ではなく、一行以上に配置すること等を含めてもよい。再配置されたプラットフォーム独立型のウィジェットは、プレゼンテーションのページが表示画面のサイズを超えるプラットフォーム特定型のウィジェットに対応する、プラットフォーム独立型のウィジェットであってもよい。一方、物理的な空間や優先度などのクライテリアは、動的な再配置用に、ページ内のプラットフォーム特定型のウィジェットに対応している、どのプラットフォーム独立型のウィジェットを選択するかを決定してもよい。

30

【0 1 3 4】

フローレイアウトマネージャは、変換マネージャモジュール2 0により起動されてもよいし、あるいは変換マネージャモジュール2 0の一部であってもよい。フローレイアウトマネージャは、Java FlowLayoutである。さらに、分割可能なプロパティとしてアプリケーション開発者に特定され、かつSGUIコンポーネント複合であるプラットフォーム独立型のウィジェットは、SGUIコンポーネント単体に分割されてもよい。SGUIコンポーネント単体がマップされるプラットフォーム特定型のウィジェットは、SGUIコンポーネント単体あるいはその他のプロパティに関連する優先プロパティに基づきながら、プレゼンテーションの複数のページ間において分配されてもよい。分割されたSGUIコンポーネント単体のレイアウトは、フローレイアウトマネージャで実行されてもよい。ここでSGUIコンポーネント複合は、アプリケーション開発者により、非分割として特定されてもよい。以下におけるカスタマイズにおいて、プレゼンテーションのページが表示画面に適さない場合、変換カスタマイズを起動してもよい。

40

【0 1 3 5】

50

変換カスタマイズは、変換ルールアプリケーションを含んでいてもよい。変換ルールは、ターゲット異種装置プラットフォームの表示画面に適合させるため、ページ内におけるプラットフォーム特定型ウィジェットの適切なサイズ変換実現の為に適用されてもよい。変換カスタマイズの目的は、プレゼンテーションのページサイズを縮小し、異種装置プラットフォームの表示画面内に丁度適合させることにより、スクロールの使用を避けることである。

#### 【0136】

変換カスタマイズ中において、変換マネージャモジュール20は、どのプラットフォーム独立型のウィジェットが変換され、そしてどのようにプラットフォーム独立型のウィジェットが変換されるかを決定してもよい。どのプラットフォーム独立型のウィジェットを変換するかの決定は、変換可能で、かつ変換ルーツをトリガするプラットフォーム独立型のウィジェットを特定する事に基づいてよい。変換ルールをトリガするプラットフォーム独立型のウィジェットは、プレゼンテーションのページサイズが表示画面のサイズを超えてしまう原因となるプラットフォーム特定型のウィジェットへマップする、非分割なSGUIコンポーネント単体、あるいは非分割なSGUIコンポーネント複合であってもよい。

#### 【0137】

プラットフォーム独立型のウィジェットがどのように変換するかは、特定されたウィジェットに基づいている。プラットフォーム独立型のウィジェットがSGUIコンポーネント単体な場合、変換ルールがそのSGUIコンポーネント単体に適用されてもよい。しかしながら、プラットフォーム独立型のウィジェットがSGUIコンポーネント複合である場合、SGUIコンポーネント複合内にあるSGUIコンポーネント単体の各々は、SGUIコアコンポーネントとSGUIオプションコンポーネントとしてカテゴリー化される。SGUIコンポーネント複合内にあるSGUIコンポーネント単体のカテゴリー化は、プレゼンテーションモデルのタスクにおいて、アプリケーション開発者が特定する重要プロパティに基づいてもよい。

#### 【0138】

プラットフォーム独立型のウィジェットを、よりコンパクトなウィジェットにする変換はGUIの有用性を劣化させるかもしれないので、変換ルールはSGUIオプションコンポーネントに適用される。もし重要プロパティによるSGUIオプションコンポーネントが特定されていない場合、あるいはページをターゲット異種装置プラットフォームへ適合させるためSGUIオプションコンポーネントのサイズを縮小することが適切でない場合、変換ルールはSGUIコアコンポーネントへ適用されてもよい。

#### 【0139】

また変換マネージャモジュール20は、複数のルールの中から最も適した変換ルールを決定することにより、プラットフォーム独立型のウィジェット(SGUIコンポーネント単体)を変換してもよい。ここで言う適切な変換ルールとは、対応するプラットフォーム特定型のウィジェットのサイズを変換し、表示画面のサイズに適合させることである。変換ルールは、次の4つのタイプにカテゴリー化されてもよい。(1)「1から1タイプへの変換ルール」(2)「1から複数タイプへの変換ルール」(3)「複数の同クラスから1タイプへの変換ルール」(4)「複数から複数タイプへの変換ルール」

#### 【0140】

「1対1タイプへの変換ルール」とは、プラットフォーム独立型のウィジェット(SGUIコンポーネント単体)を別のプラットフォーム独立型のウィジェット(SGUIコンポーネント単体)に変換することである。例を挙げると、図5における第1SGUIコンポーネント70は、1から1タイプの変換ルールにより、第2SGUIコンポーネント72であるドロップダウンボックスに変換される。「1から複数タイプへの変換ルール」とは、プラットフォーム独立型のウィジェット(SGUIコンポーネント単体)を複数のプラットフォーム独立型のウィジェット(各々のSGUIコンポーネント単体)へ変換することである。例を挙げると、データテーブルを表している単一のプラットフォーム独立型のウィジェットが、複数のデータリストを表している単一のプラットフォーム独立型のウィジェット、あるいは複数のデータリストを表している単一のプラットフォーム独立型のウィジェットに変換されてもよい

。

## 【 0 1 4 1 】

「複数の同クラスから 1 タイプへの変換ルール」とは、同じクラスに属する複数のプラットフォーム独立型のウィジェット(各々のSGUIコンポーネント単体)を、プラットフォーム独立型のウィジェット(SGUIコンポーネント単体)に変換することである。例を挙げると、同じクラスにあり、かつラジオボタンを表している複数の単一のプラットフォーム独立型のウィジェットは、リストあるいはドロップダウンボックスを表している単一のプラットフォーム独立型のウィジェットに変換されてもよい。同様に、テキスト領域あるいはラベルを表している、いくつかの単一のプラットフォーム独立型のウィジェットは、各々 1 つのテキスト領域、あるいはドロップダウンボックスを表している単一のプラットフォーム独立型のウィジェットに変換されてもよい。

10

## 【 0 1 4 2 】

「複数から複数タイプへの変換ルール」とは、異なるクラスに属する複数のプラットフォーム独立型のウィジェットを、別の対応する複数のプラットフォーム独立型のウィジェットに変換することである。各々の複数から複数タイプへの変換ルールは、「1 つ以上の複数の同クラスから 1 タイプへの変換ルール」、及びフォーム記入関係のような、異なるプラットフォーム独立型のウィジェット間における対応関係を含んでいてもよい。

## 【 0 1 4 3 】

フォーム記入とは、拡張性のあるアプリケーションのエンドユーザが入力したデータをプラットフォーム特定型のウィジェット形式で表すことを可能にする、プラットフォーム独立型のウィジェット間における関係を示している。言い換えれば、第 1 プラットフォーム特定型のプレゼンテーションにあるプラットフォーム特定型のウィジェットの状態を、第 2 プラットフォーム特定型のプレゼンテーションへ変換されたプラットフォーム特定型のウィジェットと同様の状態にすることである。プラットフォーム独立型のウィジェット間における関係は、開発者、知識ベース型システム、実験、あるいはプラットフォーム独立型のウィジェットと変換されたプラットフォーム独立型のウィジェット間において関係を作るその他のメカニズムにより、プラットフォーム独立型のプレゼンテーションモデルにおいて提供可能である。

20

## 【 0 1 4 4 】

例えば、図 6 にあるデータエントリ 8 0 にあるラベル(例:「アイテム名」、「ブランド名」)はピックリスト 8 6 に変換され、そして対応するデータエントリ領域は、拡張性のある「複数の同クラスから 1 タイプへの変換ルール」を利用している単一のデータエントリ 8 8 へと変換される。これら 2 つの変換は、プラットフォーム独立型のプレゼンテーションモデル、およびプラットフォーム特定型モデルにおけるフォーム記入関係にリンクさせてもよい。

30

## 【 0 1 4 5 】

変換マネージャモジュール 2 0 により、「複数の同クラスから 1 タイプへの変換ルール」、および「複数から複数タイプへの変換ルール」で変換されたプラットフォーム独立型のウィジェットを、他のプラットフォーム独立型のウィジェットへリンクさせてもよい。リンクは、プレゼンテーション内の 2 つ以上のプラットフォーム特定型のウィジェット間における動作関係(例:図 1 4 のバーチャルボタン 4 1 0 とバーチャルテキスト領域 4 1 2 間における動作関係)に関連している。第 1 プラットフォーム特定型のウィジェットと第 2 プラットフォーム特定型のウィジェット間におけるリンクは、第 1 プラットフォーム特定型のウィジェットの状態を変化させることになり、かつ第 2 プラットフォーム特定型のウィジェットの状態を変化させることになる。プラットフォーム特定型のウィジェット間におけるリンクは、プラットフォーム独立型のプレゼンテーションモデルにおいて、対応する第 1 & 第 2 プラットフォーム独立型のウィジェットのリンクにより提供されてもよい。

40

。

## 【 0 1 4 6 】

変換ルールが採用されると、オリジナルプラットフォーム独立型のウィジェットの実行状

50

態は別のフォームで表されてもよい。異なる変換ルールが特定のプラットフォームの制約に基づきオリジナルプラットフォーム独立型のウィジェットに対して適用可能なので、プラットフォーム特定型のウィジェットの実行状態が変換可能となる。オリジナルプラットフォーム独立型のウィジェットの実行状態は、プラットフォーム独立型のプレゼンテーションモデルに保存されてもよい。つまり、変換されたプラットフォーム独立型のウィジェットの実行状態は、オリジナルプラットフォーム独立型のウィジェットの実行状態へとマップされてもよい。

#### 【0147】

変換されたプラットフォーム独立型のウィジェットは、対応するプラットフォーム特定型のウィジェットとしてターゲット装置の表示画面上に表示される、変換ルールにより選択された装置独立型のウィジェットであってもよい。オリジナルプラットフォーム独立型のウィジェットは、表示画面上に表示される、開発者が選んだ装置独立型のウィジェットである。オリジナルプラットフォーム独立型のウィジェットは、プラットフォーム独立型のプレゼンテーションモデルに接続されてもよい。同様に、変換が成功したとき、変換されたプラットフォーム独立型のウィジェットもプラットフォーム独立型のプレゼンテーションモデルに接続されてもよい。上記の両接続は、コンポーネントノード形式で表現されてもよい。ここで各々のコンポーネントノードは、少なくとも1つのプラットフォーム独立型のウィジェットを表している。

#### 【0148】

実行状態の変換をするため、プラットフォーム独立型のプレゼンテーションモデルは、オリジナルプラットフォーム独立型のウィジェット及び変換されたプラットフォーム独立型のウィジェット両方のインスタンスを含んでいてもよい。GUIのマイグレーションの実行時間次第で、オリジナルプラットフォーム独立型のウィジェットだけのインスタンス、あるいは変換されたプラットフォーム独立型のウィジェットだけのインスタンスが、同時にアクティブ状態になってもよい。

#### 【0149】

図16は、変換されたプラットフォーム独立型のウィジェットの接続、及びアクティブ状態における変換されたプラットフォーム独立型のウィジェットの変化を表しているプラットフォーム独立型のプレゼンテーションモデル530の例である。図のプラットフォーム独立型のプレゼンテーションモデル530は、コンテナノード532、及び関連性のある複数のオリジナルコンポーネントノード534を有している。コンテナノード532は、論理パネルであるプラットフォーム独立型のウィジェットを表してもよい。また各々のオリジナルコンポーネントノード534、及び関連のあるプラットフォーム独立型のウィジェットは、パネル内のラジオボタン(RB)である、プラットフォーム独立型のウィジェットを表してもよい。

#### 【0150】

プラットフォーム独立型のプレゼンテーションモデルのコピー536による、プラットフォーム特定型のプレゼンテーションの生成中において、矢印537で示されている変換が起こり得る。プラットフォーム独立型のウィジェットへマップされるプラットフォーム特定型のウィジェット(コンテナノード532)が大きいため、ターゲット異種装置の表示画面上に配置できない時、この変換は起こる。例えば、コンテナノード532を変えずに、ラジオボタン(オリジナルコンポーネントノード534)を表している全てのプラットフォーム独立型のウィジェットを、変換されたコンポーネントノード538(コンボボックス:CB)へ変換してもよい。コンボボックスは、ラジオボタンと類似の機能を提供する小さなプラットフォーム独立型のウィジェットであってもよい。

#### 【0151】

変換されたコンポーネントノード538は、変換されたウィジェットを保存、あるいは表すためのプラットフォーム独立型のプレゼンテーションモデルのコピー536において作られた新しいコンポーネントノードであってもよい。以下の変換において、変換されたコンポーネントノード538は、「変換された子供ノード」としてコンテナノード532に

10

20

30

40

50

リンクされることにより、新しい論理関係を形成してもよい。新しい論理関係がアクティブになることにより、コンテナノード532とオリジナルコンポーネントノード534間における既存の論理関係と置き換えられてもよい。変換された子供ノード(変換されたコンポーネントノード538)は、コンテナノード532を特有の子供ノードとしてもよい。

#### 【0152】

コンテナノード532とオリジナルコンポーネントノード534間における前の論理関係は、図16において点線で表されている、プラットフォーム独立型のプレゼンテーションモデルのコピー536の論理参照として保存されてもよい。しかしながらオリジナルコンポーネントノード534は、開発者が明確に特定していない限り、コンテナノード532の子供ノードである必要はない。このようにIRツリーのレイアウト構造において、オリジナルコンポーネントノード534は含まれないことになる。またこの論理関係は、あるプラットフォーム特定型のプレゼンテーションから別のプラットフォーム特定型のプレゼンテーションへの実行状態のマイグレーション、及びイベント処理をサポートするのに保存されてもよい。

#### 【0153】

最も最適な変換ルールを選択するため、ルールには種類に応じて優先順位が設定されている。「複数から複数タイプへの変換ルール」は大幅に全体のプレゼンテーションを変え、そしてGUIの有用性を劣化させるので、「複数から複数タイプへの変換ルール」には、最も低い優先度が割り当てられてもよい。その他の変換ルールの優先度を決定するに当たって、変換マネージャモジュール20が各ルールを選択してもよい。

#### 【0154】

各変換ルールの選別機能により、オリジナルプラットフォーム独立型のウィジェットが有する機能、あるいはその他パラメータと適合性のない、変換されたプラットフォーム独立型のウィジェットを作成する変換ルールを削除してもよい。適合性のない機能には、オリジナルプラットフォーム独立型のウィジェットが、変換されたオリジナルプラットフォーム独立型のウィジェットによりサポートされていない入力メカニズムを要求する、変換ルールが含まれている。例えば、アプリケーション開発者が、アクション実行のためにマウスインイベント(例：マウスの矢印がボタンを指しているとき(クリックではない))に対応しているJ2ME DoJaボタンへ、プラットフォーム独立型のウィジェット用のマッピングを特定するとする。しかしながら、ソフトキーはマウスインイベントをサポートしないので、変換カスタマイズの間、ボタンはソフトキーに変換されることはない。このように、当該プラットフォーム独立型のウィジェット用に、かつ当該変換をサポートする変換ルールは、選別機能により削除されてもよい。

#### 【0155】

変換マネージャモジュール20は、一連のスペース縮小パラメータに基づき、変換ルールに優先順位を付けてもよい。スペース縮小パラメータの優先順位とは、変換に基づくプラットフォーム独立型のウィジェットのサイズ、及び/または変換ルールをプラットフォーム独立型のウィジェットに適用することから生じるサイズ縮小量に基づいてもよい。プラットフォーム独立型のプレゼンテーションにあるプラットフォーム独立型のウィジェットにおけるスペース縮小とは、対応するプラットフォーム独立型のウィジェットにおけるスペース縮小により特定される。

#### 【0156】

最小のスペース縮小を提供する変換ルールには、最も高い優先順位が与えられてもよい。あまりコンパクトでない対応するプラットフォーム特定型のウィジェットは、コンパクトなプラットフォーム特定型のウィジェットより高いGUIの有用性を持っているので、最も高い優先順位は最小のスペース縮小を提供する変換ルールに与えられてもよい。スペース縮小パラメータは、プラットフォーム独立型のウィジェットの幅縮小率、高さ縮小率、そして大きさ縮小率を含んでいてもよい。

#### 【0157】



変換ルールをトリガさせるプラットフォーム独立型のウィジェットの状態に基づいて、スペース縮小パラメータの優先順位は変更されてもよい。例えばプラットフォーム独立型のウィジェット用の変換ルールが、幅があって、かつ対応するプラットフォーム独立型のウィジェットによりトリガされると、幅縮小率が最も重要なパラメータとして特定されてもよい。次に高さ縮小率が2番目に重要なパラメータとして特定された場合、大きさ縮小率は無視されてもよい。同様に変換ルールが、高すぎるプラットフォーム特定型のウィジェットによりトリガされると、高さ縮小率が最も重要なパラメータとして特定され、次に幅縮小率が2番目に重要なパラメータとして特定されるので、大きさ縮小率は無視されてもよい。また変換ルールが、幅と高さが両方あり過ぎるプラットフォーム特定型のウィジェットによりトリガされると、大きさ縮小率が最も重要なパラメータとして特定され、残りのパラメータ(幅と高さ縮小率)は無視されてもよい。

10

#### 【0158】

変換ルールをタイプ化するカテゴライゼーション、選別、およびスペース縮小を実行することにより、各々に対し優先順位が付けられてもよい。また、タイプ化のカテゴライゼーション、選別、またはスペース縮小だけを利用することにより、優先順位が付けられてもよい。さらに、タイプ化のカテゴライゼーション、選別、またはスペース縮小のいかなる組み合わせや順番などを利用することにより、優先順位が付けられてもよい。

#### 【0159】

異種装置プラットフォームがJava技術を用いて拡張性のあるアプリケーションを実行することに関し、変換ルールはJavaクラスで特定されてもよい。各々の変換ルールは、1つのJavaクラスで表される。クラスの詳細に関して、実行状態及びイベント処理の変換が、プラットフォーム独立型のウィジェット用のアプリケーション開発者によって特定されてもよい。

20

#### 【0160】

例えば図6のような変換を実施する、Javaを用いて作成される変換ルールの例を以下に示す。

プライベートクラスのフォーム記入の変換

/\* プラットフォーム特定型のウィジェット変換 \*/

- ・新しいプラットフォーム独立型のウィジェット(ドロップダウンボックス(DD1))の構成
- ・バーチャルな「アイテム名」、「ブランド名」等、ラベルのプロパティ(例：フォントサイズ)をDD1へコピー

30

- ・バーチャルテキスト領域(TF1)の構成

- ・バーチャルな「アイテム名」、「ブランド名」等、テキスト領域のプロパティをTF1へコピー

/\* 実行状態の変換 \*/

- ・オリジナルのバーチャルラベルとテキスト領域のペア例を、以下のJavaハッシュテーブルに記録：

アイテム名	ボトルウォーター
ブランド名	アルハムブラ
数量	4
予想価格/ユニット	\$0.75

40

- ・拡張性のある選択イベントのリスナー(イベントライブラリからの)をDD1へ追加、そしてアクションを特定。よってエンドユーザがアイテムをDD1(例：アイテム名)から選択すると、それに対応する物(例：ボトル水)がハッシュテーブルから選択される。

- ・その物をTF1上に表示する。

/\* イベント処理の変換 \*/

- ・エンドユーザがエンターキー、あるいは相当のキーをヒットした時、「アイテム名」、あるいは「ブランド名」等、テキスト領域にある物が自動的に更新されることをアプリケ

50

ーション開発者は特定する。

- ・マイグレーション後に、エンターキーイベントが受信されたら、TF 1 上に、例えば「コーラ」のような物を入力する。
- ・例えば「アイテム名」のような、今選択されたアイテムをDD1上に入力する。
- ・ハッシュテーブル上において、選択されたアイテムのタイトルを検索し、そしてそれに対応するアイテム（物）を更新する。
- ・イベントの出所として入力された「アイテム名」、及び新しい物として入力された「コーラ」を用いてイベントを作成する。
- ・オリジナルの状態と変換された状態を同期化させるため、イベントをオリジナルのパーティクル「アイテム名」テキスト領域へ送信。

10

#### 【0161】

図17は、図13と図14を参照して説明されるレイアウトカスタマイズのプロセス、及び変換カスタマイズのプロセスの実施形態を表すフロー図である。動作はIRツリーのルートノード44aを現在ノードに設定することから始まる。(ブロック540)現在ノードの未処理である直接子供ノードがあるかどうかを判定する。(ブロック542)イエスなら、最も高い優先順位の未処理の直接子供ノードが現在ノードへ設定され(ブロック544)、処理はブロック542へ戻る。もし未処理の直接子供ノードがないなら、現在ノードはルートノード44aであるかどうかを判定する。(ブロック546)もしイエスなら、プラットフォーム特定型のプレゼンテーション、プラットフォーム独立型のプレゼンテーション、及び拡張性のあるGUIライブラリ12の関連性のある部分が、ターゲット異種装置プラットフォームへ送信される。(ブロック548)

20

#### 【0162】

もし現在ノードがルートノード44aでない場合、プラットフォーム特定型のプレゼンテーションモデル内で現在ノードと関連のあるサブノードで表されているプラットフォーム特定型のウィジェットが、アプリケーション開発者によって特定されているプロパティに基づいて、ページ上に配置されてもよい。(ブロック550)プロパティは、ページ上でプラットフォーム特定型のウィジェットの位置を示すため、レイアウトを制御するJava Gridbag に類似であってもよい。ページのサイズが算出される。(ブロック552)ページのサイズが、ターゲット表示画面のサイズと比較される。(ブロック554)もしページが、ターゲット表示画面内に適合するなら、現在ノードに関連しているプラットフォーム独立型のウィジェットが、プラットフォーム独立型のプレゼンテーションモデルのコピー内に保存される。(ブロック556)処理済みの現在ノードがマークされ(ブロック558)、別の現在ノードを選択するために、動作はブロック540へ戻る。一方ページのサイズが、ターゲット表示画面より大きいのなら(ブロック554)、フローレイアウトマネージャを使っているプラットフォーム独立型のウィジェットを再配置することにより、プラットフォーム特定型のウィジェットはページ内で動的に再配置される。(ブロック560)

30

#### 【0163】

図18に進み、ページのサイズが、再びターゲット表示画面のサイズと比較される(ブロック562)。もしページがターゲット表示画面に適合するのなら、動作はブロック556へと戻り、プラットフォーム特定型のプレゼンテーションのコピー内においてプラットフォーム独立型のウィジェットを保存する。もしページがまだターゲット表示画面のサイズより大きいのなら、現在ノード(プラットフォーム独立型のウィジェット)、あるいは他の関連するノードが分割可能かどうかを判定する(ブロック564)。ブロック566でイエスと判定されたなら、少なくとも第2ページがインスタンス化される。プレゼンテーションがページに適合するまで、特定された分割ノードに関連しているプラットフォーム特定型のウィジェットが、フローマネージャを使うことにより第1ページと少なくとも第2ページの間で分配される(ブロック568)。プラットフォーム独立型のウィジェットが、プラットフォーム独立型のプレゼンテーションのコピー内において保存される。(ブロック569)処理済みの現在ノードがマークされ(ブロック570)、別の現在ノードを選択するために、動作はブロック540へ戻る。

40

50

## 【0164】

関連のあるサブノード、あるいは現在ノードのいずれもが分割できない場合（ブロック564）、現在ノードと関連のある変換可能なプラットフォーム独立型のウィジェットが特定される。（ブロック574）プラットフォーム独立型のウィジェットが、SGUIコンポーネントであるかどうかの判定が実行される。（ブロック576）もしプラットフォーム独立型のウィジェットがSGUIコンポーネントであるなら、複合ウィジェット内にあるSGUIコンポーネント単体は、重要プロパティに基づいたSGUIコアコンポーネント、及びSGUIオプションコンポーネントとして、カテゴリー化される。（ブロック578）

## 【0165】

図19に進み、SGUIオプションコンポーネントが全て処理されたかどうかを判定する。（ブロック580）もしノーなら、次の未処理のSGUIオプションコンポーネントが選択される。（ブロック582）適用可能な全ての変換ルールが、選択された全てのSGUIコンポーネント用に検索される。（ブロック584）検索された変換ルールが、4つある変換ルールの中の1つとして特定される。（ブロック586）「複数から複数タイプへの変換ルール」に対し、最も低い優先順位が付けられる。（ブロック588）残り3つの変換ルールについては選別され、不適合なルールが除去される。（ブロック590）残り3つの変換ルールに関して、変換されたSGUIコンポーネントの予想されるスペース縮小量が算出される。（ブロック592）

## 【0166】

図20に進み、スペース縮小量に基づいて、残りの変換ルールに対しても優先順位が付けられる。（ブロック594）最も高い優先順位（最も少ないスペース縮小）を持つ変換ルールが、SGUIコンポーネントに適用される。（ブロック596）変換されたSGUIコンポーネントが、プラットフォーム特定型のウィジェットへマップされる。（ブロック598）プラットフォーム特定型のウィジェットが、フローレイアウトマネージャにより、ページ内に配置される。（ブロック599）ここでページが、ターゲット異種装置プラットフォームの表示画面より大きいかどうかを判断する。（ブロック600）もしノーなら、動作は図17のブロック556に戻り、プラットフォーム独立型のプレゼンテーションモデル等のコピー内に、現在ノードと関連したプラットフォーム独立型のウィジェットを保存する。もしページがまだ大きすぎるのなら、プラットフォーム独立型のウィジェット内の全てのSGUIコンポーネントが処理されたかどうかを判定する。（ブロック602）もしノーなら、動作は図19のブロック580へ戻る。もしプラットフォーム独立型のウィジェットにおける全てのSGUIコンポーネントが処理されているのなら、対応するプラットフォーム独立型のウィジェット用のノードが、処理されたものとしてマークされる。（ブロック604）現在ノードと関連している全ての変換可能なプラットフォーム独立型のウィジェットが、処理されたかどうかを判定する。（ブロック606）もしノーなら、動作は図18のブロック574に戻り、その他の変換可能なプラットフォーム独立型のウィジェットを特定する。もし現在ノードと関連している全ての変換可能なプラットフォーム独立型のウィジェットが処理されたなら、ノードを縮小することが不可能だというアラームがアプリケーション開発者用に生成され（ブロック608）、別の現在ノードを選択するため、動作は図17のブロック504へ戻る。

## 【0167】

図19に戻り、プラットフォーム独立型のウィジェットにおける全てのSGUIオプションコンポーネントが処理された（あるいは存在しないか）と判断されると（ブロック580）、未処理のSGUIコアコンポーネントが選択され（ブロック610）、動作はブロック584へと進む。図18に戻るが、プラットフォーム独立型のウィジェットがSGUIコンポーネント複合でなければ、動作は図19のブロック584へと進み、SGUIコンポーネント複合を持つプラットフォーム独立型のウィジェットに似た、SGUIコンポーネント単体を持つプラットフォーム独立型のウィジェットの処理をする。

## 【0168】

図13へ戻り、実行状態の変換、及びプラットフォーム独立型のウィジェットのイベント

10

20

30

40

50

処理は、変換カスタマイズの一部分として起こってもよい。前にも述べたが、マイグレーションしたプレゼンテーションは、オリジナルのプレゼンテーションから、異なる一連のプラットフォーム独立型のウィジェットを採用してもよい。エンドユーザ用にアプリケーションの継続性を保つため、オリジナルプラットフォーム特定型のプレゼンテーションにおけるプラットフォーム特定型のウィジェットの実行状態は、マイグレーションしたプラットフォーム特定型のプレゼンテーションにおけるプラットフォーム特定型のウィジェットへマップされてもよい。

#### 【0169】

実行状態のマイグレーションを、拡張性のあるGUIライブラリ12を用いて実行することにより、第1プラットフォーム特定型のプレゼンテーションにおけるプラットフォーム特定型のウィジェットの現実行状態が、プラットフォーム独立型のプレゼンテーションモデル(中間プレゼンテーション)内の対応するプラットフォーム独立型のウィジェットへマップされてもよい。プラットフォーム独立型のプレゼンテーションモデルのコピーがコピー&削除された後、ターゲット異種装置プラットフォーム用の第2プラットフォーム特定型のプレゼンテーションの開発中に、ノードに関連した対応しているプラットフォーム特定型のウィジェットが、ページ内に配置される。プラットフォーム独立型のプレゼンテーションモデルにおけるプラットフォーム独立型のウィジェットの実行状態が、第2プラットフォーム特定型のプレゼンテーションにおいて、対応しているプラットフォーム特定型のウィジェットの現実行状態として適用されてもよい。したがって、第1プラットフォーム特定型のプレゼンテーションの現実行状態が、第2プラットフォーム特定型のプレゼンテーションにおいて実装されてもよい。

#### 【0170】

マイグレーションの前に、プラットフォーム特定型のウィジェットの現実行状態と、それに対応するプラットフォーム独立型のウィジェットの現実行状態を同期させてもよい。この同期化には、外観、状態変化、あるいは/またはプラットフォーム特定型のウィジェットから入手されたデータを有する、プラットフォーム独立型のウィジェットの現実行状態を更新することを含んでもよい。ターゲット異種装置プラットフォーム用のプラットフォーム特定型プレゼンテーションの開発中において、プラットフォーム独立型のウィジェットの現実行状態が保存されてもよい。マイグレーションに続いて、プラットフォーム独立型のウィジェットの現実行状態が検索され、そしてプラットフォーム特定型のプレゼンテーションを用いる、マイグレーションしたプラットフォーム特定型のウィジェット上にプラットフォーム独立型のウィジェットの現実行状態が表わされる。

#### 【0171】

GUIのマイグレーションのイベント中に、変換ルールが適用されたとき、オリジナルプラットフォーム独立型のウィジェットの現実行状態を変換されたプラットフォーム独立型のウィジェットにマップすることにより、類似のプロセスが実行されてもよい。変換されたプラットフォーム独立型のウィジェットは、プラットフォーム特定型のプレゼンテーションにおけるプラットフォーム特定型のウィジェットへマップされてもよい。プラットフォーム特定型のプレゼンテーションを達成する為の、プラットフォーム独立型のプレゼンテーションのレイアウト及び変換が実行されてもよい。(図17から20において説明済み)

#### 【0172】

レイアウト及び変換中において、プラットフォーム独立型のウィジェット(SGUIコンポーネント)がプラットフォーム特定型のウィジェットへマップされると、SGUIコンポーネントライブラリにおけるAPIは、同様にプラットフォーム独立型のウィジェットの状態をプラットフォーム特定型のウィジェットの状態へマップしてもよい。このようにGUIのマイグレーションが起こると、実行状態は、プラットフォーム独立型のプレゼンテーションにおけるプラットフォーム独立型のウィジェットとプラットフォーム特定型のプレゼンテーションにおけるプラットフォーム特定型のウィジェットの間を往来してもよい。

#### 【0173】

実行状態のマッピングは、各々の変換ルールに対して特有のものであるので、各々の変換ルー

10

20

30

40

50

ルは実行状態を変換できるメカニズムを提供してもよい。実行状態の変換例が、Java(登録商標)コードを使って作成される変換ルールの例(前に説明済み)に含まれていてもよい。コード例に関して、全てのオリジナルのテキスト領域の状態が、Javaハッシュテーブル(既に説明済み)に保存される。変換されたテキスト領域は、ドロップダウンボックスの状態に基づき、全てのオリジナルのテキスト領域の状態を一度に表示する。

#### 【0174】

実行状態の変換、またはマッピングの手順は、異なるタイプの変換において適用されてもよい。例えば、SGUIシステムの初期使用中やマイグレーション中において、プラットフォーム独立型のウィジェットの実行状態が、プラットフォーム特定型のウィジェットの10 実行状態に変換される。さらにマイグレーションの前に、プラットフォーム特定型のウィジェットの10 実行状態が、プラットフォーム独立型のウィジェットの10 実行状態へ変換されてもよい。さらに、プラットフォーム独立型のウィジェットの10 実行状態が、別のプラットフォーム独立型のウィジェットの10 実行状態へ変換されてもよい。(例：プラットフォーム特定型のプレゼンテーションの進展中に、変換がトリガされた時)

#### 【0175】

図16を参照しながら説明した通り、例えばラジオボタンのセットのような、スペースを取り過ぎる複数の第1コンポーネントノード534は、コンボボックスのような、新しい第2コンポーネントノード単体538への変換が必要かもしれない。ラジオボタンの実行状態からコンボボックスの実行状態への変換は、プラットフォーム独立型のラジオボタンの実行状態から、マッピングを通して、プラットフォーム独立型のコンボボッ20 クスの実行状態への変換を含んでもよい。またマッピングを通して、プラットフォーム独立型のラジオボタンの実行状態が、プラットフォーム独立型のコンボボックスの実行状態へ変換されてもよい。さらに、プラットフォーム特定型のコンボボックスからプラットフォーム独立型のコンボボックスへ実行状態を変換する、実行状態のマッピングが実行されてもよい。プラットフォーム独立型をプラットフォーム独立型の実行状態の変換へ用いることにより、プラットフォーム独立型のプレゼンテーションモデルは、ウィジェットの10 実行状態を保存することが可能になる。

#### 【0176】

図21は、実行状態の変換とプラットフォーム独立型のウィジェットのマッピング間における一般的な動作を表しているプロセスのフロー図である。オリジナルプラットフォーム20 独立型のウィジェットのタイプが特定されるところで、動作は始まる。(ブロック650)例えば、ウィジェットのタイプは、ラジオボタンタイプやコンボボックスタイプなどである。オリジナルプラットフォーム独立型のウィジェットが、少なくとも1つの変換されたプラットフォーム独立型のウィジェットに変換される。(ブロック652)変換が成功すると、変換されたウィジェットのタイプが特定される。(ブロック654)オリジナルプラットフォーム独立型のウィジェットの10 実行状態が検索される。(ブロック656)実行状態が、決定された構成(例：配列)で保存される。(ブロック658)ウィジェットのタイプに応じて、オリジナルプラットフォーム独立型のウィジェットの10 実行状態が、変換されたプラットフォーム独立型のウィジェットの10 実行状態へ変換される。(ブロック660)変換された実行状態は、変換されたプラットフォーム独立型のウィジェットに関連づけて保存される。(ブロック662)変換されたプラットフォーム独立型のウィジェットの10 実行状態が、プラットフォーム特定型のプレゼンテーション内のプラットフォーム特定型のウィジェットへマップされる。(ブロック664)変換されたプラットフォーム独立型のウィジェットの10 実行状態が、プラットフォーム特定型のウィジェットの10 実行状態へマップされる。(ブロック666)変換されたプラットフォーム独立型のウィジェットの10 実行状態が、変換されたプラットフォーム特定型のウィジェット上に表される。(ブロック668)

#### 【0177】

イベント処理の変換は、GUIのマイグレーション後に、同レベルのGUIインタラクションを提供するので重要である。マウスボタンを押したり、あるいは第1異種装置プラットフォーム上におけるプラットフォーム特定型のウィジェットのタイプされた文字入力等のイベ30

ントが、バーチャルイベント形成のために抽出されてもよい。別の異種装置プラットフォームにおいて、上記プラットフォーム特定型のウィジェットに対して同じ入力を表している異なるイベントが、同様に同じバーチャルイベント形成のために抽出されてもよい。異種装置プラットフォーム上における異なるイベントが、バーチャルイベントに基づいている、プラットフォーム独立型のプレゼンテーションモデル内のプラットフォーム独立型のウィジェットの同じ入力（を表している異なるイベント）へマップされてもよい。

#### 【0178】

例えば図6が、J2SE SWINGプラットフォームからJ2ME Personal Java AWT プラットフォームまでの、GUIのマイグレーションを表していると仮定する。マイグレーションの前において、エンドユーザは、SWINGテキスト領域であるかもしれない「アイテム名」として特定されているデータを、データエントリ80に入力し、そしてSWINGイベントが生成されてもよい。SWINGイベントは、データベース内に「アイテム名」データを保存するアクションを引き起こしてもよい。マイグレーション後において、SWINGテキスト領域がAWTテキスト領域へと変換したとき、AWTテキスト領域がAWTイベントのみをサポートしていたとしても、SWINGイベントは同様のアクションを引き起こしてもよい。

#### 【0179】

拡張性のあるGUIライブラリ12を使うことにより、SWINGイベントとAWTイベントが、汎用のバーチャルイベント形成用に抽出されてもよい。このように、汎用なバーチャルイベントが受信されたとき（例：「アイテム名」データが入力されたとき）、それに関連するアクションが引き起こされる。変換ルールがプラットフォーム独立型のウィジェットに適用されたとき、上記と同様な抽出が適用されてもよい。イベントマッピングは、各々の変換ルールに対して特有のものであるので、各々の変換ルールはまたイベント処理の変換を提供してもよい。イベント処理の変換例は、Javaを用いて作成された変換ルールの例に関する個所で既に説明済みである。（図6参照）コードに関して、イベントが、変換された単一のデータエントリ88から生成されたとき、変換マネージャモジュール20は、複数のデータエントリ80内においてオリジナルのテキスト領域として動作し、そしてアクションを引き起こすためにイベントを送信してもよい。

#### 【0180】

実行状態の変換あるいはマッピングと同様に、イベント処理の変換、あるいはマッピングも各々の変換ルールに対して特有のものであってもよい。実行状態とは違い、イベント処理は実行時間により異なるものではないので、イベント処理のマイグレーションは必要ではない。代わりに、少なくとも1つのオリジナルプラットフォーム独立型のウィジェットのタイプ、及び少なくとも1つの変換されたオリジナルプラットフォーム独立型のウィジェットのタイプが特定される。そして、両方のタイプを利用できるイベント処理のメカニズムが、特定されてもよい。オリジナルプラットフォーム独立型のウィジェットのイベント処理メカニズムは、変換されたプラットフォーム独立型ウィジェットのイベント処理メカニズムにマップされてもよい。マッピングにより、オリジナルプラットフォーム独立型のウィジェットの状態変化に影響を与えている、入出力機能が可能になる。この機能は、変換されたプラットフォーム独立型ウィジェットにより実行される。

#### 【0181】

既に記載済みのSGUIシステムは、拡張性のあるアプリケーションのアプリケーションGUI用に装置プラットフォーム独立型の動作を提供している。SGUIシステムの拡張性により、拡張性のあるアプリケーションは、異なるサイズの表示画面、入力方法、及び異種装置プラットフォームのユーザインターフェイスに関連しているGUIライブラリでの動作が可能となる。さらに、SGUIシステム全体のデザインにより、拡張性のあるアプリケーションが動作可能な異種装置プラットフォームのユーザインターフェイスに関連する複雑性を回避しながら、拡張性を有するアプリケーションの開発者は、手軽にSGUIシステムを使うことができる。

#### 【0182】

SGUIシステムにより、拡張性のあるアプリケーションにおいて、装置プラットフォーム独

10

20

30

40

50

立型のアプリケーションGUIの使用が可能になる。アプリケーションGUIは、SGUIシステムがターゲット装置プラットフォーム用にカスタマイズするプラットフォーム独立型のプレゼンテーションモデルにより表されてもよい。プラットフォーム独立型のプレゼンテーションモデルのカスタマイズは、拡張性のあるアプリケーションにより特定されるプロパティ、及びターゲット装置プラットフォームの機能や性能に基づいてもよい。したがって、装置プラットフォーム独立型のアプリケーションGUIは、複数の異種装置プラットフォーム上において表示される装置プラットフォーム特定型のアプリケーションGUI用に、SGUIシステムによりカスタマイズされてもよい。

【0183】

SGUIシステムは3つの長所を持っている。1つめはGUIのマイグレーションのコンセプトである。2つめはエンドユーザが、スクロール機能を使用しなくても、変換を用いることによりプレゼンテーションのページをターゲット異種装置プラットフォームの表示画面に適合できることである。3つめは、SGUIシステムが有する柔軟性により、アプリケーション開発者は、特定の拡張性のあるアプリケーションのアプリケーションGUIをベストサポートするアプリケーション特定型の変換ルールを設定することが可能になる。

10

【0184】

本発明を説明するために、特定の実施形態を使用した。本発明の趣旨を逸脱しない範囲で、本発明に対し種々の変更または付加を施すことは、当業者にとって容易である。したがって、本出願における図面と説明は、単なる事例に過ぎず、本発明を限定するものではない。

20

【0185】

【発明の効果】

プラットフォーム独立型のプレゼンテーションモデルの実行状態を更新し、プラットフォーム独立型のプレゼンテーションモデルから前記第2異種装置プラットフォーム上で動作可能な第2プラットフォーム特定型のプレゼンテーションを生成し、そして第1異種装置プラットフォームからの現実行状態を実行することにより、SGUIウィジェットの実行状態をマイグレーションすることが可能になる。

【図面の簡単な説明】

【図1】 拡張性のあるグラフィカルユーザインターフェイスシステムのブロック図である。

30

【図2】 図1に示す拡張性のあるグラフィカルユーザインターフェイスシステムの一部のより詳細なブロック図である。

【図3】 図1に示す拡張性のあるグラフィカルユーザインターフェイスシステムが利用した中間表現の一例を示した図である。

【図4】 図1に示す拡張性のあるグラフィカルユーザインターフェイスシステムの別の部分のより詳細なブロック図である。

【図5】 図1に示す拡張性のあるグラフィカルユーザインターフェイスシステムを使っているグラフィカルユーザインターフェイスコンポーネントの変換例を示す図である。

【図6】 図1に示す拡張性のあるグラフィカルユーザインターフェイスシステムを使っているグラフィカルユーザインターフェイスコンポーネント複合の変換例を示す図である。

40

【図7】 図1に示す拡張性のあるグラフィカルユーザインターフェイスシステムの動作を示している流れ図である。

【図8】 図7に示す流れ図の続きである。

【図9】 図7に示す流れ図の続きである。

【図10】 図1に示す拡張性のあるグラフィカルユーザインターフェイスシステムで動作している装置を有している通信システムの一例である。

【図11】 図10に示す通信システム内の拡張性のあるグラフィカルユーザインターフェイスシステムの動作を示す流れ図である。

【図12】 図11に示す流れ図の続きである。

50

【図 1 3】 図 1 に示す拡張性のあるグラフィカルユーザインターフェイスシステムで動作している装置を有する通信システムの他の例である。

【図 1 4】 拡張性のあるグラフィカルユーザシステムが利用しているプラットフォーム独立型のプレゼンテーションモデルの例である。

【図 1 5】 図 1 3 の通信システム内を GUI マイグレーションしている間における拡張性のあるグラフィカルユーザインターフェイスシステムの動作を示す流れ図である。

【図 1 6】 図 1 の拡張性のあるグラフィカルユーザインターフェイスシステムを使っているプラットフォーム独立型のプレゼンテーションモデル内における変換例である。

【図 1 7】 図 1 3 のグラフィカルユーザインターフェイスシステムでページをカスタマイズしている流れ図である。

10

【図 1 8】 図 1 6 に示す流れ図の続きである。

【図 1 9】 図 1 6 に示す流れ図の続きである。

【図 2 0】 図 1 6 に示す流れ図の続きである。

【図 2 1】 図 1 の拡張性のあるグラフィカルユーザインターフェイスにおける実行状態の変換例を表しているフロー図である。

【符号の説明】

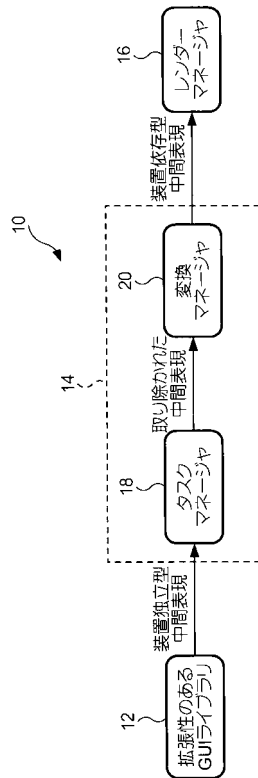
1 0 .....SGUI システム、 1 2 .....拡張性のある GUI ライブラリ(モジュール)、 1 4 .....カスタマイズモジュール、 1 6 .....レンダーマネージャ(モジュール)、 1 8 .....タスクマネージャ(モジュール)、 2 0 .....変換マネージャ(モジュール)、 2 2 .....拡張性のある GUI コンポーネントライブラリ(モジュール)、 2 4 .....中間表現、 2 6 .....拡張性のある GUI イベント変換(モジュール)、 4 0 .....IR ツリー、 4 2 .....コンポーネントノード、 4 4 .....コンテナノード、 4 6 .....タスク、 4 8 .....ページ、 5 0 .....変換ルール(モジュール)、 5 2 .....サイズ変換コンポーネント(モジュール)、 5 4 .....代替コンポーネント(モジュール)、 5 6 .....GUI 変換ルール(モジュール)単体、 5 8 .....GUI 変換ルール(モジュール)複合、 6 0 .....ダイナミックレイアウト(モジュール)、 6 2 .....スタイルガイド(モジュール)、 6 4 .....SGUI コンポーネントリスト(モジュール)、 7 0 .....第 1 SGUI コンポーネント、 7 2 .....第 2 SGUI コンポーネント、 7 6 .....第 1 SGUI コンポーネント複合、 7 8 .....第 2 SGUI コンポーネント複合、 8 0 .....(複数の)データエントリ、 8 2 .....優先度セレクション、 8 4 .....好みセレクション、 8 6 .....ピックリスト、 8 8 .....単一のデータエントリ、 2 0 0 .....通信システム、 2 0 2 .....装置プラットフォーム、 2 0 4 .....変換サーバ、 2 0 6 .....ライブラリサーバ、 2 0 8 .....ネットワーク、 2 1 0 .....拡張性のあるアプリケーション、 3 0 2 .....第 1 (異種装置)プラットフォーム、 3 0 4 .....第 2 (異種装置)プラットフォーム、 3 0 6 .....サーバコンピュータ、 4 4 a .....ルート(コンテナ)ノード、 4 4 m .....コンテナノード(アイテムタスクのサーチ)、 4 4 n .....第 1 タスクノード(タスクのサーチ)、 4 4 p .....第 2 タスクノード(結果タスクの分類)、 4 4 q .....第 3 タスクノード(結果タスクの表示)、 4 0 4 .....GUI プレゼンテーション、 4 0 8 .....バーチャルラベル(アイテムのサーチ：第 1 コンポーネントノード)、 4 1 0 .....バーチャルボタン(サーチ：第 2 コンポーネントノード)、 4 1 2 .....バーチャルテキスト領域(第 3 コンポーネントノード)

20

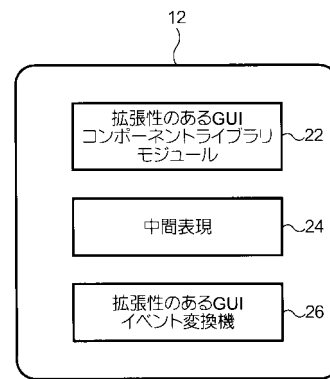
30



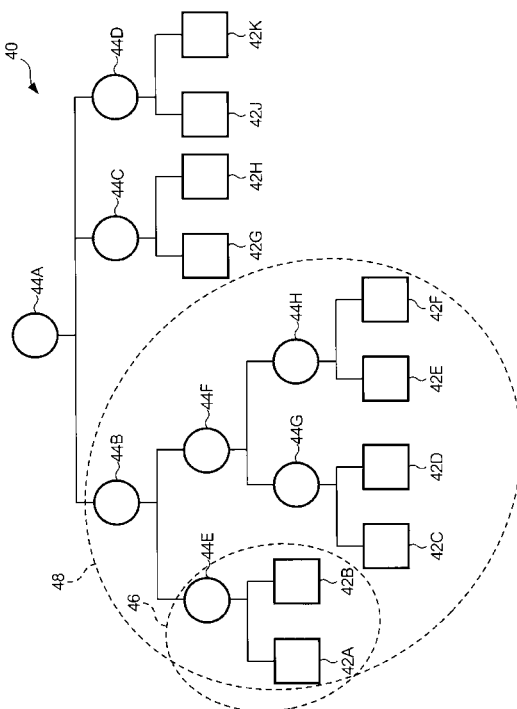
【図 1】



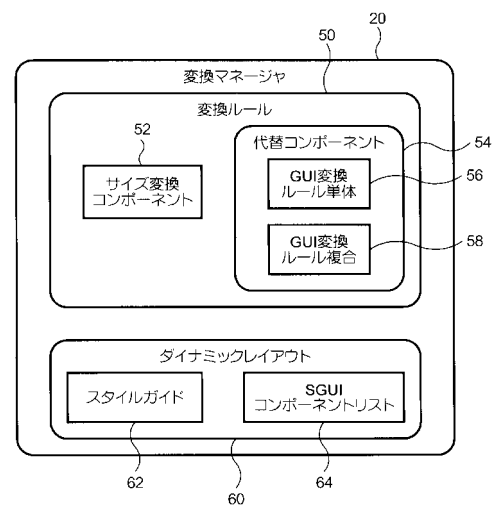
【図 2】



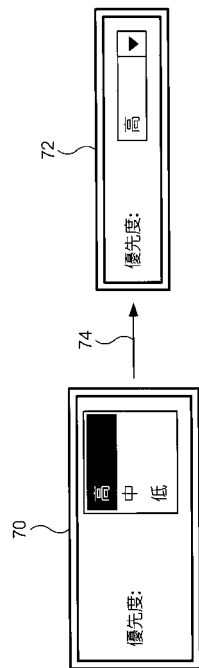
【図 3】



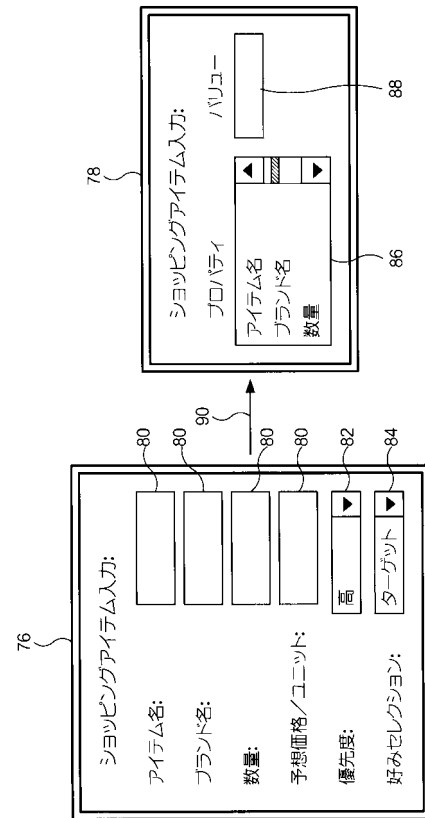
【図 4】



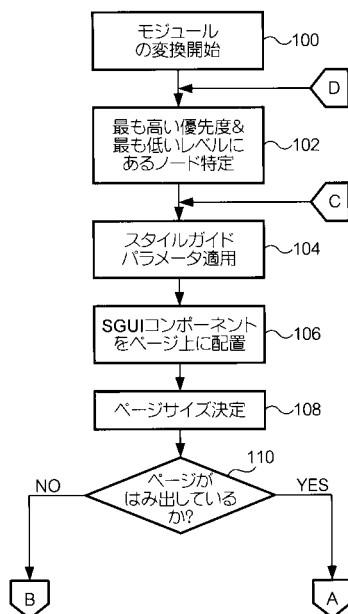
【図 5】



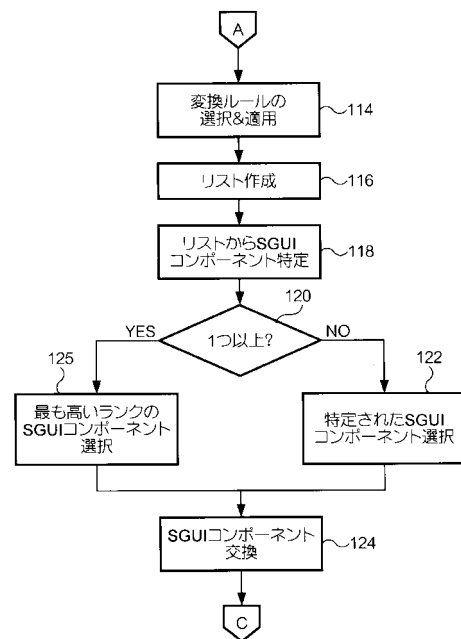
【図 6】



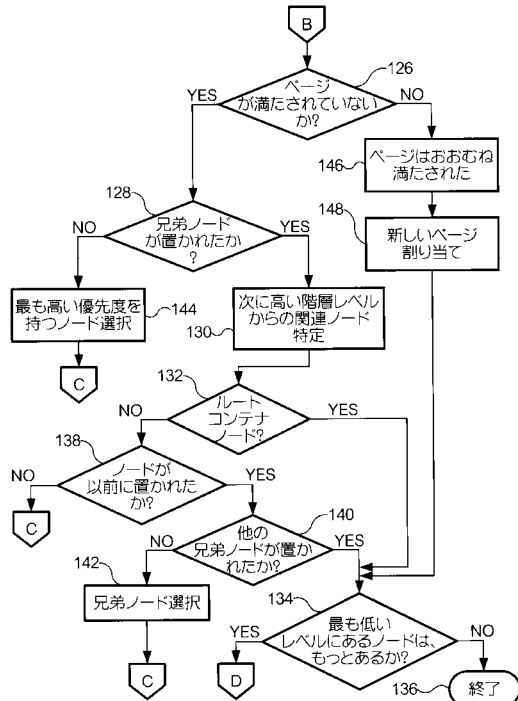
【図 7】



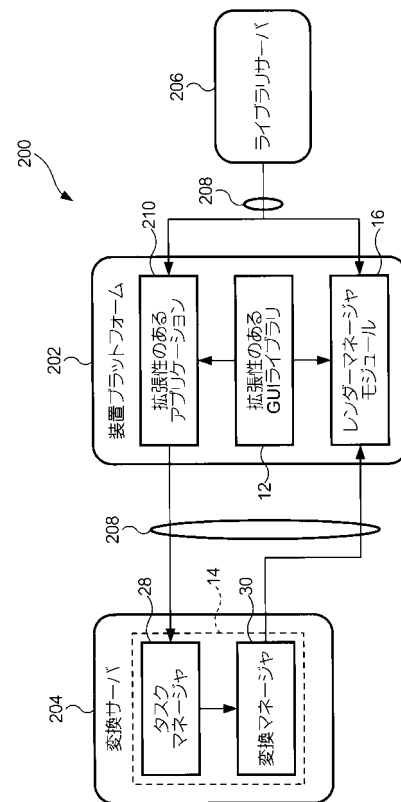
【図 8】



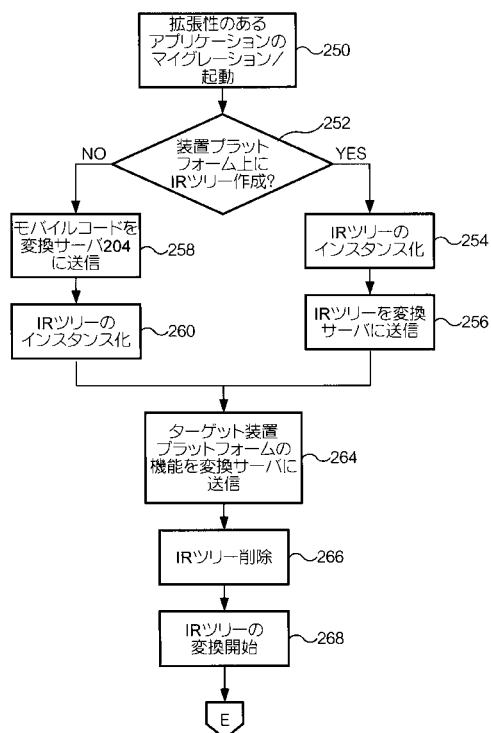
【図 9】



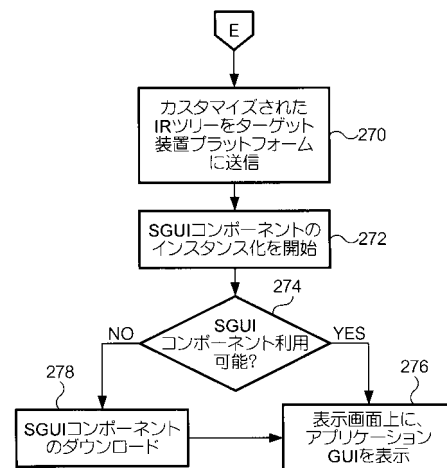
【図 10】



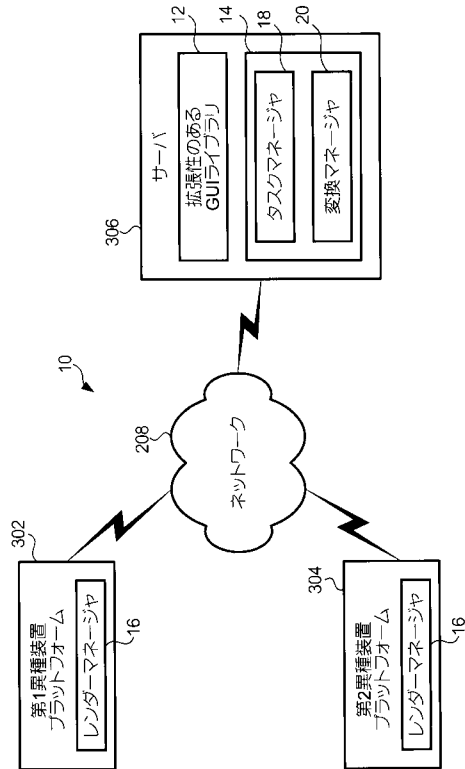
【図 11】



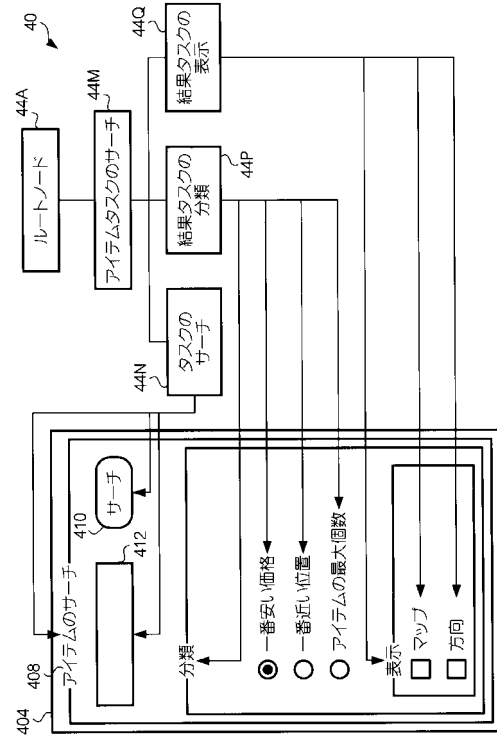
【図 12】



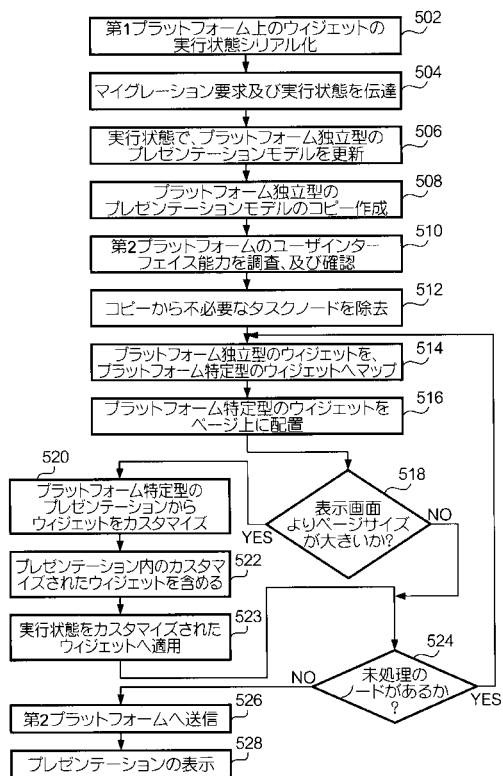
【 図 1 3 】



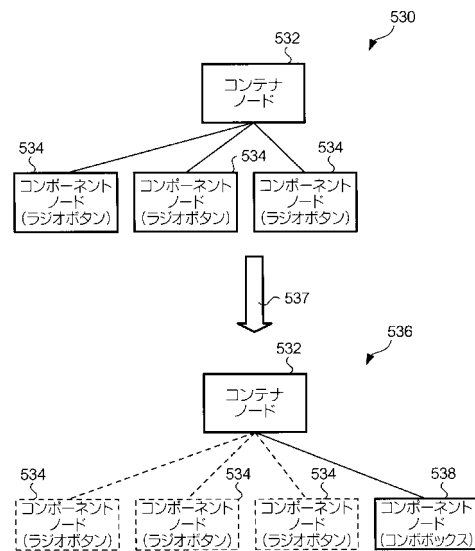
【 図 1 4 】



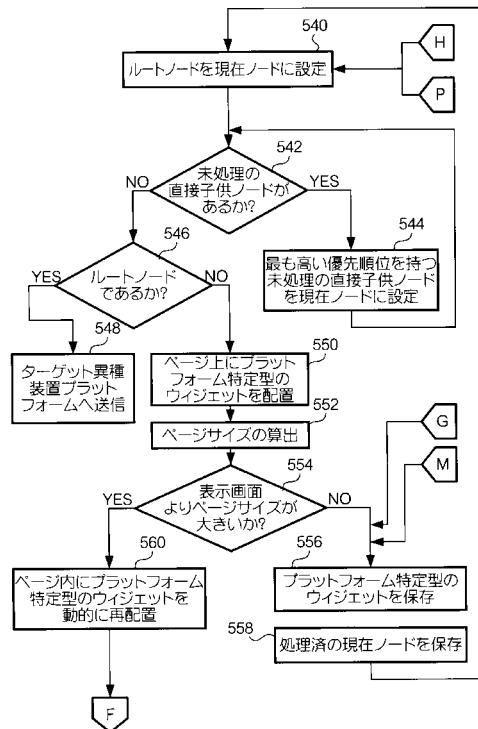
【 図 1 5 】



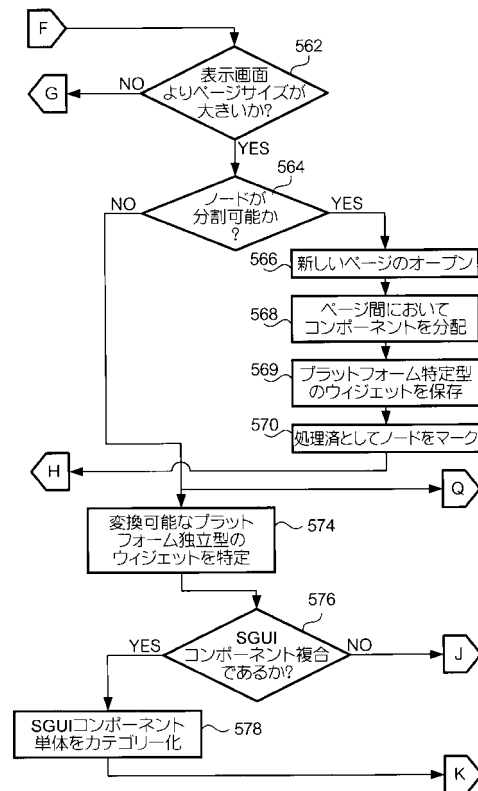
【 図 1 6 】



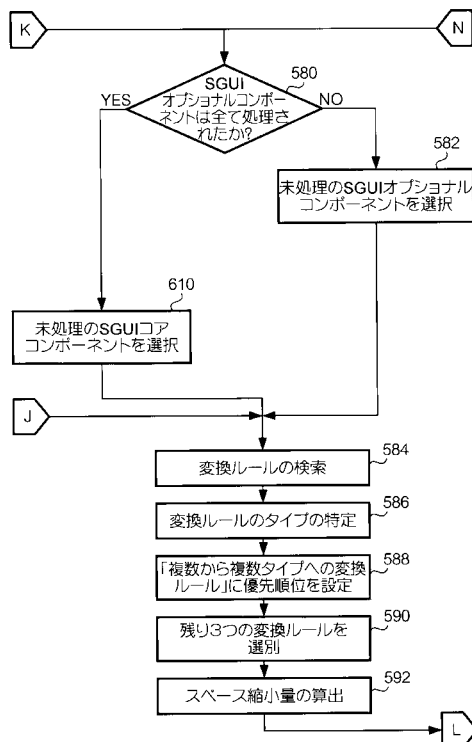
【図 17】



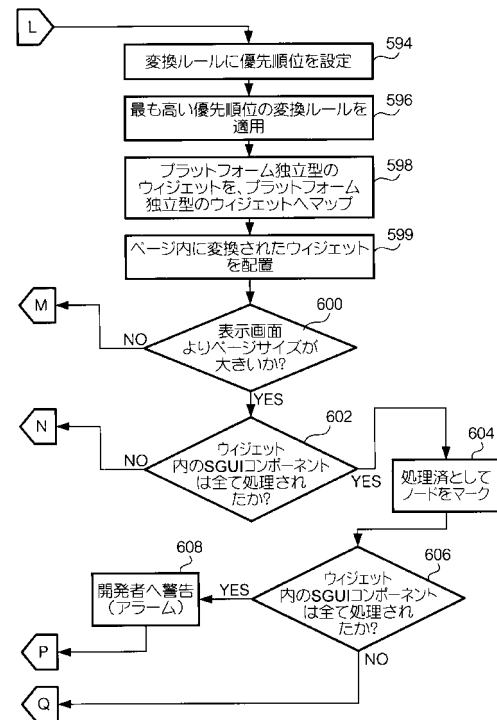
【図 18】



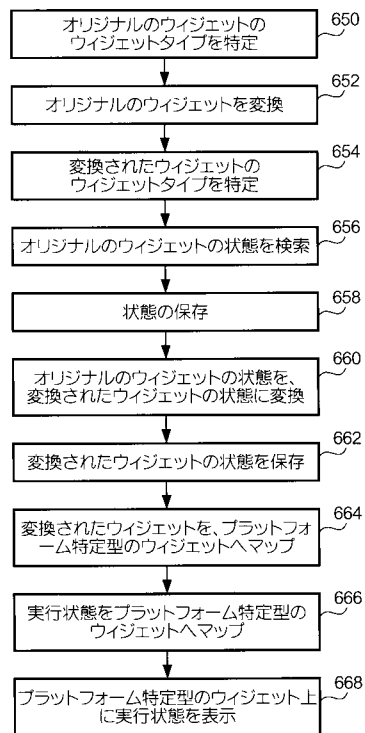
【図 19】



【図 20】



【図 21】



---

フロントページの続き

- (72)発明者 カタギリ マサジ  
アメリカ合衆国, カリフォルニア州 9 5 1 1 0 サンノゼ, スイート 3 0 0, メトロ ドライブ 1  
8 1
- (72)発明者 ソング, ユー  
アメリカ合衆国, カリフォルニア州 9 5 1 1 0 サンノゼ, スイート 3 0 0, メトロ ドライブ 1  
8 1
- (72)発明者 クラカケ ショウジ  
アメリカ合衆国, カリフォルニア州 9 5 1 1 0 サンノゼ, スイート 3 0 0, メトロ ドライブ 1  
8 1

審査官 林 毅

- (56)参考文献 特開平 0 5 - 3 2 4 2 3 2 ( J P , A )  
特開 2 0 0 0 - 0 8 2 0 3 9 ( J P , A )  
特開 2 0 0 1 - 2 9 7 0 2 9 ( J P , A )

(58)調査した分野(Int.Cl. , D B 名)

G06F 9/44

G06F 3/14

G06F 15/00