(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5149254号
(P5149254)
(45)発行日　平成25年2月20日(2013.2.20)　(24)登録日　平成24年12月7日(2012.12.7)

(51)Int.Cl.　F I
G05B 19/05 (2006.01)　G05B 19/05 W

請求項の数 13　外国語出願　(全 11 頁)

(21)出願番号　特願2009-177544(P2009-177544)
(22)出願日　平成21年7月30日(2009.7.30)
(65)公開番号　特開2010-61649(P2010-61649A)
(43)公開日　平成22年3月18日(2010.3.18)
審査請求日　平成21年10月2日(2009.10.2)
(31)優先権主張番号　GM 420/2008
(32)優先日　平成20年8月4日(2008.8.4)
(33)優先権主張国　オーストリア(AT)

(73)特許権者　398055255
アー・ファウ・エル・リスト・ゲゼルシャフト・ミト・ベシュレンクテル・ハフツング
オーストリア国、８０２０グラーツ、ハンス－リスト－プラッツ、１
(74)代理人　100069556
弁理士　江崎　光史
(74)代理人　100157440
弁理士　今村　良太
(74)代理人　100153419
弁理士　清田　栄章
(74)代理人　100111486
弁理士　鍛冶澤　實

最終頁に続く

(54)【発明の名称】実行可能な設定の生成

(57)【特許請求の範囲】
【請求項１】
ターゲットシステム上の自動化タスクを達成するために、ターゲットシステムに実行可能な設定（２５）を生成するための方法であって、－ライブラリ（２０）から、ターゲットシステムから独立したいくつかの抽出されたソフトウェア構成要素（２０－ａ，２０－ｂ，…２０－ｍ）が、選択され、かつ、ターゲットシステムの個々の機能を実行し、かつ、データチャネルへの特定のコマンドインターフェース（２２）及び特定のインターフェース（２１）を備え、－選択されたターゲットシステムから独立したソフトウェア構成要素（２０－ａ，２０－ｂ，…２０－ｍ）が、自動化タスクを実現するために、データチャネルを介して、モデル（２３）と相互接続され、－モデル（２３）は、モデル変換によって、ノード、属性、及び、エッジからなるネットリスト（２４）に、自動的に変換し、この場合、ソフトウェア構成要素（２０－ａ，２０－ｂ，…２０－ｍ）は、ノードを構成し、かつ、初期パラメータは、ネットリスト（２４）の属性とデータチャネルとエッジを構成する方法において、－ネットリスト（２４）のターゲットシステムから独立したソフトウェア構成要素（２０－ａ，２０－ｂ，…２０－ｍ）は、実行するための設定（２５）のターゲットシステムに依存したソフトウェア構成要素に変換され、－実行するための設定（２５）のターゲットシステムに依存したソフトウェア構成要素は、ターゲットシステムで構成要素の構造を保持しながらインスタント化され、かつ、コマンドインターフェースを介してパラメータ化され、
－ネットリスト（２４）から、正確なシグナルフローで処理するために、ソフトウェア構

成要素の処理順序がターゲットシステムで算出されることを特徴とする方法。
【請求項２】
請求項１に記載の方法において、フィルタ、及び／又は、サンプリング素子は、シグナルフローで周波数変動がある場合に、自動的に提案され、又は、実行可能な設定（２５）に挿入されることを特徴とする方法。
【請求項３】
請求項１又は２に記載の方法において、ソフトウェア構成要素は、ターゲットシステムで、少なくとも部分的に、異なるスレッドで処理されることを特徴とする方法。
【請求項４】
請求項３に記載の方法において、異なるスレッドは、少なくとも部分的に、異なる周波数と同期していることを特徴とする方法。
【請求項５】
請求項３又は４に記載の方法において、一般プロセスは、ターゲットシステムで実行され、処理エリア、メールボックス、及び、スレッドが利用可能であり、そこにソフトウェア構成要素は、ダイナミックロード可能なライブラリとして配置され、かつ、できるだけ多くインスタント化されることを特徴とする方法。
【請求項６】
請求項３に記載の方法において、ソフトウェア構成要素は、トリガーを備え、トリガー情報は、自動的及びユーザが無意識に、処理順序に沿って送られることを特徴とする方法。
【請求項７】
請求項６に記載の方法において、ソフトウェア構成要素のトリガーに、時間割当が割り当てられ、この時間割当を超えたソフトウェア構成要素は、一時停止される、又は、警告が送付されることを特徴とする方法。
【請求項８】
請求項１乃至７のいずれか１つに記載の方法において、ソフトウェア構成要素は、ターゲットシステムで、１サイクルで、１回以上実行されることを特徴とする方法。
【請求項９】
請求項１乃至８のいずれか１つに記載の方法において、処理順序は、新しい処理を準備する場合、２つの完全なサイクルの間の実行時間中に、切替、又は、変更され、かつ、この切替／変更が１つのサイクルの端部と次のサイクルの始めの間で同期して行われることを特徴とする方法。
【請求項１０】
請求項１乃至９のいずれか１つに記載の方法において、実行可能な設定の変更は、デルタネットリストによって達成され、そのために、デルタネットリストには、取り除かれるべきネット部分、及び、追加されたネット部分が記載され、かつ、変更は、処理順序に応じた好ましい時点で起きることを特徴とする方法。
【請求項１１】
請求項１乃至１０のいずれか１つに記載の方法において、ソフトウェア構成要素は、ターゲットシステムの異なるＣＰＵ、又は、異なるＣＰＵのコアによって、実行されることを特徴とする方法。
【請求項１２】
ターゲットシステム上の自動化タスクを達成するために、ターゲットシステムに実行可能な設定（２５）を生成するための装置であって、開発ユニット（３０）は、ターゲットシステムと接続され、かつ、－ライブラリ（２０）が備えられ、そのライブラリ（２０）から、ターゲットシステムから独立したたくさんの抽出された、いくつかのソフトウェア構成要素がユーザインターフェース（２６）を介して選択され、かつ、このソフトウェア構成要素は、ターゲットシステムの個々の機能を実行し、かつ、データチャネルへの特定のコマンドインターフェース（２２）及び特定のインターフェース（２１）を備え、かつ、ターゲットシステムから独立した選択されたソフトウェア構成要素（２０－ａ，２０－

ｂ，…２０－ｍ）は、データチャネルを介して、自動化タスクを実現するために、モデル（２３）と相互接続されており、－モデル変換ユニット（２７）が備えられ、このモデル変換ユニット（２７）は、モデル（２３）を、モデル変換によって、ノード、属性、及び、エッジからなるネットリスト（２４）に、自動的に変換し、それによって、ソフトウェア構成要素は、ノードを構成し、かつ、初期パラメータは、ネットリスト（２４）の属性とデータチャネルとエッジを構成する装置において、－ターゲットシステム変換ユニット（２８）が備えられ、このターゲットシステム変換ユニットは、ネットリスト（２４）のターゲットシステムから独立したソフトウェア構成要素（２０－ａ，２０－ｂ，…２０－ｍ）を、実行するための設定（２５）のターゲットシステムに依存したソフトウェア構成要素に変換し、このターゲットシステムに依存したソフトウェア構成要素は、ターゲットシステムで構成要素の構造を保持しながらインスタント化され、かつ、コマンドインターフェースを介してパラメータ化され、ネットリスト（２４）から、正確なシグナルフローで処理するために、ソフトウェア構成要素の処理順序がターゲットシステムで算出されることを特徴とする装置。

【請求項１３】

請求項１２に記載の装置において、ターゲットシステムは、マルチＣＰＵ又はマルチコアシステムとして設計され、かつ、ソフトウェア構成要素は、ターゲットシステムの異なるＣＰＵ、又は、ＣＰＵの異なるコア上で実行されることを特徴とする装置。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、ターゲットシステム上で自動化タスク、好ましくは、自動車、又は、自動車構成要素の開発のための自動化ユニットを実現するために、ターゲットシステム上に実行可能な設定を生成するための方法、及び、ターゲットシステム上で自動化タスク、好ましくは、自動車、又は、自動車構成要素の開発のための自動化ユニットを実現するために、ターゲットシステム上に実行可能な設定を生成、かつ、実行するための装置に関する。

【背景技術】

【０００２】

自動車、又は、自動車構成要素の開発のための現在の試験台は、多数の要求を満たさなければならなく、かつ、例えば、様々な試験台構成要素及び試験走行の制御やレギュレーション、見本で様々の測定の実行、試験台とデータの視覚化等といった多数の自動化タスクに及ばなければならない。しかし、現在の試験台の自動化ユニットは、主として、歴史的な理由のために、様々な試験台構成要素が様々なインターフェースによって接続されている中央システムである。結果として、中央データと情報集積は、全てのユーティリティとサービスが抽出可能な自動化ユニットで生成される。しかし、この結果として、新しい構成要素（ＳＷ及びＨＷ）を統合するために（例えば、新しいインターフェース、構成要素から/へデータの変化した処理、存在するシステムへの統合の適用等）、自動化ユニットに、多数の変化が必要とされる場合、柔軟性のないシステムが、限られたスケーラビリティ（拡張性）で生成される。これは、自動化ユニットの変更又は増強、及び、特別な顧客の要求を非常に高価にしている。さらに、結果として、たくさんの異なる構成が維持することが必要とされ、これは存在するシステムのメンテナンスを高価にさせる。存在する中央自動化ユニットの更なる問題は、例えば、顧客の特別のレギュレータ、又は、自動化ユニットへの特別な試験方法のような外部の機能を統合するための簡単な実現性の欠如である。その上、中央システムは、中央ですべてコンピュータ処理しなければならなく、結果として、ＣＰＵの容量の限界にすぐ達する。

【先行技術文献】

【特許文献】

【０００３】

【特許文献１】欧州特許出願公開第１３５１１０９号明細書

【発明の概要】

【発明が解決しようとする課題】
【０００４】
この理由のため、ターゲットシステムで実行可能なプログラム、特に、一方、柔軟で、他方で、インストールすることが簡単かつすばやくでき、オペレータの要求に順応可能な、自動車、又は、自動車構成要素の開発のための自動化ユニットを生成するための装置及び方法を示すことが、本発明の課題である。さらなる課題は、外部の機能でさえ、ターゲットシステムに容易に統合可能にさせるユニットをデザインすることである。
【課題を解決するための手段】
【０００５】
この課題は、ライブラリから、ターゲットシステムから独立したいくつかの抽出されたソフトウェア構成要素が、選択され、かつ、ターゲットシステムの個々の機能を実行し、かつ、データチャネルへの特定のコマンドインターフェース及び特定のインターフェースを備え、選択されたターゲットシステムから独立したソフトウェア構成要素が、自動化タスクを実現するために、データチャネルを介して、モデルと相互接続され、モデルは、モデル変換によって、ノード、属性、及び、エッジからなるネットリストに、自動的に変換し、この場合、ソフトウェア構成要素は、ノードを構成し、かつ、初期パラメータは、ネットリストの属性とデータチャネルとエッジを構成し、ターゲットシステムから独立したソフトウェア構成要素は、ターゲットシステムに依存したソフトウェア構成要素に変換され、ターゲットシステムに依存したソフトウェア構成要素は、ターゲットシステムでインスタント化され、かつ、コマンドインターフェースを介してパラメータかされ、ネットリストから、正確なシグナルフローで処理するために、ソフトウェア構成要素の処理順序が算出される方法によって解決される。装置は、開発ユニットによって、この課題を解決し、この開発ユニットは、ターゲットシステムと接続され、かつ、ライブラリが備えられ、そのライブラリから、ターゲットシステムから独立したたくさんの抽出された、いくつかのソフトウェア構成要素がユーザインターフェースを介して選択され、かつ、このソフトウェア構成要素は、、ターゲットシステムの個々の機能を実行し、かつ、データチャネルへの特定のコマンドインターフェース及び特定のインターフェースを備え、かつ、ターゲットシステムから独立した選択されたソフトウェア構成要素は、データチャネルを介して、自動化タスクを実現するために、モデルと相互接続されており、モデル変換ユニットが備えられ、このモデル変換ユニットは、モデルを、モデル変換によって、ノード、属性、及び、エッジからなるネットリストに、自動的に変換し、それによって、ソフトウェア構成要素は、ノードを構成し、かつ、初期パラメータは、ネットリストの属性とデータチャネルとエッジを構成し、ターゲットシステム変換ユニットが備えられ、このターゲットシステム変換ユニットは、ターゲットシステムから独立したソフトウェア構成要素を、ターゲットシステムに依存したソフトウェア構成要素に変換し、このターゲットシステムに依存したソフトウェア構成要素は、ターゲットシステムでインスタント化され、かつ、コマンドインターフェースを介してパラメータかされ、ネットリストから、正確なシグナルフローにより処理するために、ソフトウェア構成要素の処理順序が算出される。
【０００６】
抽出されたソフトウェア構成要素の使用の結果として、ある自動化タスクが、簡単に、かつ、柔軟性をもって実現することが可能である。それによって、ソフトウェア構成要素は、ターゲットシステムに単純な適応と外部機能の簡単な統合を可能にする必要に応じて生成可能である。例えば、すべての処理ステップの柔軟性のない配置に基づいた従来の自動化ユニットのような中央システムに対して、実際のシグナルフローに適応させることは難しく、かつ、正確なシグナルフローによる処理のみが、やっとのことで可能である。本発明による方法に由来する構成要素においては、正確なシグナルフローを用いて最適な処理が保証される。更に、例えば、全てのターゲットシステムを変更する必要なしに、個々の構成要素のみを交換することによって、システムは、部品レベルで適用可能である。それによって、実行可能な設定を生成するための極めて柔軟性のある方法を実現できる。更に、個々の構成要素が、互いに、独立、かつ、分離して開発可能であることは、例えば、

機能の拡張性を非常に容易にさせる。その上、それによって、構成要素が様々なバージョンで平行してターゲットシステムに存在可能であることは、柔軟性を向上する。さらに、この構造の結果として、簡単かつ効果的な様々のＣＰＵ/コアへの分配が可能である。

【０００７】

本発明による実行可能な設定の生成の方法の結果として、好ましくは、既に、設定が生成されている場合、ターゲットシステムの特異性、又は、仕様を考慮にいれることが可能であるため、それぞれのターゲットシステムで、最適な設定又は処理順序を可能としている。それに加えて、フィルタ、及び/又は、サンプリング素子は、シグナルフローで周波数変動がある場合に、自動的に提案される、又は、実行可能な設定に挿入されることがこの目的のために備えられている。ソフトウェア構成要素は、ターゲットシステムで、少なくとも部分的に、異なるスレッドで処理されることが可能であり、それによって、異なるスレッドは、異なる周波数と同期可能である。これは、ターゲットシステムの設定の高い性能の処理を可能としている。好ましくは、それに加えて、一般プロセスは、ターゲットシステムで実行され、処理エリア、メールボックス、及び、スレッドが利用可能であり、そこにソフトウェア構成要素は、ダイナミックロード可能なライブラリとして配置され、かつ、できるだけ多くインスタント化される。

さらに、ソフトウェア構成要素は、トリガーを備え、このトリガー情報は、自動的及びユーザが無意識に、処理順序に沿って送ることが可能であり、それは、ある出来事にソフトウェア構成要素が反応することを可能とする。ソフトウェア構成要素のトリガーに、時間割当が割り当てられ、この時間割当を超えたソフトウェア構成要素は、一時停止される、又は、警告が送付されることで処理の監視を達成できる。同様に、例えば、リソースが利用可能である場合に、必要とされるタスクのみを後で実行するために、ソフトウェア構成要素は、ターゲットシステムで、１サイクルで、１回以上実行されることが好ましく、それによって、反応速度を増加することが可能である。

【０００８】

特に、柔軟性を持たせるために、処理順序が設けられ、処理順序は、新しい処理を準備する場合、２つの完全なサイクルの間の実行時間中に、切替、又は、変更され、かつ、この切替/変更が１つのサイクルの端部と次のサイクルの始めの間で同期してオーバーヘッドすることなしに行われる。実行可能な設定の変更は、デルタネットリストを用いて、簡単に、柔軟に、かつ、迅速に達成され、そのために、デルタネットリストには、取り除かれるべきネット部分、及び、追加されたネット部分が記載され、かつ、変更は、処理順序に応じた好ましい時点で起きる。これに影響を受けないソフトウェア構成要素は、それゆえ、ランタイムシステムで、妨害されずに、実行され続ける。このように、システムでの部分的な変更は、全てのシステムを止める必要なしに可能となる。

【０００９】

実行可能な設定の処理の性能は、ソフトウェア構成要素がターゲットシステムの異なるＣＰＵ、又は、異なるＣＰＵのコアによって実行される場合に、さらに向上する。ソフトウェア構成要素の微粒子の構成の結果として、微細な分配がターゲットシステムで起きることが可能である。これは、それぞれのターゲットシステムでの設定の最適化処理を可能にもする。

【００１０】

次に、本発明は、有益な実施例を限定することなしに、図式による図１～３を用いて詳細に示されている。

【図面の簡単な説明】

【００１１】

【図１】自動化ユニットを備えた試験台のブロック図である。

【図２】本発明による処理可能なプログラムコードを生成するための処理スキームである。

【図３】自動化タスクの単純なモデルの例である。

【発明を実施するための形態】

【００１２】
図１は、試験台１００を図式的に示したものであり、この試験台で、例えば、エンジン、動力伝達装置、又は、自動車といった見本１０が、特定の機能に関してテストされる。一般に、特定の負荷状況を生成するために、動力計１１が見本１０に接続される。見本１０、及び、動力計１１は、それによって、試験目的に応じて、試験台でハードウェアとして実装され、かつ、試験台のソフトウェアが実行される自動化ユニット１（ターゲットシステム）によって調節される。自動化ユニット１で、レギュレーションユニット４が、この目的のために備えられ、インターフェース２を介して、データバスと、見本１０及び動力計１１と通信している。加えて、例えば、特に、迅速（リアルタイム）な測定（破線で示した）のためのデータ収集ユニット（ＤＡＱ）、又は、特に、分散制御又はシミュレーションのための外部制御デバイスといった外部ユニット１３が備えられ、特定のテスト目的のために必要とされ、かつ、データバス７を介して自動化ユニット１と、又は、見本１０、又は、動力計１１と接続されている。制御目的のため、一般的に、例えば、Ether CAT、profibus、CAN bus、Ethernet powerlink等の高速データバスが、一致したリアクションタイムを保証するために供えられている。高速データバスに加えて、例えば、Ether CAT、profibus、CAN busといった追加的なより低速なデータバス８が備えられ、性能データのようなタイムクリティカル（スピードが重視される）でないデータが移動可能である。通常、様々な（高速、及び/又は、低速）データバスが存在可能である。このため、自動化ユニット１で、インターフェース３が、対応する自動化ユニット１の処理ユニット５が、例えば、測定デバイス（特に、ブローバイ測定、燃費、水、オイル、空気、又は、燃料センサ、又は、排気測定等）、又は、アクチュエータ（特に、ギア転換装置、カップリング、アクセル等）といった試験台構成要素12-1から12-Nと通信可能にする手段として備えられている。

【００１３】
試験台の自動化ユニット１は、上位の試験台管理ユニット６と接続することもでき、この上位の試験台管理ユニット６は、例えば、試験設備又は試験工場の様々な試験台を制御かつ管理し、又は、データ管理タスクも実行する。

【００１４】
様々なプログラム部分からなる試験台ソフトウェアは、自動化ユニット１で実行される。このプログラムには、例えば、データ管理のためのプログラム、特定の自動化タスクを実行するためのプログラム（特に、動力計の調節、測定プロセスのサイクル制御、特定の設定点の値の規格、見本のモニタリング、例外的な状況での自動的な対応等）、上位の試験台管理ユニットとの通信プログラム等がある。これらの個々のプログラムは、周知のように、要求に依存して、リアルタイムなオペレーティングシステム（ＯＳ）、又は、伝統的なオペレーティングシステムで実行可能である。

【００１５】
さらに、追加的な開発ユニット３０が備えられ、例えば、直接、又は、ネットを介して自動化ユニット１と接続されており、かつ、特定の実行可能な設定、主に、自動化プロセスを実行するための構成が生成され、かつ、自動化ユニット１に用いられる。それによって、実行可能な設定は、本質的に、それらの仕事を共に行い、かつ、その構成が自動化タスクを実現するための処理されなければならない方法で、様々なソフトウェア及びハードウェアの構成要素を構成している。このため、ソフトウェアは、そのような構成を解釈、かつ、実行可能な自動化ユニット１でインストールされる。構成の伝達は、これによって、自然に、試験台管理ユニット６、又は、データバス７、８を介してなされる。同様に、開発ユニット３０は、自動化ユニットの全体の一部であってもよい。

【００１６】
そのような実行可能な設定を作成するために、ライブラリ（ストレージ）が、開発ユニット３０に、抽出されたソフトウェア構成要素２０－１～２０－nのために備えられている。
この抽出されたソフトウェア構成要素２０－１～２０－nは、図２で示すように、例えば

、I/0インターフェース、数学的関数、制御法、又は、外部関数といった自動化ユニット１の個々の関数を変換している。ここで、「抽出された」とは、保存されたソフトウェア構成要素２０－１～２０－nが、具体的に（ターゲットシステムに依存して）形作られているのではなく、例えば、I/0インターフェースのようで、かつ、CAN入力や出力のようではない、一般的な関数で示されている。これらのソフトウェア構成要素２０－１～２０－nは、事前に定義されるか、又は、必要とされる場合、対応するユーザインターフェースによって、新しく作成される。どのソフトウェア構成要素２０－１～２０－nも、特定の同一のコマンドインターフェース２２、及び、特定の同一のインターフェース２１を備え、自動化ユニット１のデータチャネルのために実行される。データチャネルは一般的に、スカラー又はベクトル的に構成されたデータを様々なフォーマット（例えば、整数、浮動小数点等）で、循環的にまたはイベントに対応して、１つのデータ発生器から１以上のデータ顧客に移動し、かつ、例えば、バスとして設計される。ソフトウェア構成要素は、直接又は相互の依存性を備えておらず、互いに独立性を維持している。これらのソフトウェア構成要素は、様々な特性に応じて存在してもいる。同様の場合、抽出モデル言語が、ターゲットシステムと独立して記載されている。それと接続された（記載された）場合、あるターゲットシステム（例えば、ある自動化ユニット１）のための特定の記載が存在する。その上、ターゲットシステムや独立システムで存在可能である。その場合、ターゲットシステム自身で実行可能なようにコンパイルされた実際に実行するソフトウェア構成要素もある。

【００１７】

最終的に、すべてのソフトウェア構成要素が、ターゲットシステムで実行可能な実行ファイルとして利用しなければならない。これらの実行ファイルは、ターゲットシステムで既に存在し、又は、時間内に、例えば、新しいソフトウェア構成要素のインストールの間に、後の時点で、ターゲットシステムに送信される。

【００１８】

ライブラリ２０から、遂行される自動化タスク（例えば、サンプルが、ある必要なプロフィル（例えば、トルク/回転モーメント）が、時間制御又は経路制御を割り当てられ、かつ、その作用が特性値及び特性時間で決定される自動的に走るテストラン、磨耗のサインを決定するためにギアでスイッチング操作の繰り返しのシャットダウン、又は、燃焼期間の特性曲線の決定及びパフォーマンス基準又は排出基準の自動最適化等）が、適切なユーザインターフェース２６を介して、たくさんのソフトウェア構成要素２０－ａ～２０－mが選択され、かつ、データチャネル２１を介して、記載された自動化タスクのモデル２３と相互連結されている。レギュレータの単純モデル２３は、図３で示されている。この場合、モデル２３は、４つのソフトウェア構成要素２０－a～２０－ｄからなり、それらの２つが入力（ＩＮＰＵＴ）で、ＰＩＤレギュレータの制御法のソフトウェア構成要素が２つの入力で実現され、かつ、算定された出力変数が出力（ＯＵＴＰＵＴ）によって出力される。それによって、遂行される自動化タスクに応じて、モデル２３は、自然にほとんどが複雑になる。

【００１９】

モデル変換ユニット２７を介して、モデル２３は、モデル変換によって変換され、ノード、属性、及び、エッジからなるネットリスト２４に送られる。その場合、ソフトウェア構成要素２０－aから２０－mは、ノードを構成し、かつ、ソフトウェア構成要素の初期パラメータは、ネットリスト２４の属性とデータチャネル２１とエッジを構成する。このようなモデル変換は公知であり、かつ、一般的に、いくつかの続いて起きる変換ステップからなっている。それによって、ソフトウェア構成要素の型とバージョンの明確な識別の結果として、様々のバージョンの同じソフトウェア構成要素が同じシステムで平行に実行可能である。ネットリスト２４は、例えば、ＸＭＬのような構成された記述的な言語で存在している。ネットリスト２４は、好ましくは、理論的な形で、用いられているソフトウェア構成要素２０－aから２０－mの全て、及び、それらが完全に相互接続を記述している。この単純な記載の結果、所望の機能性が非常に速く、ランタイムシステムで、構築され、

かつ、ターゲットシステムとは未だに独立している。

【００２０】

ネットリスト２４は、実装された自動化タスクの理論的な記載であり、かつ、ターゲットシステムとは独立している。そのため、ネットリスト２４は、必要の際には、様々なターゲットシステムで、実行可能な設定２５によってデコードされる。ターゲットシステム変換ユニット２８で、ネットリスト２４から、実行可能な設定２５を生成するために、ターゲットシステムの記述でない構成要素上に、ターゲットシステムのソフトウェア構成要素が記述され、つまり、一般の「Input」は、例えば、具体的な「CAN-Input」でない。このため、このために必要とされる情報は、例えば、ターゲットシステムとその構想要素とリソースが既に知られているか、開発ユニット３０がそれ自身とターゲットシステムと（例えば、所定の試験台と）結合、かつ、具体的な構成要素及びリソースがターゲットシステムで存在するすべての必要とされる情報が、関連付け等するように、問い合わせをおこなう。それによって、m:n関係、つまり、１つの構成要素が、様々のターゲットシステムの記述上にも、分配、または、集約可能である。この場合、1:1の関係、つまり、ターゲットシステムから独立した構成要素がターゲットシステムに依存した構成要素に記述されことを目指している。さらに、ターゲットシステム変換ユニット２８は、利用可能なチャネルに応じて、入力及び出力ポートを備えた構成要素のように、リソースを分離し、かつ、パラメータ、及び、ネットワーク情報、及び、処理順序から、ターゲットシステム、つまり、実行可能な設定に依存したネットリストを生成する。このターゲットシステムに依存したネットリストの構造デザインは、好ましくは、高パフォーマンス及び並列処理可能なインスタント化のために設計されている。

【００２１】

さらに、使用されるソフトウェア構成要素２０－a～２０－mは、ターゲットシステムで、インスタンスを作成し、かつ、コマンドインターフェイス２２を介して、仕様によりパラメータ化される。

【００２２】

接続されたソフトウェア構成要素２０－a～２０－mの処理順序は、ターゲットシステム変換ユニット２８で、ネットリスト２４から、例えば、グラフ理論の公知のシグナルフロー・グラフ（信号伝達線図）を用いて作成される。もちろん、同様の他の適切な方法、例えば、ネットリスト２４の従属性が算出され、かつ、再帰的にバックトラックすることで、用いられることも可能である。シグナルフロー方向にネットリスト２４を処理するための処理順序は、既にオフラインで、例えば、開発ユニット３０で、及び、ターゲットシステムに依存したネットリストでネットリスト２４の変換前にも行われている、又は、ターゲットシステム自身のみでも行われている。その際、正確なシグナルフローの処理は、算出/処理され、かつ、おそらく、例えば、位相（トリガー/ポストトリガー等）のような追加的な情報も含む構成要素の関連インスタンスを算出する連続するアクティブ算出ステップの順序を編成することを意味している。２つの分離した順序は、好ましくは、それらがデータの従属性を持たず、かつ、それゆえ互いに独立して算出される又は通り抜けることによって、それら自身を区別している。それは、マルチコア/マルチＣＰＵシステム上に、平行するための必要条件として示している。

【００２３】

ネットリスト２４又はモデル２３は、複雑な構成を備えることが可能であるため、このステップでの正確なシグナルフローの処理の場合、好ましくは、既に、様々なことが考慮可能である。シグナルフローで周波数変動の事象が起こった場合、例えば、フィルタ又はサンプリング素子が自動的に設置又は使用される。様々な周波数で処理することが可能であり、例えば、様々なスレッドが生成されることによって、その優先度が、例えば、単調なスケジューリングと格付けされるように、決定される。そうするために、ソフトウェア構成要素自身はスレッドを持たないが、その環境のみを備えていることが可能である。つまり、一般の環境は、処理エリア、スレッド管理、公知のＤＬＬ/ＲＳＬ管理、及び、１以上のメールボックスを利用可能とさせる。その際、スレッドプールは、利用可能であり

、その都度、スレッドプールから、１つのスレッドは、メールボックスでブロックされ、かつ、メッセージの場合、処理エリアにある構成要素で、このメッセージを受け取り、かつ、実行のために、構成要素にそれを割り当てる。その結果、次のスレッドがそのプールから滑り落ちて、メールボックスで待機する。それぞれ、又は、個々のソフトウェア構成要素は、トリガーを備え、その場合、トリガー情報は、自動的に、かつ、ユーザが無意識に、処理順序（トリガー順序）に沿って、例えば、ＣＰＵ－プロセス－構成要素を介して、送られる。トリガー順序で、いずれの入力にも時間割当が与えられる。この時間割当を超えた構成要素は、一時停止される、又は、警告が送付される。１以上のフェーズで、知的な分配が行われ、つまり、ソフトウェア構成要素が１サイクル（例えば、サイクルシステムでのサイクル。例えば１００Ｈｚのサイクル→１サイクルに１０msかかる。）で一回以上実行される。原理的に、ソフトウェア構成要素は、いつも、１サイクルでそれが起こり、それは、直接的に及びすぐに、次のソフトウェア構成要素のために必要となり、かつ、その時既に次に送られる。知的分配の結果として、後で必要なことが、まだ同じサイクルでなされている。同様に、処理順序は、２つの完全なサイクルの間の実行時間中に実際にオーバーヘッドすることなしに、切替、又は、変更される。このため、処理は準備され、この場合、切替/変更が１つのサイクルの端部と次のサイクルの始めの間で同期して行われる。

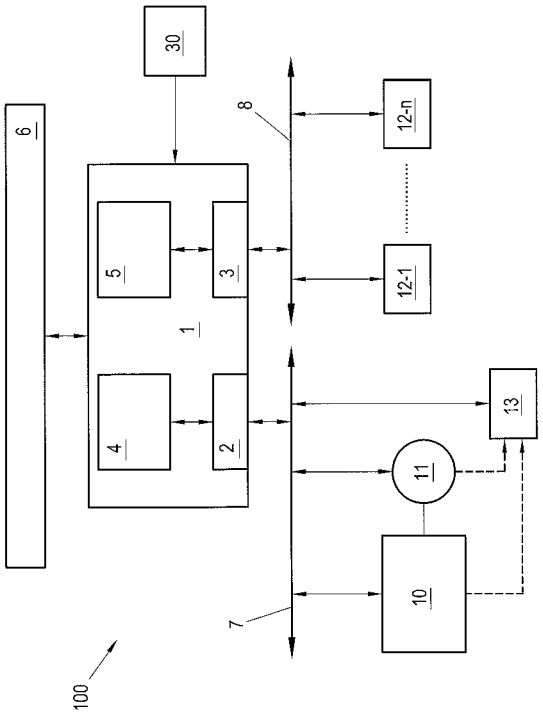
【００２４】

全てのソフトウェア構成要素の同一の構造の結果として、インスタント化の時点で、適切なアルゴリズムが、ソフトウェア構成要素が実行されるターゲットシステムのＣＰＵ上又はＣＰＵのコア上で決定している。しかし、このソフトウェア構成要素の処理の分配が同様に、ターゲットシステムでのみ起きることも可能である。それによって、分配は、好ましくは、構成要素によっておきる。つまり、それは、様々のコア/ＣＰＵ上に配置されたソフトウェア構成要素の一部ではなく、ユニットがコア/ＣＰＵに配置されているように、全てのソフトウェア構成要素である。ソフトウェア構成要素のとても粒度の細かい構成の結果として、好ましい分配が同様に可能である。このために、様々なそれ自体が公知の方法（：どの構成要素もそれ自身の限界を持っている、つまり、どのソフトウェア構成要素もそれ自身の定義された必要条件を持っている）がある。解決者は、制限を満足する課題を解決している。様々な有効な解決方法では、シミュレーションがまだ最適解を見つけている、又は、実現性を再検討している。他の方法では、例えば、同様に適応性の分配及び評価の連続するシミュレーションを用いた遺伝的アルゴリズムがある。この理由のため、複雑なネットリスト２４は、容易に、並列化でき、かつ、マルチＣＰＵ又はマルチコアシステムの間で最適に分割されることが可能である。これは、データチャネルの特性によって、全体として、すべてのＣＰＵから平等にアクセス可能なコミュニケーション手段を支援している。

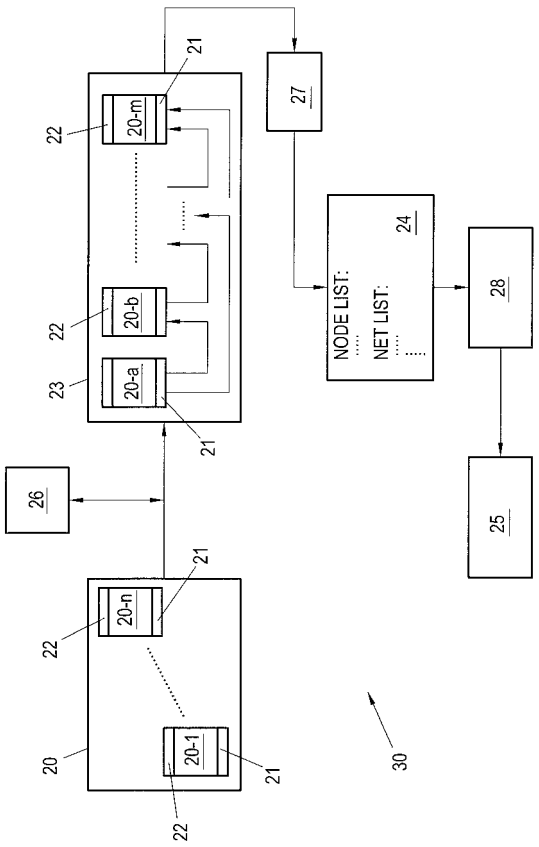
【００２５】

モデル設定の変更は、このように、容易にデルタネットリストを介して可能である。そのようなデルタネットリストは、取り除かれるべきネット部分（又は、ソフトウェア構成要素、及び、その接続）、かつ、追加されたネット部分（又は、ソフトウェア構成要素、及び、その接続）に従うことが記載されている。変更は、処理順序に応じて、好ましい時点、例えば、２つのサイクルの間で起きる。これに影響を受けないソフトウェア構成要素は、それゆえ、ランシステムで妨害されずに実行し続ける。このように、システムの部分的な変更は、全てのシステムを停止する必要なしに達成可能である。

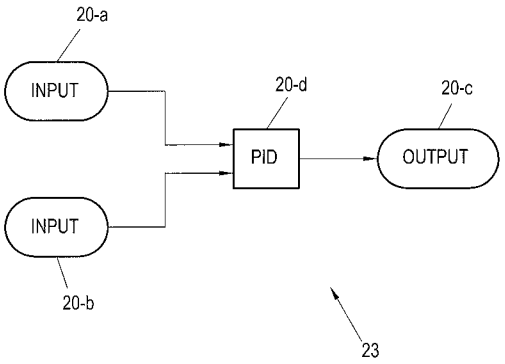
【図１】


30
6
8
12-n
12-1
5
3
1
4
2
13
11
10
7
100

【図２】


21
27
22
20-m
NODE LIST:
NET LIST:
24
28
22
20-b
23
20-a
21
26
25
21
22
20-n
20
20-1
21
22
30

【図３】


20-a
INPUT
20-d
PID
20-c
OUTPUT
INPUT
20-b
23

フロントページの続き

(72)発明者　ヴェルナー・グルーバー
オーストリア国、８６５０　キントベルク、アインデルグルントヴェーク、２２
(72)発明者　ディートマー・パインジップ
オーストリア国、８０２０　グラーツ、クライストストラーセ、７６アー
(72)発明者　ペーター・プリラー
オーストリア国、８１１１　ユーデンドルフ－ストラーセンゲール、アム・ハング、９
(72)発明者　ゲラルト・シュティーグルバウアー
オーストリア国、４９２３　ローンズブルク・アム・コーベルナウザーヴァルト、グンツィンク、４４

審査官　川東　孝至

(56)参考文献　特開２００５－２２２２２１（ＪＰ，Ａ）
特開２００２－２３６５０４（ＪＰ，Ａ）
特表２００７－５２４１７６（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０５Ｂ　１９／０４－１９／０５