

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4982020号  
(P4982020)

(45) 発行日 平成24年7月25日(2012.7.25)

(24) 登録日 平成24年4月27日(2012.4.27)

(51) Int.Cl. F I  
**G 0 6 F 9/44 (2006.01)** G O 6 F 9/06 6 2 O B  
**G 0 6 F 3/048 (2006.01)** G O 6 F 3/048 6 5 1 E

請求項の数 43 (全 26 頁)

(21) 出願番号	特願2001-519261 (P2001-519261)	(73) 特許権者	502057393
(86) (22) 出願日	平成12年8月15日 (2000.8.15)		ナショナル・インストゥルメンツ・コーポレーション
(65) 公表番号	特表2003-531414 (P2003-531414A)		アメリカ合衆国・78759-3504・テキサス州・オースティン・ノース モパック エクスプレスウェイ・11500
(43) 公表日	平成15年10月21日 (2003.10.21)	(74) 代理人	100064621
(86) 国際出願番号	PCT/US2000/022317		弁理士 山川 政樹
(87) 国際公開番号	W02001/014963	(74) 代理人	100098394
(87) 国際公開日	平成13年3月1日 (2001.3.1)		弁理士 山川 茂樹
審査請求日	平成19年7月25日 (2007.7.25)	(72) 発明者	ダイ, ロバート・イー
(31) 優先権主張番号	60/149,950		アメリカ合衆国・78749・テキサス州・オースティン・キャノン マウンテンドライブ・6009
(32) 優先日	平成11年8月19日 (1999.8.19)		
(33) 優先権主張国	米国 (US)		
(31) 優先権主張番号	09/617,600		
(32) 優先日	平成12年6月13日 (2000.6.13)		
(33) 優先権主張国	米国 (US)		
前置審査			最終頁に続く

(54) 【発明の名称】 分散ブロック・ダイアグラム実行およびユーザ・インタフェース表示を使用するグラフィカル・プログラミング・システム

(57) 【特許請求の範囲】

【請求項1】

第1のコンピュータにおいてグラフィカル・プログラムを実行し、前記第1のコンピュータに接続した2つ又はそれ以上の第2のコンピュータにおいてグラフィカル・プログラムに対応するユーザ・インタフェースを受信し且つ表示する方法であって；

前記グラフィカル・プログラムはプログラム実行ロジック、データ・フローおよび/又は制御フローをモデル化するために一緒に接続しているノード群を有しており、且つこのノード群はノードの機能形式を示すアイコンとして表示されるものであり；

更にこの方法は、

(a) 前記第1のコンピュータにおけるグラフィカル・プログラムを特定する、第2のコンピュータのうち少なくとも1つにおけるユーザ入力を受信するステップと、

(b) 前記第1のコンピュータにおいて前記グラフィカル・プログラムを実行するステップと、

(c) 前記グラフィカル・プログラムの実行中、前記ステップ(a)にตอบสนองして、前記グラフィカル・プログラムの前記ユーザ・インタフェースに関する情報を前記第2のコンピュータのそれぞれに送るステップと、

(d) 前記ステップ(c)の終了後、プログラム実行中の前記グラフィカル・プログラムへの入力を供給し、又は前記グラフィカル・プログラムからの出力を表示するために、前記グラフィカル・プログラムの前記ユーザ・インタフェースを前記第2のコンピュータのそれぞれに表示するステップとから構成され、

更にこの方法は、  
前記グラフィカル・プログラムに関連するブロック・ダイアグラムに関する情報を前記  
第2のコンピュータのそれぞれに送るステップと、  
前記ブロック・ダイアグラムに関する情報を使用して、前記ブロック・ダイアグラムを  
前記第2のコンピュータのそれぞれに表示するステップと、  
前記第2のコンピュータのいずれか一つにおいて前記ブロック・ダイアグラムへの編集  
を指定するユーザ入力を受け取るステップと、  
前記編集を指定するユーザ入力を前記第1のコンピュータに送るステップとから構成さ  
れ、

前記第1のコンピュータは前記編集を指定するユーザ入力に従って前記グラフィカル・  
プログラムを編集することが可能であることを特徴とするグラフィカル・プログラムを実  
行する方法。

【請求項2】

前記ステップ(c)の情報は、第1のコンピュータによって与えられることを特徴とする請求項1に記載の方法。

【請求項3】

前記グラフィカル・プログラムは、計測またはオートメーション機能を実行する請求項1に記載の方法。

【請求項4】

前記第1のコンピュータと前記第2のコンピュータがネットワークで接続され、前記ステップ(c)は、前記第1のコンピュータがネットワークを介して前記第2のコンピュータに前記情報を供給することを含む請求項1に記載の方法。

【請求項5】

前記第2のコンピュータが前記ネットワークを介して前記第1のコンピュータに接続するステップを更に含み、前記ステップ(c)はステップ(a)の後で且つ前記ネットワーク接続後に実行される請求項4に記載の方法。

【請求項6】

前記接続が行われたときに前記グラフィカル・プログラムがすでに前記第1のコンピュータ上ですでに実行(前記ステップ(b))されている請求項5に記載の方法。

【請求項7】

第1のコンピュータへの前記接続への応答として、前記第1のコンピュータが前記グラフィカル・プログラムの実行を開始するステップを更に含む請求項5に記載の方法。

【請求項8】

前記ステップ(a)はURLを受け取るステップを含む請求項5に記載の方法。

【請求項9】

前記URLで第1のコンピュータまたは第1のコンピュータのグラフィカル・プログラムうち1つを指定する請求項8に記載の方法。

【請求項10】

前記ネットワークはインターネットである請求項5に記載の方法。

【請求項11】

前記ステップ(d)は前記第2のコンピュータのそれぞれのWebブラウザに前記グラフィカル・プログラムのユーザ・インタフェースを表示するステップを含む請求項5に記載の方法。

【請求項12】

前記第2のコンピュータのうち少なくとも一つに表示されているユーザ・インタフェースを介して前記グラフィカル・プログラムへのユーザ入力を受け取るステップと、

前記ユーザ入力を前記第1のコンピュータに送るステップとを更に含み、

前記第1のコンピュータ上で実行されている前記グラフィカル・プログラムはユーザ入力に応答して動作可能な請求項1に記載の方法。

【請求項13】

10

20

30

40

50

前記グラフィカル・プログラムが第1の出力状態を発生し、前記ステップ(d)は第1の出力状態を示す前記ユーザ・インタフェースを表示するステップを含む請求項1に記載の方法。

【請求項14】

第2の出力状態を示すユーザ・インタフェース更新を送るステップと、前記ユーザ・インタフェースの更新に回答して前記第2のコンピュータに表示されているユーザ・インタフェースを更新するステップとを更に含み、グラフィカル・プログラムが前記第1の出力状態を発生した後、グラフィカル・プログラムが前記第2の出力状態を発生する請求項13に記載の方法。

【請求項15】

前記ステップ(a)において、前記グラフィカル・プログラムを指定するステップは、URLを受け取るステップを含む請求項1に記載の方法。

【請求項16】

前記グラフィカル・プログラムがブロック・ダイアグラム部分とユーザ・インタフェース部分を備え、前記ステップ(b)は前記第1のコンピュータ上の前記グラフィカル・プログラムの前記ブロック・ダイアグラム部分を実行するステップを含む請求項1に記載の方法。

【請求項17】

前記グラフィカル・プログラムの前記ユーザ・インタフェースは、前記ブロック・ダイアグラム部分に入力を送るための少なくとも1つの入力変数アイコンと前記ブロック・ダイアグラム部分によって発生する出力を表示するための少なくとも1つの出力変数アイコンを備える請求項16に記載の方法。

【請求項18】

前記ユーザが前記第2のコンピュータのうち少なくとも1つにおける少なくとも1つの入力変数の入力を操作するステップと、

前記少なくとも1つの入力変数の入力を前記第1コンピュータに送るステップと、

前記少なくとも1つの入力変数の入力を使用して、前記第1コンピュータにおいて前記ブロック・ダイアグラム部分を実行するステップと、

前記ブロック・ダイアグラム部分で少なくとも1つの出力変数の出力を生成するステップと、

少なくとも1つの出力変数の出力を前記第2のコンピュータに送るステップと、

少なくとも1つの出力変数の出力を前記第2のコンピュータで表示するステップとを更に含む請求項17に記載の方法。

【請求項19】

前記グラフィカル・プログラムがデータ・フローに関するグラフィカル・プログラムを含む請求項1に記載の方法。

【請求項20】

前記グラフィカル・プログラムが制御フローに関するグラフィカル・プログラムを含む請求項1に記載の方法。

【請求項21】

前記グラフィカル・プログラムが実行フローに関するグラフィカル・プログラムを含む請求項1に記載の方法。

【請求項22】

前記グラフィカル・プログラムが仮想計測器を実装し、

前記グラフィカル・プログラムのユーザ・インタフェースが仮想計測器のフロント・パネルを含む請求項1に記載の方法。

【請求項23】

メモリに接続されたプロセッサを有し、ネットワークに接続可能な第1のコンピュータと、

前記第1のコンピュータのメモリに蓄積されたグラフィカル・プログラムと、

10

20

30

40

50

前記ネットワークに接続可能な複数の第2のコンピュータと、から構成されるグラフィカル・プログラムを実行するシステムであって：

前記グラフィカル・プログラムは、プログラム実行ロジック、データ・フローおよび/又は制御フローをモデル化するために一緒に接続しているノード群を有しており、且つこのノード群はノードの機能形式を示すアイコンとして表示されるものであり：

複数の前記第2のコンピュータのそれぞれは表示装置を具備し、前記第2のコンピュータは前記第1のコンピュータ上のグラフィカル・プログラムを特定するユーザ入力を受信するように動作し、且つ

複数の前記第2のコンピュータのそれぞれは、前記第1のコンピュータ上のグラフィカル・プログラムを特定するユーザ入力を使用して、ネットワークを介して前記第1のコンピュータに接続するよう動作するものであり；

10

前記第1のコンピュータは、前記グラフィカル・プログラムを実行する動作が可能であり、また前記グラフィカル・プログラムのユーザ・インタフェースに関する情報を、前記ネットワークを介して前記グラフィカル・プログラムの実行中に複数の前記第2のコンピュータのそれぞれへ送る動作が可能であり；

複数の前記第2のコンピュータのそれぞれは、前記ユーザ・インタフェースに関する情報を受け取り、プログラム実行中の前記グラフィカル・プログラムへの入力を供給し、又は前記グラフィカル・プログラムからの出力を表示するために、前記送るステップへの応答として前記グラフィカル・プログラムのユーザ・インタフェースを表示する動作が可能であり；

20

前記第1のコンピュータは、前記グラフィカル・プログラムに関連するブロック・ダイアグラムに関する情報を、複数の前記第2のコンピュータのそれぞれに対して送る動作が可能であり；

複数の前記第2のコンピュータのそれぞれは、前記ブロック・ダイアグラムに関する情報を使用して、前記第2のコンピュータそれぞれのディスプレイに前記ブロック・ダイアグラムを表示する動作が可能であり；

複数の前記第2のコンピュータのそれぞれは、前記ブロック・ダイアグラムへの編集を指定するユーザ入力を受け取る動作が可能であり；

複数の前記第2のコンピュータのそれぞれは、編集を指定する前記ユーザ入力を前記第1のコンピュータに送る動作が可能であり；

30

前記第1のコンピュータは、編集を指定する前記ユーザ入力に従って前記グラフィカル・プログラムを編集する動作が可能ことを特徴とする、グラフィカル・プログラムを実行するシステム。

【請求項24】

前記第1のコンピュータが第1のコンピュータに接続する前記第2のコンピュータの少なくとも1つへの応答として前記グラフィカル・プログラムの実行を開始するように動作する請求項23に記載のシステム。

【請求項25】

前記ユーザ入力はURLを含む請求項23に記載のシステム。

【請求項26】

前記URLで前記第1のコンピュータまたは第1のコンピュータのグラフィカル・プログラムうち1つまたは複数指定する請求項25に記載のシステム。

40

【請求項27】

前記ネットワークがインターネットである請求項23に記載のシステム。

【請求項28】

前記第2のコンピュータのそれぞれはWebブラウザを格納し、前記グラフィカル・プログラムのユーザ・インタフェースを表示するために、Webブラウザを前記第2のコンピュータのそれぞれで実行する請求項23に記載のシステム。

【請求項29】

前記第2のコンピュータのそれぞれは、前記第2のコンピュータ上に表示される前記ユ

50

ーザ・インタフェースを介して前記グラフィカル・プログラムへのユーザ入力を受け取る動作が可能であり、

前記第2のコンピュータのそれぞれは、前記ユーザ入力を前記第1のコンピュータに送る動作が可能であり、

前記第1のコンピュータで実行されている前記グラフィカル・プログラムが前記ユーザ入力に応答する動作が可能で請求項23に記載のシステム。

【請求項30】

前記グラフィカル・プログラムを実行して第1の出力状態を発生することができ、前記第2のコンピュータのそれぞれが、前記ユーザ・インタフェースの前記第1の出力状態を表示する動作が可能で請求項23に記載のシステム。

10

【請求項31】

前記グラフィカル・プログラムが前記第1の出力状態を発生した後に第2の出力状態を発生することができるようにグラフィカル・プログラムが実行し、

前記第1のコンピュータが前記第2の出力状態を示す前記ユーザ・インタフェースの更新を送る動作が可能であり、前記第2のコンピュータのそれぞれが、前記ユーザ・インタフェースの更新への応答として前記第2のコンピュータに表示されている前記ユーザ・インタフェースを更新する動作が可能であることを特徴とする請求項30に記載のシステム。

【請求項32】

前記グラフィカル・プログラムがダイアグラム部分とユーザ・インタフェース部分を備え、前記第1のコンピュータが前記グラフィカル・プログラムの前記ダイアグラム部分を実行する動作が可能で請求項23に記載のシステム。

20

【請求項33】

前記グラフィカル・プログラムの前記ユーザ・インタフェースが、前記ブロック・ダイアグラムに入力を送るための少なくとも1つの入力変数アイコンと前記ブロック・ダイアグラムによって発生する出力を表示するための少なくとも1つの出力変数アイコンを備える請求項23に記載のシステム。

【請求項34】

前記第2のコンピュータのそれぞれが、前記第2のコンピュータ上で少なくとも1つの入力変数の入力を操作するユーザ入力を受け取る動作が可能であり、

30

前記第1のコンピュータが、前記少なくとも1つの入力変数の入力を受け取る動作が可能であり、

前記ブロック・ダイアグラムが、前記第1のコンピュータにおいて前記少なくとも1つの入力変数の入力を使用して実行する動作が可能であり、

前記ブロック・ダイアグラムが、少なくとも1つの出力変数の出力を生成する動作が可能であり、

前記第2のコンピュータのそれぞれが、前記少なくとも1つの出力変数の出力を受け取る動作が可能であり、

前記第2のコンピュータのそれぞれが、前記少なくとも1つの出力変数の出力を表示する動作が可能で請求項33に記載のシステム。

40

【請求項35】

前記グラフィカル・プログラムはデータ・フローに関するグラフィカル・プログラムを含む請求項23に記載のシステム。

【請求項36】

前記グラフィカル・プログラムは制御フローに関するグラフィカル・プログラムを含む請求項23に記載のシステム。

【請求項37】

前記グラフィカル・プログラムは実行フローに関するグラフィカル・プログラムを含む請求項23に記載のシステム。

【請求項38】

50

前記グラフィカル・プログラムは仮想計測器を実装し、  
前記グラフィカル・プログラムのユーザ・インタフェースが前記仮想計測器のフロント・パネルを含む請求項 2 3 に記載のシステム。

【請求項 3 9】

前記グラフィカル・プログラムは、計測またはオートメーション機能を実行することが可能である請求項 2 3 に記載のシステム。

【請求項 4 0】

コンピュータにより、前記コンピュータにおけるグラフィカル・プログラムを実行し、前記グラフィカル・プログラムはプログラム実行ロジック、データ・フローおよび/又は制御フローをモデル化するために一緒に接続しているノード群を有しており且つこのノード群はノードの機能形式を示すアイコンとして表示されるグラフィカル・プログラムを実行し；

複数のクライアント・コンピュータ・システムと関連するクライアント・ソフトウェアとのネットワーク接続を確立し；

前記複数のクライアント・コンピュータ・システムそれぞれの前記クライアント・ソフトウェアとのネットワーク接続を確立した後に、前記コンピュータから前記ネットワークを介して前記グラフィカル・プログラムのユーザ・インタフェースに関する情報を前記複数のクライアント・コンピュータ・システムそれぞれのクライアント・ソフトウェアへ送り；

前記グラフィカル・プログラムに関連するブロック・ダイアグラムに関する情報を、前記複数のクライアント・コンピュータ・システムそれぞれのクライアント・ソフトウェアへ送り；

前記複数のクライアント・コンピュータ・システムそれぞれのクライアント・ソフトウェアから送られる編集を指定するユーザ入力に従って、前記グラフィカル・プログラムを編集する；

ことが可能なプログラム命令を格納したメモリ媒体であって、

プログラム実行中の前記グラフィカル・プログラムへの入力を供給し、又は前記グラフィカル・プログラムからの出力を表示するために、前記ユーザ・インターフェースが使用されることを特徴とするプログラム命令を格納したメモリ媒体。

【請求項 4 1】

前記クライアント・ソフトウェアから前記グラフィカル・プログラムへのユーザ入力を受け取り、

前記ユーザ入力を前記グラフィカル・プログラムに送るために実行可能なプログラム命令を格納し、

前記グラフィカル・プログラムがユーザ入力にตอบสนองする動作が可能な請求項 4 0 に記載のメモリ媒体。

【請求項 4 2】

前記グラフィカル・プログラムが第 1 の出力状態を発生し、

前記グラフィカル・プログラムのユーザ・インタフェースに関する情報を送る前記ステップが第 1 の出力状態を示す情報を送るステップである請求項 4 0 に記載のメモリ媒体。

【請求項 4 3】

前記グラフィカル・プログラムが第 1 の出力状態を発生した後、前記グラフィカル・プログラムが第 2 の出力状態を発生し、

前記メモリ媒体がさらに、前記第 2 の出力状態を示すユーザ・インタフェースの更新を前記クライアント・ソフトウェアに送ることを実行することが可能なプログラム命令を格納する請求項 4 2 に記載のメモリ媒体。

【発明の詳細な説明】

【0 0 0 1】

(発明の分野)

本発明は、グラフィカル・プログラミングおよび仮想計装の分野に関するものである。特

10

20

30

40

50

に、本発明は、グラフィカル・プログラム・ブロック・ダイアグラムがサーバ・コンピュータ上で実行され、1つまたは複数のクライアント・コンピュータがブロック・ダイアグラムに対応するグラフィカル・プログラム・ユーザ・インタフェース・パネルを受け取って表示し、グラフィカル・プログラム・ユーザ・インタフェース・パネルを使用してブロック・ダイアグラムに入力を送ったり、ブロック・ダイアグラムからの出力を表示したりできるシステムと方法に関するものである。本発明はさらに、分散仮想計装システムに関するものであり、ブロック・ダイアグラムがサーバ・コンピュータ上で実行され、1つまたは複数のパネルがクライアント・コンピュータ上に表示される。

【0002】

(関連技術の説明)

従来、プログラマはアプリケーション・プログラムを書くのにテキストベースの高水準プログラミング言語を使用してきた。さまざまなテキストベースの高水準プログラミング言語が存在しており、たとえば、BASIC、C、FORTRAN、Pascal、COBOL、ADA、APLなどがある。こうした高水準言語で書かれたプログラムは、コンパイラあるいはインタプリタと呼ばれるトランスレータによって機械語レベルに翻訳される。このレベルのテキストベースの高水準プログラミング言語は、アセンブリ言語レベルとともに、テキストベースのプログラミング環境と呼ばれる。

【0003】

コンピュータ・プログラミング手法の高度の訓練を受けていないプログラマがコンピュータを使用し、プログラムすることが必要になる機会が増えてきている。従来のテキストベースのプログラミング環境を使用した場合、コンピュータ・システムを対話的に操作するユーザのプログラミング・スキルと能力が、コンピュータ・システムの最適な利用を制限する要因となる。

【0004】

テキストベースの環境でコンピュータ・システムを効率よくプログラムできるようになるまでにユーザが習得しなければならない微妙で複雑な事項が多数ある。プロセスのモデル作成にコンピュータ・システムをプログラムする作業は、多くの場合、プロセスの概念モデルを作成するために使用される一連の数式、数学的ステップ、あるいはその他の手順がコンピュータ・システムをプログラムしてそのようなプロセスのモデルを作成するために使用される従来のテキストベースのプログラミング手法に密に対応していないという事実によりさらに複雑なものとなる。つまり、テキストベースのプログラミング環境のユーザ・プログラムがソリューションのユーザ側での概念化とコンピュータ・プログラムでこのソリューションを達成する方法の実装との間にあるレベルの抽象化を入れる必要があるということである。そこで、ユーザは多くの場合、システムの概念モデルを作成し、そのようなシステムをモデル化するようにコンピュータをプログラムするために異なるスキルを実質的に修得する必要がある。ユーザはテキストベースの環境でコンピュータ・システムをプログラムし、自分のモデルを実装できるほどプログラミングに長けていないことが多いため、このようなモデリングを実行するのにコンピュータ・システムを利用する効率が減じることが多々ある。

【0005】

物理的システムをモデル化しかつ/または制御するのにコンピュータ・システムを採用している分野の例としては、計装、プロセス制御、工業オートメーション、およびシミュレーションなどがある。計測器や工業オートメーションのハードウェアなどのコンピュータ・モデリングまたはデバイスの制御は、使用できる計測器およびデバイスがますます複雑になり多様になってきているという点を考えると次第に望ましいものになってきている。しかし、テスト/制御状況や環境はさまざまなものが考えられ、また利用できる計測器やデバイスもさまざまなものがあるため、望むシステムを制御するプログラムをユーザが開発しなければならないことが多い。上述のように、このようなシステムを制御するのに使用されるコンピュータ・プログラムは、従来のテキストベースのプログラミング言語、たとえばアセンブリ言語、C、FORTRAN、BASIC、またはPascalなどで書か

10

20

30

40

50

なければならなかった。しかし、これらのシステムの従来からのユーザは、多くの場合プログラミング手法の高度な訓練を受けているわけではなく、また従来のテキストベースのプログラミング言語はユーザが訓練なしで利用できるほど十分に直観的なものでもなかった。したがって、このようなシステムの実装に際して、プログラマが計装または工業オートメーション・データの制御および分析用のソフトウェアを書く必要があった。そこで、これらのシステムにおけるソフトウェア要素の開発と保守が困難である場合が多かった。

【0006】

米国特許第4901221号、第4914568号、第5291587号、第5301301号、および第5301336号とりわけKodosky et alでは、プロセスをモデル化するためのグラフィカル・システムと方法、つまりユーザがプロセスを簡単にかつ直観的にモデル化できるグラフィカル・プログラミング環境を開示している。Kodosky et alで開示されているグラフィカル・プログラミング環境は、コンピュータを対話操作する最高の最も直観的な方法と考えられる。グラフィカル・プログラミング環境は、C、Pascalなどのテキストベースの高水準プログラミング言語よりも上のレベルで表現できる。Kodosky et alで開示されている方法を使用すると、ブロック・ダイアグラム・エディタを使用してダイアグラムを構築することができ、作成されたダイアグラムは、たとえば1つまたは複数の変数を操作して1つまたは複数の出力変数を出力する、ある結果を出す手順または方法をグラフィックで表示する。ユーザがブロック・ダイアグラム・エディタを使用してデータ・フロー・ダイアグラムまたはグラフィカル・プログラムを構築するのに対応して、表示される手順に対応する実行手順を特徴付けるデータ構造を自動的に構築できる。グラフィカル・プログラムは、これらのデータ構造を使用してコンピュータによってコンパイルまたは解釈できる。したがって、ユーザはグラフィカル・プログラミング環境を使用するだけでコンピュータ・プログラムを作成できる。このグラフィカル・プログラミング環境は、仮想計装システム、工業オートメーション・システム、モデリング・プロセス、およびシミュレーションの作成、さらにはさまざまな種類の一般プログラミングに使用できる。

【0007】

したがって、Kodosky et alでは、ユーザがブロック・ダイアグラム内にアイコンを入れたり、操作したりしてグラフィカル「プログラム」を作成するのにブロック・ダイアグラム・エディタを使用するグラフィカル・プログラミング環境を教示している。計測器、プロセス、または工業オートメーション・ハードウェアなどのデバイスの制御またはモデル化用のグラフィカル・プログラムを仮想計測器(VI)と呼ぶ。仮想計測器を作成する際に、ユーザはフロント・パネルまたはユーザ・インタフェース・パネルを作成できる。フロント・パネルには、コントロールやインジケータなど、グラフィカル・プログラムつまりVIが使用するそれぞれの入出力を表す、または表示する各種のフロント・パネル・オブジェクトを含み、制御されるデバイスを表す他のアイコンも入れることができる。コントロールおよびインジケータをフロント・パネル内に作成するとき、対応するアイコンまたはターミナルをブロック・ダイアグラム・エディタによりブロック・ダイアグラム内に自動的に作成することができる。それとは別に、ユーザはターミナル・アイコンまたは入出力ブロックをブロック・ダイアグラムに配置し、編集時または実行時にフロント・パネルに対応するフロント・パネル・オブジェクトを表示させることができる。

【0008】

グラフィカル・プログラムを作成するとき、ユーザは望む結果を実行するさまざまな機能を選択し、機能アイコンを接続していっしょにする。たとえば、機能をデータ・フロー形式および/または制御フロー形式で接続できる。機能は、それぞれのコントロールとインジケータのターミナルの間で接続できる。たとえば、ユーザはブロック・ダイアグラムと呼ばれる、望む機能を実行するグラフィック・データ・フローを表すデータ・フロー・プログラムを作成またはアSEMBルすることができる。アSEMBルされたグラフィック・プログラムを、コンパイラによってコンパイルするか、インタプリタによって解釈し、ブ

10

20

30

40

50



ロック・ダイアグラムに示されているような望む方法またはプロセスを実行する機械語を出力することができる。

【0009】

ユーザは、フロント・パネル・コントロールを使用して仮想計測器にデータを入力できる。この入力データは、データ・フロー・ブロック・ダイアグラムまたはグラフィカル・プログラムに伝播し、出力インジケータ上に変化として現れる。計装アプリケーションでは、フロント・パネルを計測器のフロント・パネルに似たものにできる。工業オートメーション・アプリケーションでは、フロント・パネルをデバイスのMMI（マン・マシン・インタフェース）に似たものにできる。ユーザは、フロント・パネルのコントロールを調整して、入力を制御し、それぞれのインジケータに出力を表示できる。それとは別に、フロント・パネルを入出力の表示にだけ使用し、プログラム実行中にユーザ側で入力を対話的に操作することはできないようにすることもできる。

10

【0010】

そこで、グラフィカル・プログラミングは、プログラマにとって使用可能な強力なツールとなる。National Instruments LabVIEW製品などのグラフィカル・プログラミング環境が非常に普及してきている。LabVIEWなどのツールを使用することでプログラマの生産性が大幅に向上しており、ソフトウェア・アプリケーションの開発にグラフィカル・プログラミング環境を使用するプログラマが増えてきている。特に、グラフィカル・プログラミング・ツールは、とりわけ、テストおよび測定、データ収集、プロセス制御、マン・マシン・インタフェース（MMI）、監視制御およびデータ収集（SCADA）アプリケーション、シミュレーション、およびマシン・ビジョン・アプリケーションに使用されている。

20

【0011】

多くのシナリオにおいて、グラフィカル・プログラムのフロント・パネルと上で呼ばれているユーザ・インタフェース・パネルをグラフィカル・プログラムのブロック・ダイアグラムから切り離すことが望ましいであろう。たとえば、テストおよび測定アプリケーションやプロセス制御アプリケーションなどの計装アプリケーションを開発しているユーザは、実験室や製造設備に置かれているコンピュータ上でグラフィカル・プログラムを実行したいが、ユーザのオフィスに置かれているワークステーションなどの他のコンピュータからプログラムのユーザ・インタフェース・パネルを表示してプログラムを対話操作したい場合がある。他に、プログラム開発者がグラフィカル・プログラムを構築し、他のユーザからプログラムの結果を対話的に操作したり表示したりできるようにする例もある。たとえば、プログラム開発者は、複数のインターネット・ユーザがグラフィカル・プログラムを実行しているコンピュータに接続し、グラフィカル・プログラムのユーザ・インタフェースを表示できるようにしたいと思うであろう。

30

【0012】

こうして、さまざまな種類のユーザ・インタフェース・パネルを備えるさまざまな種類のグラフィカル・プログラムで上述のようなユーザ・インタフェース・パネルをエクスポートでき、しかもプログラミング作業量が最小限度に抑えられる一般システムおよび方法を実現することが望ましい。また、共通のネットワークおよびソフトウェア標準を使用する上記の機能を備え、各種のコンピューティング・プラットフォーム上で作業しているユーザがグラフィカル・プログラムを実行しているリモート・コンピュータに接続し、グラフィカル・プログラムのユーザ・インタフェース・パネルを表示し、場合によっては、さらに、そのユーザ・インタフェース・パネルを使用してグラフィカル・プログラムを遠隔で使用または制御できるようにすることが望まれる。さらに、こうした機能を利用するのに必要なクライアント・ソフトウェアのインストール作業を最小限度に抑え、かつ/または必要クライアント・ソフトウェアを自動的にダウンロードしインストールできるようにするのが望ましい。

40

【0013】

（発明の概要）

50

本発明の一実施態様は、サーバ・コンピュータ上で実行されるグラフィカル・プログラムのユーザ・インタフェースの分散表示を可能にするシステムと方法を含む。一実施態様では、このシステムは、グラフィカル・プログラムが実行されるサーバ・コンピュータとそのサーバ・コンピュータに接続されている1つまたは複数のクライアント・コンピュータを備える。クライアント・コンピュータは、グラフィカル・プログラムに対応して、ユーザ・インタフェースたとえば1つまたは複数のユーザ・インタフェース・パネルを受け取って表示する。一実施態様では、クライアント・コンピュータからユーザ・インタフェースを使用して、プログラム実行時にグラフィカル・プログラムに入力を与え、かつそのプログラムからの出力を表示することができる。本発明の特定の一実施態様では、分散仮想計装システムを備え、グラフィカル・プログラムがサーバ・コンピュータ上で実行されて測定またはオートメーション機能を実行し、1つまたは複数のフロント・パネルがクライアント・コンピュータに表示され、これにより、一人または複数のユーザが測定またはオートメーション機能を遠隔で表示かつ/または制御できる。

10

**【0014】**

一実施態様では、クライアント・コンピュータのユーザはグラフィカル・プログラムが実行されるリモート・サーバ・コンピュータを指定する。リモート・サーバ情報はさまざまな方法で指定できる。たとえば、情報を、URL (Uniform Resource Locator)、インターネット・プロトコル (IP) アドレス、マシン名とTCP/IPポート番号などで指定できる。一実施態様では、ユーザはリモート・コンピュータを指定するのに、URLをWebブラウザやWebブラウザ機能を備えるその他のアプリケーションなどのアプリケーションに入力する。以下で述べるように、アプリケーションはURLを処理し、リモート・コンピュータに接続するため使用可能になっているプロトコル・ハンドラ・プラグインを備えることができる。

20

**【0015】**

ユーザは、グラフィカル・プログラムを実行しているリモート・コンピュータを指定するときに、望ましい特定のグラフィカル・プログラムも指定できる。たとえば、グラフィカル・プログラムの名前を示すパラメータをURLなどに付加することができる。ユーザはさらに、特定のグラフィカル・プログラムを指定することなくリモート・コンピュータを指定することもできる。たとえば、リモート・コンピュータはWebサーバを備えているとする。ユーザがそのWebサーバと関連するWebページのURLを入力すると、Webサーバは、リモート・コンピュータを実行しているグラフィカル・プログラムのリストを返す。そこでユーザはリストから1つまたは複数のグラフィカル・プログラムを選択できる。ユーザのクライアント・ソフトウェアは、ユーザのディスプレイ画面に表示されている選択されたグラフィカル・プログラムと関連するユーザ・インタフェース・パネルを表示する動作が可能である。

30

**【0016】**

一実施態様では、ユーザのクライアント・ソフトウェアは遠隔のグラフィカル・プログラムと通信するため動作可能なプラグインを内蔵するWebブラウザ(またはWebブラウザ機能を持つアプリケーション)を備える。この実施態様では、プラグインはWebブラウザのウィンドウに直接、ユーザ・インタフェース・パネルを表示できる。ユーザのクライアント・ソフトウェアは、標準のTCP/IPプロトコルに基づく通信プロトコルを使用するリモート・コンピュータ上で実行されるエージェントまたはソフトウェア・プログラムと通信するのが好ましい。ユーザが接続にリモート・コンピュータを指定した場合、リモート・コンピュータのエージェントはグラフィカル・プログラムユーザ・インタフェース・パネルの記述をユーザのクライアント・ソフトウェアに転送する。この記述は、リモート・コンピュータのユーザ・インタフェース・パネル情報を格納するのに使用しているのと同じ形式で送信できる。ユーザ・インタフェース・パネルの記述は、もちろん、他のさまざまな形式、たとえばXML記述で送信することができる。ユーザのクライアント側ソフトウェア、たとえば、Webブラウザ・プラグインは、リモート・コンピュータから受け取る可能性のあるどのようなユーザ・インタフェース・パネル記述であっても解釈

40

50

できるのが好ましく、ユーザ・インタフェース・パネルをユーザに適宜表示するよう使用可能である。

【0017】

グラフィカル・プログラムのユーザ・インタフェース・パネルを受け取ってユーザのディスプレイ画面に表示した後、グラフィカル・プログラムのブロック・ダイアグラムの実行中、ユーザ・インタフェース・パネルを動的に更新できる。たとえば、ユーザ・インタフェース・パネルには、電気信号、気象データなど、ブロック・ダイアグラムで生成されたさまざまな種類の測定データを表示するグラフを入れることができ、このグラフはグラフィカル・プログラムの実行に応じて測定データ値が変化するとユーザのディスプレイ上でスクロールする。他の例としては、新しい値で定期的に更新する数値テキスト・インジケータを備えるユーザ・インタフェース・パネルなどがある。

10

【0018】

ユーザはさらに、クライアント・コンピュータ上のユーザ・インタフェース・パネルを対話的に操作し、たとえば、ポイントしてクリックする標準タイプのGUIコマンドを発行することにより、入力をサーバ・コンピュータ上で実行中のブロック・ダイアグラムに送ることもできる。ユーザの入力は、サーバ・コンピュータ上の遠隔のグラフィカル・プログラムに渡され、グラフィカル・プログラムがそれに応じて応答する。つまり、ユーザは、ユーザのローカル・コンピュータ上で実行されている場合にプログラムを対話的に操作するのとまったく同じように遠隔のグラフィカル・プログラムを対話的に操作できる。ユーザ間の制御を調整する手段を含めることで、複数のユーザが同じグラフィカル・プログラムを対話的に操作しながら、互いのアクションに干渉しないようにできる。

20

【0019】

以下に述べるように、一実施態様では、ユーザはさらに、遠隔のグラフィカル・プログラムのブロック・ダイアグラムを要求し、受け取って、たとえば、グラフィカル・プログラムを編集やデバッグをすることができる。

【0020】

上で指摘したように、好ましい実施態様では、グラフィカル・プログラムが実行されているリモート・サーバ・コンピュータとユーザのクライアント・ソフトウェアとの間の通信にTCP/IPベースの通信プロトコルを使用する。他の実施態様では、米国特許第09/185161号で開示されているデータソケットのシステムと方法を使用して、ユーザのクライアント・ソフトウェアとグラフィカル・プログラムが実行されているリモート・コンピュータとの間の通信を簡単に行えるようにできる。データソケット・システムは、URLを使用して世界中のあらゆる場所のWebページのアドレス指定を行うのと同じような方法でURLを使用してデータ・ソース/ターゲットのアドレスを指定するクライアント・ソフトウェア・コンポーネントを備える。

30

【0021】

一実施態様では、遠隔のグラフィカル・プログラムは「VIサーバ」と呼ばれる機能を備えるグラフィカル・プログラミング環境内で実行される。VIサーバの機能を使って、ユーザ・クライアントを遠隔のグラフィカル・プログラムに接続し、それと対話的に操作するようにできる。VIサーバの詳細については、以下に参照で取り込まれている特許出願を参照されたい。

40

【0022】

本発明には、さまざまな修正および他の態様がありえるが、その特定の実施態様は図面の内の例を用いて示されているため、ここで詳述する。ただし、図面および詳細な説明は発明を開示されている特定の態様に制限する意図のないことを理解していただきたい。それどころか、本発明は付属の請求項で定義されているように本発明の精神と範囲内にあるすべての修正物、等価物、および代替物を適用範囲とする。

【0023】

本発明は、次の図面とともに好ましい実施形態の次の詳細な説明を考察するとより容易に理解できるであろう。

50

## 【 0 0 2 4 】

( 好ましい実施形態の詳細な説明 )

図 1 - ネットワークに接続されたコンピュータ・システム

図 1 は、コンピュータ・システム 8 2 がネットワーク 8 4 を介して第 2 のコンピュータ・システム 8 6 に接続されているコンピュータ・ネットワークの例の図である。コンピュータ・システム 8 2 と第 2 のコンピュータ・システム 8 6 は、望むとおりどのような種類のものでよい。ネットワーク 8 4 はまた、LAN ( ローカル・エリア・ネットワーク )、WAN ( ワイド・エリア・ネットワーク )、さらにはインターネットとすることができる。

## 【 0 0 2 5 】

コンピュータ・システム 8 2 のユーザは、ここで述べているシステムと方法に従って、コンピュータ・システム 8 6 に接続できる。コンピュータ・システム 8 2 は、クライアント・コンピュータ・システム 8 2 とも呼ばれ、グラフィカル・プログラムのユーザ・インタフェース・パネルの記述を受け取り、コンピュータ・システム 8 2 のディスプレイ画面にそのパネルを表示できるクライアント・ソフトウェアを備える。たとえば、クライアント・ソフトウェアは、Web ブラウザ・プラグインを内蔵する Web ブラウザとすることができる。Web ブラウザとしては、たとえば、Microsoft Internet Explorer があり、プラグインを Microsoft の Asynchronous Pluggable Protocols 仕様に従って構築できる。

## 【 0 0 2 6 】

コンピュータ・システム 8 6 は、サーバ・コンピュータ・システム 8 6 と呼ぶこともできるが、グラフィカル・プログラムとともにサーバ側プログラムまたはエージェントを備え、コンピュータ・システム 8 2 のユーザは本発明に従ってコンピュータ・システム 8 6 と通信することができる。たとえば、コンピュータ・システム 8 6 は、上述のように V I サーバ機能を備えることができる。

## 【 0 0 2 7 】

上述のようにコンピュータ・システム 8 6 に接続されているクライアントが 1 つだけ示されているが、複数のクライアントをコンピュータ 8 6 に接続して、グラフィカル・プログラムのユーザ・インタフェース・パネルを表示したりグラフィカル・プログラムを対話的に操作することができる。コンピュータ・システム 8 6 は、複数の遠隔地のユーザ間でグラフィカル・プログラムの制御を調整するメカニズムを備えるのが好ましい。たとえば、コンピュータ・システム 8 6 では、ラウンドロビン方式、優先順位付きラウンドロビン方式などのさまざまな方法またはアルゴリズムを使用して、グラフィカル・プログラムの制御を複数のユーザに分散させることができる。さまざまな種類の特権または許可を、異なるユーザに割り当て、グラフィカル・プログラムに対する異なるレベルで制御することもできる。たとえば、プログラム作成者が他のユーザをロックアウトし、プログラムの完全な制御権を奪うことを許可できる。他のユーザに対しては、グラフィカル・プログラムのユーザ・インタフェース・パネルの表示のみを許可し、グラフィカル・プログラムの制御に使用することは許可せず、たとえば、これらのユーザに対しグラフィカル・プログラムへの入力を行うことだけを許可できる。

## 【 0 0 2 8 】

図 2 A および 2 B - 計装および工業オートメーション・システム

図 2 A および 2 B は、本発明に従ってプログラムを格納または使用できるシステムの例を示している。これらのシステム例は、計装、プロセス制御、またはその他の目的専用のシステムを説明している。図 2 A および 2 B は、サーバ・コンピュータ・システムの例である。そこで、上述のサーバ・コンピュータ 8 6 は、計装または工業オートメーション・システム内に装備することができ、本発明では、テストまたはオートメーション・アプリケーションの分散制御が可能である。本発明は、もちろん、望ましいその他の種類のアプリケーションでも使用できる。

## 【 0 0 2 9 】

図 2 A は、計装制御システム 100 を示している。システム 100 はホスト・コンピュータ 86 (サーバ・コンピュータ 86) を備える。これは 1 つまたは複数の計測器に接続されている。ホスト・コンピュータ 86 は、図に示されているように CPU、ディスプレイ画面、メモリ、およびマウスまたはキーボードなど 1 つまたは複数の入力デバイスを備える。コンピュータ 86 は、1 つまたは複数の計測器に接続され、テスト対象のユニット (UUT) またはプロセス 150 を分析し、測定し、または制御する。

【 0030 】

1 つまたは複数の計測器が、GPIB 計測器 112 および関連する GPIB インタフェース・カード 122、データ収集ボード 114 および関連する信号調整回路 124、VXI 計測器 116、PXI 計測器 118、ビデオデバイス 132 および関連する画像収集カード 134、モーション制御デバイス 136 および関連するモーション制御インタフェース・カード 138、および/または、1 つまたは複数のコンピュータベースの計測器カード 142 を備えることができる。

10

【 0031 】

GPIB 計測器 112 は、コンピュータ 86 に装備されている GPIB インタフェース・カード 122 を介してコンピュータ 86 に結合されている。同様の方法で、ビデオ・デバイス 132 は、画像収集カード 134 を介してコンピュータ 86 に結合され、モーション制御デバイス 136 は、モーション制御インタフェース・カード 138 を介してコンピュータ 86 に結合されている。データ収集ボード 114 は、コンピュータ 86 に結合され、信号調整回路 124 を通じて UUT にインタフェースすることができる。信号調整回路 124 は、1 つまたは複数の SCXI モジュール 126 を備える SCXI (計装用信号調整拡張) シャシを備えるのが好ましい。

20

【 0032 】

GPIB カード 122、画像収集カード 134、モーション制御インタフェース・カード 138、および DAQ カード 114 は、通常、コンピュータ 86 内の I/O スロット、たとえばコンピュータ 86 に用意されている PCI バス・スロット、PC カード・スロット、または ISA、EISA、または Micro Channel バス・スロットなどに差し込まれる。ただし、これらのカード 122、134、138、および 114 は、説明のため、コンピュータ 86 の外部に示されている。

【 0033 】

VXI シャシまたは計測器 116 は、コンピュータ 86 に用意されている VXI バス、MXI バス、またはその他のシリアルまたはパラレル・バスを介してコンピュータ 86 に結合されている。コンピュータ 86 は、VXI シャシ 116 にインタフェースする VXI、MXI、または GPIB インタフェース・カード (図には示されていない) などの VXI インタフェース・ロジックを備えるのが好ましい。PXI シャシまたは計測器は、コンピュータの PCI バスを介してコンピュータ 86 に結合するのが好ましい。

30

【 0034 】

シリアル計測器 (図には示されていない) も、コンピュータ 86 に装備されている RS-232 ポート、USB (ユニバーサル・シリアル・バス)、IEEE 1394 または 1394.2 バスなどのシリアル・ポートを介してコンピュータ 86 に結合できる。通常の計装制御システムでは、それぞれのインタフェース・タイプの計測器が存在せず、実際、多くのシステムで GPIB 計測器のみなど、単一のインタフェース・タイプの計測器を 1 つまたは複数備えるだけである。

40

【 0035 】

計測器は、テスト対象ユニット (UUT) またはプロセス 150 に結合されるか、またはトランスデューサによって通常生成されるフィールド信号を受信するように結合される。システム 100 は、データ収集および制御アプリケーション、テストおよび測定アプリケーション、プロセス制御アプリケーション、またはマン・マシン・インタフェース・アプリケーションで使用できる。

【 0036 】

50

図2Bは工業オートメーション・システム160の例の図である。工業オートメーション・システム160は、図2Aに示されている計装またはテストおよび測定システム100に似ている。図2Aの要素に似ている、または同一の要素は、便宜上同じ参照番号が振られている。システム160は、1つまたは複数のデバイスまたは計測器に接続されるコンピュータ86を備える。コンピュータ86は、図に示されているようにCPU、ディスプレイ画面、メモリ、およびマウスまたはキーボードなど1つまたは複数の入力デバイスを備える。コンピュータ86は、1つまたは複数のデバイスを通じて、プロセスまたはデバイス150に接続され、MMI（マン・マシン・インタフェース）、SCADA（監視制御およびデータ収集）、ポータブルまたは分散データ収集、プロセス制御、詳細分析、またはその他の制御などのオートメーション機能を実行する。

10

**【0037】**

1つまたは複数のデバイスが、データ収集ボード114および関連する信号調整回路124、PXI計測器118、ビデオデバイス132および関連する画像収集カード134、モーション制御デバイス136および関連するモーション制御インタフェース・カード138、フィールドバス・デバイス170および関連するフィールドバス・インタフェース・カード172、PLC（プログラマブル・ロジック・コントローラ）176、シリアル計測器182および関連するシリアル・インタフェース・カード184、または分散データ収集システム、たとえば、とりわけNational Instruments社製のFieldpointシステムなどを備えることができる。

**【0038】**

DAQカード114、PXIシャシ118、ビデオ・デバイス132、および画像収集カード136を上述のようにコンピュータ86に接続するのが好ましい。シリアル計測器182は、コンピュータ86に装備されているシリアル・インタフェース・カード184または、RS-232ポートなどのシリアル・ポートを介してコンピュータ86に結合されている。PLC 176は、シリアル・ポート、Ethernet（登録商標）ポート、または専用インタフェースを介してコンピュータ86に結合される。フィールドバス・インタフェース・カード172をコンピュータ86内に装備し、フィールドバス・ネットワークを介して1つまたは複数のフィールドバス・デバイスにインタフェースするのが好ましい。通常、DAQカード114、シリアル・カード184、フィールドバス・カード172、画像収集カード134、およびモーション制御カード138はそれぞれ、上述のように、コンピュータ86内のI/Oスロットに差し込まれる。ただし、これらのカード114、184、172、134、および138は、説明のため、コンピュータ86の外部に示されている。通常の工業オートメーション・システムでは、それぞれのインタフェース・タイプのデバイスが存在せず、実際、多くのシステムはPLCのみなど、単一のインタフェース・タイプのデバイスを1つまたは複数備えるだけである。これらのデバイスはデバイスまたはプロセス150に結合されている。

20

30

**【0039】**

再び図2Aおよび2Bを参照する。サーバ・コンピュータ・システム86は、本発明による1つまたは複数のコンピュータ・プログラムまたはソフトウェア・コンポーネントが格納されるメモリ媒体を備えるのが好ましい。「メモリ媒体」という用語は、インストール媒体、たとえば、CD-ROM、フロッピー（登録商標）・ディスク104、またはテープ・デバイス、コンピュータ・システム・メモリ、またはDRAM、SRAM、EDORAM、Rambus RAMなど、あるいは磁気媒体、たとえばハードディスク・ドライブや光記憶装置などの不揮発性メモリを含むことを意図している。メモリ媒体は、他のタイプのメモリ、あるいはその組み合わせとしてもよい。

40

**【0040】**

さらに、メモリ媒体は、プログラムが実行される第1のコンピュータに配置しても、またインターネットなどのネットワークを介して第1のコンピュータに接続している異なる第2のコンピュータに配置してもよい。後者の場合、第2のコンピュータは実行のためプログラム命令を第1のコンピュータに送る。サーバ・コンピュータ・システム86は、さま

50

ざまな形態をとりうる。同様に、クライアント・コンピュータ・システム 8 2 は、パーソナル・コンピュータ・システム、ワークステーション、ネットワーク・アプライアンス、インターネット・アプライアンス、パーソナル・デジタル・アシスタント ( P D A )、テレビジョン・システムまたはその他のデバイスをはじめとする、さまざまな形態をとりうる。一般に、「コンピュータ・システム」という用語は、メモリ媒体にある命令を実行する少なくとも 1 つのプロセッサを備えるデバイスを包含するように広い意味で定義することができる。

【 0 0 4 1 】

一実施形態では、サーバ・コンピュータ 8 6 のメモリ媒体は、本発明によりクライアント・コンピュータ・システム 8 2 と通信するためのソフトウェア・プログラムを格納する。たとえば、サーバ・コンピュータ 8 6 は、ネットワーク通信ソフトウェア、たとえば、TCP / IP ソフトウェアを格納し、またリモート・コンピュータとの通信を可能にするグラフィカル・プログラミング・システムなどのアプリケーション・レベルのソフトウェアを格納することもできる。

10

【 0 0 4 2 】

一実施形態では、クライアント・コンピュータ 8 2 のメモリ媒体は、本発明によりサーバ・コンピュータ・システム 8 6 と通信するためのソフトウェア・プログラムを格納する。たとえば、クライアント・コンピュータ 8 2 は、Web ブラウザまたは Web ブラウズ機能を備えるその他のアプリケーションなどの標準ユーザ・エージェントと、場合によっては、サーバ・コンピュータとの通信用の専用ブラウザ・プラグインを格納できる。

20

【 0 0 4 3 】

一実施形態では、ユーザが遠隔地で表示または制御できるグラフィカル・プログラムは、データ収集 / 生成、分析および / または表示用のプログラム、あるいは計装または工業オートメーション・ハードウェアの制御またはモデル化用のプログラムである。たとえば、好ましい実施形態では、グラフィカル・プログラムは、計装および工業オートメーション・アプリケーションの開発者専用のサポートが行われている National Instruments LabVIEW というグラフィカル・プログラミング環境アプリケーションを使用して作成されるプログラムである。

【 0 0 4 4 】

ただし、本発明は、多数のアプリケーションで使用することができ、計装または工業オートメーション・アプリケーションに限定されないことに注意されたい。つまり、図 2 A および 2 B は例にすぎず、ユーザはさまざまなアプリケーションのどれについてもいろいろな種類の目的のためにグラフィカル・プログラムを遠隔で対話的に操作することができる。

30

【 0 0 4 5 】

図 3 - コンピュータ・システムのブロック図

図 3 は、図 1、2 A、および 2 B に示されているコンピュータ・システムのブロック図である。いかなる種類のコンピュータ・システム構成またはアーキテクチャも、望む通りに使用でき、図 3 は代表的な PC 実施形態を説明していることに注意されたい。また、コンピュータ・システムは、図 2 A および 2 B に示されているような汎用コンピュータ、V X I シャシに装着された V X I カードに実装されたコンピュータ、P X I シャシに装着された P X I カードに実装されたコンピュータ、またはその他の種類の実施形態である。本発明を理解するうえで必要のないコンピュータの要素は、簡略化のため省いてある。

40

【 0 0 4 6 】

コンピュータ 8 6 ( または 8 2 ) は、プロセッサまたはホスト・バス 1 6 2 に結合された少なくとも 1 つの中央処理装置つまり CPU 1 6 0 を備える。CPU 1 6 0 は、x 8 6 プロセッサ、たとえば Pentium ( 登録商標 ) クラス、Power PC プロセッサ、SPARC RISC プロセッサ・ファミリの CPU などさまざまな種類のいずれでもかまわない。メイン・メモリ 1 6 6 は、メモリ・コントローラ 1 6 4 を使用してホスト・バス 1 6 2 に結合されている。

50

## 【0047】

メイン・メモリ166は本発明によるコンピュータ・プログラムを格納する。メイン・メモリ166は、さらに、当業者にはよく知られているように、オペレーティング・システム・ソフトウェアとともにコンピュータ・システムを動作させるためのソフトウェアも格納する。本発明のコンピュータ・プログラムについて、以下の段落で詳しく説明する。

## 【0048】

ホスト・バス162は、バス・コントローラ168またはバス・ブリッジ・ロジックにより拡張バスまたは入出力バス170に結合されている。拡張バス170は、PCI (Peripheral Component Interconnect) 拡張バスであるのが好ましいが、他の種類のバスも使用できる。拡張バス170は、データ収集ボード114 (図2Aの)、GP I B計測器112 (図2Aの)へのGP I Bバス・インタフェースであるGP I Bインタフェース・カード122、およびV X I計測器を受け入れるためのV X Iシャシ116に結合されているV X IまたはM X Iバス・カード186などのさまざまなデバイス用のスロットを備える。コンピュータ86ではさらに、ビデオ表示サブシステム180とハードディスク・ドライブ182が拡張バス170に結合されている。

## 【0049】

図4～6：ユーザによるグラフィカル・プログラムの対話的作成

図4は、グラフィカル・プログラムを対話的にまたは手で作成または編集する方法の一実施形態を説明する流れ図である。流れ図に示され、以下で説明するように、ユーザはグラフィカル・プログラムに対話的にさまざまなオブジェクトを追加し、接続してつなぎ合わせる。図4のさまざまなステップはいろいろな順序で実行したり、必要に応じて省くこともできることに注意されたい。

## 【0050】

図4に示されている実施形態のステップは、グラフィカル・プログラミング環境でグラフィカル・プログラムを作成または編集する開発者によって実行される。図に示されているように、ステップ420で、開発者はグラフィカル・ユーザ・インタフェースを表示するユーザ・インタフェース・パネルを作成または編集できる。ユーザ・インタフェース・パネルは、ユーザ入力を受け入れたり、プログラムの出力などの情報を表示したり、あるいはその両方をコントロールすることができる。たとえば、ユーザ・インタフェース・パネルは、ボタン、選択可能リスト、テキスト・ボックス、グラフ・コントロール、画像などを備えることができる。開発者は、たとえばコントロール・パレットから目的のコントロールを選択して、ユーザ・インタフェース・パネルにさまざまなコントロールまたはその他のオブジェクトを「ドロップ」する。図5は、単純なユーザ・インタフェース・パネルを示している。ステップ420は必ずしも実行されない。たとえば、ユーザ・インタフェース・パネルが望まれない場合があったり、ユーザ・インタフェース・パネルがブロック・ダイアグラムの作成時に本質的に指定する場合があったり、ユーザがグラフィカル・プログラムの実行可能部分を作成するときにユーザ・インタフェース・パネルが自動的に作成されたりする場合がある。

## 【0051】

ステップ422で、開発者は、ブロック・ダイアグラムと呼ばれるグラフィカル・プログラムの実行可能部分を作成または編集する。グラフィカル・プログラムは、ここで「ノード」と呼ぶオブジェクトを含むブロック・ダイアグラムを備えることができ、これらのオブジェクト同士を接続して、プログラム実行ロジック、データ・フロー、および/または制御フローをモデル化する。ブロック・ダイアグラムは、ノードのタイプまたは機能を表すアイコンとして表示することができる。図6は、単純なブロック・ダイアグラムを示している。開発者がユーザ・インタフェース・パネルにオブジェクトを追加すると、グラフィカル・プログラミング環境により対応するオブジェクトがブロック・ダイアグラム上に自動的に作成される。ユーザ・インタフェース・パネル・オブジェクトに対応するこのようなブロック・ダイアグラム・ノードは、ここでは、ユーザ・インタフェース・ノードまたはターミナルと呼ぶ。たとえば、「The result of 2.0 + 3.0 w



as : ( 2 . 0 + 3 . 0 の結果 : ) 」というラベルの付いている図 6 のブロック・ダイアグラム・ノードは図 5 のユーザ・インタフェース出力インジケータに対応するユーザ・インタフェース・ノードである。ユーザ・インタフェース・ノードをブロック・ダイアグラム内の他のオブジェクトまたはノードと接続して、プログラム・ロジックおよびデータ/制御フローに加えることができる。ユーザ・インタフェース・パネルでは、入出力をユーザ・インタフェース・パネルとブロック・ダイアグラムとの間にマッピングできる。たとえば、図 6 のユーザ・インタフェース・ノードでは、データを受け取り、そのデータを図 5 内の対応するユーザ・インタフェース・インジケータに表示する。

#### 【 0 0 5 2 】

図 4 のステップ 4 2 2 で、開発者は他のオブジェクト/ノードをグラフィカル・プログラムに追加したり、グラフィカル・プログラムの他のオブジェクト/ノードを編集する。これらのオブジェクトまたはノードは、数値関数、ブール関数、文字列関数、配列関数、エラー関数、ファイル関数、アプリケーション制御関数などの所定の関数演算を実行する関数ノードを含む。たとえば、図 6 に示されているブロック・ダイアグラムでは、加算関数を使用して、2 つの定数を足し合わせている。ステップ 4 2 2 で、開発者はさらに、他の種類のノードをグラフィカル・プログラムに追加できる。たとえば、数値定数を表すノードを追加できる。図 6 は、浮動小数点定数 2 . 0 と 3 . 0 を表す数値定数ノードを示している。

10

#### 【 0 0 5 3 】

追加できるノードとしては他に、グラフィカル・サブプログラムを呼び出すサブプログラム・ノード、変数を定義し使用するグローバルまたはローカル変数ノードなどがある。ステップ 4 2 2 で、開発者はさらに、他の種類のオブジェクトをグラフィカル・プログラムに追加できる。たとえば、ループ、While ループ、Case 構文などのプログラム構造を表すオブジェクトも追加できる。開発者は、グラフィカル・プログラムにノードや他の種類のオブジェクトを追加でき、その際にさまざまな方法、たとえば、さまざまなノードおよびオブジェクトを表すアイコンを表示しているパレットからノードまたはオブジェクトを選択する方法を採用できる。

20

#### 【 0 0 5 4 】

図 4 のステップ 4 2 2 では、開発者はさらに、目的の実行可能ロジック、データ・フロー、および/または制御フローを得るためにグラフィカル・プログラムのオブジェクトを接続、つまり「配線」することもできる。たとえば、オブジェクトには、入出力ターミナルを入れることができ、開発者は、一方のノードの出力ターミナルを他方のノードの入力ターミナルに接続するといったことができる。図 6 は、オブジェクト同士を接続する方法の一実施形態を示している。この例では、2 つの数値定数ノードの出力ターミナルを加算関数ノードの入力ターミナルに接続している。加算関数ノードは、数値入力に対し加算を実行する。加算関数ノードの出力ターミナルは、ユーザ・インタフェース・インジケータ・ノードの入力に接続されており、加算の結果が図 5 に示されているユーザ・インタフェース・パネルに表示される。

30

#### 【 0 0 5 5 】

プログラム構造オブジェクトはさらに、グラフィカル・プログラムの他のオブジェクトと統合するターミナルを備えることができる。たとえば、While ループは、ループがいつ終了するかを示すブール値を与えるノードの出力ターミナルを接続できる条件ターミナルを備えることができる。

40

#### 【 0 0 5 6 】

グラフィカル・プログラムを作成または編集する一実施形態に関する詳細は、引用により本発明に取り込まれている、さまざまな LabVIEW ユーザマニュアルおよび開発者マニュアルおよび、National Instruments Corporation で販売している LabVIEW バージョン 5 . 1 を参照していただきたい。

#### 【 0 0 5 7 】

図 4 のステップ 4 2 6 で開発者は、グラフィカル・プログラムを保存または実行する。グ

50

グラフィカル・プログラムはいろいろな形式で保存できる。たとえば、グラフィカル・プログラムのさまざまな要素とそれらの要素間の関係を表すデータ構造のツリーを構築し、そのデータ構造をバイナリ形式またはテキスト形式で保存できる。これらのデータ構造は、機械語コードにコンパイルするか、または解釈実行される。グラフィカル・プログラムにユーザ・インタフェース・パネルが含まれる場合、これらのパネルも保存できる。ステップ426で開発者は、グラフィカル・プログラムを実行することもできる。開発者は、グラフィカル・プログラムをいろいろな方法で実行できる。たとえば、グラフィカル・プログラミング環境で、開発環境内からプログラムを実行することができ、また開発者はスタンドアローンのプログラムを作成して、そのプログラムを実行するなどできる。

【0058】

ステップ420～426は通常、反復実行され、また実行順序はさまざまであるのがふつうであることに注意されたい。たとえば、開発者はユーザ・インタフェース・コントロールをユーザ・インタフェース・パネルに追加し、そのコントロールに対応するユーザ・インタフェース・ノードを他のノードに接続し、関数ノードをプログラムに追加して接続し、プログラムを実行してそれをテストし、ノードの接続の仕方を変更するといったことができる。また、上で指摘したように、ステップ420をステップ422への応答として自動的に（たとえば、プログラムで）実行するようにできる。さらに、ユーザ・インタフェース・パネルは、編集時に自動的に作成するようにしたり、あるいは実行時に自動的に生成するようにもできる。従って、図4の流れ図は例であり、さまざまなステップを組み合わせる、省く、追加する、修正するという操作を必要に応じてあるいは望ましい場合に実行し、異なるグラフィカル・プログラムを開発したり、グラフィカル・プログラム開発環境の異なる実施形態を使用することができる。

【0059】

図7 - 遠隔のグラフィカル・プログラムへのアクセス

図7は、遠隔のグラフィカル・プログラムにユーザがアクセスする一実施形態を説明する流れ図である。他の実施形態では、図7のさまざまなステップを組み合わせる、変更する、省くという操作を行ったり、または異なる順序で実行することができる。

【0060】

図に示されているように、図7のステップ450で、ユーザはリモート・コンピュータを指定する。ステップ452で、ユーザはリモート・コンピュータのグラフィカル・プログラムを指定する。ステップ450および452を1つのステップに組み合わせることができる。上述のように、ステップ450および452はいろいろな方法で実行できる。たとえば、ユーザがリモート・コンピュータまたは遠隔のグラフィカル・プログラムを指し示すURLを指定することにより、リモート・コンピュータおよび/または遠隔のグラフィカル・プログラムを暗黙のうち指定することができる。ステップ450および452は必ずしもユーザによって直接実行されるわけではなく、プログラムで実行するようにもできることに注意されたい。たとえば、ユーザは、以下で説明するリモート・コンピュータおよび遠隔のグラフィカル・プログラムへの参照をユーザのマシン上で実行されるクライアント・ソフトウェアに送るアプリケーションを操作できる。

【0061】

好ましい実施形態では、ユーザはステップ450と452を実行する。その際に、WebブラウザやWebブラウズ機能を備えるアプリケーション、たとえば、Microsoft Internet Explorerコードベースを使用するアプリケーションなどの標準の一般に使用可能なクライアント・ソフトウェアを対話的に操作する。たとえば、ユーザはURLをブラウザ・アプリケーションに送り、ブラウザ・アプリケーションはWebサーバに接続し、そのWebサーバ・コンピュータまたは他のコンピュータ上で実行されているグラフィカル・プログラムのリストを受け取ることができる。そこでユーザは、たとえばハイパーテキスト・リンクなどをクリックすることにより、1つまたは複数のグラフィカル・プログラムを選択できる。グラフィカル・プログラムを選択すると、ユーザのブラウザ・アプリケーションがブラウザ・プラグインを起動し、図7の残りのステッ

10

20

30

40

50

ブを処理する。

【0062】

ステップ450～452の他の実施形態も考慮される。たとえば、ユーザはまだWebブラウザ環境のコンテキスト内で作業しているかもしれないが、いずれの時点でもWebサーバと対話していない場合がある。たとえば、ユーザが、そのWebブラウザ内部ではサポートしていないプロトコル方式が使われるURLをWebブラウザに送る場合がある。応答として、WebブラウザはURLをプロトコル・ハンドラ・プラグインに任せる。たとえば、このようなプロトコル・ハンドラ・プラグインをMicrosoft Asynchronous Pluggable Protocols仕様に従って構築できる。このプラグインは、URLで参照している資源、たとえばグラフィカル・プログラムが置かれて

10

【0063】

ステップ454で、ユーザのクライアント・ソフトウェア、たとえばWebブラウザ・プラグインはリモート・コンピュータに接続する。リモート・コンピュータは、図7に示されているクライアント側の演算に対応するサーバ側の演算をサポートする動作が可能なアプリケーションまたはエージェントを備えることができる。クライアント・ソフトウェアとサーバ・ソフトウェアとの間の通信には、さまざまなアプリケーション・レベルのプロトコルを使用できる。好ましい実施形態では、TCP/IPプロトコルに基づく通信プロトコルをリモート・コンピュータとの通信に使用している。接続時に、遠隔のグラフィカル・プログラムは、すでに、リモート・コンピュータで実行されている可能性があり、またリモート・コンピュータはクライアント・コンピュータの接続に

20

【0064】

ステップ456で、ユーザのクライアント・ソフトウェアは、ステップ452で指定されているグラフィカル・プログラムに関連付けられているユーザ・インタフェース・パネルの記述を送信するようリモート・コンピュータに要求する。ステップ456は、ステップ454と組み合わせることができる。この要求に応じて、リモート・コンピュータはユーザ・インタフェース・パネルの記述を送信する。

【0065】

ステップ458で、ユーザのクライアント・ソフトウェア、たとえばWebブラウザ・プラグインはユーザ・インタフェース・パネルの記述を受け取り、ユーザ・インタフェース・パネルを適宜表示する。好ましい実施形態では、クライアント・ソフトウェアが受け取るユーザ・インタフェース・パネルの記述は、リモート・コンピュータがユーザ・インタフェース・パネル情報を永続的に格納するために使用する記述に基づく、またはそれと同一の記述である。つまり、グラフィカル・プログラムとそのユーザ・インタフェース・パネルをリモート・コンピュータで作成し保存すると、ユーザ・インタフェース・パネルを記述する情報が特定の方法で構造化されるということである。好ましい実施形態では、ユーザのクライアント・ソフトウェアはこの構造化された情報の構文解析を行い、ユーザ・インタフェース・パネルをユーザのディスプレイ画面、たとえばユーザのWebブラウザのウィンドウ内に適宜表示する。

30

40

【0066】

ただし、他の実施形態では、リモート・コンピュータはユーザ・インタフェース・パネルの記述を変換してから、クライアント・コンピュータに記述を送る場合があることに注意されたい。たとえば、ユーザ・インタフェース・パネルをリモート・コンピュータにバイナリ形式で格納することができるが、テキスト形式、たとえばマークアップ言語記述に変換することができ、クライアント・コンピュータはこれを処理する動作を行ってパネルを適宜表示できる。このような実施形態では、異なる種類の表示デバイス、たとえばさまざまな種類の無線デバイスに内蔵される小型スクリーンを備えるクライアント・コンピュータが特定の表示デバイスの機能に応じて、ユーザ・インタフェース・パネル記述を異なる

50

形で容易に解釈し表示できるようにすると便利である。

【 0 0 6 7 】

ステップ 4 6 0 で、ユーザのクライアント・ソフトウェアはリモート・コンピュータからデータ更新を受け取り、それに従ってユーザ・インタフェース・パネルの表示を更新することができる。たとえば、上述のように、グラフィカル・プログラムはライブ・データ・ソースからのデータの測定に関連付けることができ、また常時または定期的にユーザ・インタフェース・パネルにライブ・データを表示する動作が可能である。さまざまなデータ・プロトコルをデータ更新の転送および表示に使用できる。

【 0 0 6 8 】

ステップ 4 6 0 についての上の説明は、クライアント・コンピュータに表示されるユーザ・インタフェース・パネルをパネルに表示される実際のデータから「分離する」実施形態に関係している。つまり、クライアント・コンピュータはそれ自体パネル記述とは独立にユーザ・インタフェース・パネルに表示するデータを受け取ることができ、またデータに応じてパネルの表示を更新し、遠隔のグラフィカル・プログラムの出力を反映させることができる。他の実施形態としては、プログラム出力をパネル記述と結合するものもある。たとえば、パネル記述は、プログラム出力を反映する画像として受け取ることができる。そこで、データ更新を受け取るときに、クライアント・コンピュータはユーザ・インタフェース・パネルの更新された記述を受け取り、更新されたパネルを再表示することができる。

【 0 0 6 9 】

ステップ 4 6 2 で、ユーザはユーザ・インタフェース・パネルを、たとえば GUI スタイルの「ポイントしてクリックする」方法で操作することができる。ユーザのクライアント・ソフトウェアは、この GUI 操作をリモート・コンピュータ 8 6 へ仲介する。たとえば、上述のように、ユーザのクライアント・ソフトウェアはサーバ側のエージェントと通信し、コマンドを遠隔のグラフィカル・プログラムに転送することができる。その後、その結果に応じて遠隔のグラフィカル・プログラムはコマンドに回答する。多くの場合、ステップ 4 6 2 のユーザのコマンドを使用すると、グラフィカル・プログラムはその出力表示を変更するため、出力表示はユーザのディスプレイ画面に反映される。つまり、ユーザがクライアント・コンピュータ 8 2 に表示されるユーザ・インタフェースでの入力を操作することに回答して、ユーザ入力はサーバ・コンピュータ 8 6 で実行されているグラフィカル・プログラムに送られるため、グラフィカル・プログラムの表示される出力に影響を及ぼすことがある。表示された出力は、サーバ・コンピュータ 8 6 に送られ、クライアント・コンピュータ 8 2 に表示されるユーザ・インタフェースに表示される。その後、ユーザは他の入力をグラフィカル・ユーザ・インタフェースなどに送ることができる。ステップ 4 6 0 および 4 6 2 は、反復して実行することができる。

【 0 0 7 0 】

データソケット

他の実施形態では、米国特許第 0 9 / 1 8 5 1 6 1 号で開示されているデータソケットのシステムと方法を使用して、ユーザのクライアント・ソフトウェアとグラフィカル・プログラムが実行されているリモート・コンピュータとの間の通信を簡単に行えるようにできる。データソケット・システムはクライアント・ソフトウェア・コンポーネントを備える。URL を使用して世界中のあらゆる場所の Web ページのアドレス指定を行うのと同じような方法で、クライアント・ソフトウェア・コンポーネントは、URL を使用してデータ・ソース/ターゲットのアドレスを指定する。入力ソースから読み取るときに、データソケットは、さまざまな入力ソースから生データを読み取り、データを構文解析して、ユーザのアプリケーションで直接使用可能な形式で戻す。たとえば、本発明の実施形態の一実施形態に関して、データソケットを使用して、遠隔のグラフィカル・プログラムのユーザ・インタフェース・パネルの記述を受け取り、この記述を、たとえば、動作可能な Web ブラウザ・プラグインに仲介し、Web ブラウザのウィンドウ内にユーザ・インタフェース・パネルを表示できる。ユーザ・インタフェース・パネルが表示されたら、データソ

10

20

30

40

50

ケットは遠隔のグラフィカル・プログラムからデータ更新を受け取ることができる。これはユーザ・インタフェース・パネルに表示される。

【0071】

出力ターゲットに書き込むと、データソケットはユーザによって提供されるデータを特定のターゲットに適切な生データ形式に変換するのに必要なすべての作業を実行する。たとえば、本発明の一実施形態に関して、データソケットはユーザの入力コマンドを適切な形式に整え、遠隔のグラフィカル・プログラムに送ることができる。データソケットのシステムと方法の詳細については、上記の特許出願を参照されたい。

【0072】

遠隔のグラフィカル・プログラムのブロック・ダイアグラムの受取

10

一実施形態では、ユーザはさらに、遠隔のグラフィカル・プログラムのブロック・ダイアグラムを要求し受け取ることができる。ブロック・ダイアグラムは、単純な非対話的画像として表示することができ、たとえば、グラフィカル・プログラムの実装方法をユーザに理解させるために使用できる。たとえば、遠隔のグラフィカル・プログラムを、グラフィカル・プログラムをプログラムで編集する機能を備えるグラフィカル・プログラミング環境内で実行する。そこで、ユーザはブロック・ダイアグラムの表から得た情報を使用して、グラフィカル・プログラムを遠隔から編集することができる。グラフィカル・プログラムの動的作成または編集の詳細については、「System and Method for Programmatically Creating a Graphical Program」という表題の上述の特許出願第09/518492号を参照されたい

20

【0073】

他の実施形態では、クライアント・コンピュータは実際のブロック・ダイアグラムを受け取って表示することができ、それにより、ユーザはクライアント・コンピュータを使用してブロック・ダイアグラムを表示し編集することができる。その後、クライアント・コンピュータのユーザは、編集されたブロック・ダイアグラムをサーバ・コンピュータに転送して返す。

【0074】

他の実施形態では、ユーザは、グラフィカル・プログラムがリモート・コンピュータで実行されている間に、プログラムのデバッグなどの操作を対話的に実行できる。クライアント・ソフトウェアは、リモート・コンピュータと通信して、ブレーク・ポイントなどのデバッグ情報を指定し、ブレーク・ポイントから実行を継続するなどのプログラム実行を制御することができる。クライアント・ソフトウェアは、遠隔のグラフィカル・プログラムの動作をさまざまな方法で示す操作が可能であり、たとえば、実行ハイライト表示を使用して、ブロック・ダイアグラムの表示を更新し、リアルタイムの実行またはデータ・フローを示すことなどができる。

30

【0075】

図8～10 - グラフィカル・プログラムの例

図8～図10は、本発明のシステムと方法を適用できるグラフィカル・プログラムのいくつかの例を示している。各図は、プログラムおよび関連するユーザ・インタフェース・パネルのブロック図である。上述のように、グラフィカル・プログラムを一方のコンピュータ上で実行しておいて、一人または複数のエンド・ユーザが遠隔地で、異なるコンピュータからグラフィカル・プログラムのユーザ・インタフェース・パネルを表示したり、対話的に操作したりできる。また、エンド・ユーザは、グラフィカル・プログラムのブロック・ダイアグラムを遠隔地で表示かつ/または編集することもできる。各グラフィカル・プログラムの例について以下の段落で簡単に説明する。

40

【0076】

図8Bに示されているブロック図は、タンク制御アプリケーションをシミュレートするものである。図8Aの関連するユーザ・インタフェース・パネルは、タンク制御アプリケーションの流入量、レベル、および温度の履歴を表示する。

50

## 【0077】

図9Bに示されているブロック図は、GPIB計測器を使用してテスト対象のユニット(UUT)に周波数応答テストを実行するアプリケーションをシミュレートするものである。ファンクション・ジェネレータが、正弦波入力をUUT(この例では帯域通過フィルタ)に供給し、デジタル・マルチメータでUUTの出力電圧を測定する。

## 【0078】

図10Bに示されているブロック図は、温度分析アプリケーションをシミュレートするものである。このプログラムは、シミュレートされた温度を読み取り、所定の範囲を外れていた場合にアラームを発生し、温度履歴の平均値、標準偏差、およびヒストグラムを決定する。

10

## 【0079】

図8~10に示されているグラフィカル・プログラムの例は、計装、工業オートメーション、またはプロセス制御アプリケーションを指向している。これらのプログラムのユーザ・インタフェース・パネルは、ハードウェアによる計測器またはコンソールにあるのと似たさまざまなコントロールまたは表示読み取り値を含む。しかし、上述のように、さまざまな分野で仕事をしているプログラム開発者およびエンド・ユーザは、本発明で説明しているシステムと方法を利用し、さまざまなアプリケーションに対応するグラフィカル・プログラムのユーザ・インタフェースを分散表示および/制御することが可能である。

## 【0080】

本発明のシステムと方法は好ましい実施形態に関して説明されているが、ここで述べた特定の形態に限定する意図はなく、それどころか、添付の請求項により定められているように本発明の精神と範囲に含めることが妥当である限りそのような代替物、修正物、等価物を対象とすることを意図している。

20

## 【図面の簡単な説明】

【図1】 ネットワークを介して第2のコンピュータ・システムに接続されているコンピュータ・システムの図である。

【図2A】 さまざまなI/Oインタフェース・オプションを備える代表的な計装およびプロセス制御システムの図である。

【図2B】 さまざまなI/Oインタフェース・オプションを備える代表的な計装およびプロセス制御システムの図である。

30

【図3】 図1、2A、および2Bのコンピュータ・システムのブロック図である。

【図4】 グラフィカル・プログラムを対話的に作成または編集する一実施形態を説明する流れ図である。

【図5】 ユーザ・インタフェース・パネルとブロック・ダイアグラムを備える単純なグラフィカル・プログラムの図である。

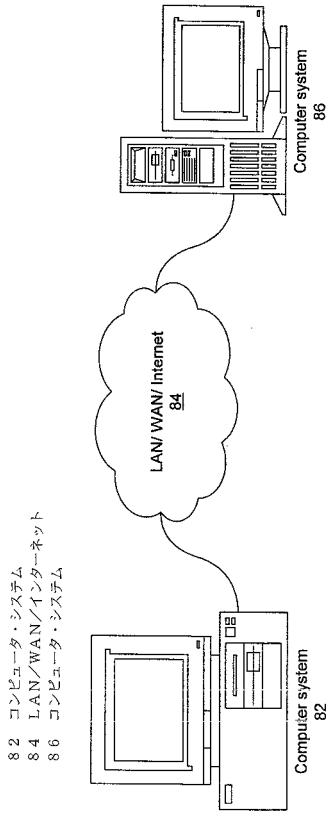
【図6】 ユーザ・インタフェース・パネルとブロック・ダイアグラムを備える単純なグラフィカル・プログラムの図である。

【図7】 遠隔のグラフィカル・プログラムにユーザがアクセスする一実施形態を説明する流れ図である。

【図8】 ~ 【図10】 グラフィカル・プログラムとその関連するユーザ・インタフェースの例を示す図である。

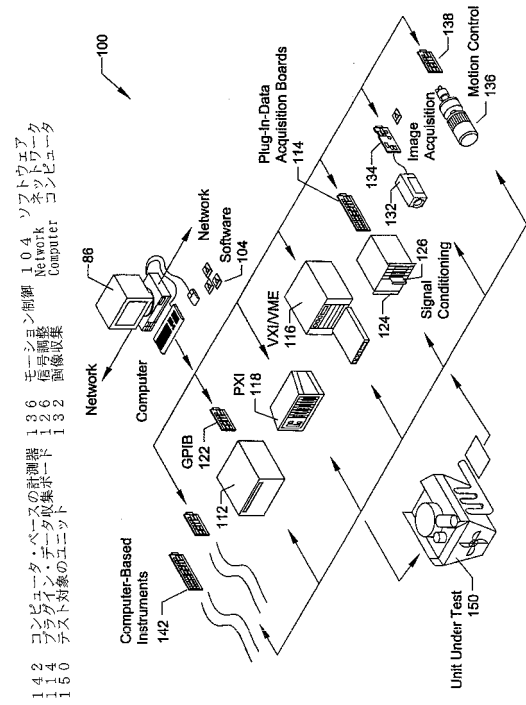
40

【図1】



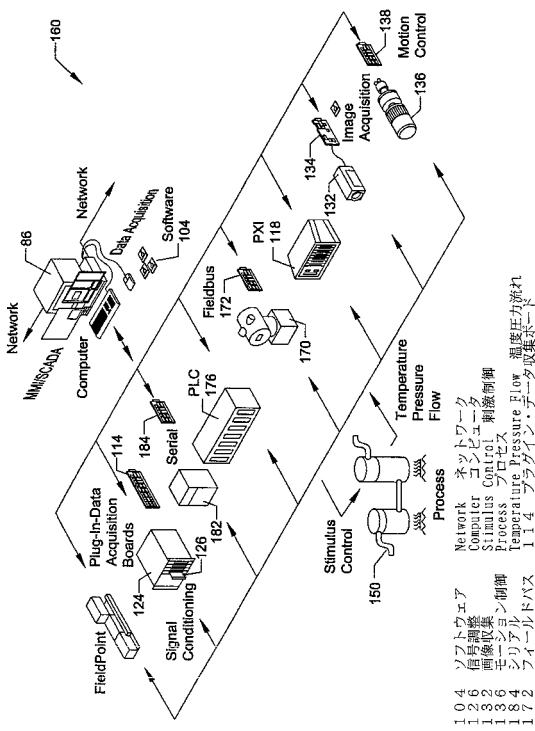
- 82 コンピュータ・システム
- 84 LAN/WAN/インターネット
- 86 コンピュータ・システム

【図2A】



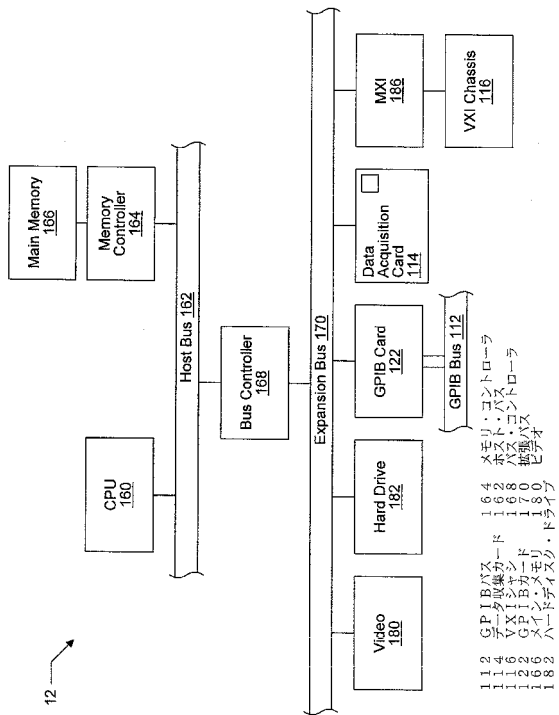
- 100
- 86 コンピュータ・システム
- 104 ネットワークソフトウェア
- 114 プラグイン・データ取得ボード
- 122 GPIB
- 118 PXI
- 116 VXI/VME
- 126 シグナル調整
- 134 イメージ取得
- 138 モーション制御
- 142 コンピュータベースの計測器
- 150 テスト対象のユニット

【図2B】



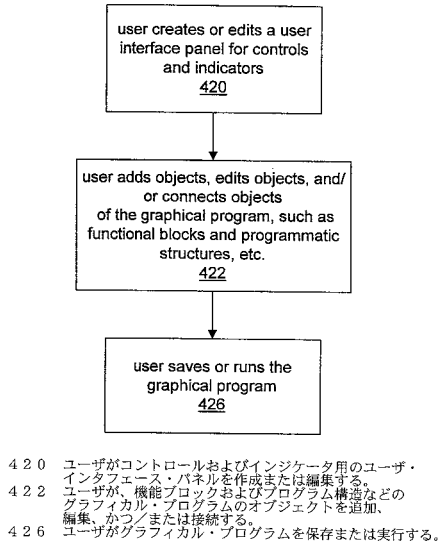
- 104 ソフトウェア
- 124 フィールドポイント
- 134 イメージ取得
- 138 モーション制御
- 142 コンピュータベースの計測器
- 150 スティムルス制御
- 170 温度圧力流れ
- 172 フィールドバス
- 184 シリアル
- 188 ファイバーストック・ドライバ

【図3】

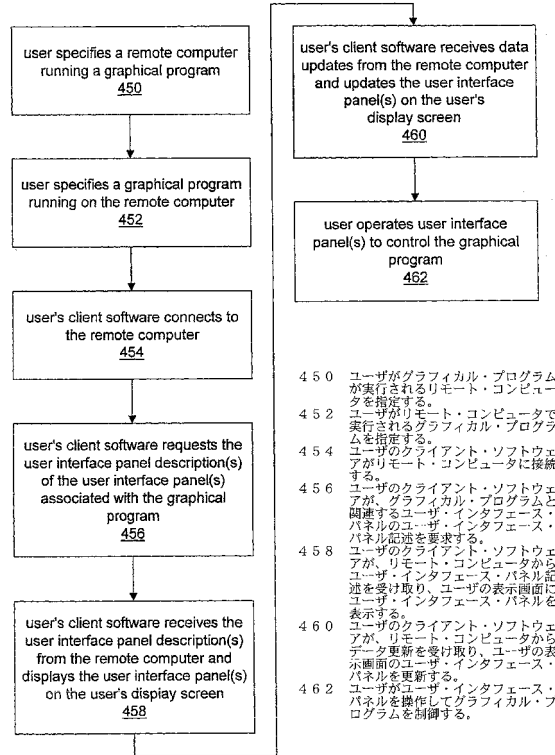


- 112 GPIBバス
- 114 データ取得カード
- 116 VXIシャーシ
- 122 GPIBカード
- 162 ホストバス
- 164 メモリコントローラ
- 166 メモリ
- 168 バスコントローラ
- 180 ビデオ
- 182 ハードディスク・ドライブ
- 186 MXI
- 188 CPU

【 図 4 】



【 図 7 】



【 図 5 】

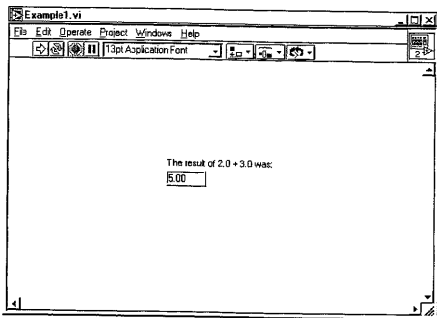


FIG. 5

【 図 6 】

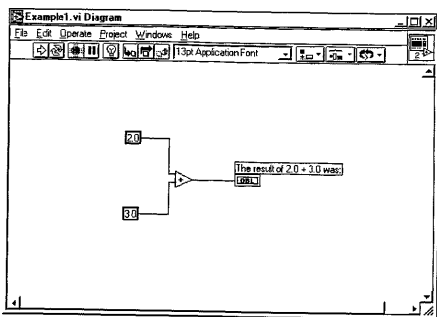


FIG. 6

【 図 8 A 】

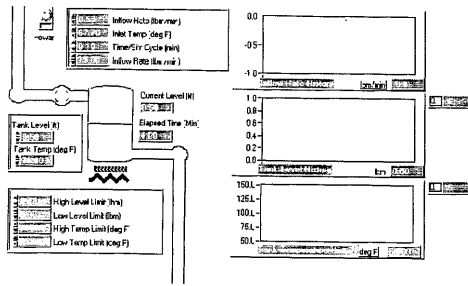


FIG. 8A

【 図 8 B 】

FIG. 8B



【 9 A 】

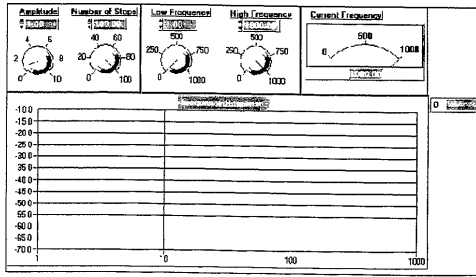


FIG. 9A

【 9 B 】

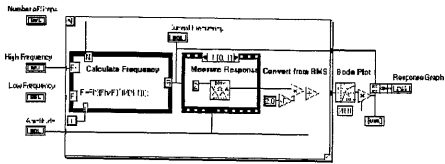


FIG. 9B

【 10 A 】

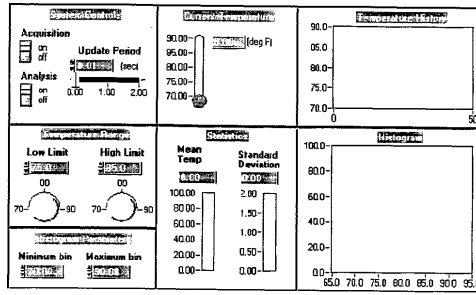


FIG. 10A

【 10 B 】

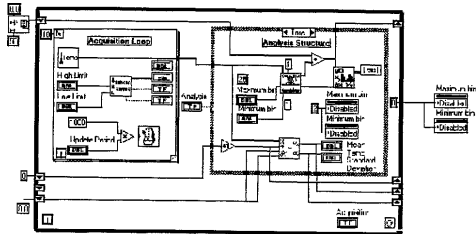


FIG. 10B

## フロントページの続き

- (72)発明者 シャー, ダーシャン  
アメリカ合衆国・78681・テキサス州・ラウンド ロック・ホーク コート・1004
- (72)発明者 ロジャーズ, スティーブ  
アメリカ合衆国・78757・テキサス州・オースティン・ボウリング グリーン ドライブ・8  
309
- (72)発明者 リチャードソン, グレッグ  
アメリカ合衆国・78681・テキサス州・ラウンド ロック・スプリング ホロウ パス・21  
12
- (72)発明者 ルイック, ディーン・エイ  
アメリカ合衆国・78759・テキサス州・オースティン・スパイスウッド スプリングズ ロー  
ド・6429

審査官 稲垣 良一

- (56)参考文献 特開平10-240337(JP, A)  
特開平6-230947(JP, A)  
特開平10-315171(JP, A)  
関野 敏正 他, 特集: オープンコンピューティングによる制御技術 (4)計測制御ソフトウェア・  
OS, エレクトロニクス, 株式会社オーム社, 1996年11月 1日, 第41巻, 第11号, pp.7  
3-85

## (58)調査した分野(Int.Cl., DB名)

G06F 9/44  
G06F 3/048  
G06F 13/00  
G06F 15/00  
G06F 17/50