

(19)日本国特許庁(JP)

## (12)特許公報(B2)

(11)特許番号

特許第7381502号  
(P7381502)

(45)発行日 令和5年11月15日(2023.11.15)

(24)登録日 令和5年11月7日(2023.11.7)

(51)国際特許分類

F I

G 0 6 F 8/34 (2018.01)

G 0 6 F 8/34

G 1 6 Y 40/30 (2020.01)

G 1 6 Y 40/30

G 0 6 F 9/48 (2006.01)

G 0 6 F 9/48 3 7 0

G 0 6 F 9/50 (2006.01)

G 0 6 F 9/50 1 2 0 A

請求項の数 38 (全55頁)

(21)出願番号 特願2020-567518(P2020-567518)

(86)(22)出願日 令和1年6月5日(2019.6.5)

(65)公表番号 特表2021-526688(P2021-526688  
A)

(43)公表日 令和3年10月7日(2021.10.7)

(86)国際出願番号 PCT/US2019/035502

(87)国際公開番号 WO2019/236657

(87)国際公開日 令和1年12月12日(2019.12.12)

審査請求日 令和4年5月16日(2022.5.16)

(31)優先権主張番号 62/681,397

(32)優先日 平成30年6月6日(2018.6.6)

(33)優先権主張国・地域又は機関  
米国(US)

(31)優先権主張番号 62/741,326

(32)優先日 平成30年10月4日(2018.10.4)

最終頁に続く

(73)特許権者 520469664

プレシエント デバイシーズ, インコー  
ポレイテッドアメリカ合衆国 マサチューセッツ 01  
450, グロットン, リバーバンド  
ドライブ 281

(74)代理人 100078282

弁理士 山本 秀策

(74)代理人 100113413

弁理士 森下 夏樹

(74)代理人 100181674

弁理士 飯田 貴敏

(74)代理人 100181641

弁理士 石川 大輔

(74)代理人 230113332

最終頁に続く

(54)【発明の名称】 分散型異種コンピューティングおよび制御システムを設計する方法およびシステム

## (57)【特許請求の範囲】

## 【請求項1】

分散型異種コンピューティングおよび制御システムを設計するコンピュータ実装方法であって、前記分散型異種コンピューティングおよび制御システムは、アプリケーションおよびハードウェアコンテキストの両方を含み、前記ハードウェアコンテキストにおいて前記アプリケーションを構成し、前記方法は、双方向ディスプレイに関連付けられたコンピュータノードにおいて実行する設計ソフトウェア命令によって実装され、コンピュータプロセスを利用して双方向環境を確立し、前記コンピュータプロセスは、

前記コンピュータノードが、前記双方向環境において、機能モジュールの組およびプリミティブモジュールの組へのアクセスを提供することであって、前記モジュールのいずれも、ユーザによってグラフィカルに選択されることができる、ことと、

前記コンピュータノードが、前記双方向環境において、前記機能モジュールおよび前記プリミティブモジュールのうちの所望のものの選択を受信し、デバイスの組の回路図として、前記分散型異種コンピューティングおよび制御システムを特徴づける様式でそれらを順序付けることと、

前記コンピュータノードが、前記回路図を解析し、サブ回路図の組を生成することであって、各サブ回路図は、前記分散型異種コンピューティングおよび制御システムにおけるデバイスの組における異なるデバイスに対応する、ことと、

前記コンピュータノードが、前記回路図に従って、前記デバイスの各々を特徴づけるために、前記サブ回路図の組を前記デバイスの組に伝送することと、

前記コンピュータノードが、前記デバイスの内側で起動する対応する制御プログラムを介して、前記サブ回路図を実行するように各デバイスを構成することとを含む、コンピュータ実装方法。

【請求項 2】

前記コンピュータノードは、独立型コンピュータである、請求項 1 に記載のコンピュータ実装方法。

【請求項 3】

前記コンピュータノードは、ネットワーク環境において、クライアントーサーバの組み合わせによって実装される、請求項 1 に記載のコンピュータ実装方法。

【請求項 4】

前記コンピュータプロセスは、

前記コンピュータノードが、ネットリストによって前記回路図を特徴づけることと、

前記コンピュータノードが、前記回路図を前記ネットリストとして記憶することと

をさらに含む、請求項 1 に記載のコンピュータ実装方法。

【請求項 5】

前記コンピュータプロセスは、

前記コンピュータノードが、前記双方向環境において、記憶されたネットリストを読み出し、処理することによって、前記回路図の表示をレンダリングすることをさらに含む、請求項 1 に記載のコンピュータ実装方法。

【請求項 6】

前記双方向環境は、設計キャンバス、ライブラリマネージャ、メニューのグラフィカルコンポーネントを含み、前記グラフィカルコンポーネントのいずれも、ユーザによってグラフィカルに選択されることができる、請求項 1 に記載のコンピュータ実装方法。

【請求項 7】

前記コンピュータプロセスは、

前記コンピュータノードが、前記設計キャンバスにおいて、前記回路図、対応するネットリスト、およびそれらの組み合わせから成る群から選択されるメンバーを表示することをさらに含む、請求項 6 に記載のコンピュータ実装方法。

【請求項 8】

前記コンピュータプロセスは、

前記コンピュータノードが、一意のデバイス識別子 (ID) を前記デバイスの組における各デバイスに割り当ててをさらに含む、請求項 1 に記載のコンピュータ実装方法。

【請求項 9】

前記コンピュータプロセスは、

前記コンピュータノードが、デバイス名を前記デバイスの組における少なくとも 1 つのデバイスに割り当ててをさらに含む、前記デバイス名は、ユーザによってグラフィカルに記入されることができる、請求項 1 に記載のコンピュータ実装方法。

【請求項 10】

前記機能モジュールおよび前記プリミティブモジュールのうちの所望のものの前記選択を受信することは、前記所望のものに対応するモジュールインスタンスの作成をもたらし、前記分散型異種コンピューティングおよび制御システムを前記回路図として特徴づける様式で前記モジュールインスタンスを順序付ける、請求項 1 に記載のコンピュータ実装方法。

【請求項 11】

前記モジュールインスタンスを順序付けることは、互いに通信すべき一対の前記モジュールインスタンスに対して、仮想ワイヤを前記対間に構成し、そのような通信を示すことを含む、請求項 10 に記載のコンピュータ実装方法。

【請求項 12】

前記コンピュータプロセスは、

前記コンピュータノードが、所与のモジュールのグラフィカル表現を前記双方向環境の

10

20

30

40

50

設計キャンバスにおいて表示された前記回路図の表示にドラッグし、ドロップする入力を受信することに続いて、前記所与のモジュールのモジュールインスタンスが前記回路図において作成され、順序付けられるようにすることをさらに含む、請求項 1 に記載のコンピュータ実装方法。

【請求項 1 3】

前記コンピュータプロセスは、

前記コンピュータノードが、（i） プロパティテーブルにおける少なくとも 1 つのプロパティ、（i i） グラフィカル表現、（i i i） 少なくとも 1 つの回路図、（i v） 少なくとも 1 つのデバイス依存ブロック、および、（i v） それらの組み合わせから成る群から選択されるグラフィカルに選択可能なコンポーネントを使用して、所与のモジュールのモジュールインスタンスを特徴づけることをさらに含む、請求項 1 に記載のコンピュータ実装方法。

10

【請求項 1 4】

前記プロパティテーブルは、前記少なくとも 1 つの回路図を特徴づけるためのプロパティを含み、前記プロパティは、タイプおよびバージョンを含む、請求項 1 3 に記載のコンピュータ実装方法。

【請求項 1 5】

前記所与のモジュールが、プリミティブモジュールであるとき、前記プロパティテーブルは、前記所与のモジュールに関連付けられた各デバイス依存ブロックのためのプロパティを含み、各デバイス依存ブロックのための前記プロパティは、デバイスタイプリストおよびバージョンを含む、請求項 1 3 に記載のコンピュータ実装方法。

20

【請求項 1 6】

前記プロパティテーブルは、（i） インスタンス名、（i i） 前記所与のモジュールをインスタンス化するためのインスタンスタイプ、（i i i） 前記モジュールインスタンスを実行すべき特定のデバイス、（i v） 少なくとも 1 つのインスタンスパラメータ、および、（v） それらの組み合わせから成る群から選択される前記モジュールインスタンスのためのユーザ選択可能プロパティを含む、請求項 1 3 に記載のコンピュータ実装方法。

【請求項 1 7】

前記特定のデバイスの少なくとも 1 つのプロパティは、前記モジュールインスタンスから作成された各子モジュールインスタンスによって継承される、請求項 1 6 に記載のコンピュータ実装方法。

30

【請求項 1 8】

前記コンピュータプロセスは、

前記コンピュータノードが、デバイス依存ブロックを所与のプリミティブモジュールのモジュールインスタンスに関連付けることをさらに含む、

前記関連付けることは、

前記モジュールインスタンスのためのプロパティテーブルにおいて、前記モジュールインスタンスに対応するデバイスプロパティを見つけることであって、前記デバイスプロパティは、前記デバイスの組のうちの 1 つのためのデバイス識別子（ID）を規定することと、

40

前記デバイス ID をデバイスタイプにマッピングし、所与のモジュールのための前記プロパティテーブルにおいて、前記デバイスタイプに関連付けられた前記デバイス依存ブロックを見つけることと、

前記モジュールインスタンスのための前記プロパティテーブルにおいて、前記モジュールインスタンスに対応するタイププロパティを前記デバイス依存ブロックに割り当てられた名称で自動的に更新することと

を含む、請求項 1 に記載のコンピュータ実装方法。

【請求項 1 9】

前記タイププロパティを自動的に更新することは、前記タイププロパティのユーザ選択によって上書きされる、請求項 1 8 に記載のコンピュータ実装方法。

50

## 【請求項 20】

前記コンピュータプロセスは、

前記コンピュータノードが、前記双方向環境において、

前記回路図を特徴づけるモジュールインスタンスの階層構造を含むツリーグラフと、  
正しく選定されたデバイス依存ブロックのグラフィカル表現と

を表示することをさらに含み、

前記正しく選定されたデバイス依存ブロックのグラフィカル表現は、前記ツリーグラフ  
の下方に位置付けられ、前記ツリーグラフにおける対応するプリミティブモジュールイン  
スタンスにリンクされている、請求項 1 に記載のコンピュータ実装方法。

## 【請求項 21】

前記サブ回路図の組を前記デバイスの組に伝送することは、プリミティブモジュールイン  
スタンスを有する各サブ回路図に関して、デバイス依存ブロックを前記サブ回路図に対  
応する異なるデバイスに伝送することを含む、請求項 1 に記載のコンピュータ実装方法。

## 【請求項 22】

前記回路図は、モジュールインスタンスを含むコンポーネントを有し、

前記コンピュータプロセスは、

前記コンピュータノードが、前記回路図の所与のコンポーネントに、(i) デバイスの  
デバイス ID、(i i) 前記所与のコンポーネントに到達するために前記回路図内で順に  
トラバースされるべき各コンポーネントの名称を定義するパス、および、(i i i) 前記  
所与のコンポーネントの名称の組み合わせに基づいて、異なるアドレスを割り当てること  
をさらに含む、請求項 1 に記載のコンピュータ実装方法。

## 【請求項 23】

前記回路図は、モジュールインスタンスを含むコンポーネントを有し、

前記コンピュータプロセスは、

前記コンピュータノードが、所与のモジュールインスタンスに、(i) デバイスのデバ  
イス ID、および、(i i) 前記所与のモジュールインスタンスの ID の組み合わせに基  
づいて、異なるアドレスを割り当てることと、

前記コンピュータノードが、モジュールインスタンスではない所与のコンポーネントに  
、(i) デバイスのデバイス ID、(i i) 前記所与のコンポーネントを含むモジュール  
インスタンスの ID、および、(i i i) 前記所与のコンポーネントの名称の組み合わせ  
によって定義される一意のアドレスを割り当てることと

をさらに含む、請求項 1 に記載のコンピュータ実装方法。

## 【請求項 24】

前記回路図を解析することは、

前記デバイスの組における各デバイスのために、前記デバイスのためのサブ回路図を作  
成することと、

前記回路図の各モジュールインスタンスのために、前記モジュールインスタンスをサブ  
回路図の中にコピーすることであって、前記サブ回路図において、前記サブ回路図の前記  
デバイスは、前記モジュールインスタンスのデバイス ID に対応している、ことと、

前記回路図における第 1 のモジュールインスタンスと第 2 のモジュールインスタンスとを  
接続する仮想ワイヤを処理することと

を含み、

前記仮想ワイヤを処理することは、

前記第 1 のモジュールインスタンスおよび前記第 2 のモジュールインスタンスの両方が  
所与のサブ回路図の中にあるとき、前記仮想ワイヤを前記所与のサブ回路図において保存  
することと、

前記仮想ワイヤが、(i) 第 1 のサブ回路図の中の前記第 1 のモジュールインスタンス  
の出力ピンを (i i) 第 2 のサブ回路図の中の前記第 2 のモジュールインスタンスの入力  
ピンに接続するとき、前記第 1 のサブ回路図と前記第 2 のサブ回路図との間の c o m m 接  
続を作成することと

10

20

30

40

50

を含む、請求項 1 に記載のコンピュータ実装方法。

【請求項 2 5】

前記 c o m m 接続を作成することは、

第 1 の c o m m モジュールインスタンスを前記第 1 のサブ回路図に追加し、仮想ワイヤを前記第 1 のモジュールインスタンスの前記出力ピンと前記第 1 の c o m m モジュールインスタンスの入力ピンとの間に追加することと、

第 2 の c o m m モジュールインスタンスを前記第 2 のサブ回路図に追加し、仮想ワイヤを前記第 2 の c o m m モジュールインスタンスの出力ピンと前記第 2 のモジュールインスタンスの前記入力ピンとの間に追加することと、

前記第 1 の c o m m モジュールインスタンスに含まれる宛先アドレスを前記第 2 のモジュールインスタンスの前記入力ピンのアドレスに設定することと、

前記第 2 の c o m m モジュールインスタンスに含まれる受信アドレスを前記第 2 のモジュールインスタンスの前記入力ピンのアドレスに設定することと

を含む、請求項 2 4 に記載のコンピュータ実装方法。

【請求項 2 6】

前記コンピュータプロセスは、

前記コンピュータノードが、前記宛先アドレスおよび受信アドレスの各々に、( i ) デバイスのデバイス ID、( i i ) 標的化されたピンに到達するために前記回路図内で順にトラバースされるべき各コンポーネントの名称を定義するパス、および、( i i i ) 前記標的化されたピンの名称の組み合わせに基づいて、異なるアドレスを割り当てることをさらに含む、請求項 2 5 に記載のコンピュータ実装方法。

【請求項 2 7】

前記コンピュータプロセスは、

前記コンピュータノードが、前記宛先アドレスおよび受信アドレスの各々に、( i ) デバイスのデバイス ID、( i i ) 標的化されたピンを含むモジュールインスタンスの ID、および、( i i i ) 前記標的化されたピンの名称の組み合わせに基づいて、異なるアドレスを割り当てることをさらに含む、請求項 2 5 に記載のコンピュータ実装方法。

【請求項 2 8】

前記 c o m m モジュールは、複数の機能回路図に関連付けられており、各機能回路図は、異なる通信プロトコルを実装し、前記双方向環境は、( i ) 所与の c o m m モジュールインスタンスのための c o m m モジュールインスタンスタイプを個々に変更するユーザ選択と、( i i ) 各 c o m m モジュールインスタンスのための前記 c o m m モジュールインスタンスタイプを大域的に変更するユーザ選択との両方のために構成されている、請求項 2 5 に記載のコンピュータ実装方法。

【請求項 2 9】

前記コンピュータプロセスは、前記コンピュータノードが、中央 c o m m ブローカを確立することをさらに含み、前記中央 c o m m ブローカは、第 1 の c o m m モジュールインスタンスと第 2 の c o m m モジュールインスタンスとの間の通信のためにパブリッシュサブスクライブ通信プロトコルを利用し、

前記パブリッシュサブスクライブ通信プロトコルは、

第 2 の c o m m モジュールインスタンスを前記中央 c o m m ブローカにサブスクライブするための要求を受信することであって、前記要求は、前記第 2 の c o m m モジュールインスタンスの受信アドレスを規定することを含み、前記受信アドレスは、前記第 2 の c o m m モジュールに接続されたモジュールインスタンスの入力ピンのアドレスである、ことと、

データパケットを第 1 の c o m m モジュールインスタンスから前記中央 c o m m ブローカに伝送することであって、前記データパケットは、データペイロード、宛先アドレス、ソースアドレス、およびそれらの組み合わせから成る群から選択されるデータを含み、前記ソースアドレスは、前記第 1 の c o m m モジュールインスタンスに接続されたモジュールインスタンスの出力ピンのアドレスであり、前記宛先アドレスは、意図される宛先モ

10

20

30

40

50

ジュールインスタンスの前記入カピンのアドレスである、ことと、

前記データパケットにおける前記宛先アドレスが前記第2のc o m mモジュールインスタンスの前記受信アドレスに合致しているとき、前記中央c o m mブローカに前記データパケットを前記第2のc o m mモジュールインスタンスに転送させることと

を行うように構成されている、請求項1に記載のコンピュータ実装方法。

【請求項30】

前記サブ回路図は、伝送パスおよび受信パスを含むc o m mモジュールインスタンスが前記サブ回路図の複数のモジュールインスタンス間で共有されるように構成され、前記コンピュータプロセスは、

前記コンピュータノードが、前記デバイスの組のデバイスにおけるc o m mモジュールインスタンスに前記デバイスの識別を含む宛先アドレスを伴う所与のデータパケットを伝送することをさらに含み、

前記c o m mモジュールインスタンスは、

前記受信パスを通して、前記デバイスの識別を含む宛先アドレスを有する任意のデータパケットを受信することであって、前記任意のデータパケットは、前記所与のデータパケットを含む、ことと、

前記所与のデータパケットにおける前記宛先アドレスを分析し、前記所与のデータパケットを前記宛先アドレスに対応する受信アドレスを伴う前記サブ回路図のモジュールインスタンスに転送することと

を行うように構成されている、請求項1に記載のコンピュータ実装方法。

【請求項31】

前記コンピュータプロセスは、

前記コンピュータノードが、前記デバイスの組のデバイスに前記デバイス上にロードされた新しい回路図を実行させることをさらに含み、前記デバイスは、制御プログラムと、s y sモジュールインスタンスに接続されたc o m mモジュールインスタンスを有するi n i t回路図とを事前にロードされ、前記c o m mモジュールインスタンスの受信アドレスは、宛先アドレスが前記デバイスの識別および前記s y sモジュールインスタンスを含む任意のデータパケットを受信するように設定され、

前記コンピュータプロセスは、

前記コンピュータノードが、前記デバイスに、新しいサブ回路図と、前記デバイスの識別および前記s y sモジュールインスタンスを含む前記宛先アドレスとを含むデータペイロードを有するデータパケットを伝送することを含み、

前記宛先アドレスは、(i)前記データパケットを受信することと、(i i)前記デバイス上に前記新しいサブ回路図をロードすることと、(i i i)前記制御プログラムに前記新しいサブ回路図を実行するように要求することとを前記s y sモジュールインスタンスに行わせる、請求項1に記載のコンピュータ実装方法。

【請求項32】

前記コンピュータプロセスは、

前記コンピュータノードが、デバイスに回路図を実行させることをさらに含み、

前記デバイスは、

制御プログラムおよびi n i t回路図を事前にロードすることと、

前記デバイスの電源がオンにされると、前記制御プログラムによる前記i n i t回路図の実行を引き起こすことと、

前記i n i t回路図を使用して、新しいサブ回路図を受信し、ロードすることと、

前記制御プログラムによる前記新しいサブ回路図の実行を引き起こす再ロードコマンドを受信することと

を行うように構成されている、請求項1に記載のコンピュータ実装方法。

【請求項33】

サブ回路図における各プリミティブモジュールインスタンスに関して、前記デバイスにおいて、前記プリミティブモジュールインスタンスに関連付けられたデバイス依存プロッ

10

20

30

40

50

クを実行し、前記デバイス依存ブロックは、前記関連付けられたプリミティブモジュールインスタンスのタイプモジュールインスタンスプロパティを使用して、前記制御プログラムによってロードされる、請求項 3 2 に記載のコンピュータ実装方法。

【請求項 3 4】

前記コンピュータプロセスは、

前記コンピュータノードが、サブ回路図をバイナリコードに変換するコンパイラを実行することと、

前記コンピュータノードが、前記回路図に対応する低能力デバイスに前記デバイス上での実行のためのバイナリコードを伝送することと

をさらに含む、請求項 1 に記載のコンピュータ実装方法。

【請求項 3 5】

前記コンピュータプロセスは、

前記コンピュータノードが、前記双方向環境において、前記分散型異種コンピューティングおよび制御システムを前記回路図としてさらに特徴づけるための可視化モジュールインスタンスの選択を受信することと、

前記コンピュータノードが、前記回路図における前記可視化モジュールインスタンスのデバイスプロパティをシミュレーションコンテキストを示す値に設定することと、

前記コンピュータノードが、前記回路図を解析し、前記サブ回路図の組を生成することであって、前記サブ回路図の組を生成することは、仮想デバイスのためのサブ回路図を生成することを含む、ことと、

前記コンピュータノードが、前記仮想デバイスを通して、前記シミュレーションコンテキストにおいて、前記サブ回路図を実行することと

をさらに含む、請求項 1 に記載のコンピュータ実装方法。

【請求項 3 6】

前記可視化モジュールインスタンスの前記デバイスプロパティは、少なくとも 1 つのデバイスの ID に設定され、前記コンピュータプロセスは、前記可視化モジュールインスタンスが実行のために前記少なくとも 1 つのデバイスに伝送されるようにすることをさらに含む、請求項 3 5 に記載のコンピュータ実装方法。

【請求項 3 7】

前記コンピュータプロセスは、

前記コンピュータノードが、前記双方向環境において、シミュレーションメニューオプションの選択を受信することと、

前記回路図における各モジュールインスタンスに関して、

前記コンピュータノードが、前記モジュールインスタンスのデバイスプロパティをシミュレーションコンテキストにおける前記モジュールインスタンスの実行を示す値に設定することと、

デフォルト挙動回路図が前記モジュールインスタンスに関連付けられているとき、前記モジュールインスタンスのタイププロパティをデフォルト挙動回路図を示すコンポーネント名に設定することと、

前記回路図を解析し、仮想デバイスのための 1 つのみのサブ回路図を生成することと、

前記コンピュータノードが、前記仮想デバイスを通して、前記シミュレーションコンテキストにおいて、前記サブ回路図の各モジュールインスタンスを実行することと

をさらに含む、請求項 1 に記載のコンピュータ実装方法。

【請求項 3 8】

前記モジュールインスタンスのタイププロパティは、前記デフォルト挙動回路図の前記コンポーネント名と異なる値にユーザ選択可能である、請求項 3 7 に記載のコンピュータ実装方法。

【発明の詳細な説明】

【技術分野】

【0001】

(優先権)

本特許出願は、2018年6月6日に出願された米国仮特許出願62/681,397号、および2018年10月4日に出願された米国仮特許出願62/741,326号の利益を主張する。これらの出願の各々は、参照することによってその全体として本明細書に組み込まれる。

(技術分野)

【0002】

本発明は、分散型異種コンピューティングシステムのためのアプリケーションを設計する方法に関する。

【背景技術】

【0003】

ハードウェアおよびソフトウェアコンポーネントの両方を有するコンピューティングおよび制御システムの概念が、当技術分野において公知である。コンピューティングおよび制御システムは、機器、デバイス、およびコンピュータプロセスの協調制御を提供するために使用される。また、そのような機器、デバイス、およびシステムの内側のコンピュータプロセスによって収集されたデータ上にデータ処理能力を提供する。コンピューティングおよび制御システムは、複数のデバイスが、システム内に存在し、ある協調が、これらのデバイス間に生じる場合、分散型と見なされる。デバイスは、概して、別個に位置し、通信ネットワークを通して、互いに通信する。コンピューティングおよび制御システムは、複数のデバイスが、システム内に存在し、デバイスのうちの少なくとも1つが、他のデバイスのうちの少なくとも1つと異なるハードウェア構成、ソフトウェア構成、オペレーティングシステム、プログラミング方法、開発ツール等を有する場合、異種である。

【0004】

例示的分散型異種コンピューティングおよび制御システムは、その中に、概して、少なくとも1つのサーバコンピュータおよび複数のエッジデバイスが存在する、モノのインターネット（IoT）システムである。少なくとも1つのサーバコンピュータが、概して、強力なハードウェアリソースを有する一方、エッジデバイスの各々は、概して、制約されたハードウェアリソースを有する。エッジデバイスはまた、ハードウェアリソース、ソフトウェア構成、オペレーティングシステム、プログラミング方法等において互いに異なり得る。

【0005】

そのような分散型異種コンピューティングおよび制御システムのためのアプリケーションを設計することは、システムの分散型および異種性質に起因して、困難である。従来のプログラミング方法は、通常、それらが、シングルデバイス上で起動するように意図されるアプリケーションプログラムを生成するので、分散型異種システムにあまり好適ではない。すなわち、従来のプログラミング方法は、異なるハードウェア構成、ソフトウェア構成、オペレーティングシステム等を有する複数のデバイス上で起動するアプリケーションプログラムの生成をサポートしない。

【0006】

例えば、開発チームは、以下の分散型異種システムのためにアプリケーションを実装する必要がある。システムの第1のエッジデバイスが、産業機械の本体に取り付けられる、温度センサを有し、第1のエッジデバイスは、温度をセンサから周期的に読み取る。温度が、閾値を上回る場合、第1のエッジデバイスは、アラートメッセージをサーバコンピュータに送信する。第2のエッジデバイスが、シリアルポートを通して、産業機械と通信し、産業機械の内部レジスタから読み取ることを、またはそれに書き込むことができる。第2のエッジデバイスは、産業機械の内部ステータスレジスタを周期的に読み取る。ステータスが、異常になる場合、第2のエッジデバイスは、アラートメッセージをサーバコンピュータに送信する。第3のエッジデバイスが、ビデオカメラに取り付けられ、それは、産業機械を包囲するエリアの写真またはビデオを撮影することができる。サーバコンピュータが、アラートメッセージを第1のエッジデバイスまたは第2のエッジデバイスのいず

10

20

30

40

50

れかから受信する場合、サーバコンピュータは、第3のエッジデバイスが写真を撮影することを要求する。サーバコンピュータは、アラートメッセージおよび写真の両方をクライアントデバイスに送信する。オペレータが、クライアントデバイス上のアラートメッセージおよび写真をチェックし、オペレータが、状況が緊急であることを決定する場合、オペレータは、コマンドを第2のエッジデバイスに送信し、産業機械をシャットダウンする。

【0007】

上記の例を従来のアプリケーション内に実装するために、少なくとも5つの別個のアプリケーションプログラム、すなわち、第1のエッジデバイスのための1つのプログラム、第2のエッジデバイスのための1つのプログラム、第3のエッジデバイスのための1つのプログラム、およびサーバコンピュータのための2つのプログラムが、実装される必要がある。サーバコンピュータのための第1のプログラムは、第1、第2、および第3のエッジデバイスから生じるデータを処理し得、サーバコンピュータのための第2のプログラムは、クライアントデバイスのためのウェブアプリケーションであり得る。本例示的アプリケーションを従来のプログラミング方法を使用して異なるデバイス上に実装することは、複雑かつインコヒーレントなプロセスである。

【0008】

例えば、アプリケーションプログラムが、互いに通信する分散型異種システムの異なるデバイス上で実行されると、アプリケーションプログラムは、データフォーマット適合性に非常に慎重である必要がある。上記の例において、第1のエッジデバイスのプログラム内の温度は、浮動小数点数として表され得、典型的に、通信インターフェースを通して送信されるために、文字(char)に変換される必要がある。温度を受信するサーバコンピュータ上で実行されるアプリケーションプログラムは、次いで、文字を浮動小数点数に逆変換しなければならない。この変換プロセスは、複数の分散型アプリケーションプログラムが関わる時、非常にエラーを被りやすい。さらに、異なるデバイスは、イーサネット(登録商標)、Wi-Fi、セルラー、Zigbee(登録商標)等の異なるネットワークインターフェースを使用し得る。したがって、各デバイスは、1つ以上のネットワークデバイスドライバをインストールし、1つ以上の通信データフォーマットを使用する必要がある。異なるデバイス上で実行されるアプリケーションプログラムはまた、異なるアプリケーションプログラムを実装し、ネットワークインターフェースに応じて、異なる通信データフォーマット間で伝送されるデータを変換する必要がある。

【0009】

異なるデバイス上で実行されるアプリケーションプログラムはまた、異なるプログラミング言語で実装され得る。上記の例において、第1のエッジデバイスは、Pythonで実装されるアプリケーションプログラムを実行し得、第2のエッジデバイスは、Luaで実装されるアプリケーションプログラムを実行し得、第3のエッジデバイスは、C++で実装されるアプリケーションプログラムを実行し得、サーバコンピュータは、JAVAScript(登録商標)で実装されるアプリケーションプログラムを実行し得る。異なるプログラミング言語は、異なる性能パラメータを最適化し、それは、これらの異なるプログラミング言語を使用する実装を非常に複雑にする。さらに、異なるデバイス上で実行される異なるアプリケーションプログラムは、遠隔で更新される。例えば、アプリケーションプログラムは、概して、開発チームによってアクセス可能な1つ以上のコンピュータ内で開発され、開発されたプログラムは、次いで、オリジナルフォーマットまたはコンパイルされたフォーマットのいずれかにおいて、それぞれのデバイスにプッシュ配信される。より多くのアプリケーションプログラムが、異なるデバイス間に存在するほど、更新プロセスは、より複雑になる。

【0010】

これらの課題に起因して、分散型異種システムのためのアプリケーションプログラムを実装することは、複雑で、冗漫、かつエラーを被りやすいプロセスである。概して、複数の開発ツール、複数の開発プロセス、および異なる技能セットを伴う複数の開発者が、これらのアプリケーションプログラムを実装するために要求され、それは、分散型異種シス

10

20

30

40

50

テムを実装することを高価かつ時間がかかるプロセスにする。

【発明の概要】

【課題を解決するための手段】

【0011】

本発明の一実施形態によると、コンピュータ実装方法は、アプリケーションおよびハードウェアコンテキストの両方を含み、ハードウェアコンテキストにおいて、アプリケーションを構成する、分散型異種コンピューティングおよび制御システムを設計する。本方法は、双方向ディスプレイに関連付けられたコンピュータノードにおいて実行され、コンピュータプロセスを利用して双方向環境を確立する設計ソフトウェア命令によって実装される。コンピュータプロセスは、コンピュータノードによって、双方向環境において、機能モジュールの組およびプリミティブモジュールの組へのアクセスを提供することであって、モジュールのいずれも、ユーザによってグラフィカルに選択されることができる、ことを含む。コンピュータプロセスはさらに、コンピュータノードによって、双方向環境において、デバイスの組の回路図として、分散型コンピューティングおよび制御システムを特徴づける様式において、機能モジュールおよびプリミティブモジュールのうちの所望のものを選択を受信し、それらを順序付けることを含む。コンピュータプロセスはまた、コンピュータノードによって、回路図を解析し、サブ回路図の組を生成することであって、各サブ回路図は、分散されるコンピューティングおよび制御システムにおけるデバイスの組における異なるデバイスに対応することを含む。コンピュータプロセスはさらに、コンピュータノードによって、回路図に従って、デバイスの各々を特徴づけるように、サブ回路図の組をデバイスの組に伝送することを含む。コンピュータプロセスはまた、コンピュータノードによって、デバイスの内側で起動する対応する制御プログラムを介してサブ回路図を実行するように、各デバイスを構成することを含む。

【0012】

随意に、コンピュータノードは、独立型コンピュータである。代替として、または加えて、コンピュータノードは、ネットワーク環境において、クライアントーサーバの組み合わせによって実装される。代替として、または加えて、コンピュータプロセスは、コンピュータノードによって、ネットリストによって回路図を特徴づけることと、コンピュータノードによって、回路図をネットリストとして記憶することとを含む。代替として、または加えて、コンピュータプロセスはさらに、コンピュータノードによって、双方向環境において、記憶されたネットリストを読み出し、処理することによって、回路図の表示をレンダリングすることを含む。代替として、または加えて、双方向環境は、設計キャンバス、ライブラリマネージャ、およびメニューのグラフィカルコンポーネントを含み、グラフィカルコンポーネントのいずれも、ユーザによってグラフィカルに選択されることができる。代替として、または加えて、コンピュータプロセスはさらに、コンピュータノードによって、設計キャンバスにおいて、回路図、対応するネットリスト、およびそれらの組み合わせから成る群から選択されるメンバーを表示することを含む。

【0013】

代替として、または加えて、コンピュータプロセスはさらに、コンピュータノードによって、一意のデバイス識別子（ID）をデバイスの組における各デバイスに割り当てることを含む。代替として、または加えて、コンピュータプロセスはさらに、コンピュータノードによって、デバイス名をデバイスの組における少なくとも1つのデバイスに割り当てることであって、デバイス名は、ユーザによってグラフィカルに記入されることができることを含む。

【0014】

代替として、または加えて、機能モジュールおよびプリミティブモジュールのうちの所望のものを選択を受信することが、所望されるものに対応するモジュールインスタンスの作成をもたらす、分散型コンピューティングおよび制御システムを回路図として特徴づける様式でモジュールインスタンスを順序付ける。代替として、または加えて、モジュールインスタンスを順序付けることは、互いに通信すべき一対のモジュールインスタンスに対

10

20

30

40

50

して、仮想ワイヤを対間に構成し、そのような通信を示すことを含む。代替として、または加えて、コンピュータプロセスはさらに、コンピュータノードによって、所与のモジュールのグラフィカル表現を双方向環境の設計キャンバスにおいて表示される回路図の表示にドラッグし、ドロップする入力を受信することによって、所与のモジュールのモジュールインスタンスが回路図において作成され、順序付けられるようにすることを含む。

【0015】

代替として、または加えて、コンピュータプロセスはさらに、コンピュータノードによって、(i) プロパティテーブルにおける少なくとも1つのプロパティ、(ii) グラフィカル表現、(iii) 少なくとも1つの回路図、(iv) 少なくとも1つのデバイス依存ブロック、および、(v) それらの組み合わせから成る群から選択されるグラフィカルに選択可能なコンポーネントを使用して、所与のモジュールのモジュールインスタンスを特徴づけることを含む。代替として、または加えて、プロパティテーブルは、タイプおよびバージョンを含む、少なくとも1つの回路図を特徴づけるためのプロパティを含む。代替として、または加えて、所与のモジュールが、プリミティブモジュールであるとき、プロパティテーブルは、所与のモジュールに関連付けられた各デバイス依存ブロックのためのプロパティを含み、各デバイス依存ブロックのためのプロパティは、デバイスタイプリストおよびバージョンを含む。代替として、または加えて、プロパティテーブルは、(i) インスタンス名、(ii) 所与のモジュールをインスタンス化するためのインスタンスタイプ、(iii) モジュールインスタンスを実行すべき特定のデバイス、(iv) 少なくとも1つのインスタンスパラメータ、および、(v) それらの組み合わせから成る群から選択されるモジュールインスタンスのためのユーザ選択可能プロパティを含む。代替として、または加えて、特定のデバイスの少なくとも1つのプロパティは、モジュールインスタンスから作成された各子モジュールインスタンスによって継承される。

【0016】

代替として、または加えて、コンピュータプロセスはさらに、コンピュータノードによって、デバイス依存ブロックを所与のプリミティブモジュールのモジュールインスタンスに関連付けることを含み、その中で、コンピュータプロセスは、以下を含む。コンピュータプロセスは、モジュールインスタンスのためのプロパティテーブルにおいて、モジュールインスタンスに対応するデバイスプロパティを見つけることであって、デバイスプロパティは、デバイスの組のうちの1つのためのデバイス識別子(ID)を規定する、ことを含む。コンピュータプロセスはまた、デバイスIDをデバイスタイプにマッピングし、所与のモジュールのためのプロパティテーブルにおいて、デバイスタイプに関連付けられたデバイス依存ブロックを見つけることを含む。コンピュータプロセスはさらに、モジュールインスタンスのためのプロパティテーブルにおいて、モジュールインスタンスに対応するタイププロパティをデバイス依存ブロックに割り当てられた名称で自動的に更新することを含む。随意に、タイププロパティを自動的に更新することは、タイププロパティのユーザ選択によって上書きされる。

【0017】

代替として、または加えて、コンピュータプロセスはさらに、コンピュータノードによって、双方向環境において、回路図を特徴づけるモジュールインスタンスの階層構造を含むツリーグラフと、ツリーグラフの下方に位置付けられ、ツリーグラフにおける対応するプリミティブモジュールインスタンスにリンクされている、正しく選定されたデバイス依存ブロックのグラフィカル表現とを表示することを含む。代替として、または加えて、サブ回路図の組をデバイスの組に伝送することは、プリミティブモジュールインスタンスを有する各サブ回路図に関して、デバイス依存ブロックをサブ回路図に対応する異なるデバイスに伝送することを含む。代替として、または加えて、回路図は、モジュールインスタンスを含むコンポーネントを有し、コンピュータプロセスはさらに、コンピュータノードによって、回路図の所与のコンポーネントに、(i) デバイスのデバイスID、(ii) 所与のコンポーネントに到達するために回路図内で順にトラバースされるべき各コンポーネントの名称を定義するパス、および、(iii) 所与のコンポーネントの名称の組み合

10

20

30

40

50

わせに基づいて、異なるアドレスを割り当てることを含む。代替として、または加えて、回路図は、モジュールインスタンスを含むコンポーネントを有し、コンピュータプロセスはさらに、コンピュータノードによって、所与のモジュールインスタンスに、(i) デバイスのデバイスID、および、(ii) 所与のモジュールインスタンスのIDの組み合わせに基づいて、異なるアドレスを割り当てることと、コンピュータノードによって、モジュールインスタンスではない所与のコンポーネントに、(i) デバイスのデバイスID、(ii) 所与のコンポーネントを含むモジュールインスタンスのID、および、(iii) 所与のコンポーネントの名称の組み合わせによって定義される一意のアドレスを割り当てることを含む。

【0018】

代替として、または加えて、回路図を解析することは、以下を含む。デバイスの組における各デバイスのために、コンピュータプロセスはさらに、デバイスのためのサブ回路図を作成することを含む。回路図の各モジュールインスタンスのために、コンピュータプロセスはさらに、モジュールインスタンスを、サブ回路図のデバイスがモジュールインスタンスのデバイスIDに対応するサブ回路図の中にコピーすることを含む。コンピュータプロセスはさらに、回路図内において第1のモジュールインスタンスと第2のモジュールインスタンスとを接続する仮想ワイヤを処理することを含み、それは、以下を含む。第1のモジュールインスタンスおよび第2のモジュールインスタンスの両方が、所与のサブ回路図の中にあるとき、コンピュータプロセスはさらに、仮想ワイヤを所与のサブ回路図において保存することを含む。仮想ワイヤが、(i) 第1のサブ回路図内の第1のモジュールインスタンスの出力ピンを(ii) 第2のサブ回路図内の第2のモジュールインスタンスの入力ピンに接続するとき、コンピュータプロセスはさらに、第1のサブ回路図と第2のサブ回路図との間のcomm接続を作成することを含む。

【0019】

代替として、または加えて、comm接続を作成することは、以下を含む。コンピュータプロセスはさらに、第1のcommモジュールインスタンスを第1のサブ回路図に追加し、仮想ワイヤを第1のモジュールインスタンスの出力ピンと第1のcommモジュールインスタンスの入力ピンとの間に追加することを含む。コンピュータプロセスはさらに、第2のcommモジュールインスタンスを第2のサブ回路図に追加し、仮想ワイヤを第2のcommモジュールインスタンスの出力ピンと第2のモジュールインスタンスの入力ピンとの間に追加することを含む。コンピュータプロセスはさらに、第1のcommモジュールインスタンスに含まれる宛先アドレスを第2のモジュールインスタンスの入力ピンのアドレスに設定することを含む。コンピュータプロセスはさらに、第2のcommモジュール内に含まれる受信アドレスを第2のモジュールインスタンスの入力ピンのアドレスに設定することを含む。代替として、または加えて、コンピュータプロセスはさらに、コンピュータノードによって、宛先アドレスおよび受信アドレスの各々に、(i) デバイスのデバイスID、(ii) 標的化されたピンに到達するために回路図内で順にトラバースされるべき各コンポーネントの名称を定義するパス、および、(iii) 標的化されたピンの名称の組み合わせに基づいて、異なるアドレスを割り当てることを含む。代替として、または加えて、コンピュータプロセスはさらに、コンピュータノードによって、宛先アドレスおよび受信アドレスの各々に、(i) デバイスのデバイスID、(ii) 標的化されたピンを含むモジュールインスタンスのID、および、(iii) 標的化されたピンの名称の組み合わせに基づいて、異なるアドレスを割り当てることを含む。代替として、または加えて、commモジュールは、複数の機能回路図に関連付けられ、各機能回路図は、異なる通信プロトコルを実装し、双方向環境は、(i) 所与のcommモジュールインスタンスのためのcommモジュールインスタンスタイプを個々に変更するユーザ選択と、(ii) 各commモジュールインスタンスのためのcommモジュールインスタンスタイプを大域的に変更するユーザ選択の両方のために構成される。

【0020】

代替として、または加えて、コンピュータプロセスはさらに、コンピュータノードによ

10

20

30

40

50

って、第1のc o m mモジュールインスタンスと第2のc o m mモジュールインスタンスとの間の通信のためにパブリッシュサブスクライブ通信プロトコルを利用する、中央c o m mブローカを確立することを含む。パブリッシュサブスクライブ通信プロトコルは、中央c o m mブローカを用いて、第2のc o m mモジュールインスタンスをサブスクライブするための要求を受信するように構成され、要求は、第2のc o m mモジュールインスタンスのための受信アドレスを規定することを含み、受信アドレスは、第2のc o m mモジュールに接続されるモジュールインスタンスの入力ピンのアドレスである。パブリッシュサブスクライブ通信プロトコルはさらに、データパケットを第1のc o m mモジュールインスタンスから中央c o m mブローカに伝送するように構成され、データパケットは、データペイロード、宛先アドレス、ソースアドレス、およびそれらの組み合わせから成る群から選択されるデータを含み、ソースアドレスは、第1のc o m mモジュールインスタンスに接続される、モジュールインスタンスの出力ピンのアドレスであって、宛先アドレスは、意図される宛先モジュールインスタンスの入力ピンのアドレスである。データパケットにおける宛先アドレスが、第2のc o m mモジュールインスタンスの受信アドレスに合致しているとき、パブリッシュサブスクライブ通信プロトコルはまた、c o m mブローカに、データパケットを第2のc o m mモジュールインスタンスに転送させるように構成される。

#### 【0021】

代替として、または加えて、サブ回路図は、伝送パスおよび受信パスを含むc o m mモジュールインスタンスが、サブ回路図の複数のモジュールインスタンス間で共有されるように構成され、コンピュータプロセスはさらに、コンピュータノードによって、デバイスの組のデバイスにおけるc o m mモジュールインスタンスに、デバイスの識別を含む宛先アドレスを伴う所与のデータパケットを伝送することを含む。c o m mモジュールインスタンスは、受信パスを通して、所与のデータパケットを含む、デバイスの識別を含む宛先アドレスを有する任意のデータパケットを受信するように構成される。c o m mモジュールインスタンスはさらに、所与のデータパケットにおける宛先アドレスを分析し、所与のデータパケットにおける宛先アドレスを分析し、所与のデータパケットを、宛先アドレスに対応する受信アドレスを伴うサブ回路図のモジュールインスタンスに転送するように構成される。

#### 【0022】

代替として、または加えて、コンピュータプロセスはさらに、コンピュータノードによって、デバイスの組のデバイスにデバイス上にロードされた新しい回路図を実行させることを含み、デバイスは、制御プログラムと、s y sモジュールインスタンスに接続されるc o m mモジュールインスタンスを有するi n i t回路図とを事前にロードされ、c o m mモジュールインスタンスの受信アドレスは、以下のように、宛先アドレスが、デバイスの識別およびs y sモジュールを含む任意のデータパケットを受信するように設定される。コンピュータプロセスは、コンピュータノードによって、デバイスに、新しいサブ回路図と、デバイスの識別およびs y sモジュールを含む宛先アドレスとを含むデータペイロードを有するデータパケットを伝送することであって、宛先アドレスは、s y sモジュールインスタンスに、( i ) データパケットを受信させ、( i i ) デバイス上に、新しいサブ回路図をロードさせ、( i i i ) 制御プログラムに、新しいサブ回路図を実行するように要求させる、ことを含む。

#### 【0023】

代替として、または加えて、コンピュータプロセスはさらに、コンピュータノードによって、デバイスに、回路図を実行させることを含み、デバイスは、制御プログラムおよびi n i t回路図を事前にロードするように構成される。デバイスの電源が、オンになると、デバイスはさらに、制御プログラムによってi n i t回路図の実行を生じさせるように構成される。デバイスはまた、i n i t回路図を使用して、新しいサブ回路図を受信およびロードするように構成される。デバイスはさらに、制御プログラムによって新しいサブ回路図の実行を生じさせる、再ロードコマンドを受信するように構成される。代替として

、または加えて、サブ回路図における各プリミティブモジュールインスタンスに関して、デバイスにおいて、プリミティブモジュールインスタンスに関連付けられたデバイス依存ブロックを実行し、デバイス依存ブロックは、関連付けられたプリミティブモジュールインスタンスのタイプモジュールインスタンスプロパティを使用して、制御プログラムによってロードされる。

【0024】

代替として、または加えて、コンピュータプロセスはさらに、コンピュータノードによって、サブ回路図をバイナリコードに変換するコンパイラを実行することと、コンピュータノードによって、回路図に対応する低能力デバイスに、デバイス上での実行のためのバイナリコードを伝送することを含む。代替として、または加えて、コンピュータプロセスはさらに、以下を含む。コンピュータプロセスはさらに、コンピュータノードによって、双方向環境において、可視化モジュールインスタンスの選択を受信し、さらに、分散型コンピューティングおよび制御システムを回路図として特徴づけることを含む。コンピュータプロセスはさらに、コンピュータノードによって、回路図内の可視化モジュールインスタンスのデバイスプロパティをシミュレーションコンテキストを示す値に設定することを含む。コンピュータプロセスはさらに、コンピュータノードによって、仮想デバイスのためのサブ回路図を生成することを含む、回路図を解析し、サブ回路図の組を生成することを含む。コンピュータプロセスはさらに、コンピュータノードによって、仮想デバイス上で、シミュレーションコンテキストにおいて、サブ回路図を実行することを含む。代替として、または加えて、可視化モジュールインスタンスのデバイスプロパティは、少なくとも1つのデバイスのIDに設定され、コンピュータプロセスはさらに、可視化モジュールを実行のために少なくとも1つのデバイスに伝送させることを含む。

【0025】

代替として、または加えて、コンピュータプロセスはさらに、以下を含む。コンピュータプロセスはさらに、コンピュータノードによって、双方向環境において、シミュレーションメニューオプションの選択を受信することを含む。回路図における各モジュールインスタンスに関して、コンピュータプロセスはさらに、コンピュータノードによって、モジュールインスタンスのデバイスプロパティをシミュレーションコンテキストにおけるモジュールインスタンスの実行を示す値に設定することを含む。回路図における各モジュールインスタンスに関して、コンピュータプロセスはさらに、デフォルト挙動回路図が、モジュールインスタンスに関連付けられているとき、モジュールインスタンスのタイププロパティをデフォルト挙動回路図を示すコンポーネント名に設定することを含む。コンピュータプロセスはさらに、回路図を解析し、仮想デバイスのための1つのみのサブ回路図を生成することを含む。コンピュータプロセスはさらに、コンピュータノードによって、仮想デバイスを通して、シミュレーションコンテキストにおいて、サブ回路図の各モジュールインスタンスを実行することを含む。代替として、または加えて、モジュールインスタンスのタイププロパティは、デフォルト挙動回路図のコンポーネント名と異なる値にユーザ選択可能である。

本発明は、例えば、以下を提供する。

(項目1)

分散型異種コンピューティングおよび制御システムを設計するコンピュータ実装方法であって、前記システムは、アプリケーションおよびハードウェアコンテキストの両方を含み、前記ハードウェアコンテキストにおいて前記アプリケーションを構成し、前記方法は、双方向ディスプレイに関連付けられたコンピュータノードにおいて実行する設計ソフトウェア命令によって実装され、コンピュータプロセスを利用して双方向環境を確立し、前記コンピュータプロセスは、

前記コンピュータノードによって、前記双方向環境において、機能モジュールの組およびプリミティブモジュールの組へのアクセスを提供することであって、前記モジュールのいずれも、ユーザによってグラフィカルに選択されることができる、ことと、

前記コンピュータノードによって、前記双方向環境において、前記機能モジュールおよ

10

20

30

40

50

び前記プリミティブモジュールのうちの所望のものの選択を受信し、デバイスの組の回路図として、前記分散型コンピューティングおよび制御システムを特徴づける様式でそれらを順序付けることと、

前記コンピュータノードによって、前記回路図を解析し、サブ回路図の組を生成することであって、各サブ回路図は、前記分散型コンピューティングおよび制御システムにおけるデバイスの組における異なるデバイスに対応する、ことと、

前記コンピュータノードによって、前記回路図に従って、前記デバイスの各々を特徴づけるために、前記サブ回路図の組を前記デバイスの組に伝送することと、

前記コンピュータノードによって、前記デバイスの内側で起動する対応する制御プログラムを介して、前記サブ回路図を実行するように各デバイスを構成することと

を含む、コンピュータ実装方法。

(項目 2)

前記コンピュータノードは、独立型コンピュータである、項目 1 に記載のコンピュータ実装方法。

(項目 3)

前記コンピュータノードは、ネットワーク環境において、クライアントーサーバの組み合わせによって実装される、項目 1 に記載のコンピュータ実装方法。

(項目 4)

前記コンピュータプロセスは、

前記コンピュータノードによって、ネットリストによって前記回路図を特徴づけることと、

前記コンピュータノードによって、前記回路図を前記ネットリストとして記憶することとをさらに含む、項目 1 に記載のコンピュータ実装方法。

(項目 5)

前記コンピュータプロセスは、

前記コンピュータノードによって、前記双方向環境において、記憶されたネットリストを読み出し、処理することによって、前記回路図の表示をレンダリングすることをさらに含む、項目 1 に記載のコンピュータ実装方法。

(項目 6)

前記双方向環境は、設計キャンバス、ライブラリマネージャ、およびメニューのグラフィカルコンポーネントを含み、前記グラフィカルコンポーネントのいずれも、ユーザによってグラフィカルに選択されることができる、項目 1 に記載のコンピュータ実装方法。

(項目 7)

前記コンピュータプロセスは、

前記コンピュータノードによって、前記設計キャンバスにおいて、前記回路図、対応するネットリスト、およびそれらの組み合わせから成る群から選択されるメンバーを表示することをさらに含む、項目 6 に記載のコンピュータ実装方法。

(項目 8)

前記コンピュータプロセスは、

前記コンピュータノードによって、一意のデバイス識別子 (ID) を前記デバイスの組における各デバイスに割り当てることをさらに含む、項目 1 に記載のコンピュータ実装方法。

(項目 9)

前記コンピュータプロセスは、

前記コンピュータノードによって、デバイス名を前記デバイスの組における少なくとも 1 つのデバイスに割り当てることをさらに含み、前記デバイス名は、ユーザによってグラフィカルに記入されることができる、項目 1 に記載のコンピュータ実装方法。

(項目 10)

前記機能モジュールおよび前記プリミティブモジュールのうちの所望のものの前記選択を受信することは、前記所望のものに対応するモジュールインスタンスの作成をもたらす

10

20

30

40

50

、前記分散型コンピューティングおよび制御システムを前記回路図として特徴づける様式で前記モジュールインスタンスを順序付ける、項目 1 に記載のコンピュータ実装方法。

(項目 1 1)

前記モジュールインスタンスを順序付けることは、互いに通信すべき一対の前記モジュールインスタンスに対して、仮想ワイヤを前記対間に構成し、そのような通信を示すことを含む、項目 1 0 に記載のコンピュータ実装方法。

(項目 1 2)

前記コンピュータプロセスは、

前記コンピュータノードによって、所与のモジュールのグラフィカル表現を前記双方向環境の設計キャンバスにおいて表示された前記回路図の表示にドラッグし、ドロップする入力を受信することに続いて、前記所与のモジュールのモジュールインスタンスが前記回路図において作成され、順序付けられるようにすることをさらに含む、項目 1 に記載のコンピュータ実装方法。

(項目 1 3)

前記コンピュータプロセスは、

前記コンピュータノードによって、(i) プロパティテーブルにおける少なくとも 1 つのプロパティ、(i i) グラフィカル表現、(i i i) 少なくとも 1 つの回路図、(i v) 少なくとも 1 つのデバイス依存ブロック、および、(i v) それらの組み合わせから成る群から選択されるグラフィカルに選択可能なコンポーネントを使用して、所与のモジュールのモジュールインスタンスを特徴づけることをさらに含む、項目 1 に記載のコンピュータ実装方法。

(項目 1 4)

前記プロパティテーブルは、前記少なくとも 1 つの回路図を特徴づけるためのプロパティを含み、前記プロパティは、タイプおよびバージョンを含む、項目 1 3 に記載のコンピュータ実装方法。

(項目 1 5)

前記所与のモジュールが、プリミティブモジュールであるとき、前記プロパティテーブルは、前記所与のモジュールに関連付けられた各デバイス依存ブロックのためのプロパティを含み、各デバイス依存ブロックのための前記プロパティは、デバイスタイプリストおよびバージョンを含む、項目 1 3 に記載のコンピュータ実装方法。

(項目 1 6)

前記プロパティテーブルは、(i) インスタンス名、(i i) 前記所与のモジュールをインスタンス化するためのインスタンスタイプ、(i i i) 前記モジュールインスタンスを実行すべき特定のデバイス、(i v) 少なくとも 1 つのインスタンスパラメータ、および、(v) それらの組み合わせから成る群から選択される前記モジュールインスタンスのためのユーザ選択可能プロパティを含む、項目 1 3 に記載のコンピュータ実装方法。

(項目 1 7)

前記特定のデバイスの少なくとも 1 つのプロパティは、前記モジュールインスタンスから作成された各子モジュールインスタンスによって継承される、項目 1 6 に記載のコンピュータ実装方法。

(項目 1 8)

前記コンピュータプロセスは、

前記コンピュータノードによって、デバイス依存ブロックを所与のプリミティブモジュールのモジュールインスタンスに関連付けることをさらに含み、前記関連付けることは、前記モジュールインスタンスのためのプロパティテーブルにおいて、前記モジュールインスタンスに対応するデバイスプロパティを見つけることであって、前記デバイスプロパティは、前記デバイスの組のうちの 1 つのためのデバイス識別子 (ID) を規定する、ことと、

前記デバイス ID をデバイスタイプにマッピングし、前記所与のモジュールのための前記プロパティテーブルにおいて、前記デバイスタイプに関連付けられた前記デバイス依存

10

20

30

40

50

ブロックを見つけることと、

前記モジュールインスタンスのための前記プロパティテーブルにおいて、前記モジュールインスタンスに対応するタイププロパティを前記デバイス依存ブロックに割り当てられた名称で自動的に更新することと

を含む、項目 1 に記載のコンピュータ実装方法。

(項目 19)

前記タイププロパティを自動的に更新することは、前記タイププロパティのユーザ選択によって上書きされる、項目 18 に記載のコンピュータ実装方法。

(項目 20)

前記コンピュータプロセスは、

前記コンピュータノードによって、前記双方向環境において、

前記回路図を特徴づけるモジュールインスタンスの階層構造を含むツリーグラフと、

正しく選定されたデバイス依存ブロックのグラフィカル表現と

を表示することをさらに含む、

前記正しく選定されたデバイス依存ブロックのグラフィカル表現は、前記ツリーグラフの下方に位置付けられ、前記ツリーグラフにおける対応するプリミティブモジュールインスタンスにリンクされている、項目 1 に記載のコンピュータ実装方法。

(項目 21)

前記サブ回路図の組を前記デバイスの組に伝送することは、プリミティブモジュールインスタンスを有する各サブ回路図に関して、デバイス依存ブロックを前記サブ回路図に対応する異なるデバイスに伝送することを含む、項目 1 に記載のコンピュータ実装方法。

(項目 22)

前記回路図は、モジュールインスタンスを含むコンポーネントを有し、前記コンピュータプロセスは、

前記コンピュータノードによって、前記回路図の所与のコンポーネントに、(i) デバイスのデバイス ID、(ii) 前記所与のコンポーネントに到達するために前記回路図内で順にトラバースされるべき各コンポーネントの名称を定義するパス、および、(iii) 前記所与のコンポーネントの名称の組み合わせに基づいて、異なるアドレスを割り当てることをさらに含む、項目 1 に記載のコンピュータ実装方法。

(項目 23)

前記回路図は、モジュールインスタンスを含むコンポーネントを有し、前記コンピュータプロセスは、

前記コンピュータノードによって、所与のモジュールインスタンスに、(i) デバイスのデバイス ID、および、(ii) 前記所与のモジュールインスタンスの ID の組み合わせに基づいて、異なるアドレスを割り当てることと、

前記コンピュータノードによって、モジュールインスタンスではない所与のコンポーネントに、(i) デバイスのデバイス ID、(ii) 前記所与のコンポーネントを含むモジュールインスタンスの ID、および、(iii) 前記所与のコンポーネントの名称の組み合わせによって定義される一意のアドレスを割り当てることと

をさらに含む、項目 1 に記載のコンピュータ実装方法。

(項目 24)

前記回路図を解析することは、

前記デバイスの組における各デバイスのために、前記デバイスのためのサブ回路図を作成することと、

前記回路図の各モジュールインスタンスのために、前記モジュールインスタンスをサブ回路図の中にコピーすることであって、前記サブ回路図において、前記サブ回路図の前記デバイスは、前記モジュールインスタンスのデバイス ID に対応している、ことと、

前記回路図における第 1 のモジュールインスタンスと第 2 のモジュールインスタンスとを接続する仮想ワイヤを処理することと

を含み、

10

20

30

40

50

前記仮想ワイヤを処理することは、

前記第 1 のモジュールインスタンスおよび前記第 2 のモジュールインスタンスの両方が所与のサブ回路図の中にあるとき、前記仮想ワイヤを前記所与のサブ回路図において保存することと、

前記仮想ワイヤが、( i ) 第 1 のサブ回路図の中の前記第 1 のモジュールインスタンスの出力ピンを ( i i ) 第 2 のサブ回路図の中の前記第 2 のモジュールインスタンスの入力ピンに接続するとき、前記第 1 のサブ回路図と前記第 2 のサブ回路図との間の c o m m 接続を作成することと

を含む、項目 1 に記載のコンピュータ実装方法。

(項目 2 5)

前記 c o m m 接続を作成することは、

第 1 の c o m m モジュールインスタンスを前記第 1 のサブ回路図に追加し、仮想ワイヤを前記第 1 のモジュールインスタンスの前記出力ピンと前記第 1 の c o m m モジュールインスタンスの入力ピンとの間に追加することと、

第 2 の c o m m モジュールインスタンスを前記第 2 のサブ回路図に追加し、仮想ワイヤを前記第 2 の c o m m モジュールインスタンスの出力ピンと前記第 2 のモジュールインスタンスの前記入力ピンとの間に追加することと、

前記第 1 の c o m m モジュールインスタンスに含まれる宛先アドレスを前記第 2 のモジュールインスタンスの前記入力ピンのアドレスに設定することと、

前記第 2 の c o m m モジュール内に含まれる受信アドレスを前記第 2 のモジュールインスタンスの前記入力ピンのアドレスに設定することと

を含む、項目 2 4 に記載のコンピュータ実装方法。

(項目 2 6)

前記コンピュータプロセスは、

前記コンピュータノードによって、前記宛先アドレスおよび受信アドレスの各々に、( i ) デバイスのデバイス ID、( i i ) 標的化されたピンに到達するために前記回路図内で順にトラバースされるべき各コンポーネントの名称を定義するパス、および、( i i i ) 前記標的化されたピンの名称の組み合わせに基づいて、異なるアドレスを割り当てることをさらに含む、項目 2 5 に記載のコンピュータ実装方法。

(項目 2 7)

前記コンピュータプロセスは、

前記コンピュータノードによって、前記宛先アドレスおよび受信アドレスの各々に、( i ) デバイスのデバイス ID、( i i ) 標的化されたピンを含むモジュールインスタンスの ID、および、( i i i ) 前記標的化されたピンの名称の組み合わせに基づいて、異なるアドレスを割り当てることをさらに含む、項目 2 5 に記載のコンピュータ実装方法。

(項目 2 8)

前記 c o m m モジュールは、複数の機能回路図に関連付けられており、各機能回路図は、異なる通信プロトコルを実装し、前記双方向環境は、( i ) 所与の c o m m モジュールインスタンスのための c o m m モジュールインスタンスタイプを個々に変更するユーザ選択と、( i i ) 各 c o m m モジュールインスタンスのための前記 c o m m モジュールインスタンスタイプを大域的に変更するユーザ選択との両方のために構成されている、項目 2 5 に記載のコンピュータ実装方法。

(項目 2 9)

前記コンピュータプロセスは、前記コンピュータノードによって、中央 c o m m ブローカを確立することをさらに含み、前記中央 c o m m ブローカは、第 1 の c o m m モジュールインスタンスと第 2 の c o m m モジュールインスタンスとの間の通信のためにパブリッシュ-サブスクライブ通信プロトコルを利用し、前記パブリッシュ-サブスクライブ通信プロトコルは、

第 2 の c o m m モジュールインスタンスを前記中央 c o m m ブローカにサブスクライブするための要求を受信することであって、前記要求は、前記第 2 の c o m m モジュールイ

10

20

30

40

50

インスタンスの受信アドレスを規定することを含み、前記受信アドレスは、前記第2のc o m mモジュールに接続されたモジュールインスタンスの入力ピンのアドレスである、ことと、

データパケットを第1のc o m mモジュールインスタンスから前記中央c o m mブローカに伝送することであって、前記データパケットは、データペイロード、宛先アドレス、ソースアドレス、およびそれらの組み合わせから成る群から選択されるデータを含み、前記ソースアドレスは、前記第1のc o m mモジュールインスタンスに接続されたモジュールインスタンスの出力ピンのアドレスであり、前記宛先アドレスは、前記意図される宛先モジュールインスタンスの前記入力ピンのアドレスである、ことと、

前記データパケットにおける宛先アドレスが前記第2のc o m mモジュールインスタンスの前記受信アドレスに合致しているとき、前記c o m mブローカに前記データパケットを前記第2のc o m mモジュールインスタンスに転送させることと

を行うように構成されている、項目1に記載のコンピュータ実装方法。

(項目30)

前記サブ回路図は、伝送パスおよび受信パスを含むc o m mモジュールインスタンスが前記サブ回路図の複数のモジュールインスタンス間で供給されるように構成され、前記コンピュータプロセスは、

前記コンピュータノードによって、前記デバイスの組のデバイスにおけるc o m mモジュールインスタンスに前記デバイスの識別を含む宛先アドレスを伴う所与のデータパケットを伝送することをさらに含み、前記c o m mモジュールインスタンスは、

前記受信パスを通して、前記デバイスの識別を含む宛先アドレスを有する任意のデータパケットを受信することであって、前記任意のデータパケットは、前記所与のデータパケットを含む、ことと、

前記所与のデータパケットにおける前記宛先アドレスを分析し、前記所与のデータパケットを前記宛先アドレスに対応する受信アドレスを伴う前記サブ回路図のモジュールインスタンスに転送することと

を行うように構成されている、項目1に記載のコンピュータ実装方法。

(項目31)

前記コンピュータプロセスは、

前記コンピュータノードによって、前記デバイスの組のデバイスに前記デバイス上にロードされた新しい回路図を実行させることをさらに含み、前記デバイスは、制御プログラムと、s y sモジュールインスタンスに接続されたc o m mモジュールインスタンスを有するi n i t回路図とを事前にロードされ、前記c o m mモジュールインスタンスの受信アドレスは、前記宛先アドレスが前記デバイスの識別および前記s y sモジュールを含む任意のデータパケットを受信するように設定され、

前記コンピュータプロセスは、

前記コンピュータノードによって、前記デバイスに、新しいサブ回路図と、前記デバイスの識別および前記s y sモジュールを含む宛先アドレスとを含むデータペイロードを有するデータパケットを伝送することを含み、

前記宛先アドレスは、(i)前記データパケットを受信することと、(i i)前記デバイス上に前記新しいサブ回路図をロードすることと、(i i i)前記制御プログラムに前記新しいサブ回路図を実行するように要求することとを前記s y sモジュールインスタンスに行わせる、項目1に記載のコンピュータ実装方法。

(項目32)

前記コンピュータプロセスは、

前記コンピュータノードによって、デバイスに回路図を実行させることをさらに含み、前記デバイスは、

制御プログラムおよびi n i t回路図を事前にロードすることと、

前記デバイスの電源がオンにされると、前記制御プログラムによる前記i n i t回路図の実行を引き起こすことと、

10

20

30

40

50

前記init回路図を使用して、新しいサブ回路図を受信し、ロードすることと、  
前記制御プログラムによる前記新しいサブ回路図の実行を引き起こす再ロードコマンド  
を受信することと

を行うように構成されている、項目1に記載のコンピュータ実装方法。

(項目33)

サブ回路図における各プリミティブモジュールインスタンスに関して、前記デバイスに  
おいて、前記プリミティブモジュールインスタンスに関連付けられたデバイス依存ブロッ  
クを実行し、前記デバイス依存ブロックは、前記関連付けられたプリミティブモジュール  
インスタンスの前記タイプモジュールインスタンスプロパティを使用して、前記制御プロ  
グラムによってロードされる、項目32に記載のコンピュータ実装方法。

(項目34)

前記コンピュータプロセスは、

前記コンピュータノードによって、サブ回路図をバイナリコードに変換するコンパイラ  
を実行することと、

前記コンピュータノードによって、前記回路図に対応する低能力デバイスに前記デバイ  
ス上での実行のためのバイナリコードを伝送することと

をさらに含む、項目1に記載のコンピュータ実装方法。

(項目35)

前記コンピュータプロセスは、

前記コンピュータノードによって、前記双方向環境において、前記分散型コンピューテ  
ィングおよび制御システムを前記回路図としてさらに特徴づけるための可視化モジュール  
インスタンスの選択を受信することと、

前記コンピュータノードによって、前記回路図における前記可視化モジュールインスタ  
ンスのデバイスプロパティをシミュレーションコンテキストを示す値に設定することと、

前記コンピュータノードによって、前記回路図を解析し、前記サブ回路図の組を生成す  
ることとあって、前記サブ回路図の組を生成することは、仮想デバイスのためのサブ回路  
図を生成することを含む、ことと、

前記コンピュータノードによって、前記仮想デバイスを通して、前記シミュレーション  
コンテキストにおいて、前記サブ回路図を実行することと

をさらに含む、項目1に記載のコンピュータ実装方法。

(項目36)

前記可視化モジュールインスタンスの前記デバイスプロパティは、少なくとも1つのデ  
バイスのIDに設定され、前記コンピュータプロセスは、前記可視化モジュールが実行の  
ために前記少なくとも1つのデバイスに伝送されるようにすることをさらに含む、項目3  
5に記載のコンピュータ実装方法。

(項目37)

前記コンピュータプロセスは、

前記コンピュータノードによって、前記双方向環境において、シミュレーションメニ  
ューオプションの選択を受信することと、

前記回路図における各モジュールインスタンスに関して、

前記コンピュータノードによって、前記モジュールインスタンスのデバイスプロパティ  
をシミュレーションコンテキストにおける前記モジュールインスタンスの実行を示す値に  
設定することと、

デフォルト挙動回路図が前記モジュールインスタンスに関連付けられているとき、前記  
モジュールインスタンスのタイププロパティをデフォルト挙動回路図を示すコンポーネン  
ト名に設定することと、

前記回路図を解析し、仮想デバイスのための1つのみのサブ回路図を生成することと、

前記コンピュータノードによって、前記仮想デバイスを通して、前記シミュレーション  
コンテキストにおいて、前記サブ回路図の各モジュールインスタンスを実行することと

をさらに含む、項目1に記載のコンピュータ実装方法。

10

20

30

40

50

(項目 3 8)

前記モジュールインスタンスのタイププロパティは、前記デフォルト挙動回路図の前記コンポーネント名と異なる値にユーザ選択可能である、項目 3 7 に記載のコンピュータ実装方法。

【図面の簡単な説明】

【0026】

前述の実施形態の特徴は、付随の図面の参照と併せて、以下の詳細な説明を参照することによって、より容易に理解されるであろう。

【0027】

【図 1】図 1 は、本発明のある実施形態による、分散型異種コンピューティングおよび制御のためのシステムのブロック図であり、そのためにアプリケーションが設計される。

10

【0028】

【図 2】図 2 は、従来技術における分散型異種コンピューティングおよび制御システムのためのアプリケーションを設計する方法を図示するフローチャートである。

【0029】

【図 3】図 3 は、本発明のある実施形態による、分散型異種コンピューティングおよび制御システムのためのアプリケーションを設計する方法を図示するフローチャートである。

【0030】

【図 4】図 4 は、本発明のある実施形態による、設計ソフトウェアを実行するために使用されるシステムのブロック図である。

20

【0031】

【図 5】図 5 は、本発明のある実施形態による、分散型異種コンピューティングおよび制御システムのためのアプリケーションを設計するために使用されるグラフィカルユーザインターフェースである。

【0032】

【図 6】図 6 は、本発明のある実施形態による、図 5 のグラフィカルユーザインターフェース内の回路図および対応するネットリストのディスプレイを図示する。

【0033】

【図 7】図 7 は、本発明のある実施形態による、モジュールコンポーネントのブロック図である。

30

【0034】

【図 8】図 8 は、本発明のある実施形態による、モジュールプロパティを定義するプロパティテーブルのブロック図である。

【0035】

【図 9】図 9 は、本発明のある実施形態による、異なるタイプのデバイスにおける回路図の実行を図示するブロック図である。

【0036】

【図 10】図 10 は、本発明のある実施形態による、モジュールプロパティを定義する別のプロパティテーブルのブロック図である。

【0037】

40

【図 11】図 11 は、本発明のある実施形態による、適合性があるデバイスにおけるデバイス依存ブロックの実行を図示するブロック図である。

【0038】

【図 12】図 12 は、本発明のある実施形態による、モジュールのグラフィカル表現のブロック図である。

【0039】

【図 13】図 13 は、本発明のある実施形態による、回路図のブロック図である。

【0040】

【図 14】図 14 は、本発明のある実施形態による、モジュールインスタンスのためのプロパティを定義するプロパティテーブルを提示するグラフィカルユーザインターフェース

50

である。

【0041】

【図15】図15は、本発明のある実施形態による、回路図の階層構造を表すツリーグラフである。

【0042】

【図16】図16は、本発明のある実施形態による、回路図を利用して、分散型異種コンピューティングおよび制御システムのためのアプリケーションを設計する方法35を図示するフローチャートである。

【0043】

【図17】図17は、本発明のある実施形態による、図13の回路図から解析されたサブ回路図のブロック図である。

10

【0044】

【図18】図18は、本発明のある実施形態による、回路図をサブ回路図に解析する方法を図示するフローチャートである。

【0045】

【図19】図19は、本発明のある実施形態による、回路図のコンポーネントをアドレスするための論理フローを図示するブロック図である。

【0046】

【図20】図20は、本発明のある実施形態による、回路図をサブ回路図に解析するとき使用されるアドレスの方法を図示するブロック図である。

20

【0047】

【図21】図21は、本発明のある実施形態による、commモジュールのコンポーネントのブロック図である。

【0048】

【図22】図22は、本発明のある実施形態による、パブリッシュ／サブスクライブプロトコルを使用してデータを通信する方法を図示するフローチャートである。

【0049】

【図23】図23は、本発明のある実施形態による、モジュールインスタンス間の仮想ワイヤを置換する論理フローを図示するブロック図である。

【0050】

30

【図24】図24は、本発明のある実施形態による、デバイスにおける実行のためのサブ回路図を更新する方法を図示するフローチャートである。

【0051】

【図25】図25は、本発明のある実施形態による、init回路図のブロック図である。

【0052】

【図26】図26は、本発明のある実施形態による、インスタンス化されたsysモジュールを図示するブロック図である。

【0053】

【図27】図27は、本発明のある実施形態による、デバイスにおいて制御プログラムを実行する方法を図示するフローチャートである。

40

【0054】

【図28】図28は、本発明のある実施形態による、低能力デバイスのための実行可能コードを提供する方法を図示するフローチャートである。

【0055】

【図29】図29は、本発明のある実施形態による、分散型異種システムのためのデータ可視化を提供するモジュールインスタンスを含む回路図のブロック図である。

【0056】

【図30】図30は、本発明のある実施形態による、データ可視化を実施するためのサブ回路図のブロック図である。

【0057】

50

【図 3 1】図 3 1 は、本発明のある実施形態による、仮想デバイスを通したシミュレーションコンテキストにおける実行のための回路図を更新する方法を図示するフローチャートである。

【発明を実施するための形態】

【0058】

定義。本説明および付随の請求項において使用されるように、以下の用語は、文脈によって別様に要求されない限り、示される意味を有するものとする。

「セット」は、少なくとも 1 つのメンバーを含む。

システムは、物理的に異なる場所に、ネットワークを経由して互いに通信する、複数のノードを含む場合、「分散型」である。

システムは、システム内の 1 つ以上のデバイスが、ハードウェア構成またはソフトウェア構成における差異に起因して、異なるアプリケーションプログラミングコードを要求する場合、「異種」である。

「アプリケーション」は、デバイスまたはシステムが遂行する必要がある、タスクまたはタスクの組である。

「アプリケーションコード」のブロックは、ソフトウェアアプリケーションが 1 つ以上のデバイス内で実行されるために書き込まれる、ソフトウェアプログラムコードのブロックである。

「デバイス」は、ハードウェアのアイテムのインスタンスであってハードウェアの構成だけではなく、また、ハードウェアに関連付けられる任意のソフトウェア環境も含む。

「実デバイス」は、物理的に異なる、デバイスである。

「仮想デバイス」は、動作上異なる、デバイスである。

「通信ネットワーク」または「ネットワーク」は、分散型デバイスをともに接続するために使用される媒体である。これは、インターネット、または産業イーサネット（登録商標）等の特殊ネットワークプロトコル、または複数のネットワークプロトコルの組み合わせであることができる。これは、有線、無線、または有線および無線の組み合わせであることができる。

「サーバ」は、ソフトウェアプログラムを起動するために使用されるコンピュータである。これは、ローカルまたはクラウド内にあることができる。

「データベース」は、電子的に記憶およびアクセスされる、編成されたデータの集合である。

「クライアントデバイス」は、人物がソフトウェアプログラムと相互作用するために使用されるデバイスである。これは、コンピュータまたはスマートフォン等のモバイルデバイスであることができる。

「エッジコンピュータ」または「エッジデバイス」は、複数の I/O 接続端末を有するコンピュータまたはマイクロコントローラデバイスであって、種々のセンサ、スイッチ、オーディオ／ビデオデバイス、機器、または測定デバイスに接続するために使用され、一方方向性または双方向性に、これらのデバイスと通信することができる。

「開発者」は、システムのためのアプリケーションを設計する人物である。

「設計ソフトウェア」は、ユーザが、システムのためのアプリケーションを設計および管理することを可能にするソフトウェアプログラムである。本発明において、「設計ソフトウェア」は、分散型異種コンピューティングおよび制御システムのためのアプリケーションを設計するためのソフトウェアプログラム実施形態を指す。

「ユーザ」は、設計ソフトウェアを使用して、システムのためのアプリケーションを設計または管理する、人物である。

「回路図」は、特定の機能の仮想化されたグラフィカル表現であって、ハードウェアコンテキストおよびソフトウェア構成から独立する。仮想化は、階層であることができる。

「機能回路図」は、特定のアプリケーション機能を実施するように構成される回路図であり、ハードウェアコンテキストおよびソフトウェア構成から独立する。

「挙動回路図」は、簡略化されたモデルまたは統計的モデル等、「機能回路図」をモデル

10

20

30

40

50

化するように構成される回路図である。

「システム回路図」は、分散型異種コンピューティングおよび制御システム全体のアプリケーションを表す設計ソフトウェア内に実装される回路図である。システム回路図は、階層であり、全ての階層内に全ての要素を含む。

「サブ回路図」は、システム回路図から解析され、分散型異種コンピューティングおよび制御システム内の特定のまたは仮想デバイスのアプリケーションを表す回路図である。追加のモジュールおよび他の要素が、サブ回路図の中に追加され、機能性を完成または向上させることができる。

「ネットリスト」は、回路図のテキスト表現である。ネットリストおよびその対応する回路図は、機能的に均等物である。

「モジュール」または「設計モジュール」は、以下のコンポーネント、すなわち、プロパティテーブル、シンボル、機能回路図、挙動回路図、および「デバイス依存」ブロックのうちの1つ以上のものを含む構造である。

「モジュールシンボル」は、モジュールのブロック形状表現である。これは、回路図内で使用される。

「モジュールインスタンス」は、回路図内のモジュールのインスタンス化である。これは、回路図内のモジュールシンボルとして現れ、対応するモジュールの回路図コンポーネントまたはデバイス依存ブロックコンポーネントのいずれかの中にインスタンス化する。回路図は、同一モジュールから複数のモジュールインスタンスを含むことができる。

「デバイス依存」ブロックは、デバイスのために書き込まれるアプリケーションコードのブロックであって、コードの構造およびコンテンツは、デバイスのハードウェア環境、コードのためのソフトウェア環境、およびそれらの組み合わせから成る群から選択されるメンバーの構成に依存し、典型的に、異なるタイプのデバイスのために異なる。

「機能モジュールインスタンス」または「機能モジュール」は、機能回路図の中にインスタンス化するモジュールインスタンスである。

「挙動モジュールインスタンス」または「挙動モジュール」は、挙動回路図の中にインスタンス化するモジュールインスタンスである。

「プリミティブモジュールインスタンス」または「プリミティブモジュール」または「プリミティブ」は、デバイス依存ブロックの中にインスタンス化するモジュールインスタンスである。

モジュールの「ピン」は、モジュールの入力、または出力、または双方向性ポートを表す。これは、外部信号が接続され得る場所である。モジュールは、複数のピンを有することができ、それは、典型的に、回路図およびシンボルの両方において、小正方形または他の形状として描かれる。

「仮想ワイヤ」は、2つのモジュール間の接続を表す。これは、データが、第1のモジュールの出力ピンから第2のモジュールの入力ピンの中にフローすることを示す。仮想ワイヤはまた、複数のモジュールを接続して、マルチセグメント化されることができる。

回路図の「要素」は、回路図の内側の任意の個々のエンティティを指す。これは、プロパティ、ピン、インスタンスシンボル、仮想ワイヤ、パラメータ等を含むことができる。

回路図内の要素の「アドレス」は、その要素のための一意の識別子である。

「デバイスタイプ」は、物理的または仮想デバイスのタイプのための識別子である。各「デバイス依存」ブロックは、1つ以上のデバイスタイプに結び付けられ、適合性を示す。

「デバイスID」または「デバイス\_\_id」は、システム内のデバイスのための一意の識別子である。

「モジュールID」または「モジュール\_\_id」は、回路図の内側のモジュールインスタンスのための一意の識別子である。モジュールインスタンスの「モジュールID」は、モジュールインスタンスの「アドレス」と同一である必要はない。

「解析」は、サブ回路図をシステム回路図から抽出するためのプロセスである。追加のモジュールおよび他の要素が、プロセスにおいてサブ回路図に追加されることができる。

「通信モジュール」または「commモジュール」は、その一次機能が、ネットワークを

10

20

30

40

50

介して、データをその入力ピンから遠隔宛先に送信する、またはネットワークを介して、データを遠隔宛先から受信し、そのデータをその出力ピンに送達することであるモジュールである。c o m mモジュールはまた、伝送モジュールおよび受信モジュールとして別個に実装されることができることに留意されたい。

「通信ブローカ」または「c o m mブローカ」は、ネットワークを経由した複数のデバイス間の双方向性通信を管理する、ソフトウェアプログラムである。全てのデバイスが、c o m mブローカを用いて、1つ以上のメッセージトピックをサブスクライブする。各デバイスは、データペイロード+メッセージトピックをc o m mブローカに送信することができ、c o m mブローカは、データペイロードを、該メッセージトピックをサブスクライブする、デバイスに転送するであろう。

10

「i n i t回路図」は、デバイスの中に事前にロードされ、制御プログラムによって実行される、回路図である。その機能は、限定ではないが、サブ回路図更新およびシステムコマンド実行を含む。

「s y sモジュール」は、その機能が、限定ではないが、サブ回路図更新およびシステムコマンド実行を含む、モジュールである。

「制御プログラム」は、サブ回路図およびi n i t回路図をロードおよび実行することが可能であるデバイス内で起動されるソフトウェアプログラムである。制御プログラムの異なる実装が、異なるデバイスまたはさらに同一デバイスに関して存在し得る。

「コンパイラ」は、意図されるデバイス上で実行されるために、プログラムコードを機械コードまたはより低次のコードに変換する、ソフトウェアプログラムである。

20

「チャート」は、その入力ピンに関するデータのプロットを生成する、モジュールである。データは、回路図ウィンドウとともに、または異なるウィンドウ内で、プロットされることができる。プロットは、時系列プロット、ヒストグラム、円グラフ等であるように選定されることができる。

「プローブ」は、その入力ピンに到着するデータを印刷するモジュールである。データは、回路図ウィンドウの内側、別のウィンドウ内、またはファイルに印刷されることができる。

「コンピュータプロセス」は、ソフトウェアまたはファームウェアまたはこれらのいずれかの組み合わせの制御下で動作する、または前述のいずれかの制御外で動作し得る、コンピュータハードウェア（プロセッサ、フィールドプログラマブルゲートアレイまたは他の電子組み合わせ論理、または類似デバイス等）を使用するコンピュータ内の説明される機能の実施である。説明される機能の全部または一部は、トランジスタまたは抵抗器等の能動または受動電子コンポーネントによって実施され得る。用語「コンピュータプロセス」を使用する際、必ずしも、スケジュール可能エンティティまたはコンピュータプログラムもしくはその一部の動作を要求するわけではないが、いくつかの実施形態において、コンピュータプロセスは、スケジュール可能エンティティまたはコンピュータプログラムもしくはその一部の動作によって実装され得る。さらに、文脈によって別様に要求されない限り、「プロセス」は、2つ以上のプロセッサまたは2つ以上の（シングルまたはマルチプロセッサ）コンピュータを使用して実装され得る。

30

【0059】

40

本発明の実施形態は、分散型異種システムにおいて実行されるためのアプリケーションを設計するためのシステムおよび方法を提供する。これらの実施形態において、ユーザは、システムの分散型または異種性質のいずれも懸念する必要がない。代わりに、ユーザは、分散型異種システムにおいて実行されるためのアプリケーションを定義する、システム回路図を実装することができ、システム回路図は、分散型異種システムの対応するデバイス内での実行のためのサブ回路図に解析される。これは、ユーザが、通信、データ変換、遠隔更新等の複雑な問題点の代わりに、そのようなアプリケーションを実装する際に重要であるデバイス間のデータ処理およびデータフローに焦点を当て、システムのためのアプリケーションを実装することを可能にする。

【0060】

50

図 1 は、本発明のある実施形態による、そのためにアプリケーションが設計される、分散型異種コンピューティングおよび制御システム 10 のブロック図である。システム 10 は、通信ネットワーク 14 を通して、クライアントデバイス 17、エッジコンピュータ 11、12、13、およびデータベース 15 に通信可能に結合される、集中型サーバ 16 を含む。デバイス 11、12、13、15、16、17 の各々は、物理的デバイスまたは仮想デバイスであり得、デバイス 11、12、13、15、16、17 のいくつかまたは全ては、異種であり得る（可変デバイスタイプ、デバイスアプリケーション等を伴う）。通信ネットワーク 14 は、1 つ以上のタイプのネットワークデバイスおよび 1 つ以上のネットワークプロトコルを用いて構成され得る。例示的实施形態において、システム 10 は、全て同一デバイスタイプであるデバイスを用いて構成される特殊な例の分散型コンピューティングおよび制御システムである。

10

#### 【0061】

いくつかの実施形態において、分散型システムとして、システム 10 は、集中型サーバ 16、クライアントデバイス 17、エッジコンピュータ 11、12、13、およびデータベース 15 を横断して実装される、アプリケーションを実行し、これらのデバイス間に協調を要求する。例えば、システム 10 は、デバイス 11、12、13、15、16、17 間の協調を要求する、産業機械を監視するためのアプリケーションを実行し得る。本アプリケーションを実行するために、アプリケーションコードが、各エッジコンピュータ 11、12、13 内に実装され、読取値を産業機械の異なるセンサから求め、読取値がそのセンサのための閾値を上回ると、アラートを集中型サーバ 16 に送信する。エッジコンピュータ 11、12、13 内に実装されるアプリケーションコードは、センサのための閾値にデータベース 15 からアクセスし得る。アプリケーションコードはまた、集中型サーバ 16 内に実装され、アラートを各エッジコンピュータ 11、12、13 から受信し、協調させ、産業機械のエラー条件を決定する。集中型サーバ 16 のアプリケーションコードはさらに、クライアントデバイス 17 に、エラー条件を送信し、クライアントデバイス 17 上に実装されるアプリケーションコードは、エラー条件を受信し、産業機械のオペレータに提示する。

20

#### 【0062】

異種システムとして、エッジコンピュータ 11、12、13 の各々は、異なるハードウェア構成、ソフトウェア構成、オペレーティングシステム、プログラミング方法、開発ツール等を有し得る。例えば、エッジコンピュータ 11 は、低能力マイクロコントローラ（MCU）であり得、エッジコンピュータ 212 は、Linux（登録商標）オペレーティングシステムを起動させる、シングルボードコンピュータであり得る。別の例に関して、エッジコンピュータ 2 およびエッジコンピュータ 3 13 は、同一ハードウェア構成を有し得るが、エッジコンピュータ 2 12 は、C プログラミング言語でアプリケーションを実装し得る一方、エッジコンピュータ 3 13 は、Python プログラミング言語でアプリケーションを実装し得る。慣例上、システム 10 のデバイス 11、12、13、15、16、17 間のハードウェア構成、ソフトウェア構成等における差異は、異なるプログラミング言語、ソフトウェア開発ツール等が、デバイス 11、12、13、15、16、17 を横断して実装されるためにアプリケーションを設計することにおいて使用されることを要求し、それは、同一アプリケーション機能を異なるデバイス上に実施するためのアプリケーション内に異なるコード構造をもたらす。当業者によって公知であるように、慣例上、異種システムは、シングルプログラミング言語またはソフトウェア開発ツールで設計されることができない。

30

40

#### 【0063】

本発明の実施形態において、ユーザは、デバイス 11、12、13、15、16、17 毎にサブ回路図に解析される、アプリケーション機能のシステム回路図を定義することによって、システム 10 のデバイス 11、12、13、15、16、17 を横断して実装されるようにアプリケーションを設計する。いくつかの実施形態において、各サブ回路図は、その対応するデバイスのために実装される制御プログラムによって、対応するデバイス

50

11、12、13、15、16、17内にロードされ、実行される。他の実施形態において、各サブ回路図は、対応するデバイス内にロードされ、実行される、バイナリコードにコンパイルされる。

【0064】

図2は、先行技術における、分散型異種コンピューティングおよび制御システムのためのアプリケーションを設計する方法20を図示するフローチャートである。方法20の第1のプロセス21において、開発者は、それぞれ、システム内の特定のデバイスタイプ、特定のデバイスアプリケーション等に対応する複数の別個のアプリケーションコードプログラムを実装することによって、アプリケーションを設計する。アプリケーションコードプログラムの実装は、典型的に、異なるプログラミング言語、開発ツール、および開発フレームワークを要求し、それは、設計を複雑かつ困難にする。複数のアプリケーションコードプログラムの存在により、第2のプロセス22において、開発者は、各アプリケーションコードプログラムを実装し、システム内の異なるデバイスによって使用される通信プロトコルおよびデータ通信フォーマットを協調させなければならず、それは、エラーを被りやすいプロセスである。第3のプロセス23において、開発者は、アプリケーションコードプログラムを別個のファイルまたはリポジトリ内に記憶する。第4のプロセス24において、開発者は、別個のプログラム更新プロシージャを実装し、各アプリケーションコードプログラムを、アプリケーションコードプログラムに対応するデバイスタイプ、デバイスアプリケーション等を有するシステム内の1つ以上のデバイスに分散させる。

【0065】

図3は、本発明のある実施形態による、分散型異種コンピューティングおよび制御システムのためのアプリケーションを設計する方法30を図示するフローチャートである。方法30の第1のプロセス31において、ユーザは、設計ソフトウェア内の分散型異種コンピューティングおよび制御システムを横断して実行するためのアプリケーションの回路図表現であるシステム回路図を実装することによって、アプリケーションを設計する。ユーザは、設計ソフトウェアと通信するように構成されるユーザインターフェースを通して、システム回路図を実装し得る。システム回路図の使用は、アプリケーションを設計するための複数のプログラミング言語、開発ツール、および開発フレームワークを介して別個のアプリケーションコードプログラムを実装することを排除する。

【0066】

第2のプロセス32において、ユーザは、分散型異種コンピューティングおよび制御システムを横断して、設計ソフトウェア内にシステム回路図を展開する。システム回路図は、各サブ回路図が、デバイスの組の1つ以上のデバイスに対応し、それらのデバイスに伝送されるように、サブ回路図の組として展開される。第3のプロセス33において、設計ソフトウェアは、展開されるサブ回路図を通して、通信プロトコル、データ通信フォーマット、およびハードウェア依存性をハンドリングする。第4のプロセス34において、設計ソフトウェアは、分散型異種コンピューティングおよび制御システムを横断して、デバイス内のサブ回路図の分布および同期を管理する。

【0067】

図4は、本発明のある実施形態による、設計ソフトウェアを実行するために使用されるシステム40のブロック図である。システム40は、コンピュータノードとも称される。図4のシステム40は、ネットワーク環境内のクライアント-サーバ組み合わせであって、それは、ネットワーク43を通してクライアントデバイス42に通信可能に結合されるサーバ44を含む。クライアントデバイス42は、ユーザ41によって動作される。設計ソフトウェアは、サーバ44が、ディスプレイユーザ41に、クライアントデバイス42上で表示させるグラフィカルユーザインターフェースに関連付けられる。クライアントデバイス42は、ウェブブラウザまたはローカルソフトウェアプログラムを実行するコンピュータ、アプリを実行するモバイルデバイス等であり得る。他の実施形態において、システムは、独立型コンピュータであり得る。

【0068】

サーバ４４は、設計ソフトウェアを実行し、分散型異種コンピューティングおよび制御システムのためのアプリケーションのシステム回路図を実装する、割り振る、展開する、および実行する。ユーザは、クライアントデバイス４２上の、グラフィカルユーザインターフェースを通して、入力を設計ソフトウェアに提供し得る。サーバ４４は、完全または部分的に、設計ソフトウェアを記憶し得る。クライアントデバイス４２がウェブブラウザである実施形態において、サーバ４４は、完全設計ソフトウェアを記憶し、ネットワーク４３を通して、設計ソフトウェアの一部をクライアントデバイス４２に提供する。クライアントデバイス４２がローカルソフトウェアプログラムまたはアプリである実施形態において、サーバ４４は、設計ソフトウェアの一部を記憶し、クライアントデバイス４２は、設計ソフトウェアの残りを記憶する。他の実施形態において、完全設計ソフトウェアは、サーバに接続されずに、クライアントデバイス上に記憶される。

10

【００６９】

図５は、本発明のある実施形態による、分散型異種コンピューティングおよび制御システムのためのアプリケーションを設計するために使用されるグラフィカルユーザインターフェース５０である。本発明の実施形態において、ユーザは、回路図を設計し、アプリケーション機能を表すことによって、分散型異種システムのためのアプリケーションを設計する。グラフィカルユーザインターフェース５０は、回路図を設計および展開するために、双方向環境をユーザに提供する。グラフィカルユーザインターフェース５０は、限定ではないが、以下のコンポーネント、すなわち、設計キャンバス５２、ライブラリマネージャサブパネル５４、およびメニューバー５６を含む。設計キャンバス５２において、ユーザは、分散型異種システムのための回路図を形成する。形成されるシステム回路図は、ライブラリ内に、将来的回路図を設計することにおいて使用するための新しいモジュールとして保存され得る。設計キャンバス５２において、複数の回路図タブが、同時に、複数の回路図を設計するために開かれることができる。

20

【００７０】

ライブラリマネージャサブパネル５４は、ライブラリ内にグループ化される、設計モジュール（例えば、機能モジュール、プリミティブモジュール、挙動モジュール等）を表示する。各ライブラリは、１つのライブラリ名下にグループ化されるモジュールの組である。ユーザは、設計キャンバス内に開かれる、または設置されるためのモジュールをライブラリマネージャから選択することができる。例えば、ユーザは、モジュールをライブラリマネージャサブパネル５４からドラッグし、モジュールを設計キャンバス５２の中にドロップすることができる。モジュールのそのような選択は、モジュールのインスタンスを作成し得る。ユーザは、作成されたモジュールインスタンスを設計キャンバス５２内で順序付け、構造化し、分散型異種システムのための回路図を設計し得る。メニューバー５６は、複数のメニューオプションおよびプルダウンメニューから成り、それは、回路図を設計するシミュレートする、および展開するためのユーザオプションを提供する。

30

【００７１】

図６は、本発明のある実施形態による、グラフィカルユーザインターフェース６０内の回路図６２および対応するネットリスト６４のディスプレイを図示する。回路図６２内のグラフィカルに表されるアプリケーション機能およびデバイスは、分散型異種システムを横断して実行されるアプリケーションを備え得る。回路図６２は、対応するテキストベースのネットリスト６４に機能的に匹敵する。図６において、ネットリスト６４の一部が、回路図６２内の「クラウド論理」モジュールインスタンスのために表示され、それは、識別子、名称、タイプ、設計キャンバス内の座標、およびこのモジュールインスタンスを他のモジュールインスタンスに接続する仮想ワイヤの識別子を含むこのモジュールインスタンスのためのパラメータを示す。ある実施形態において、回路図は、メモリ内に、ネットリスト６４として記憶され、設計ソフトウェアは、記憶されたネットリスト６４をグラフィカルユーザインターフェース６０内に表示される回路図６２の形態の中にレンダリングする。ユーザは、グラフィカルユーザインターフェース６０を通して、回路図６２またはネットリスト６４の一方または両方を可視化および編集し得る。回路図６２またはネット

40

50

リスト 64 のうち的一方が、編集されると、設計ソフトウェアは、編集に従って、他方を自動的に更新する。

#### 【0072】

グラフィカルユーザインターフェースに関連付けられる、設計ソフトウェアは、当技術分野において公知である。しかしながら、既存の設計ソフトウェアは、分散型異種システムの設計をサポートしない。大部分の既存の設計ソフトウェアは、同一コンピュータ内で起動され、設計を実行する。本発明による設計ソフトウェアは、設計（例えば、回路図として）をコンポーネント（例えば、サブ回路図）に解析し、コンポーネントを分散型異種デバイスの組に伝送し、コンポーネントをこれらの分散型異種デバイス内で実行する。

#### 【0073】

図 7 は、本発明のある実施形態による、モジュールコンポーネント 70 のブロック図である。モジュールは、分散型異種システムの 1 つ以上のアプリケーション機能を含む、ユニットである。モジュールは、システム回路図内に設置され、順序付けられる、モジュールインスタンスの中にインスタンス化され得る。図 7 に示されるように、モジュールのコンポーネントは、プロパティテーブル 71、グラフィカル表現（モジュールシンボルとも称される）72、回路図 73、74、およびデバイス依存ブロック 75、76（プリミティブモジュールまたはプリミティブモジュールインスタンスの場合）を含む。回路図は、機能回路図、挙動回路図等であり得る。モジュールは、限定ではないが、他のタイプのコンポーネントを含み得る。

#### 【0074】

図 8 は、本発明のある実施形態による、モジュールプロパティを定義する、プロパティテーブル 80 のブロック図である。プロパティテーブル 80 は、モジュール名、モジュールライブラリ、およびコンポーネントプロパティのモジュールプロパティを含む。プロパティテーブル 80 は、モジュール「モジュール 1」のためのものである。プロパティテーブル 80 において、モジュールプロパティ名は、「モジュール 1」に設定され、ライブラリは、「Lib 2」に設定され、「モジュール 1」が、「Lib 2」と命名されたライブラリ内に記憶されることを示す。プロパティテーブル 80 のコンポーネントプロパティは、「モジュール 1」内に含まれる 2 つの回路図を含む。第 1 の回路図は、「回路図」と命名され、第 1 の回路図が機能回路図であることを示す、「機能」に設定されたタイププロパティを有し、「2. 1」に設定されたバージョンプロパティを有する。第 2 の回路図は、「回路図 2」と命名され、第 2 の回路図が挙動回路図であることを示す、「挙動」に設定されるタイププロパティを有し、「1. 0」に設定されたバージョンプロパティを有する。

#### 【0075】

図 9 は、本発明のある実施形態による、異なるタイプのデバイス内の回路図の実行 90 を図示するブロック図である。本発明の実施形態における回路図（例えば、機能回路図および挙動回路図）は、デバイス独立性であって、それは、同一回路図が、異なるタイプのデバイス内で実行されることができるとを意味する。図 9 は、回路図 92 を含む、「モジュール 1」と命名されたモジュールインスタンスを示す。図 9 はまた、それぞれ、異なるタイプのデバイスである 3 つの異種デバイス 93、94、95 を含む、分散型異種システムを示す。具体的に、デバイス 1 93 は、タイプ「MCU 1」であって、デバイス 2 94 は、タイプ「P E C 1」であり、およびデバイス 3 は、タイプ「P E C 2」である。回路図 92 は、デバイス独立性であるので、デバイスの各々が、異なるタイプのデバイスである場合でも、異種デバイス 93、94、95 のそれぞれ上にロードされ、実行されることができる。

#### 【0076】

制御プログラムは、回路図を実行する各デバイス上で起動されるように実装される。具体的に、制御プログラム 1 は、デバイス 1 93 上に実装され、制御プログラム 2 は、デバイス 2 94 上に実装され、制御プログラム 3 は、デバイス 3 95 上に実装される。異なるタイプのデバイス 93、94、95 との適合性のために、各制御プログラムは、異な

10

20

30

40

50

るプログラミング言語で実装される、異なる開発ツールを使用する、異なるハードウェアアーキテクチャ内に実装されること等ができる。制御プログラムは、そのデバイスタイプに基づいて、異なる実装を有するが、それぞれ、同一回路図 9 2 を実行することが可能である（デバイス 9 3、9 4、9 5 毎に異なる回路図を要求するのではなく）。

【0077】

図 10 は、本発明のある実施形態による、モジュールプロパティを定義するプロパティテーブル 100 のブロック図である。プロパティテーブル 100 は、モジュール名、モジュールライブラリ、およびコンポーネントプロパティのモジュールプロパティを含む。プロパティテーブル 80 は、プリミティブモジュールであるモジュール「モジュール 2」のものである。プロパティテーブル 100 において、名称のモジュールプロパティは、「モジュール 2」に設定され、ライブラリのモジュールプロパティは、「Lib 2」に設定され、「モジュール 2」が、「Lib 2」と命名されたライブラリ内に記憶されることを示す。プロパティテーブル 80 のコンポーネントプロパティは、「モジュール 2」内に含まれる、2つのデバイス依存ブロックを含む。第 1 のデバイス依存ブロックは、「依存」と命名され、「PEC 1」および「PEC 2」に設定されたデバイスタイププロパティを有し、「3. 4」に設定されたバージョンプロパティを有する。本デバイスタイププロパティ設定は、デバイス依存ブロック「依存」が、2つのタイプのデバイス「PEC 1」および「PEC 2」と適合性があることを示す。第 2 のデバイス依存ブロックは、「依存 2」と命名され、「MCU 1」に設定されたデバイスタイププロパティを有し、「2. 7」に設定されたバージョンプロパティを有する。本デバイスタイププロパティ設定は、依存ブロック「依存 2」が、1つのタイプのデバイス「MCU 1」と適合性があることを示す。

【0078】

図 10 に示されるように、「モジュール 2」等のモジュールは、1つ以上のタイプのデバイスと適合性がある、複数のデバイス依存ブロックを含むことができる。いくつかの実施形態において、モジュール内の異なるデバイス依存ブロックは、同一アプリケーション機能であるが、異なるデバイスのコンテキストにおいて実施する。他の実施形態において、モジュール内のデバイス依存ブロックのいくつかまたは全ては、異なるデバイスのコンテキストにおいて、異なるアプリケーション機能を実施する。

【0079】

図 11 は、本発明のある実施形態による、適合性があるデバイス 114、115、116 内におけるデバイス依存ブロック 112、113 の実行を図示するブロック図である。デバイス依存ブロックは、その名称の通り、デバイス依存性であって、それは、デバイス依存ブロックが、デバイス依存ブロックと適合性があるデバイスのタイプ内においてのみ実行することを意味する。図 10 に示されるように、デバイス依存ブロックは、デバイス依存ブロックと適合性があるデバイスタイプに設定される、デバイスタイププロパティを有する。図 11 は、「モジュール 2」と命名されたモジュール 111 を示し、それは、デバイス依存ブロック「依存」112 および「依存 2」113 を含む。図 10 に示されるように、デバイス依存ブロック「依存」112 は、「PEC 1」および「PEC 2」のデバイスタイプと適合性があり、デバイス依存ブロック「依存 2」113 は、デバイスタイプ「MCU 1」と適合性がある。

【0080】

図 11 はまた、それぞれ、異なるタイプのデバイスである 3つの異種デバイス 114、115、116 を含む、分散型異種システムを示す。具体的に、デバイス 1 114 は、タイプ「MCU 1」であって、デバイス 2 115 は、タイプ「PEC 1」であって、デバイス 3 116 は、タイプ「PEC 2」である。デバイス依存ブロック「依存」は、「PEC 1」および「PEC 2」のデバイスタイプのみと適合性があるので、デバイス 2 115 およびデバイス 3 116 上にのみロードされ、実行されることができる。さらに、デバイス依存ブロック「依存 2」は、「MCU 1」のデバイスタイプのみと適合性があるので、デバイス 1 114 上にのみロードされ、実行される。したがって、モジュールインスタンス「モジュール 2」111 をこれらのデバイス 114、115、116 内で使

用するために、デバイス依存ブロック「依存」は、デバイス2 115およびデバイス3 116上にロードするために選択され、ブロック「依存2」は、デバイス1 114上にロードするために選択される。デバイス依存ブロック「依存」は、デバイス1 114内では正しく実行されず、デバイス依存ブロック「依存2」は、デバイス2 115およびデバイス3 116内では正しく実行されないであろうことに留意されたい。

【0081】

制御プログラムは、適合性があるデバイス依存ブロックを実行する各デバイス上で起動するように実装される。具体的に、制御プログラム1は、デバイス1 114上に実装され、制御プログラム2は、デバイス2 115上に実装され、制御プログラム3は、デバイス3 116上に実装される。異なるタイプのデバイス114、115、116との適合性のために、各制御プログラムは、異なるプログラミング言語で実装される、異なる開発ツールを使用する、異なるハードウェアアーキテクチャ内に実装されること等ができる。

【0082】

図12は、本発明のある実施形態による、モジュールのグラフィカル表現のブロック図である。本同一グラフィカル表現は、モジュールおよびモジュールのインスタンスのために使用され得る。グラフィカル表現（モジュールシンボルとも称される）は、縁に位置付けられる正方形形状の入力および出力ピンを伴う矩形ブロックである。他の実施形態において、グラフィカル表現およびピンは、他の形状として提示されることができる。

【0083】

図13は、本発明のある実施形態による、回路図130のブロック図である。本回路図130は、分散型異種コンピューティングおよび制御システム10を表すシステム回路図である。回路図130は、それぞれ、モジュールのインスタンスをグラフィカルに表す複数のモジュールシンボルを含む。所与のモジュールのモジュールシンボルは、各使用が、所与のモジュールのインスタンス化またはインスタンスであるように、複数回、回路図内で使用され得る。モジュールインスタンスは、仮想ワイヤによって互いに接続され、モジュールインスタンス間のデータフローおよび通信を表す。例えば、回路図130は、データベースモジュールインスタンス135、ダッシュボードモジュールインスタンス136、クラウド論理モジュールインスタンス134、エッジ論理1モジュールインスタンス131、エッジ論理2モジュールインスタンス132、およびエッジ論理3モジュールインスタンス133を含む、システム回路図であって、これらの各々は、モジュールシンボルとして提示される。モジュールインスタンス131、132、133、134、135、136は、階層的に構造化され、分散型異種コンピューティングおよび制御システム内で実行されるためのアプリケーション機能を定義する。回路図130は、クラウド論理モジュールインスタンス134とエッジ論理1モジュールインスタンス131との間等の対のモジュールインスタンス間に仮想ワイヤを含み、対間のデータフローまたは通信を表す。

【0084】

図14は、本発明のある実施形態による、モジュールインスタンスのためのプロパティ141、142、143、144を定義するプロパティテーブル140を提示するグラフィカルユーザインターフェースである。モジュールの各インスタンスは、インスタンスプロパティを定義する、対応するプロパティテーブル140を有する。ある実施形態において、図14に示されるように、モジュールインスタンスプロパティテーブルは、グラフィカルユーザインターフェース内に表示されるポップアップウィンドウとして実装される。グラフィカルユーザインターフェースは、ユーザが、プロパティテーブル140内のモジュールインスタンスのプロパティおよびパラメータを修正することを可能にする。グラフィカルユーザインターフェースはまた、ユーザが、カスタムプロパティおよびパラメータをプロパティテーブル140に追加することを可能にする。プロパティテーブル140は、モジュールインスタンス「クラウド論理」のための名称141、タイプ142、デバイス143、およびパラメータ144のモジュールインスタンスプロパティを含む。プロパティテーブル140において、名称141のモジュールインスタンスプロパティは、「クラウド論理」に設定され、タイプ142のモジュールインスタンスプロパティは、「Lib

10

20

30

40

50

2 / モジュール 1 / 回路図」に設定される。

【0085】

タイププロパティは、モジュールインスタンスを作成することにおいて使用される親モジュールから選択されるインスタンス化のタイプを定義する。例えば、親モジュールは、機能回路図、挙動回路図、およびデバイス依存ブロック等のコンポーネントを含有し得る（図8における「モジュール1」および図10におけるモジュール2に関して示されるように）。モジュールインスタンスタイププロパティは、ユーザによって、親モジュールのコンポーネントから選択され、モジュールインスタンスのためのインスタンス化タイプを定義し得る。図14において、モジュールインスタンス「クラウド論理」は、図8の親モジュール「モジュール1」からインスタンス化され、そのプロパティタイプ142は、「Lib2」内に位置する、「回路図」と命名された1つの回路図を含む。したがって、モジュールインスタンス「クラウド論理」のためのタイププロパティは、「Lib2 / モジュール1 / 回路図」に設定される。モジュールインスタンスのためのタイプセットが、図14におけるように、機能回路図であるとき、モジュールインスタンスは、機能モジュールインスタンスとして構成される。モジュールインスタンスのためのタイプセットが、挙動回路図であるとき、モジュールインスタンスは、挙動モジュールインスタンスとして構成される。モジュールインスタンスのためのタイプセットが、デバイス依存ブロックであるとき、モジュールインスタンスは、プリミティブモジュールインスタンスとして構成される、（また、単に、「プリミティブ」と呼ばれる）。

【0086】

機能モジュールインスタンスは、システム回路図であり得る、タイププロパティ142内に設定される機能回路図に対してインスタンス化される。システム回路図等の回路図は、複数の機能モジュールインスタンスを含有し得、これの各々は、1つ以上の機能回路図を含有し得る。したがって、システム回路図に対してインスタンス化されるモジュールインスタンスは、機能回路図の階層構造（階層）を含有し得る。プリミティブモジュールインスタンスは、そのような階層構造を有していない、デバイス依存ブロックに対してインスタンス化される。本発明の実施形態の文脈において参照される、「階層」または「階層構造」は、（本発明のいくつかの実施形態で使用されるような）デバイス依存ブロックの内側の階層コード構造ではなく、モジュールインスタンスに対する設計階層を意味することに留意されたい。システム回路図の最低階層レベルは、全てのモジュールインスタンスパスをプリミティブモジュールインスタンス内で終了させる。

【0087】

プロパティテーブル140において、デバイス143のモジュールインスタンスプロパティは、「サーバ5」に設定され、その中でモジュールインスタンスが実行されるべき、デバイスを示す。設計ソフトウェアは、システム回路図の解析の間、デバイスプロパティを使用して、モジュールインスタンスを伝送する（およびモジュールインスタンスをそのデバイスのための関連付けられるサブ回路図に配分する）ための標的デバイスを決定する。いくつかの実施形態において、モジュールインスタンスのデバイスプロパティは、デフォルトによって、モジュールインスタンスが、その親モジュールと同一デバイスプロパティを有するように、インスタンス化時のその親モジュールから継承される。これらの実施形態において、ユーザは、グラフィカルユーザインターフェースを通して、モジュールインスタンスのためのデバイスプロパティを手動で設定することによって、デフォルト設定をオーバーライドすることができる。

【0088】

プロパティテーブル140において、パラメータのモジュールインスタンスプロパティは、モジュールプロパティのための2つのパラメータ、すなわち、「雑音」および「pi」に設定される。パラメータ「雑音」は、モジュールインスタンス「クラウド論理」のために、「2.5」のデフォルト値に設定され、パラメータ「pi」は、「3.14」のデフォルト値に設定される。ユーザは、グラフィカルユーザインターフェースを通して、パラメータのための値を手動で設定することによって、デフォルト値を変更させることがで

きる。

【0089】

図15は、本発明のある実施形態による、システム回路図151の階層構造を表すツリーグラフ150である。図15に示されるように、システム回路図151は、モジュールインスタンスの階層であって、それは、回路図、機能モジュールインスタンス、およびプリミティブモジュールインスタンスを含むが、デバイス依存ブロックを含有しない。ツリーグラフ150内の本構造は、システム回路図151が、デバイス独立性であることを可能にする。構造内の識別された上位レベル回路図は、システム回路図151の上位レベル階層における回路図である。上位レベル回路図は、3つのモジュールインスタンス、すなわち、「デバイス1」内の実行のための機能モジュールインスタンスであるF1、「デバイス2」内の実行のための機能モジュールインスタンスであるF2、および「デバイス3」内の実行のためのプリミティブモジュールインスタンスであるP1を含む。「デバイス1」、「デバイス2」、および「デバイス3」の各々は、異なるデバイスである。

【0090】

第2のレベルの階層において、機能モジュールインスタンスF1は、2つのモジュールインスタンス、すなわち、プリミティブモジュールインスタンスであるP2、および機能モジュールインスタンスであるF4を含む。機能モジュールインスタンスF2は、2つのモジュールインスタンス、すなわち、プリミティブモジュールインスタンスの両方であるP2およびP4を含む。プリミティブモジュールインスタンスP2は、機能モジュールF1およびF2の両方内にあることに留意されたい。第3のレベルの階層において、機能モジュールインスタンスF4は、シングルプリミティブモジュールインスタンスP3を含む。P1、P2、P3、およびP4は、リーフモジュールインスタンスであって、それは、任意の子モジュールインスタンスを有していない。モジュールインスタンスP2、F4、およびP3は全て、モジュールインスタンスF1の子モジュールインスタンスであるので、それらの各々は、F1のデバイスプロパティ「デバイス1」を継承する。モジュールインスタンスP2およびP4は全て、モジュールインスタンスF2の子モジュールインスタンスであるので、それらの各々は、F2のデバイスプロパティ「デバイス2」を継承する。ユーザは、これらのモジュールインスタンスに対応するインスタンスプロパティテーブル内のデバイスプロパティの手動選択を行うことによって、P2、F4、P3、P4のためのデバイスプロパティをオーバーライドし得る。

【0091】

システム回路図151の構造において、各階層パスは、152によって示されるように、デバイス依存ブロックに関連付けられたプリミティブモジュールインスタンス（リーフモジュールインスタンスP1、P2、P3、およびP4）内で終了する。すなわち、P1は、デバイス依存ブロックD1.3に関連付けられ、P2は、デバイス依存ブロックD2.1およびD2.2に関連付けられ、P3は、デバイス依存ブロックD3.1に関連付けられ、P4は、デバイス依存ブロックD4.2に関連付けられる。実施形態において、デバイス依存ブロックは、プリミティブモジュールインスタンス毎に構成されなければならない。システム回路図151において、プリミティブモジュールインスタンスP2は、機能モジュールインスタンスF1内で1回および機能モジュールインスタンスF2内で1回の2回、使用されることに留意されたい。F1およびF2は、異なるタイプのデバイス、すなわち、それぞれ、デバイス1およびデバイス2内で実行されるため、2つの異なるデバイス依存ブロック、すなわち、D2.1およびD2.2が、P2内に含まれなければならない。F1が、「デバイス1」にロードされると、デバイス1のデバイスタイプのためのデバイス依存ブロックD2.1もまた、デバイス1内にロードされる。F2が、「デバイス2」にロードされると、デバイス2のデバイスタイプのためのデバイス依存ブロックD2.2もまた、「デバイス2」内にロードされる。

【0092】

システム回路図151が、デバイス独立性（すなわち、デバイス依存ブロックが、152によって示されるように、システム回路図151内の階層構造内に含まれない）場合で

も、設計ソフトウェアおよびデバイスは両方とも、分散型異種コンピューティングおよび制御システムにおけるデバイスのための正しいデバイス依存ブロックを見つけるための十分なコンテキストを有する。分散型異種システム内の各デバイスは、設計ソフトウェアを実行するコンピュータノードによって、一意の識別子（ID）を割り当てられ、また、対応するデバイス名（グラフィカルインターフェースを通して、ユーザによってグラフィカルに記入され得る）を割り当てられ得る。デバイス依存ブロックの正しい選択が、そのプロパティテーブル内のモジュールインスタンス毎に規定されたデバイスプロパティに基づいて行われる。デバイスプロパティは、デバイスのデバイスIDまたはデバイス名を含む。デバイス名が、デバイスプロパティ内に規定されるとき、設計ソフトウェアは、対応するデバイスIDを特定することができる。デバイスIDを使用して、設計ソフトウェアは、そのデバイスIDに対応するデバイスタイプをルックアップすることができる。デバイスタイプが、見出されると、設計ソフトウェアは、そのデバイスタイプに対応するデバイス依存ブロックを見つけることができる。

#### 【0093】

本発明のいくつかの実施形態において、システム回路図151内に含まれるモジュールインスタンスが、デバイスに関連付けられるとき、デバイスのためのデバイス依存ブロックは、デバイス内に事前にロードされる。いくつかの実施形態において、システム回路図内に含まれるモジュールインスタンスが、デバイスに関連付けられるとき、設計ソフトウェアは、対応するデバイス依存ブロックをデバイスに伝送する。

#### 【0094】

図16は、本発明のある実施形態による、回路図を利用して、分散型異種コンピューティングおよび制御システムのためのアプリケーションを設計する方法160を図示するフローチャートである。ユーザは、コンピュータノードにおいて実行され、関連付けられるユーザインターフェースを通してユーザにアクセス可能である設計ソフトウェア命令を使用して、システムのための回路図を実装し、ユーザのための双方向環境を確立する。システム回路図の実装を通して、ユーザは、システムの各デバイスのハードウェアがシステム回路図内で表されるように、ハードウェアコンテキストにおいて、アプリケーションを設計する。

#### 【0095】

システム回路図を実装するために、方法160の第1のプロセス161において、コンピュータノードが、設計ソフトウェアを実行し、システム回路図を実装する。そのために、コンピュータノードは、双方向環境において、システム回路図を定義するための機能モジュールの組およびプリミティブモジュールの組へのユーザアクセスを提供する。機能モジュールの組は、1つ以上の機能回路図を含む。各機能回路図は、アプリケーション機能のグラフィカル表現であって、ハードウェアコンテキストおよびソフトウェア構成から独立する。プリミティブモジュールの組は、1つ以上のデバイス依存ブロックを含む。各デバイス依存ブロックは、分散型異種コンピューティングおよび制御システムの特定のデバイスのために書き込まれるアプリケーションコードであって、コードの構造およびコンテンツは、デバイスのためのハードウェア環境、コードのためのソフトウェア環境、およびそれらの組み合わせから成る群から選択されるメンバーの構成に依存し、典型的に、異なるタイプのデバイスに関して異なる。

#### 【0096】

実施形態において、挙動回路図と称される、特殊タイプの回路図もまた、存在する。挙動回路図は、典型的に、簡略化されたモデルまたは統計的モジュール等の機能回路図またはデバイス依存ブロックのモデルである。機能回路図と異なり、挙動回路図は、分散型異種システムの物理的デバイス上での実行のためではなく、シミュレーションコンテキストにおいて使用される。シミュレーションコンテキストは、典型的に、統計的変動、コーナーケース、および特殊入力条件等の分散型異種システムに関連付けられる問題点を理解するために、設計ソフトウェア内で実施される。ユーザはまた、1つ以上の挙動回路図を含む、挙動モジュールの組へのアクセスを提供され得る。ユーザはまた、システム回路図を

実装することにおいて使用される他のコンポーネントへのアクセスを提供され得る。モジュールおよびコンポーネントのいずれも、双方向環境を通して、ユーザによってグラフィカルに選択されることができる。

【0097】

コンピュータノードは、双方向環境において、機能モジュールと、プリミティブモジュールと、それらを順序付け、システム回路図を実装する、入力との一連の所望のものを受信する。コンピュータノードはまた、双方向環境において、挙動モジュールと、それらを順序付け、システム回路図を実装する、入力との一連の所望のものを受信し得る。いくつかの実施形態において、一連の所望のものは、対応する機能モジュールおよびプリミティブモジュールと、モジュールインスタンスを順序付け、システム回路図を実装する、入力とのモジュールインスタンスを作成する。コンピュータノードはまた、双方向環境において、一連の仮想ワイヤと、システム回路図内において、仮想ワイヤをシステム内で互いに通信することになる一対のモジュール（またはモジュールインスタンス）間に構成するための入力とを受信し得る。

【0098】

第2のプロセス162において、コンピュータノードは、各サブ回路図が、分散型異種コンピューティングおよび制御システムにおけるデバイスの組のデバイスに対応するように、システム回路図を解析し、サブ回路図の組を生成する。第3のプロセス163において、コンピュータノードは、デバイスが、サブ回路図内のデバイスに配分される、設計されるアプリケーションのアプリケーション機能を実行し得るように、各サブ回路図をその対応するデバイスに伝送する。第4のプロセス163において、コンピュータノードは、各デバイスを構成し、デバイスの内側で起動する対応する制御プログラムを介して、その対応するサブ回路図を実行する。

【0099】

図17は、本発明のある実施形態による、図13のシステム回路図130から解析される、サブ回路図のブロック図である。システム回路図130等のシステム回路図が、設計されると、システム回路図は、サブ回路図に解析されなければならない。解析されたサブ回路図の各々は、特定の分散型異種デバイスのために標的化されたシステム回路図の一部である。設計ソフトウェアは、サブ回路図をその分散型異種デバイスに伝送し、サブ回路図は、それぞれのデバイス内で起動される制御プログラムを介して実行される。

【0100】

図17は、システム回路図130に対応する例示的解析されたサブ回路図を示す。図13のモジュールインスタンスは、分散型異種システム内の4つのデバイス、すなわち、サーバ、エッジコンピュータ1、エッジコンピュータ2、およびエッジコンピュータ3に関連付けられる。サブ回路図は、4つのデバイス毎に、システム回路図から解析される。すなわち、サブ回路図171は、サーバのためにシステム回路図から解析され、サブ回路図172は、エッジコンピュータ1のためにシステム回路図から解析され、サブ回路図173は、エッジコンピュータ2のためにシステム回路図から解析され、サブ回路図174は、エッジコンピュータ3のためにシステム回路図から解析される。

【0101】

解析の間、システム回路図130内のモジュールインスタンスは、これらのモジュールインスタンスが実行されるべきデバイスに関連付けられる、サブ回路図に割り当てられる。具体的に、モジュールインスタンスデータベース、ダッシュボード、およびクラウド論理が、デバイスサーバのために、サブ回路図171に割り当てられる。他の実施形態において、「データベース」および「ダッシュボード」のモジュールインスタンスは、「クラウド論理」と同一サーバデバイス内で実行されないこともあることに留意されたい。さらに、モジュールインスタンスエッジ論理1は、デバイスエッジコンピュータ1のために、サブ回路図172に割り当てられ、モジュールインスタンスエッジ論理2は、デバイスエッジコンピュータ2のために、サブ回路図173に割り当てられ、モジュールインスタンスエッジ論理3は、デバイスエッジコンピュータ3のために、サブ回路図174に割り当

てられる。仮想ワイヤが、システム回路図130内の一対のモジュールインスタンス間に存在するが、対の各モジュールインスタンスが、2つの別個のサブ回路図に解析される場合、仮想ワイヤが、2つのサブ回路図の各々内のcommモジュールインスタンスによって置換される。commモジュールインスタンスは、2つの別個のサブ回路図間に通信およびデータフローを提供するように構成される。例えば、システム回路図130において、クラウド論理が、仮想ワイヤを介して、エッジ論理1に接続されるクラウド論理は、commモジュールとともにサブ回路図171に解析され、エッジ論理1は、commモジュールとともにサブ回路図172に解析される。サブ回路図171および172内のcommモジュールは、サブ回路図171および172間の通信およびデータフローを促進するように構成される。

#### 【0102】

図18は、本発明のある実施形態による、回路図をサブ回路図に解析する方法180を図示するフローチャートである。図17に示されるように、解析プロセスは、2つの別個のプロセス、すなわち、モジュールインスタンスの解析および仮想ワイヤの解析を伴う。方法180の第1のプロセス181において、設計ソフトウェアを実行するコンピュータノードは、回路図内のモジュールインスタンスをサブ回路図に解析する。本第1のプロセス181において、コンピュータノードは、そのデバイスプロパティに基づいて、モジュールインスタンスをサブ回路図に解析する。各サブ回路図は、分散型異種システム内の特定のデバイスのために構成される。各モジュールインスタンスは、モジュールインスタンスに関連付けられるデバイスを示す一意のデバイス識別子(ID)に設定されるデバイスプロパティを含む。同一デバイスに関連付けられる各モジュールインスタンスは、そのデバイスのための同一サブ回路図に解析される。このプロシージャは、全ての階層レベルのシステム回路図を横断して、全てのモジュールインスタンスに適用される。

#### 【0103】

第2のプロセス182において、コンピュータノードは、システム回路図内の仮想ワイヤ毎に解析する。仮想ワイヤを解析する際、システム回路図内の各仮想ワイヤが、検査される。仮想ワイヤが、2つのモジュールインスタンスを同一デバイスプロパティと接続する(すなわち、2つのモジュールインスタンスが、同一サブ回路図に解析される)場合、第3のプロセス183において、コンピュータノードは、仮想ワイヤをサブ回路図内の2つのモジュールインスタンス間の場所に保存する。仮想ワイヤが、2つのモジュールインスタンスを異なるデバイスプロパティと接続する(すなわち、2つのモジュールインスタンスは、別個のサブ回路図に解析される)場合、第4のプロセス184において、コンピュータノードは、commモジュールインスタンスを別個のサブ回路図の各々に追加する。各モジュールインスタンスは、次いで、その対応するサブ回路図に追加される、commモジュールインスタンスに接続される。すなわち、仮想ワイヤは、システム回路図内において、第1のモジュールインスタンスの出力を第2のモジュールインスタンスの入力に接続する。第1のモジュールインスタンスは、第1のサブ回路図に解析され、第2のモジュールインスタンスは、第2のサブ回路図に解析される。第1のcommモジュールインスタンスが、第1のサブ回路図内で作成され、仮想ワイヤが、第1のモジュールインスタンスの出力から第1のcommモジュールインスタンスの入力に追加される。第2のcommモジュールインスタンスが、第2のサブ回路図内で作成され、仮想ワイヤが、第2のcommモジュールインスタンスの出力から第2のモジュールインスタンスの入力に追加される。

#### 【0104】

第4のプロセス185において、コンピュータノードは、第1のモジュールインスタンスに接続される第1のcommモジュールインスタンスの宛先アドレスを第2のモジュールインスタンスのアドレスに設定する。第5のプロセス186において、コンピュータノードは、第2のモジュールインスタンスに接続される第2のcommモジュールインスタンスの受信アドレスを第2のモジュールインスタンスのアドレスに設定する。本アドレス指定に基づいて、第1のcommモジュールは、第1のモジュールインスタンスから受信

10

20

30

40

50

されたデータを、データを第2のモジュールインスタンスに伝送する異なるサブ回路図およびデバイス内の第2のcommモジュールに通信し得る。

【0105】

図19は、本発明のある実施形態による、回路図のコンポーネントをアドレス指定するための論理フロー190を図示するブロック図である。システム回路図において、コンピュータノードは、各コンポーネント（例えば、モジュール、ピン、仮想ワイヤ等）に、一意のアドレスを割り当てる。割り当てられたアドレスは、システム回路図内のコンポーネントの精密な場所を識別するために使用される。後に説明されるであろうように、本アドレスはまた、サブ回路図に解析されるとき、コンポーネントの精密な場所を識別し、コンポーネントのための通信アドレスを設定するために使用される。

【0106】

本発明におけるアドレス指定方法の実施形態は、(i) デバイスID、(ii) コンポーネントへのパス、および、(iii) コンポーネントの名称の組み合わせとして、回路図のコンポーネントをアドレス指定する。システム回路図において、デバイスIDは、設計者のシミュレーション仮想デバイスIDに設定される。サブ回路図において、デバイスIDは、サブ回路図に関連付けられる分散型異種システム内のデバイスを識別する。コンポーネントのパスは、パスが、所与のコンポーネントに到達するシステム回路図またはサブ回路図内の階層順序でトラバースされるために各モジュールインスタンスの名称を定義するように、階層パスとして定義される。図19は、モジュールインスタンスであるモジュール\_\_1を含む、システム回路図を図示する。モジュールインスタンスであるモジュール\_\_1は、デバイス\_\_A内で実行されるように構成される。システム回路図において、モジュール\_\_1は、2つのモジュールインスタンス、すなわち、モジュール\_\_2およびモジュール\_\_3を含む、回路図の中にインスタンス化される。システム回路図において、モジュール\_\_2はさらに、2つのモジュールインスタンス、すなわち、モジュール\_\_5およびモジュール\_\_6を含む、回路図の中にインスタンス化される。モジュール\_\_6は、「入力」と命名された入力ピンと、「出力」と命名された出力ピンとを有する。

【0107】

システム回路図において、シミュレーション仮想デバイスのデバイスIDは、「設計者」であって、ピン「入力」への階層パスは、「モジュール\_\_1／モジュール\_\_2／モジュール\_\_6」である。したがって、システム回路図において、ピン「入力」のアドレスは、「設計者／モジュール\_\_1／モジュール\_\_2／モジュール\_\_6／入力」である。システム回路図をサブ回路図に解析する結果として、「モジュール\_\_1」（全てのその子とともに）は、「デバイス\_\_A」に対応するサブ回路図の中に設置される。本サブ回路図において、デバイスIDは、「デバイス\_\_A」であって、ピン「入力」への階層パスは、依然として、「モジュール\_\_1／モジュール\_\_2／モジュール\_\_6」である。したがって、本サブ回路図において、ピン「入力」のアドレスは、「デバイス\_\_A／モジュール\_\_1／モジュール\_\_2／モジュール\_\_6／入力」である。システム回路図およびサブ回路図におけるピン「入力」間のアドレスの唯一の変更は、デバイスIDである一方、アドレスの残りは、保存されることに留意されたい。これは、システム回路図内の任意のコンポーネントが対応するサブ回路図内で容易に特定されることを可能にするので、本アドレス指定方法の大きな利点である。

【0108】

図20は、本発明のある実施形態による、回路図201をサブ回路図202、203に解析するとき使用されるアドレス指定の方法200を図示するブロック図である。例えば、図20は、回路図201を解析するときの図19の一意のアドレス指定方法の使用を図示する。図20は、2つのモジュールインスタンス、すなわち、「エッジ論理1」および「クラウド論理」を含む例示的システム回路図201を示す。これらの2つのモジュールインスタンスは、2つの異なるデバイス、すなわち、それぞれ、「エッジコンピュータ1」および「サーバ」内で実行されるように構成される。これらの2つのモジュールインスタンスは、「エッジ論理1」の出力が、「クラウド論理」の入力に接続されるように構

成される。

#### 【0109】

システム回路図201は、エッジコンピュータ1のための第1のサブ回路図202およびサーバ203のための第2のサブ回路図203に解析される。第1のサブ回路図202は、モジュールインスタンス「エッジ論理1」およびcommモジュールインスタンスを含み、システム回路図201内の「クラウド論理」モジュールインスタンスと「エッジ論理1」モジュールインスタンスとの間の仮想ワイヤを置換する。第1のサブ回路図202において、仮想ワイヤは、「エッジ論理1」モジュールインスタンスの出力をcommモジュールインスタンスの入力に接続する。第2のサブ回路図203は、モジュールインスタンス「クラウド論理」およびcommモジュールインスタンスを含む。第2のサブ回路図203において、仮想ワイヤは、commモジュールインスタンスの出力を「クラウド論理」モジュールインスタンスの入力に接続し、回路図201内の「クラウド論理」モジュールインスタンスと「エッジ論理1」モジュールインスタンスシステムとの間の仮想ワイヤを置換する。図19のアドレス指定方法を使用して、「クラウド論理」モジュールインスタンスの入力ピンは、「／サーバ／クラウド論理／入力」となる。

#### 【0110】

第1のサブ回路図202において、commモジュールインスタンスの宛先アドレスは、「クラウド論理」モジュールインスタンスの入力ピンのアドレス「／サーバ／クラウド論理／入力」に設定される。第2のサブ回路図202において、commモジュールの受信アドレスもまた、「クラウド論理」モジュールインスタンスの入力ピンのアドレス「／サーバ／クラウド論理／入力」に設定される。このアドレス指定に基づいて、commモジュールは、異なるサブ回路図およびデバイスにおいて、データを「クラウド論理」モジュールインスタンスと「エッジ論理1」モジュールインスタンスとの間で通信し得る。本実施形態において、第1のサブ回路図202の宛先アドレスおよび第2のサブ回路図の受信アドレスは両方とも、受信モジュールインスタンス（「クラウド論理」）上の入力ピンのアドレスである。図19に示されるように、このアドレスは、入力ピンのアドレスおよび受信モジュールインスタンスのデバイスプロパティに基づいて、容易に生成される。さらに、本アドレスを使用したアドレス指定方法は、システム回路図が、複雑な階層構造を伴う多数のモジュールインスタンスを含む場合でも、良好に機能する。

#### 【0111】

本発明におけるアドレス指定方法の別の実施形態は、(i) デバイスID、(ii) モジュールインスタンスID、および、(iii) コンポーネント名の組み合わせとして、回路図のコンポーネントをアドレス指定する。コンポーネントが、モジュールインスタンスである場合、コンポーネント名は、アドレスから省略されることができる。システム回路図において、デバイスIDは、「設計者」のシミュレーション仮想デバイスIDに設定される。サブ回路図において、デバイスIDは、サブ回路図に関連付けられる分散型異種コンピューティングおよび制御システムにおけるデバイスを識別する。コンポーネントが、モジュールインスタンスである場合、モジュールインスタンスIDは、そのモジュールインスタンスのためのものである。コンポーネントが、モジュールインスタンスではない場合、モジュールインスタンスIDは、コンポーネントに関連付けられるモジュールインスタンスのためのものである。モジュールインスタンスIDは、システム回路図内のモジュールインスタンスのための一意の識別子であって、典型的に、設計ソフトウェアによって自動生成され、ランダムストリングまたは数である。モジュールインスタンスIDは、モジュールインスタンス名がシステム回路図内で一意である必要がないという点で、モジュールインスタンス名と異なる。モジュールインスタンス名は、典型的に、ユーザによって定義され、人間によって可読である。

#### 【0112】

本アドレス指定方法を図19のシステム回路図に適用すると、「モジュール\_\_6」のピン「入力」のアドレスは、「／デバイス\_\_A／29f.ab2／入力」に設定され、「29f.ab2」は、「モジュール\_\_6」の一意的モジュールインスタンスIDである。本

10

20

30

40

50

アドレス指定方法の利点は、システム回路図が、多くの階層レベルを有するとき、アドレスがはるかに短くなり得ることである。本発明の別の実施形態において、設計ソフトウェアは、図19のパスベースのアドレス指定方法およびこのモジュールインスタンスIDベースの方法の両方を使用し得る。アドレス指定方法のそのような適用は、モジュールインスタンスIDベースのアドレスがパスベースのアドレスと区別され得るように、モジュールインスタンスIDが、一意の規則的表現、例えば、3つの英数字の後にドットが続き、その後3つのさらなる英数字を有するとき、可能性として考えられる。

#### 【0113】

図21は、本発明のある実施形態による、commモジュール210のコンポーネント211、212、213、214のブロック図である。コンポーネント211、212、213、214は、commモジュール210の機能回路図を形成する、モジュールインスタンスである。機能回路図は、ネットワークを経由して、異なるデバイス内で実行されるcommモジュール210の2つのインスタンス間の通信およびデータフローを可能にする。

10

#### 【0114】

commモジュール210の機能回路図は、受信パスを有し、それは、comm受信モジュールインスタンス211および受信処理モジュールインスタンス212を含む。comm受信モジュールインスタンス211は、ネットワークを経由してcommモジュール210に結合されるモジュールインスタンスに伝送される、データを受信する。受信処理モジュールインスタンス212は、受信されたデータを後処理する（例えば、commモジュール210に結合されるモジュールインスタンスによる使用のためのデータを準備する）。commモジュール210の機能回路図はまた、伝送パスを有し、それは、comm伝送モジュールモジュールインスタンス214および伝送処理モジュールインスタンス213を含む。伝送処理モジュールインスタンス213は、ネットワークを経由した伝送のためにデータを前処理する（例えば、ネットワークを経由した通信のために、commモジュール210に結合されるモジュールインスタンスからのデータを準備する）。comm伝送モジュールモジュールインスタンス214は、ネットワークを経由して、結合されるモジュールインスタンスからのデータを伝送する。

20

#### 【0115】

実施形態において、commモジュール210の複数の機能回路図は、ネットワークを経由してデータを通信するために、TCP、HTTP、MQTT等の異なる通信プロトコルをサポートするために実装されることができる。実施形態において、ユーザは、commモジュールインスタンスプロパティテーブル内のcommモジュールインスタンスタイププロパティを使用して、commモジュール210のインスタンスのための通信プロトコルを設定することができる。実施形態において、設計ソフトウェアに関連付けられるグラフィカルユーザインターフェースを通して、ユーザは、所与のcommモジュールインスタンスのためのcommモジュールインスタンスタイプを変更させることができる。実施形態において、設計ソフトウェアに関連付けられるグラフィカルユーザインターフェースを通して、ユーザは、全てのcommモジュールインスタンスのためのモジュールインスタンスタイプを大域的に変更させることができる。そのような様式においてcommモジュールインスタンスタイプを変更させることは、ユーザが、通信プロトコルを容易に切り替えることを可能にする。

30

40

#### 【0116】

図22は、本発明のある実施形態による、パブリッシュ／サブスクライブプロトコルを使用してデータを通信する方法220を図示するフローチャートである。実施形態において、コンピュータノードは、コンピューティングデバイス上に、中央通信(comm)ブローカを構成し、それは、ソースcommモジュールインスタンスと宛先commモジュールインスタンスとの間の通信のために、パブリッシュ／サブスクライブプロトコルを利用する。commモジュールインスタンスは、パブリッシュ機能を通信の伝送側上で実施し、サブスクライブ機能を受信側上で実施する。方法220の実行に先立って、宛先co

50

mmモジュールインスタンスは、中央c o m mブローカに、宛先c o m mモジュールインスタンスの少なくとも1つの受信アドレスを規定する、サブスクリプション要求を送送することによって、中央c o m mブローカとサブスクライブする。

【0117】

方法220の第1のプロセス121において、ソースc o m mモジュールインスタンスは、ネットワークを経由して、伝送のためのデータパケットをアSEMBルする。データパケットは、データペイロードと、宛先アドレスと、随意に、ソースアドレスとを含む。実施形態において、ソースアドレスは、ソースc o m mモジュールインスタンスに結合される、モジュールインスタンスの出力ピンのアドレスである。第2のプロセス222において、ソースc o m mモジュールインスタンスは、データパケットを中央c o m mブローカに送信する。データパケットにおける宛先アドレスが、宛先c o m mモジュールインスタンスの受信アドレスに合致しているとき、第3のプロセス223において、中央c o m mブローカは、処理するために、データパケットを宛先c o m mモジュールインスタンスに転送する。

【0118】

図22の「パブリッシュ／サブスクライブ」通信プロトコルは、図18に説明される仮想ワイヤ置換方法の代替として使用され得る。実施形態において、図19および20に説明されるアドレス指定方法は、データパケットにおける宛先アドレスおよびソースアドレスおよび宛先c o m mモジュールインスタンスの受信アドレスを構成するために使用され得る。

【0119】

図23は、本発明のある実施形態による、モジュールインスタンス間の仮想ワイヤを置換する論理フロー230を図示するブロック図である。図23の仮想ワイヤ置換方法は、図18に説明される方法の代替である。図18に示される方法180において、各サブ回路図において、設計ソフトウェアは、仮想ワイヤ毎に、一対のc o m mモジュールインスタンスを構成し、それは、システムアプリケーションへの多数のc o m mモジュールインスタンスの追加をもたらす。図23に示される方法において、所与のサブ回路図のために、設計ソフトウェアは、複数のモジュールインスタンス間で共有される、c o m mモジュールインスタンスを構成する。

【0120】

図23は、3つのモジュールインスタンス、すなわち、モジュール\_\_1 231、モジュール\_\_2 232、およびモジュール\_\_3 233、c o m mモジュールインスタンス234、およびアドレス解析モジュールインスタンス235を有する、サブ回路図を示す。設計ソフトウェアは、モジュールインスタンス（モジュール\_\_1 231、モジュール\_\_2 232、およびモジュール\_\_3 233）を構成し、c o m mモジュールインスタンス234の伝送パスを共有する。具体的に、モジュールインスタンス（モジュール\_\_1 231、モジュール\_\_2 232、およびモジュール\_\_3 233）が、仮想ワイヤ接続を経由して、データを伝送する必要があるとき、モジュールインスタンスは、その宛先アドレスおよびデータペイロードを同一c o m mモジュールインスタンス234に伝送する。c o m mモジュールインスタンス234は、次いで、宛先アドレスおよびデータペイロードを含む、データパケットを形成し、ネットワークを経由して、データパケットを送送する。

【0121】

設計ソフトウェアは、c o m mモジュールインスタンス234の受信パスを構成し、モジュールインスタンス（モジュール\_\_1 231、モジュール\_\_2 232、およびモジュール\_\_3 233）を宛先とする、全てのデータパケットを受信する。具体的に、設計ソフトウェアは、c o m mモジュールインスタンス234の受信アドレスを構成し、モジュールインスタンス（モジュール\_\_1 231、モジュール\_\_2 232、およびモジュール\_\_3 233）の入力ピンを宛先とするものを含む、サブ回路図を実行するデバイス内の任意のサブ回路図コンポーネントを宛先とする、全てのデータパケットを受信する。例えば、サブ回路図を実行するデバイスが、「デバイス\_\_A」である場合、設計ソフトウェア

は、c o m mモジュールインスタンス2 3 4が、アドレス「／デバイス\_\_A／モジュール\_\_1／入1」、「／デバイス\_\_A／モジュール\_\_2／入2」、「／デバイス\_\_A／モジュール\_\_3／入3」等の「デバイス\_\_A」内のサブ回路図のコンポーネントに合致する宛先アドレスを用いて、任意のデータパケットを受信するように、c o m mモジュールインスタンス2 3 4の受信アドレスを「／デバイス\_\_A／\*」に設定する。

【0 1 2 2】

c o m mモジュールインスタンス2 3 4は、次いで、データパケットをアドレス解析モジュール2 3 5に転送する。アドレス解析モジュール2 3 5は、データパケットの完全宛先アドレスを検査し、サブ回路図内のデータパケットの意図される宛先を決定し、データパケットをその意図される宛先に転送する。例えば、パケットの宛先アドレスが、「／デバイス\_\_A／モジュール\_\_2／入2」であるとき、アドレス解析モジュール2 3 5は、データパケットをモジュール\_\_2の入力ピン（「入2」）に転送する。

【0 1 2 3】

図2 4は、本発明のある実施形態による、デバイス内で実行するためのサブ回路図を更新する方法2 4 0を図示するフローチャートである。方法2 4 0の実行に先立って、分散型異種システム内のデバイスの組の各デバイスは、制御プログラムおよびi n i t回路図を事前にロードされる。i n i t回路図は、s y sモジュールインスタンスに結合される、c o m mモジュールインスタンスを有し、これはさらに、更新モジュールインスタンスに結合される。更新モジュールインスタンスは、デバイス上で実行されるサブ回路図の更新を実施するように構成される。各デバイスにおいて、デバイスの電源が、オンになると、制御プログラムは、デバイス上に事前にロードされたi n i t回路図を実行する。

【0 1 2 4】

設計ソフトウェアが、システム回路図をサブ回路図に解析すると、設計ソフトウェアは、実行のために、サブ回路図をその対応するデバイスに送信する。サブ回路図毎に、方法2 4 0の第1のプロセス2 4 1において、設計ソフトウェアを実行するコンピュータノードは、データパケットを対応するデバイスに送信する。データパケットは、サブ回路図と、対応するデバイス上で実行される更新モジュールインスタンスに設定される宛先アドレスとを含むデータペイロードを有する。例えば、サブ回路図が、「デバイス\_\_i d」に設定されたデバイスI Dを伴う対応するデバイス上での実行のためのものである場合、宛先アドレスは、「／デバイス\_\_i d／s y s／更新」に設定される。ある実施形態において、サブ回路図を表す回路図のテキスト表現であるネットリストが、データパケット内のデータペイロードとして使用される。第2のプロセス2 4 2において、各デバイスは、デバイスに合致するデバイス\_\_i dを含む宛先アドレスを含む、それぞれのデータパケットを受信する。宛先アドレスに基づいて、デバイスは、受信されたデータパケットをi n i t回路図のs y sモジュールインスタンスに結合される更新モジュールインスタンスに転送する。更新モジュールインスタンスは、データパケットのデータペイロードとして含まれるサブ回路図をロードし、制御プログラムにロードされたサブ回路図を実行させることによって、更新をデバイスに実施する。

【0 1 2 5】

変更が、システム回路図に行われる場合、設計ソフトウェアが、変更されたシステム回路図をサブ回路図に解析すると、方法2 4 0（プロセス2 4 1および2 4 2）は、繰り返され、対応するデバイスを変更されたシステム回路図で更新する。実施形態において、ロードされたサブ回路図はまた、s y sモジュールインスタンスに結合されるc o m mモジュールインスタンスを有し得、これはさらに、更新モジュールインスタンスに結合される。これらの実施形態のうちのいくつかにおいて、方法2 4 0を繰り返す際、デバイスは、受信されたデータパケットを、サブ回路図を実行するs y sモジュールインスタンスに結合される、更新モジュールインスタンスに転送する。

【0 1 2 6】

図2 5は、本発明のある実施形態による、i n i t回路図2 5 0のブロック図である。i n i t回路図2 5 0は、分散型異種システムのデバイス内にロードされ、実行される。

ある実施形態において、`init`回路図は、デバイスの電源が、オンにされると、デバイスが、`init`回路図を実行するように、各デバイスの中に事前にロードされる。そのような実行は、デバイスが、サブ回路図が、デバイス内にまだロードされず、実行されないときでも、設計ソフトウェアからサブ回路図更新を受信することを可能にする。

【0127】

図25に示されるように、`init`回路図250は、`sys`モジュールインスタンス251と結合される、`comm`モジュールインスタンス252を含む。デバイスID「デバイス\_\_id」を伴う所与のデバイス上で実行される、結合される`sys`モジュールインスタンス215（`comm`モジュールインスタンス252と結合される）の受信アドレスは、「／デバイス\_\_id／sys／＊」に設定される。`init`回路図250において、本受信アドレスを宛先とするサブ回路図更新が、`sys`モジュールインスタンス251によって受信され、それは、サブ回路図更新を実施し、更新内に含まれるサブ回路図をロードし、実行する。

【0128】

図26は、本発明のある実施形態による、インスタンス化された`sys`モジュール260を図示するブロック図である。ある実施形態において、インスタンス化された`sys`モジュール260は、図25の`sys`モジュールインスタンス251である。図26のインスタンス化された`sys`モジュール260（`sys`モジュールインスタンスとも呼ばれる）は、前処理モジュールインスタンス261を含む。データパケットは、データパケットの宛先アドレスがアドレス「／デバイス\_\_id／sys／＊」に合致するとき、`sys`モジュールインスタンス260によって受信され、デバイス\_\_idは、`sys`モジュールインスタンス260を実行するデバイスの識別子である。前処理モジュールインスタンス260は、受信されたデータを`sys`モジュールインスタンス260内に含まれる処理モジュールインスタンス（例えば、更新モジュールインスタンス262、`cmd`モジュールインスタンス、263等）にルーティングする。処理モジュールインスタンス262、263等の各々は、受信アドレスを有し、データパケットの具体的処理を実施するように構成される。例えば、更新モジュールインスタンス262の受信アドレスは、「／デバイス\_\_id／sys／更新」であって、サブ回路図更新を実施するように構成される。別の例に関して、`cmd`モジュールインスタンス263の受信アドレスは、「／デバイス\_\_id／sys／cmd」であって、データパケットのデータペイロード内に含まれる1つ以上のシステムコマンドを実施するように構成される。

【0129】

前処理モジュールインスタンス261は、処理モジュールインスタンスに合致するデータパケットにおける宛先アドレスに基づいて、受信されたデータパケットを処理モジュールインスタンスにルーティングする。例えば、データパケットにおける宛先アドレスが、「／デバイス\_\_id／sys／更新」に合致するとき、前処理モジュール261は、受信されたデータパケットを更新モジュールインスタンス262に転送し、それは、サブ回路図更新を実施する。本発明の一実施形態において、更新モジュールインスタンス262は、データパケットのデータペイロード内に含まれる、受信されたサブ回路図を、ファイルとして、デバイスのハードドライブ上に保存する。更新モジュールインスタンス262は、次いで、デバイス内で起動される制御プログラムに、サブ回路図を実行するように要求する（例えば、再ロード要求を介して）。別の例に関して、データパケットにおける宛先アドレスが、「／デバイス\_\_id／sys／cmd」に合致するとき、前処理モジュール261は、受信されたデータパケットを`cmd`モジュールインスタンス263に転送し、それは、データパケット内に含まれる1つ以上のシステムコマンドを実行する。

【0130】

図27は、本発明のある実施形態による、デバイス内で制御プログラムを実行する方法270を図示するフローチャートである。制御プログラムは、特定のタイプのデバイスにロードされ、そのデバイス内でサブ回路図を実行するデバイス依存ソフトウェアプログラムである。デバイスの電源が、オンにされると、方法270の第1のプロセス271にお

いて、制御プログラムは、デバイス上に事前にロードされた `init` 回路図を開始および実行する。第2のプロセス272において、制御プログラムは、次いで、サブ回路図がデバイス上に存在するかどうかをチェックする。ある実施形態において、サブ回路図が、存在する場合、ファイルとして、デバイスのハードドライブ上に保存される。制御プログラムが、そのようなサブ回路図を特定する場合、第3のプロセス273において、制御プログラムは、サブ回路図を実行する。制御プログラムが、そのようなサブ回路図を特定しない場合、第4のプロセス274において、制御プログラムは、追加のアクションを行わない。制御プログラムは、次いで、実行を継続する。

#### 【0131】

制御プログラムが、再ロードコマンドを受信すると、第5のプロセス275において、制御プログラムは、新しいサブ回路図を再ロードし、再ロードされたサブ回路図を実行する。制御プログラムは、典型的に、再ロードコマンドを更新モジュールインスタンス（例えば、図26の更新モジュールインスタンス262）から受信するが、再ロードコマンドは、デバイスの内外の他のソースから受信され得る。制御プログラムが、サブ回路図を実行すると、また、任意のプリミティブモジュールインスタンスに関連付けられたデバイス依存ブロックをサブ回路図内にロードする。制御プログラムは、サブ回路図内の各プリミティブモジュールインスタンスに関連付けられるタイププロパティに基づいて、正しいデバイス依存ブロックを選択することが可能である。

#### 【0132】

図28は、本発明のある実施形態による、低能力デバイスのための実行可能コードを提供する方法280を図示するフローチャートである。分散型異種システム内に、回路図を実行するための制御プログラムを起動する能力（例えば、処理能力、メモリ能力等）を有していない、いくつかの低能力デバイスが存在する。そのような低能力デバイスの例は、オペレーティングシステムを伴わずに構成される低電力マイクロコントローラである。

#### 【0133】

そのようなデバイスに関して、設計ソフトウェアは、図28の方法280に示されるように、制御プログラムを伴わずにこれらのデバイス上で実行可能な形態に、サブ回路図をさらに処理する。方法280の第1のプロセス281において、設計ソフトウェアを実行するコンピュータノードは、そのようなデバイスのためのサブ回路図を生成する。第2のプロセス282において、コンピュータノードは、次いで、サブ回路図をコンパイラと適合性があるコードに処理する。ある実施形態において、サブ回路図は、設計ソフトウェアによって内部から処理され、別の実施形態において、サブ回路図は、外部コンバータソフトウェアを通して処理される。第3のプロセス283において、コンピュータノードは、外部コンパイラソフトウェアを起動し、コンパイラと適合性があるコードをデバイスコードの中にコンパイルし、それは、サブ回路図に対応する標的低能力デバイス内で実行されることができる。第4のプロセス284において、コンピュータノードは、デバイスコードを標的デバイスに送信し、それは、デバイスコードをロードおよび実行し得る、ブートローダとともに構成される。

#### 【0134】

図29は、本発明のある実施形態による、分散型異種システムのためのデータ可視化を提供するモジュールインスタンスを含む回路図のブロック図である。当業者によって公知のように、従来の設計ソフトウェアは、非分散型システムのためのデータ可視化能力を提供し得、データの可視化は、中央場所において実施されることができる。しかしながら、データ可視化は、分散型異種システムにおいて、それほど簡単ではない。分散型異種システムにおいて、可視化コンポーネントは、そのような中央場所がそのようなシステム内でデータを可視化するために存在しない、分散型デバイス内で実行されなければならない。本発明の設計ソフトウェアは、可視化のために、分散型デバイスからの全ての可視化コンポーネントを設計ソフトウェアの中に集中させる様式において、可視化機能性を提供する。

#### 【0135】

本発明の設計ソフトウェアにおいて、ユーザは、設計ソフトウェアに関連付けられるグ

10

20

30

40

50

グラフィカルユーザインターフェースを通して、可視化モジュールインスタンスを回路図内のモジュールインスタンスのピンおよび仮想ワイヤに追加するためのオプションを提供される。図29は、システム回路図290を示し、それは、2つの可視化モジュールインスタンス、すなわち、プローブモジュールインスタンス291およびチャートモジュールインスタンス292の追加を伴う図13のシステム回路図130である。プローブモジュールインスタンス291は、「クラウド論理」モジュールインスタンスおよび「エッジ論理1」モジュールインスタンスを接続する第1の仮想ワイヤに結合されて追加される。プローブモジュールインスタンス291は、第1の仮想ワイヤ上で通信されるデータを捕捉および印刷する。チャートモジュールインスタンス292は、「クラウド論理」モジュールインスタンスおよび「エッジ論理3」モジュールインスタンスを接続する第2の仮想ワイヤに結合されて追加される。チャートモジュールインスタンス292は、第2の仮想ワイヤ上で通信されるデータを捕捉およびプロットする。チャートモジュールインスタンス292は、設計ソフトウェアに関連付けられるグラフィカルユーザインターフェースからユーザ選択可能なあるフォーマットでデータをプロットし得る。モジュールインスタンス291および292がデータを受信した、仮想ワイヤは、実際に、分散型異種デバイス上の実行されるサブ回路図内に存在しないので、プローブおよびチャートモジュールインスタンス291、292は、従来の設計ソフトウェアにおけるものと同一様式で機能しない。そのような可視化モジュールインスタンスの構成および機能はさらに、図30に説明される。

#### 【0136】

図30は、本発明のある実施形態による、データ可視化を実施するためのサブ回路図300のブロック図である。実施形態において、本発明の設計ソフトウェアは、「設計者」に設定されたデバイス\_\_idを伴う仮想デバイスを作成する。ユーザが、回路図内において、プローブモジュールインスタンス301およびチャートモジュールインスタンス302を仮想ワイヤに接続することを選択すると、設計ソフトウェアは、これらのモジュールインスタンス291、292のためのデバイスプロパティを「設計者」のデバイス\_\_idに設定する。解析プロセス中、設計ソフトウェアは、設計者デバイスのために、図30のサブ回路図300を生成し、このデバイスプロパティに設定される全てのモジュールを、プローブモジュールインスタンス301およびチャートモジュールインスタンス302を含む、サブ回路図300に解析する。解析プロセス中、設計ソフトウェアはまた、図18に説明されるようなcommモジュールインスタンス303、304を追加し、それに対してプローブおよびプロットモジュールインスタンス301、302が接続される、仮想ワイヤを置換する。設計ソフトウェアは、プローブモジュールインスタンス301を追加されたcommモジュールインスタンス303に結合し、チャートモジュールインスタンスを追加されたcommモジュールインスタンス304に結合する。

#### 【0137】

ある実施形態において、設計者デバイスのためのサブ回路図は、ユーザが設計ソフトウェアの内側のデータを可視化することを可能にするシミュレーションコンテキストにおいて、設計ソフトウェアによって実行される。本実施形態において、設計者デバイスは、デバイスが、設計ソフトウェアの内側で実行され、分散型異種システム内の実デバイスを表さないため、仮想デバイスと呼ばれる。しかしながら、仮想化能力を実行するためのデバイスのタイプに関する設計ソフトウェア内に、制限が存在しない。別の実施形態において、仮想デバイス識別子は、実デバイス識別子によって置換され得、サブ回路図300は、その実デバイス内で実行されることができる。

#### 【0138】

図31は、本発明のある実施形態による、仮想デバイスを通したシミュレーションコンテキストにおける実行のために回路図を更新する方法310を図示するフローチャートである。実施形態において、挙動回路図と称される、特殊タイプの回路図もまた、存在する。挙動回路図は、典型的に、簡略化されたモデルまたは統計的モジュール等の機能回路図またはデバイス依存ブロックのモデルである。機能回路図と異なり、挙動回路図は、分散

型異種システムの物理的デバイス上での実行のためではなく、シミュレーションコンテキストにおいて使用される。シミュレーションコンテキストは、典型的に、統計的変動、コーナーケース、および特殊入力条件等の分散型異種システムに関連付けられる問題点を理解するために、設計ソフトウェア内で実施される。

#### 【0139】

方法310の第1のプロセス311において、設計ソフトウェアを実行するコンピュータノードは、システム回路図内の各モジュールインスタンスを特定する。第2のプロセス312において、コンピュータノードは、各特定されたモジュールインスタンスのデバイスプロパティを「設計者」の値に設定する。設計者デバイスは、シミュレーションコンテキストにおいて設計ソフトウェアの内側で実行される、仮想デバイスである。第3のプロセス313において、システム回路図内にインスタンスを伴うモジュール毎に、コンピュータノードは、モジュールが、「回路図2」と命名された挙動回路図に設定されるコンポーネントを含むかどうかを決定する。方法310において、「回路図2」は、挙動回路図のためのデフォルト名称であるが、他の方法において、挙動回路図のデフォルト名称は、限定ではないが、任意の他の名称であり得る。「回路図2」が、所与のモジュール内に含まれる場合、第4のプロセス314において、設計ソフトウェアは、システム回路図内の所与のモジュールのインスタンス毎のモジュールインスタンスタイプを「回路図2」に設定する。「回路図2」が、所与のモジュール内に含まれない場合、第5のプロセス315において、設計ソフトウェアは、システム回路図内のそのモジュールのインスタンスにアクションを行わない。

#### 【0140】

設計ソフトウェアが、システム回路図を解析すると、仮想設計者デバイスに対応する1つのみのサブ回路図が、生成され、各モジュールインスタンスは、「回路図2」コンポーネントが利用可能である場合、「回路図2」に設定される。1つのサブ回路図が、設計ソフトウェアの内部で展開され、設計ソフトウェアが、本サブ回路図を実行すると、シミュレーションコンテキスト内で起動される。方法310は、種々の他の実施形態において実装され得る。ある実施形態において、挙動回路図は、設計ソフトウェアの内側のシミュレーションコンテキストにおける実行に制限される必要はない。本実施形態において、ユーザは、設計ソフトウェアに関連付けられるグラフィカルユーザインターフェースを通して、1つ以上のデバイスタイプを手動で選択し得、挙動回路図は、シミュレーションコンテキストにおいて、デバイス内において実行される、または選択された1つまたはそれを上回るデバイスタイプに対応する複数のデバイスを通して分散されることができる。別の実施形態において、ユーザは、「回路図2」以外の挙動回路図を選択することができ、それは、複数の挙動回路図を要求する、複数の挙動がシミュレートされる必要があるとき、有用である。

#### 【0141】

本発明は、プロセッサ（例えば、マイクロプロセッサ、マイクロコントローラ、デジタル信号プロセッサ、または汎用コンピュータ）とともに使用するためのコンピュータプログラム論理、プログラム可能論理デバイス（例えば、フィールドプログラマブルゲートアレイ（FPGA）または他のPLD）とともに使用するためのプログラム可能論理、離散構成要素、集積回路（例えば、特定用途向け集積回路（ASIC））、またはそれらの任意の組み合わせを含む任意の他の手段を含むが、決してそれらに限定されない、多くの異なる形態で具現化され得る。

#### 【0142】

本明細書で以前に説明されている機能性の全体または一部を実装するコンピュータプログラム論理は、ソースコード形態、コンピュータ実行可能形態、および種々の中間形態（例えば、アセンブラ、コンパイラ、ネットワークカ、またはロケータによって生成される形態）を含むが、決してそれらに限定されない、種々の形態で具現化され得る。ソースコードは、種々のオペレーティングシステムまたはオペレーティング環境とともに使用するための種々のプログラミング言語（例えば、オブジェクトコード、アセンブリ言語、または

Fortran、C、C++、JAVA（登録商標）、あるいはHTML等の高レベル言語）のうちのいずれかで実装される、一連のコンピュータプログラム命令を含み得る。ソースコードは、種々のデータ構造および通信メッセージを定義し、使用し得る。ソースコードは、（例えば、インタープリタを介した）コンピュータ実行可能形態であり得、またはソースコードは、（例えば、翻訳装置、アセンブラ、またはコンパイラを介して）コンピュータ実行可能形態に変換され得る。

【0143】

コンピュータプログラムは、半導体メモリデバイス（例えば、RAM、ROM、PROM、EEPROM、またはフラッシュプログラマブルRAM）、磁気メモリデバイス（例えば、ディスクまたは固定ディスク）、光学メモリデバイス（例えば、CD-ROM）、PCカード（例えば、PCMCIAカード）、または他のメモリデバイス等の、有形記憶媒体の中で永久的または一時的に、任意の形態（例えば、ソースコード形態、コンピュータ実行可能形態、または中間形態）で固定され得る。コンピュータプログラムは、アナログ技術、デジタル技術、光学技術、無線技術、ネットワーキング技術、およびインターネットワーキング技術を含むが、決してそれらに限定されない、種々の通信技術のうちのいずれかを使用してコンピュータに伝送可能である信号の中で、任意の形態で固定され得る。コンピュータプログラムは、添付印刷または電子文書（例えば、収縮包装されたソフトウェアまたは磁気テープ）を伴う可撤性記憶媒体として任意の形態で配信され、（例えば、システムROMまたは固定ディスク上で）コンピュータシステムに事前搭載され、または通信システム（例えば、インターネットまたはワールドワイドウェブ）上でサーバまたは電子掲示板から配信され得る。

【0144】

本明細書で以前に説明されている機能性の全体または一部を実装するハードウェア論理（プログラム可能論理デバイスとともに使用するためのプログラム可能論理を含む）は、従来の手動方法を使用して設計され得、あるいは、コンピュータ支援設計（CAD）、ハードウェア記述言語（例えば、VHDLまたはAHDL）、またはPLDプログラミング言語（例えば、PALASM、ABEL、またはCUPPL）等の種々のツールを使用して、設計され、捕捉され、シミュレートされ、または電子的に文書化され得る。

【0145】

本発明は、具体的実施形態を参照して具体的に示され、説明されているが、添付請求項によって定義されるような本発明の精神および範囲から逸脱することなく、形態および詳細の種々の変更が行われてもよいことが、当業者によって理解されるであろう。これらの実施形態のいくつかは、請求項ではプロセスステップによって説明されているが、下記の請求項におけるプロセスステップを実行することが可能な関連ディスプレイを伴うコンピュータを備える装置も、本発明に含まれる。同様に、以下の請求項におけるプロセスステップを実行するためのコンピュータ実行可能命令を含み、コンピュータ可読媒体上に記憶される、コンピュータプログラム製品が、本発明内に含まれる。

【0146】

上記に説明される本発明の実施形態は、単に、例示的であることが意図される。多数の変形例および修正が、当業者に明白となるであろう。全てのそのような変形例および修正は、任意の添付の請求項に定義される本発明の範囲内であることが意図される。

10

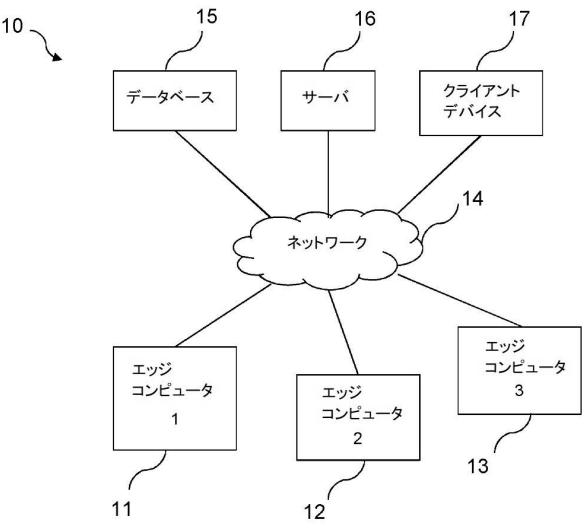
20

30

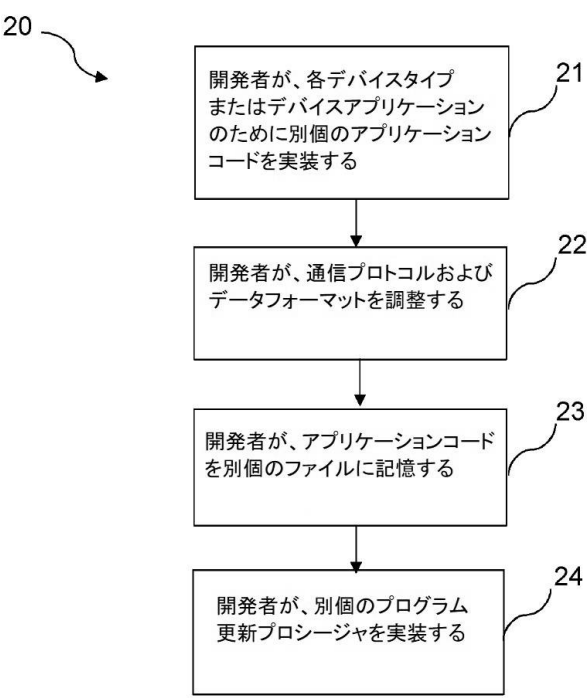
40

50

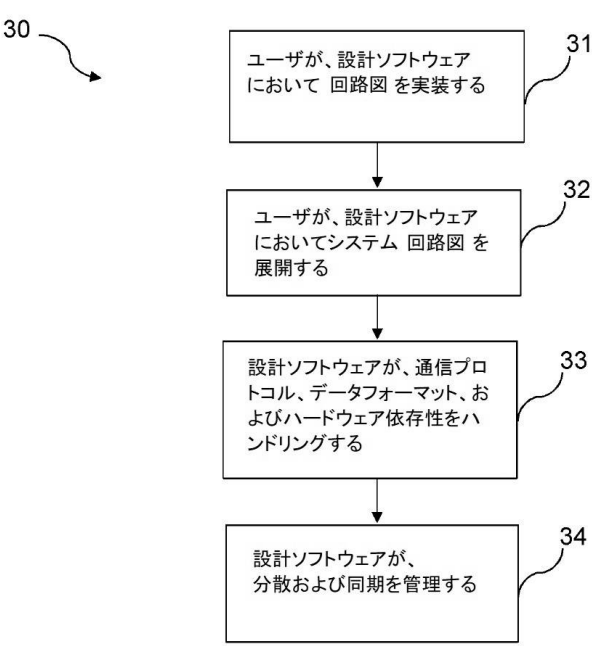
【図面】  
【図 1】



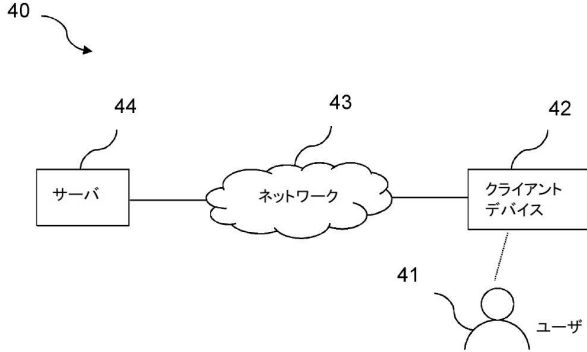
【図 2】



【図 3】



【図 4】



10

20

30

40

50

【図 5】

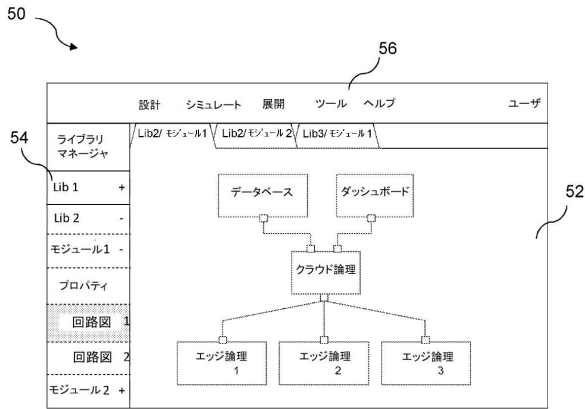


Fig. 5

【図 6】

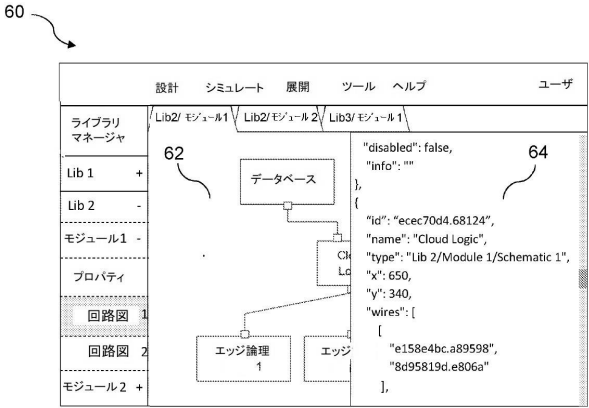


Fig. 6

【図 7】

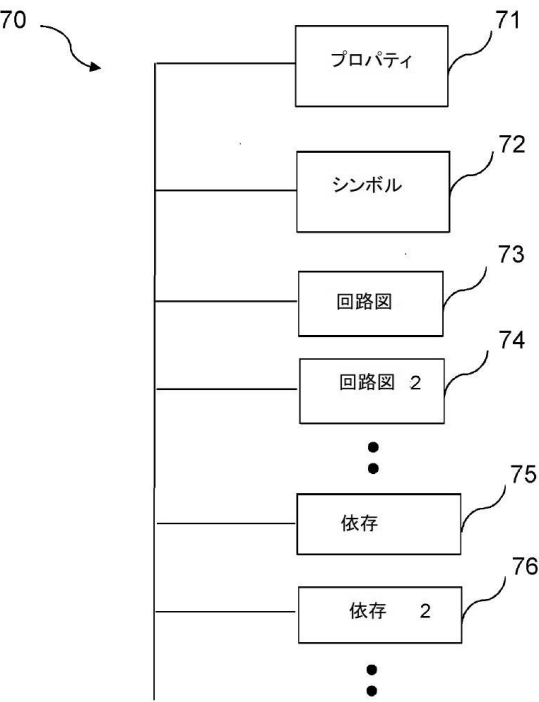


Fig. 7

【図 8】

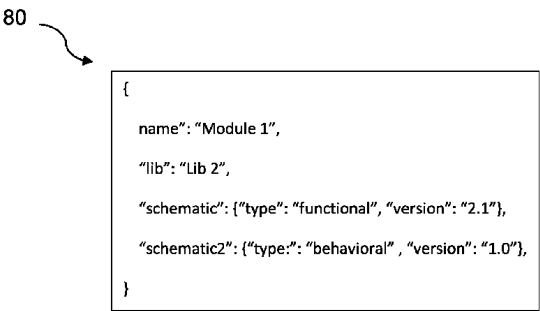


Fig. 8

【図 9】

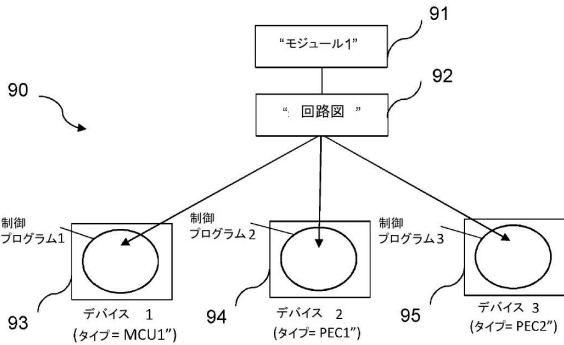


Fig. 9

【図 10】

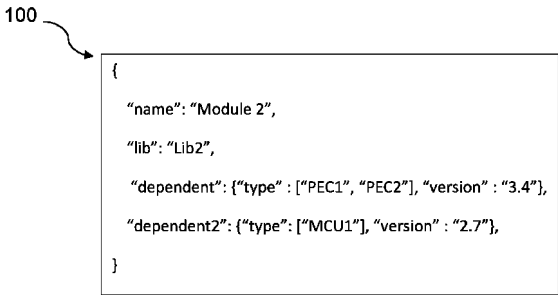


Fig. 10

【図 11】

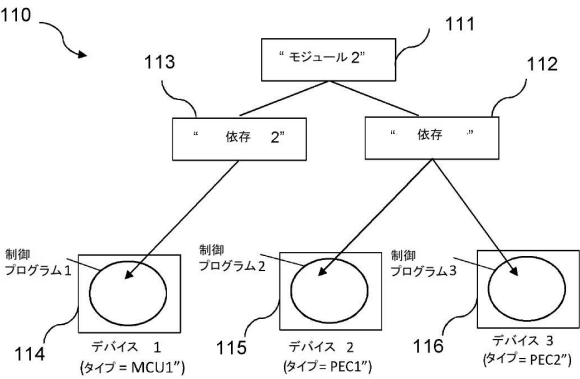


Figure 11

【図 12】

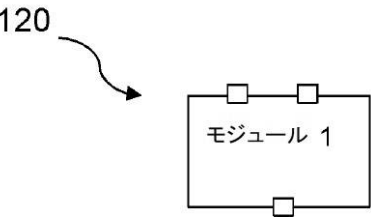


Fig. 12

【図 1 3】

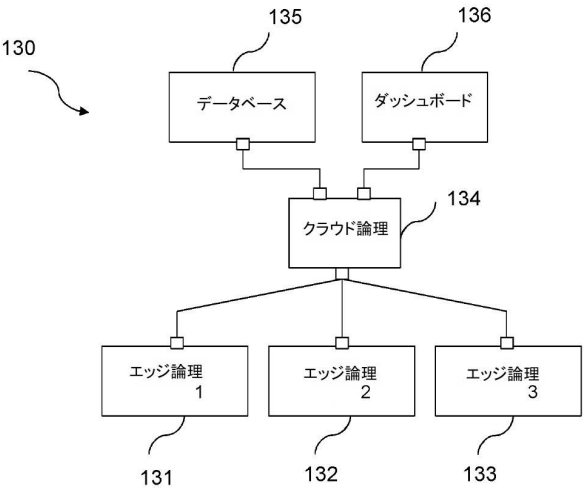


Fig. 13

【図 1 4】

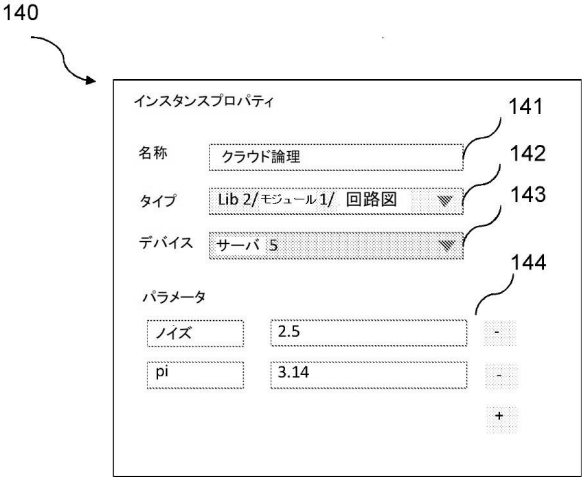


Fig. 14

【図 1 5】

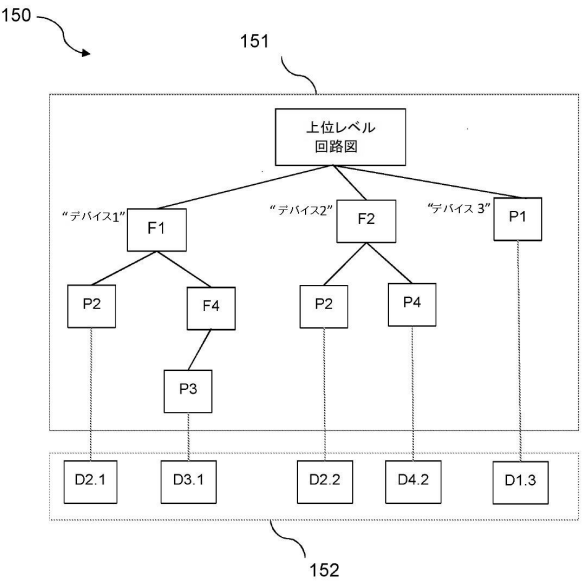


Fig. 15

【図 1 6】

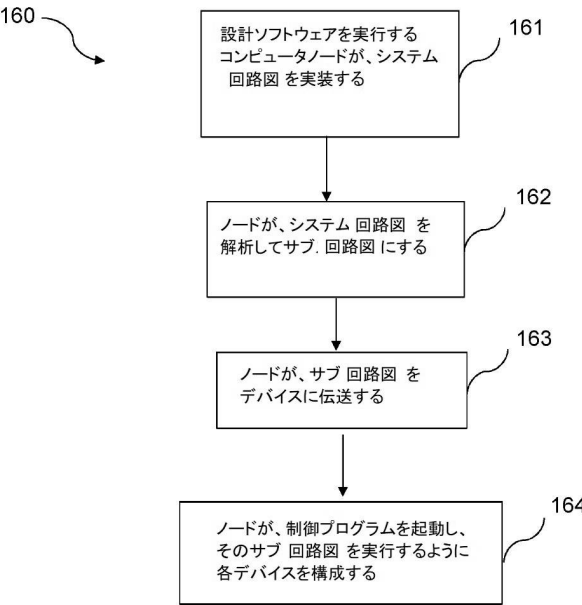


Fig. 16

10

20

30

40

50

【図 17】

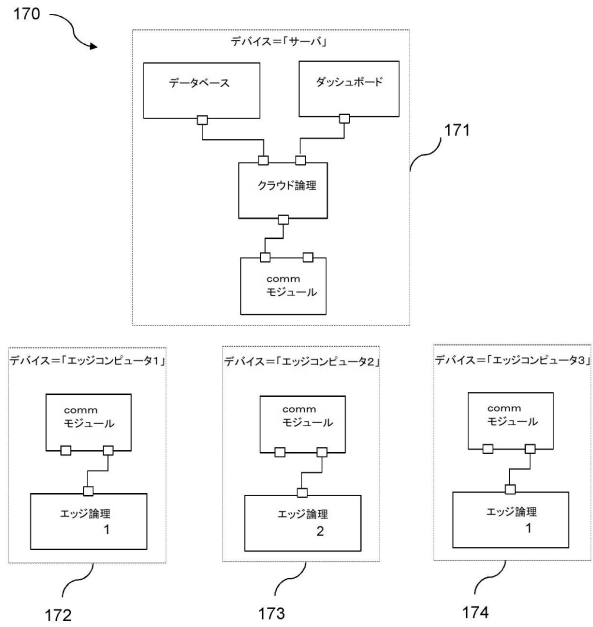


Fig. 17

【図 18】

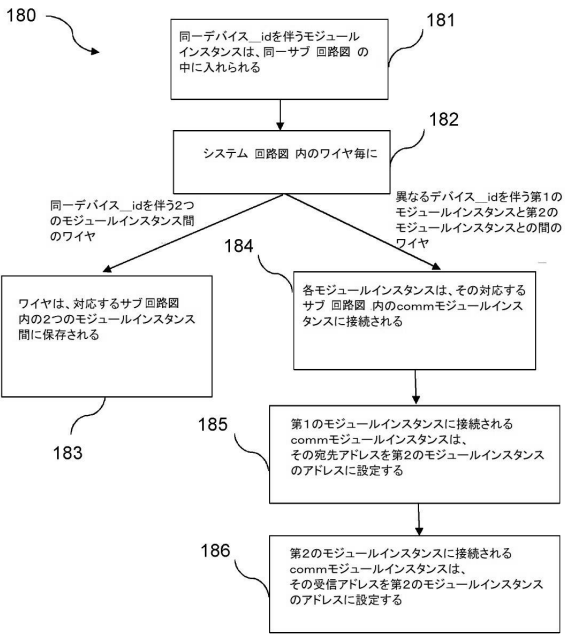


Fig. 18

【図 19】

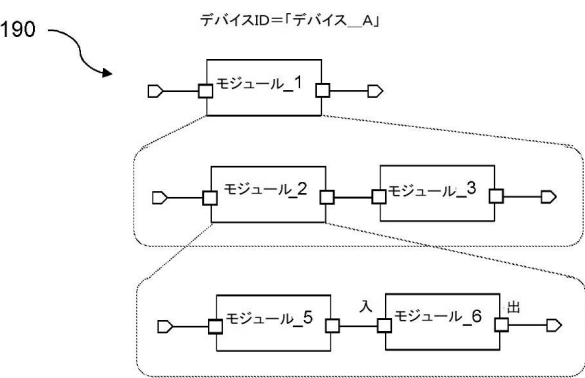


Fig. 19

【図 20】

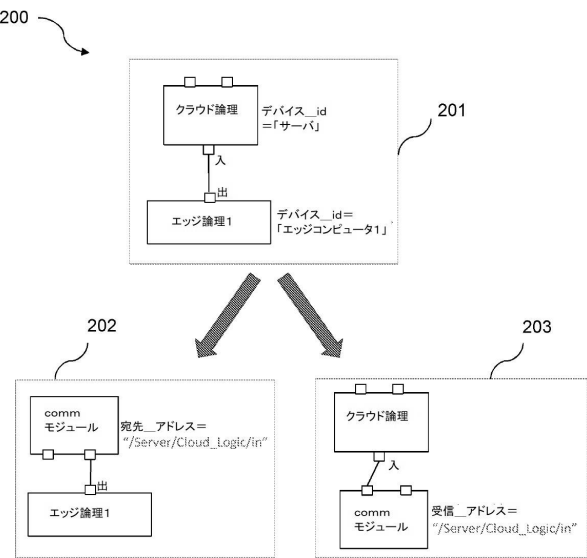


Fig. 20

【図 2 1】

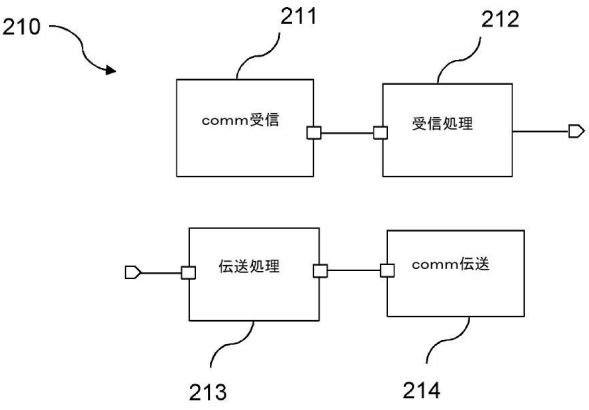


Fig. 21

【図 2 2】

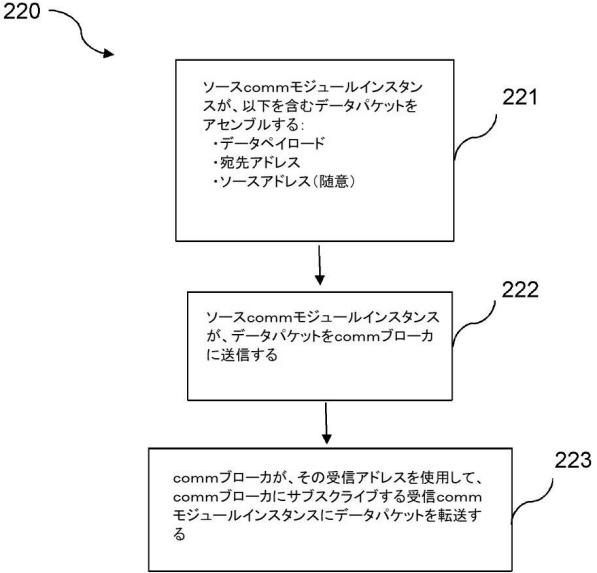


Fig. 22

【図 2 3】

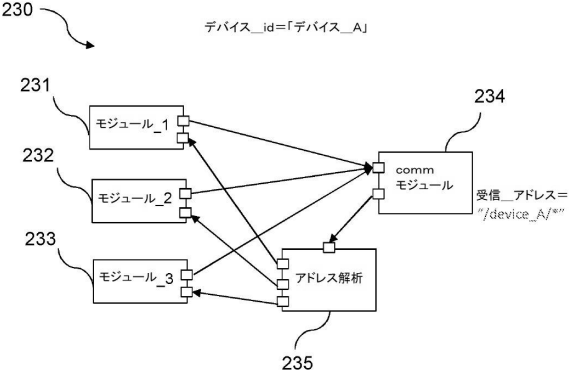


Fig. 23

【図 2 4】

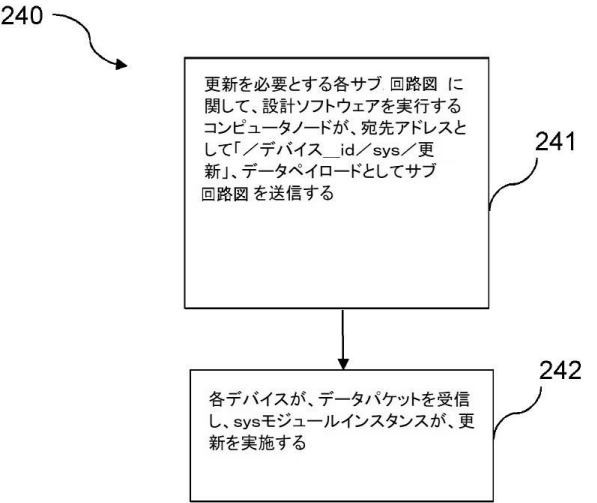


Fig. 24

10

20

30

40

50

【図 2 5】

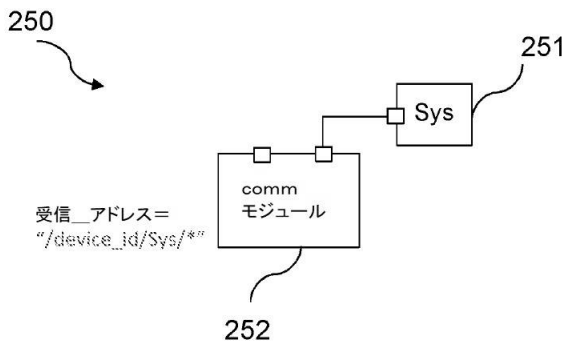


Fig. 25

【図 2 6】

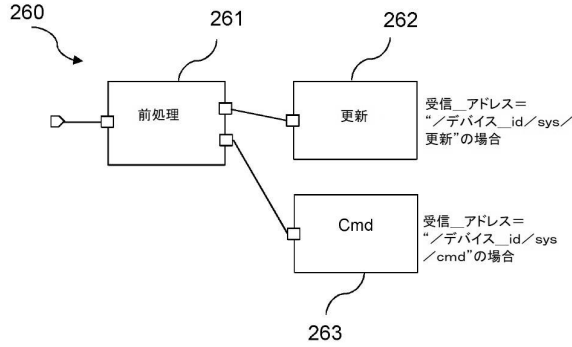


Fig. 26

【図 2 7】

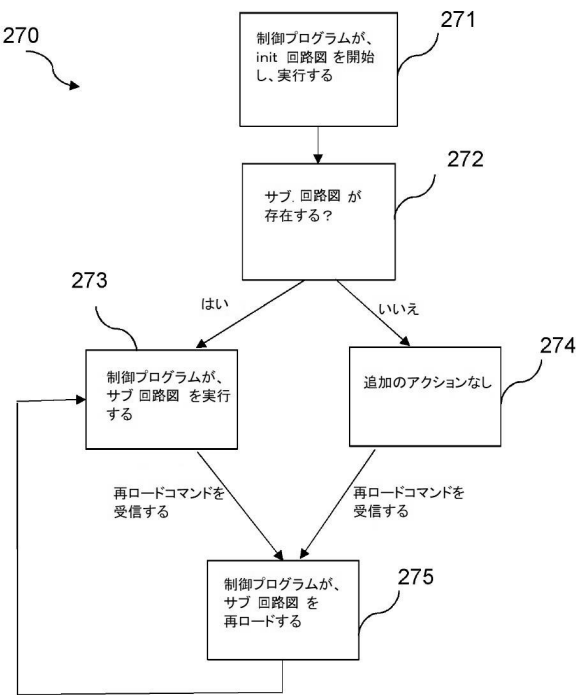


Fig. 27

【図 2 8】

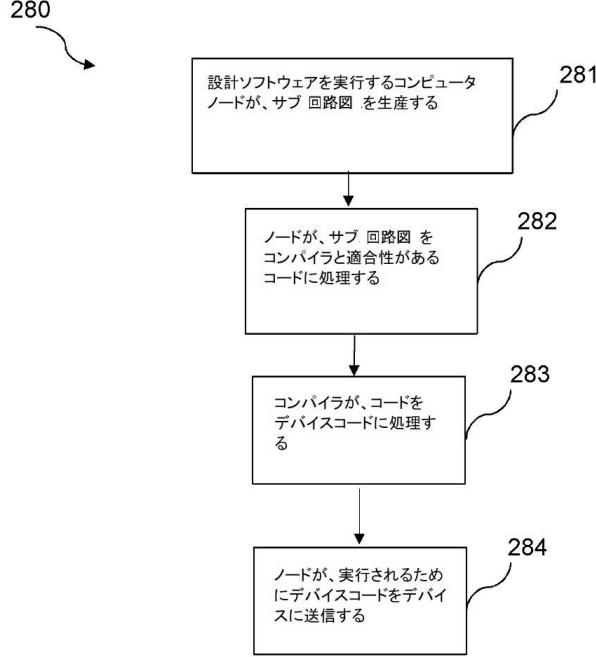


Fig. 28

10

20

30

40

50

【図 29】

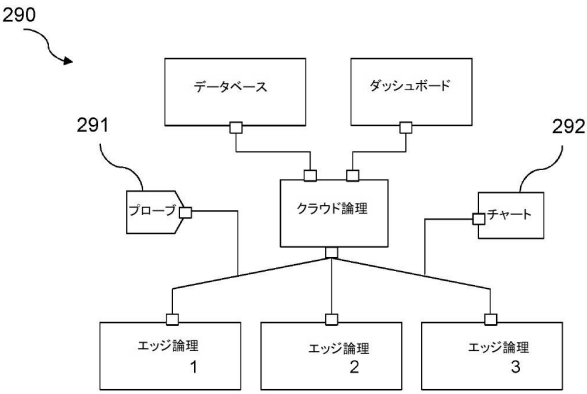


Fig. 29

【図 30】

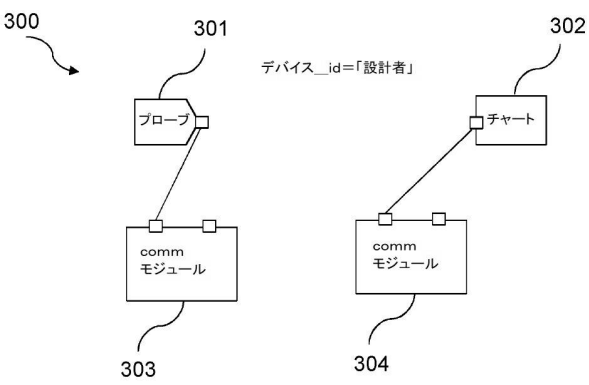


Fig. 30

【図 31】

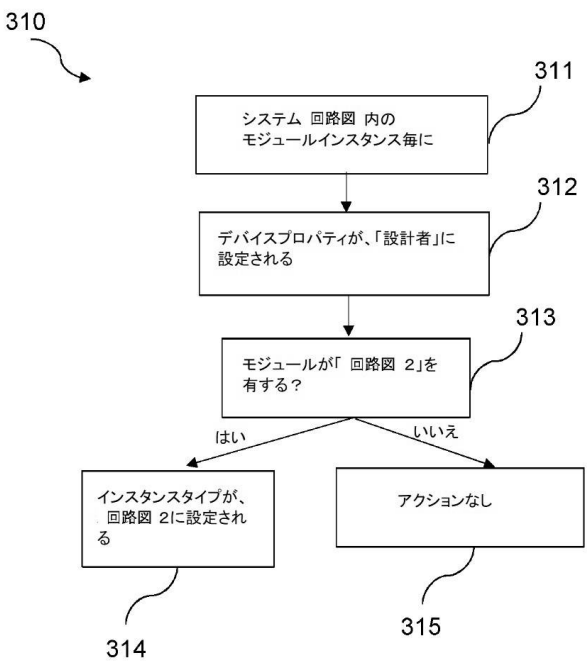


Fig. 31

---

フロントページの続き

(33)優先権主張国・地域又は機関

米国(US)

弁護士 山本 健策

(72)発明者 ワン, アンドリュー

アメリカ合衆国 マサチューセッツ 01450, グロットン, リバーベンド ドライブ 281

審査官 円子 英紀

(56)参考文献 国際公開第2017/106219 (WO, A1)

特開2004-094950 (JP, A)

米国特許出願公開第2007/0294074 (US, A1)

(58)調査した分野 (Int.Cl., DB名)

G06F 8/00-8/38

G06F 8/60-8/77

G06F 9/44-9/54

G06F 30/00-30/398

G16Y 10/00-40/60