

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5271994号

(P5271994)

(45) 発行日 平成25年8月21日(2013.8.21)

(24) 登録日 平成25年5月17日(2013.5.17)

(51) Int.Cl.

F I

G 0 6 F 9/48 (2006.01)

G 0 6 F 9/46 4 5 7

請求項の数 25 外国語出願 (全 20 頁)

(21) 出願番号 特願2010-260082 (P2010-260082)
 (22) 出願日 平成22年11月22日(2010.11.22)
 (65) 公開番号 特開2011-134310 (P2011-134310A)
 (43) 公開日 平成23年7月7日(2011.7.7)
 審査請求日 平成22年11月22日(2010.11.22)
 (31) 優先権主張番号 10 2009 047 025.5
 (32) 優先日 平成21年11月23日(2009.11.23)
 (33) 優先権主張国 ドイツ(DE)

(73) 特許権者 505344063
 ベックホフ オートメーション ゲーエム
 ベーハー
 ドイツ連邦共和国, 33415 フェアル
 , アイザーシュトラッセ 5
 (74) 代理人 110000338
 特許業務法人原謙三国際特許事務所
 (72) 発明者 ディーク ヤンセン
 ドイツ連邦共和国, 33415 フェアル
 , ヴァルスヴェーク 40
 (72) 発明者 ゲルト ホップ
 ドイツ連邦共和国, 33378 レーダー
 ウィーデンプリュック, リューマーンヴェ
 ーク 19

最終頁に続く

(54) 【発明の名称】 リアルタイム・ランタイムシステムおよびそのようなランタイムシステムのための機能モジュール

(57) 【特許請求の範囲】

【請求項1】

複数の機能モジュールと、

管理モジュールとを備えたリアルタイム・ランタイムシステムであって、

上記複数の機能モジュールは、初期化状態から前動作状態までの間の遷移状態において、割り当てられたモジュール識別子を用いて管理モジュールにログオンするとともに、前動作状態から初期化状態までの間の遷移状態においてログオフするように構成されており、

上記複数の機能モジュールのうち少なくとも1つは、前動作状態から試動作状態までの間の遷移状態において上記管理モジュールを介して他の機能モジュールとの通信接続を確立し、試動作状態から前動作状態までの間の遷移状態において通信接続を切断し、試動作状態からリアルタイム動作状態までの間の遷移状態において上記他の機能モジュールにログオンし、リアルタイム動作状態から試動作状態までの間の遷移状態においてログオフするように構成されており、

各機能モジュールは、

初期化状態、前動作状態、試動作状態、およびリアルタイム動作状態、ならびに初期化状態から前動作状態までの間の遷移状態、前動作状態から試動作状態までの間の遷移状態、および試動作状態からリアルタイム動作状態までの間の遷移状態を含み、上記初期化状態、上記前動作状態および上記試動作状態、ならびに、初期化状態から前動作状態までの間の上記遷移状態および前動作状態から試動作状態までの間の上記遷移状態は非リアルタ

10

20

タイムコンテキストに割り当てられ、上記リアルタイム動作状態、および試動作状態からリアルタイム動作状態までの間の上記遷移状態がリアルタイムコンテキストに割り当てられる有限状態機械と、

上記機能モジュールのプロパティ、および一意なモジュール識別子を含むモジュール記述と、

上記有限状態機械および上記モジュール記述にアクセスするための機能モジュールインターフェースと、を備えている、ことを特徴とするリアルタイム・ランタイムシステム。

【請求項 2】

制御プログラムの記述に基づいて、上記複数の機能モジュールを用いてリアルタイムでシステム制御プログラムを実行するように構成され、

上記管理モジュールが上記制御プログラムの記述に基づいて上記複数の機能モジュールを管理することを特徴とする、請求項 1 に記載のリアルタイム・ランタイムシステム。

【請求項 3】

上記管理モジュールが、エンティティの記述に基づいて機能モジュールの上記エンティティを生成するように構成されていることを特徴とする、請求項 1 または 2 に記載のリアルタイム・ランタイムシステム。

【請求項 4】

上記複数の機能モジュールのうち少なくとも 1 つは、上記システム制御プログラムの基本サービスを提供する機能システムモジュールとして構成されていることを特徴とする、請求項 2、または請求項 2 を引用する請求項 3 に記載のリアルタイム・ランタイムシステム。

【請求項 5】

上記機能システムモジュールの上記基本サービスがリアルタイム制御であることを特徴とする、請求項 4 に記載のリアルタイム・ランタイムシステム。

【請求項 6】

上記複数の機能モジュールのうち少なくとも 1 つは、少なくとも 1 つの装置機能をイメージで表示する機能ユーザモジュールとして構成されている、ことを特徴とする請求項 1 から 5 のいずれか 1 項に記載のリアルタイム・ランタイムシステム。

【請求項 7】

少なくとも 1 つの他の有限状態機械は、上記機能ユーザモジュールの上記有限状態機械の下位にあり、

上記他の有限状態機械は、上記機能ユーザモジュールの上記有限状態機械が上記リアルタイム動作状態である間に起動する、ことを特徴とする請求項 6 に記載のリアルタイム・ランタイムシステム。

【請求項 8】

上記リアルタイム・ランタイムシステムは、データ処理システムを動作させるためのリアルタイム・ランタイムシステムであり、

上記複数の機能モジュールのうち少なくとも 1 つは、初期化状態から前動作状態までの間の上記遷移状態においてデータ処理システムのリソースを占有し、前動作状態から初期化状態までの間の上記遷移状態において上記データ処理システムの占有された上記リソースを解放する、ことを特徴とする請求項 1 から 7 のいずれか 1 項に記載のリアルタイム・ランタイムシステム。

【請求項 9】

上記複数の機能モジュールのうち少なくとも 1 つは、試動作状態からリアルタイム動作状態までの間の上記遷移状態において他の機能モジュールのリソースを占有し、リアルタイム動作状態から試動作状態までの間の上記遷移状態において上記他の機能モジュールの占有された上記リソースを解放する、ことを特徴とする請求項 1 から 8 のいずれか 1 項に記載のリアルタイム・ランタイムシステム。

【請求項 10】

複数の機能モジュールを備え、制御プログラムの記述に基づいてリアルタイムでシステ

10

20

30

40

50

ム制御プログラムを実行するランタイムシステムであって、
各機能モジュールは、

初期化状態、前動作状態、試動作状態、およびリアルタイム動作状態、ならびに初期化状態から前動作状態までの間の遷移状態、前動作状態から試動作状態までの間の遷移状態、および試動作状態からリアルタイム動作状態までの間の遷移状態を含み、上記初期化状態、上記前動作状態、および上記試動作状態、ならびに初期化状態から前動作状態までの間の上記遷移状態および前動作状態から試動作状態までの間の上記遷移状態は非リアルタイムコンテキストに割り当てられ、上記リアルタイム動作状態、および試動作状態からリアルタイム動作状態までの間の上記遷移状態がリアルタイムコンテキストに割り当てられている有限状態機械と、

10

上記機能モジュールのプロパティおよび一意なモジュール識別子を含むモジュール記述と、

上記有限状態機械および上記モジュール記述にアクセスするための機能モジュールインターフェースとを備えている、ことを特徴とするランタイムシステム。

【請求項 1 1】

管理モジュールインターフェースを有し、上記制御プログラムの記述に基づいて上記複数の機能モジュールを管理する管理モジュールをさらに備えている、ことを特徴とする請求項 1 0 に記載のランタイムシステム。

【請求項 1 2】

上記複数の機能モジュールは、初期化状態から前動作状態までの間の上記遷移状態においてモジュール識別子を用いて管理モジュールにログオンし、前動作状態から初期化状態までの間の上記遷移状態においてログオフするように構成されており、

20

上記複数の機能モジュールのうち少なくとも 1 つは、前動作状態から試動作状態までの上記遷移状態において上記管理モジュールを介して他の機能モジュールと通信接続を確立し、試動作状態から前動作状態までの間の上記遷移状態において上記通信接続を切断し、試動作状態からリアルタイム動作状態までの間の上記遷移状態において上記他の機能モジュールにログオンし、リアルタイム動作状態から試動作状態までの間の上記遷移状態においてログオフするように構成されている、ことを特徴とする請求項 1 1 に記載のランタイムシステム。

【請求項 1 3】

30

上記管理モジュールが、エンティティの記述に基づいて機能モジュールの上記エンティティを生成するように構成されている、ことを特徴とする請求項 1 1 に記載のランタイムシステム。

【請求項 1 4】

少なくとも 1 つの機能モジュールが、上記システム制御プログラムの基本サービスを提供する機能システムモジュールとして構成されている、ことを特徴とする請求項 1 0 から 1 3 のいずれか 1 項に記載のランタイムシステム。

【請求項 1 5】

上記機能システムモジュールの上記基本サービスがリアルタイム制御である、ことを特徴とする請求項 1 4 に記載のランタイムシステム。

40

【請求項 1 6】

少なくとも 1 つの機能モジュールは、少なくとも 1 つの装置機能をイメージで表示する機能ユーザモジュールとして構成されている、ことを特徴とする請求項 1 0 から 1 5 のいずれか 1 項に記載のランタイムシステム。

【請求項 1 7】

少なくとも 1 つの他の有限状態機械は、上記機能ユーザモジュールの上記有限状態機械の下位にあり、

上記他の有限状態機械は、上記機能ユーザモジュールの上記有限状態機械が上記リアルタイム動作状態である間に起動する、ことを特徴とする請求項 1 6 に記載のランタイムシステム。

50

【請求項 1 8】

上記ランタイムシステムは、データ処理システムを動作させるためのランタイムシステムであり、

上記複数の機能モジュールのうち少なくとも1つは、初期化状態から前動作状態までの間の上記遷移状態においてデータ処理システムのリソースを占有し、前動作状態から初期化状態までの間の上記遷移状態において上記データ処理システムの占有された上記リソースを解放するように構成されている、ことを特徴とする請求項 1 0 から 1 7 のいずれか1項に記載のランタイムシステム。

【請求項 1 9】

上記複数の機能モジュールのうち少なくとも1つは、試動作状態からリアルタイム動作状態までの間の上記遷移状態において他の機能モジュールのリソースを占有し、リアルタイム動作状態から試動作状態までの間の上記遷移状態において上記他の機能モジュールの占有された上記リソースを解放するように構成されている、ことを特徴とする請求項 1 0 から 1 8 のいずれか1項に記載のランタイムシステム。

【請求項 2 0】

制御プログラムの記述に基づいて、システム制御プログラムを実行するように構成されているリアルタイム・ランタイムシステムにおいて使用される機能モジュールであって、

初期化状態、前動作状態、試動作状態、およびリアルタイム動作状態、ならびに初期化状態から前動作状態までの間の遷移状態、前動作状態から試動作状態までの間の遷移状態、および試動作状態からリアルタイム動作状態までの間の遷移状態を含み、上記初期化状態、上記前動作状態、および上記試動作状態、ならびに初期化状態から前動作状態までの間の上記遷移状態および前動作状態から試動作状態までの間の上記遷移状態が非リアルタイムコンテキストに割り当てられ、上記リアルタイム動作状態、および試動作状態からリアルタイム動作状態までの間の上記遷移状態がリアルタイムコンテキストに割り当てられている有限状態機械と、

上記機能モジュールのプロパティおよび一意なモジュール識別子を含むモジュール記述と、

上記有限状態機械および上記モジュール記述にアクセスするための機能モジュールインターフェースとを備えている、ことを特徴とする機能モジュール。

【請求項 2 1】

初期化状態から前動作状態までの間の上記遷移状態においてモジュール識別子を用いてログオンし、前動作状態から初期化状態までの間の上記遷移状態においてログオフし、

前動作状態から試動作状態までの間の上記遷移状態において管理モジュールを介して他の機能モジュールとの通信接続を確立し、試動作状態から前動作状態までの間の上記遷移状態において通信接続を切断し、

試動作状態からリアルタイム動作状態までの間の上記遷移状態において他の機能モジュールにログオンし、リアルタイム動作状態から試動作状態までの間の上記遷移状態においてログオフするように構成されている、ことを特徴とする請求項 2 0 に記載の機能モジュール。

【請求項 2 2】

少なくとも1つの装置機能をイメージで表示するように構成されている、ことを特徴とする請求項 2 0 または 2 1 に記載の機能モジュール。

【請求項 2 3】

インターフェース、インターフェース指定子、パラメータ領域、データ領域、データ領域指定子、時間コンテキスト記述領域、設定領域および参照カウンタの少なくとも1つを備えている、ことを特徴とする請求項 2 0 から 2 2 のいずれか1項に記載の機能モジュール。

【請求項 2 4】

上記参照カウンタは、上記機能モジュールの寿命をモニターし、インターフェースがログオンするときにカウンタ値を1増数し、インターフェースがログオフするときに上記カ

10

20

30

40

50

ウンタ値を1減数し、

上記機能モジュールは、上記カウンタ値が0に達したときに自身を削除可能になっている、ことを特徴とする請求項23に記載の機能モジュール。

【請求項25】

制御プログラムの記述に基づいてリアルタイムでシステム制御プログラムを実行するように構成されているランタイムシステムの一部である、ことを特徴とする請求項20から24のいずれか1項に記載の機能モジュール。

【発明の詳細な説明】

【技術分野】

10

【0001】

本発明は、制御プログラムの記述に基づいて、リアルタイムでシステム制御プログラムを実行するデータ処理システムのためのランタイムシステム、およびそのようなランタイムシステムにおいて用いられる機能モジュールに関する。

【背景技術】

【0002】

工業オートメーションにおいて、メモリプログラマブルな制御および数値制御（SPS／NC）は、工業用PCなどの高性能標準ハードウェアにソフトウェアとして導入されることが多くなってきている。独立した演算ユニット上で実行される従来のSPS／NCとは対照的に、いわゆるソフトSPS／NCは、Windows（登録商標）またはUnix（登録商標）などのデータ処理システムのホストオペレーティングシステム上で動作する。このようにするために、ホストオペレーティングシステムは、リアルタイムカーネルによって拡張される。このリアルタイムカーネルは、ソフトSPS／NCが提供してもよく、ホストオペレーティングシステムによってSPS／NCに与えられる演算時間を制御する。

20

【0003】

従来のSPS／NCと比べ、ソフトSPS／NCは、用いられるハードウェアに対し広範囲にわたって独立性を有することが利点として挙げられる。このことにより、必要に応じて、ハードウェアを容易に取り替えたり、より高性能なハードウェアと交換したりすることができる。したがって、ソフトSPS／NCは、ハードウェアの着実な性能向上による利益を自動的に享受することになる。さらに、通常、ソフトSPS／NCはユーザに対しハードウェア提供者からの独立性を与える。その他、ソフトSPS／NCは、制御されるプロセスへの変更を従来のSPS／NCで行うのに比べて容易に実現できる。通常、従来のSPS／NCは、SPS／NCの要素をプログラムするのに特定のプログラミングデバイスが必要である。また、SPS／NCはコンピュータアセンブリにおける複数のコンピュータに頻繁に分配されるが、SPS／NCの要素もまた様々なコンピュータ言語を用いて頻繁に生成されるため、様々な構成ツールが必要となる。一方で、ソフトSPS／NCの場合、構成ツールは制御ソフトウェアと同じハードウェアに配置してもよい。

30

【0004】

従来およびソフトSPS／NCソリューションにおいて複雑な機械および装置を開ループ制御または閉ループ制御するためのプログラミングに要する時間および労力を減らすべく、モジュール化の概念が次第に用いられるようになってきた。個々の機械の集合体もしくは設備ユニットおよび／または機能性をそれぞれ独立したモジュールと見做し、各モジュールについて外部でカプセル化された制御ソフトウェアを生成する。したがって、個々のプログラム部分は様々な開発ツールおよび／またはコンピュータ言語で作成することが可能であり、そのソースの文字列に関連するコンパイラが実行可能プログラムのコードに翻訳する。これらの実行可能プログラムのモジュールをあわせて、システム制御プログラムを形成する。リアルタイムでシステム制御プログラムを実行するランタイムシステムにおいて、追加で提供されたソフトウェア環境が、起動時に、全てのプログラムモジュールをロードし実行する。ここでは、個々のプログラムモジュールは、互いにおよびソフト

40

50

ウェア環境と通信する規定のインターフェースを備えている。

【先行技術文献】

【特許文献】

【0005】

【特許文献1】 欧州特許出願公開第1182529号明細書

【特許文献2】 独国特許出願公開第10065419号明細書

【特許文献3】 独国特許出願公開第10065418号明細書

【特許文献4】 欧州特許出願公開第2093663号明細書

【発明の概要】

【発明が解決しようとする課題】

10

【0006】

S P S / N C のモジュール構成は個々のモジュールを別々に開発することを可能とするため、その個々のモジュールの開発に必要な最も適したコンピュータ言語および／または開発ツールをそれぞれ選択することができる。したがって、標準化された要素を採用することができる。ロジック制御のための開発ツールとしては、例えば高水準言語が用いられ、閉ループ制御処理用のボード線図もしくはモデルレギュレータが用いられる。しかしながら、個々のプログラムモジュールは、別のコンピュータによって S P S / N C のフレームワーク内にて静的かつ不変な、実行可能なプログラムに変換される。S P S / N C を起動するときには、全てのプログラムモジュールが既に実行可能な形態で存在していなければならない。また、プログラムモジュールを別々に実行するにはプログラムモジュール間においてデータ交換が正しく行われる必要がある。このため、用いられるデータタイプが互換性を有することが確実に保障されるように、特に注意が払われなければならない。個々のプログラムモジュールが完全に別々であるために、データ交換を、インターフェースを介して行うことは避けられない。さらに、プログラムモジュールを相互に関連づけるとともに、結合して共有プログラムを得るには、いわゆるリンカーが必要である。このリンカーは、1つのプログラムモジュールが別のプログラムモジュールを用いる場合に、モジュールの関数および変数のアドレスをメモリアドレスに変換する。

20

【0007】

本発明は、制御プログラムの記述に基づいてリアルタイムで機械プログラムを実行するデータ処理システム用のランタイムシステムであって、より柔軟性があり拡張が容易なランタイムシステムに関する。また、本発明は、よりオープンなランタイムシステムにおいて用いられる、より柔軟性があり拡張が容易な機能モジュールに関する。

30

【課題を解決するための手段】

【0008】

本発明の一実施形態は、複数の機能モジュールおよび1つの管理モジュールを備えるリアルタイム・ランタイムシステムを提供する。上記複数の機能モジュールは、初期化状態から前動作状態までの間の遷移状態において、割り当てられたモジュール識別子を用いて管理モジュールにログオンし、前動作状態から初期化状態までの間の遷移状態においてログオフするように構成されている。上記複数の機能モジュールのうち少なくとも1つは、前動作状態から試動作状態までの間の遷移状態において管理モジュールを介して他の機能モジュールとの通信接続を確立し、試動作状態から前動作状態までの間の遷移状態において上記通信接続を切断し、試動作状態からリアルタイム動作状態までの間の遷移状態において上記他の機能モジュールにログオンし、リアルタイム動作状態から試動作状態までの間の遷移状態においてログオフするように構成されている。

40

【0009】

本発明の別の実施形態は、複数の機能モジュールを備え、制御プログラムの記述に基づいてリアルタイムでシステム制御プログラムを実行するように構成されたランタイムシステムを提供する。各機能モジュールは、初期化状態、前動作状態、試動作状態およびリアルタイム動作状態、ならびに初期化状態から前動作状態までの間の遷移状態、前動作状態から試動作状態までの間の遷移状態、および試動作状態からリアルタイム動作状態までの

50

間の遷移状態を含む有限状態機械を備え、初期化状態、前動作状態および試動作状態、ならびに初期化状態から前動作状態までの間の遷移状態、および前動作状態から試動作状態までの間の遷移状態は、非リアルタイムコンテキストに割り当てられ、リアルタイム動作状態、および試動作状態からリアルタイム動作状態までの間の遷移状態はリアルタイムコンテキストに割り当てられる。各機能モジュールはさらに、モジュールのプロパティおよび一意なモジュール識別子を含むモジュール記述と、上記有限状態機械および上記モジュール記述にアクセスするための機能モジュールインターフェースとを備えている。

【0010】

本発明のさらに別の実施形態は、初期化状態、前動作状態、試動作状態およびリアルタイム動作状態、ならびに初期化状態から前動作状態までの間の遷移状態、前動作状態から試動作状態までの間の遷移状態、および試動作状態からリアルタイム動作状態までの間の遷移状態を含む有限状態機械であって、初期化状態、前動作状態および試動作状態、ならびに、初期化状態から前動作状態までの間の遷移状態、および前動作状態から試動作状態までの間の遷移状態は非リアルタイムコンテキストに割り当てられ、リアルタイム動作状態、および試動作状態からリアルタイム動作状態までの遷移状態がリアルタイムコンテキストに割り当てられている有限状態機械を備える機能モジュールを提供する。上記機能モジュールはさらに、モジュールのプロパティ、および一意なモジュール識別子を含むモジュール記述と、上記有限状態機械および上記モジュール記述にアクセスするための機能モジュールインターフェースとを備えている。

【図面の簡単な説明】

【0011】

【図1】本発明に係るランタイムシステムの実施形態を示す図である。

【図2】本発明に係る機能モジュールについて採用可能な設計を示す図である。

【図3】本発明に係る機能モジュールの有限状態機械を示す図である。

【発明を実施するための形態】

【0012】

本発明の上記特徴が詳細に理解されるべく、上記要約された本発明のさらに詳しい説明を実施形態で示し、そのいくつかは添付の図面にて図示されている。なお、添付図面は本発明の典型的な実施形態のみを図示しているにすぎず、本発明の範囲を限定するものとして考慮されるものではなく、本発明は他の同様の効果を奏する実施形態においても実現可能である。

【0013】

以下の説明では本発明の実施形態を参照する。しかしながら、本発明は説明された特定の実施形態に限定されない。それどころか、他の実施形態に関連するか否かに関わらず、以下に説明される特徴および要素の任意の組み合わせは、本発明を実施および実行するものとみなされる。さらに、本発明は、様々な実施形態において従来技術に対して幾つもの利点を提供する。ただし、本発明の実施形態が他の可能な解決法および／もしくは従来技術に対して利点をもたらすものの、特定の実施形態によって特定の利点を得られるか否かは本発明を制限するものではない。したがって、以下に記載の様態、特徴、実施形態および利点は単に例示的なものであり、請求項に明示的に記載されていない限りは、添付の請求項の要素または限定であると理解されない。同様に、「本発明」の表現は、本明細書に記載されたいかなる発明的な構成要素を一般化したものと解釈されるべきではなく、請求項で明示的に記載されていない限りは、添付の請求項の要素または限定として解釈すべきではない。

【0014】

工業オートメーション、すなわち技術的処理の制御およびモニタリングにおいて、メモリプログラマブルな数値制御（SPS/NC）が長い間に亘って用いられてきた。そのため、制御およびモニタリングされる技術的処理は、機械または複雑な生産ラインにおいて動作し得る。しかしながら、SPS/NCは、在庫管理もしくは建築管理システムとしても用いられてもよい。以下において、「システム」および「システム制御プログラム」の

表現は技術的処理が動作する対象を包括的に示す。したがって、SPS/NCは、独自のオペレーティングシステムと実行すべきシステム制御プログラムをロードするのに用いられるインターフェースとを有する独立した演算ユニットとして、様々な方法によって実現可能である。また、SPS/NCの入力部は、制御する装置内のイベントをSPS/NCに知らせるセンサーに接続されている。SPS/NCの出力部は、制御する装置のアクチュエーターに接続されている。また、制御する機械は、入出力部の代わりとして、フィールドバスシステムを介して接続されていてもよい。

【0015】

ここでは、制御される機械が所望の様態で動作するよう、どの信号を出力部から出力するかをシステム制御プログラムが入力部からの信号に基づいて決定している。SPS/NCのランタイムシステムは、システム制御プログラムが正しく処理され、システム制御プログラムに対して現在の処理状態が常にリアルタイムで示されることを保障する。

10

【0016】

内蔵ユニットの代わりに、SPS/NCは、例えばWindowsまたはUnixといったホストオペレーティングシステムが動作可能な工業用PCもしくは組み込みPCなどの標準データ処理システム上で動作する単なるソフトウェアエミュレーション（ソフトSPS/NC）として構成されていてもよい。したがって、ソフトSPS/NCはホストオペレーティングシステムのリアルタイム拡張を提供し、入出力部および／またはフィールドバスシステムを、制御する機械のセンサーおよび／またはアクチュエーターに接続する。

20

【0017】

従来のSPS/NCと同様に、ソフトSPS/NCのランタイムシステムは、制御する装置の処理順を制御プログラムの記述に基づいて決定する。したがって、制御プログラムの記述を独立したエンジニアリングツールによって外部的に生成してもよく、またはソフトSPS/NCのインターフェースが提供してもよく、またはソフトSPS/NC自体の対応するプログラミングツールによって生成してもよい。

【0018】

とりわけソフトSPS/NCによってリアルタイムで複雑な装置を制御し、同時にプログラムの作成中における工学的な処理に要する時間および労力を削減するべく、SPS/NCの構成においてモジュール化の概念が用いられる。このために、制御対象となる装置の個々の機能、集合体およびユニットは、内蔵機能モジュールが生成される別々のアプリケーションであると判断される。これに関して、各機能モジュールは、互いに通信したり、自身が組み込まれる追加的に提供されたソフトウェア環境と通信したりするためのインターフェースを備えている。ここでは、制御対象となる装置内で頻繁に実行される機能を実行する基本機能モジュールと、基本機能モジュールの上位にあって基本機能モジュールに依存して複雑なタスクを実行する機能モジュールと、を含む、階層的な機能モジュールを構成することができる。

30

【0019】

実施形態によれば、データ処理システム用のランタイムシステムは、制御プログラムの記述に基づいてリアルタイムでシステム制御プログラムを実行する複数の機能モジュールを備える。各機能モジュールは、初期化状態、前動作状態、試動作状態およびリアルタイム動作状態、ならびに初期化状態から前動作状態までの間の遷移状態、前動作状態から試動作状態までの間の遷移状態、および試動作状態からリアルタイム動作状態までの間の遷移状態を含む有限状態機械であって、初期化状態、前動作状態および試動作状態、ならびに、初期化状態から前動作状態までの間の遷移状態、および前動作状態から試動作状態までの間の遷移状態が非リアルタイムコンテキストに割り当てられ、リアルタイム動作状態、および試動作状態からリアルタイム動作状態までの間の遷移状態がリアルタイムコンテキストに割り当てられる有限状態機械と、モジュールのプロパティおよび一意なモジュール識別子を含むモジュール記述と、上記有限状態機械および上記モジュール記述にアクセスするための機能モジュールインターフェースとを備える。

40

50

【0020】

さらなる実施形態によれば、上記ランタイムシステム内には管理モジュールが設けられている。この管理モジュールは管理モジュールインターフェースを備え、制御プログラムの記述に基づいて上記複数の機能モジュールを管理する。ここでは、上記複数の機能モジュールは、初期化状態から前動作状態までの間の遷移状態においてモジュール識別子を用いて管理モジュールにログオンし、前動作状態から初期設定状態までの間の遷移状態にある間にログオフし、前動作状態から試動作状態までの間の遷移状態において管理モジュールを介して他の機能モジュールとの通信接続を確立し、試動作状態から前動作状態までの間の遷移状態において通信接続を切断し、試動作状態からリアルタイム動作状態までの間の遷移状態において他の機能モジュールにログオンし、リアルタイム動作状態から試動作状態までの遷移状態においてログオフするように構成される。

10

【0021】

制御プログラムの記述に基づいてリアルタイムでシステム制御プログラムを実行するデータ処理システム用のランタイムシステムによって、ランタイムシステムにおけるシステム制御プログラムを実行する機能モジュールを動的に管理することができる。このことで、機能モジュールを、各々独立して実行可能なプログラムコードに翻訳する必要がなくなり、ランタイムシステムを起動する前にこれらを結合して共有ランタイムシステムを得る必要もない。機能モジュールは、実行中に生成され、シャットダウンおよび電源オフ中に起動および実行が行われてもよい。さらに、個々の機能モジュールは様々なツールおよび／または様々なコンピュータ言語を用いて生成することができる。各機能モジュールは、さらにリンクしなくともデータ交換ができるように、同じ時間コンテキストで動作し、メモリ領域およびデータタイプを共用する。全ての機能モジュールの設計が標準化されていることにより、変更および適合はランタイムシステム内にて容易に行うことができる。

20

【0022】

さらなる実施形態によれば、上記管理モジュールはエンティティの記述に基づいて機能モジュールのエンティティを生成するように構成されている。したがって、実行するアプリケーションをリアルタイムコンテキストにて自由にコンパイルすることができる。

【0023】

さらなる実施形態によれば、少なくとも1つの機能モジュールは、システム制御プログラムの、好ましくはリアルタイム制御であるような基本サービスを提供する機能システムモジュールとして構成されている。機能システムモジュールを機能モジュールとして扱うことにより、容易に基本サービスをランタイムシステム内に追加または変更することができる。

30

【0024】

さらなる実施形態によれば、少なくとも1つの機能モジュールは、少なくとも1つの装置機能を示す機能ユーザモジュールとして構成されている。ここで、さらなる有限状態機械が機能ユーザモジュールの有限状態機械の下位にあってもよく、上記さらなる有限状態機械は上記機能ユーザモジュールの上記有限状態機械がリアルタイム動作状態にある間に起動されてもよい。この機能ユーザモジュールの構成によって、複雑にふるまうアプリケーションをも、容易かつ信頼性を有する様態で作成することができる。

40

【0025】

さらなる実施形態によれば、上記機能モジュールは、初期化状態から前動作状態までの間の遷移状態において上記データ処理システムのリソースを占有し、前動作状態から初期化状態までの間の遷移状態において占有していた上記リソースを解放する。さらに、上記機能モジュールは、試動作状態からリアルタイム動作状態までの間の遷移状態において他の機能モジュールのリソースを確保し、リアルタイム動作状態から試動作状態までの間の遷移状態において、占有した上記リソースを返却する。この手順はランタイムシステムの信頼性の高い動作を確保するだけでなく、効果的なリソースの管理をも可能とする。

【0026】

さらなる実施形態によれば、上記機能モジュールは、さらなる要素として、インターフ

50

エース、インターフェース指定子、パラメータ領域、データ領域、データ領域指定子、時間コンテキスト記述領域、設定領域および参照カウンタを備える。したがって、上記参照カウンタは、上記機能モジュールの寿命をモニターし、インターフェースがログオンしたときにカウンタ値を1増数させ、インターフェースがログオフしたときに上記カウンタ値を1減数させるように構成される。上記カウンタ値が0に達したとき、上記機能モジュールは自身を独立して削除してもよい。このようにして、上記機能モジュールはモジュールシステムの形態で容易に生成される。

【0027】

実施形態によれば、データ処理システム用のランタイムシステムは、制御プログラムの記述に基づいてリアルタイムでシステム制御プログラムを実行する複数の機能モジュールを備える。各機能モジュールは、初期化状態、前動作状態、試動作状態およびリアルタイム動作状態、ならびに初期化状態から前動作状態までの間の遷移状態、前動作状態から試動作状態までの間の遷移状態、および試動作状態からリアルタイム動作状態までの間の遷移状態を含み、初期化状態、前動作状態および試動作状態ならびに初期化状態から前動作状態までの間の遷移状態、および前動作状態から試動作状態までの間の遷移状態は非リアルタイムコンテキストに割り当てられ、リアルタイム動作状態、および試動作状態からリアルタイム動作状態までの遷移状態がリアルタイムコンテキストに割り当てられる有限状態機械と、モジュールのプロパティおよび一意なモジュール識別子を含むモジュール記述と、上記有限状態機械および上記モジュール記述にアクセスするための機能モジュールインターフェースとを備えている。

【0028】

さらなる実施形態によれば、上記ランタイムシステム内には管理モジュールが設けられている。この管理モジュールは管理モジュールインターフェースを備え、上記制御プログラムの記述に基づいて上記複数の機能モジュールを管理する。ここでは、上記複数の機能モジュールは、初期化状態から前動作状態までの間の遷移状態において自身のモジュール識別子を用いて上記管理モジュールにログオンし、前動作状態から初期化状態までの間の遷移状態の間にログオフし、前動作状態から試動作状態までの間の遷移状態において管理モジュールを介して他の機能モジュールとの通信接続を確立し、試動作状態から前動作状態までの間の遷移状態において通信接続を切断し、試動作状態からリアルタイム動作状態までの間の遷移状態において他の機能モジュールにログオンし、リアルタイム動作状態から試動作状態までの遷移状態においてログオフするように構成される。

【0029】

制御プログラムの記述に基づいてリアルタイムでシステム制御プログラムを実行するデータ処理システム用のランタイムシステムによって、ランタイムシステムにおけるシステム制御プログラムを実行する機能モジュールを動的に管理することができる。このことで、ランタイムシステムを起動する前に、上記機能モジュールを、各々独立して実行可能なプログラムコードに、翻訳する必要がない。上記機能モジュールは、実行中、または、シャットダウンおよび電源OFF中にそれぞれ生成、起動実行されてもよい。さらに、個々の機能モジュールは様々なツールおよび／または様々なコンピュータ言語を用いて生成することができる。全ての機能モジュールの設計が標準化されていることにより、変更および適合はランタイムシステム内にて容易に行うことができる。

【0030】

さらなる実施形態によれば、管理モジュールはエンティティの記述に基づいて機能モジュールの上記エンティティを生成するように構成されている。したがって、実行アプリケーションをリアルタイムコンテキストにて自由にコンパイルすることができる。

【0031】

さらなる実施形態によれば、少なくとも1つの機能モジュールは、システム制御プログラムの、好ましくはリアルタイム制御であるような基本サービスを提供する機能システムモジュールとして構成されている。機能システムモジュールを機能モジュールとして扱うことにより、容易に基本サービスをランタイムシステム内に付加または変更することがで

きる。

【0032】

さらなる実施形態によれば、少なくとも1つの機能モジュールは、少なくとも1つの装置機能を示す機能ユーザモジュールとして構成されている。そのため、さらなる有限状態機械が機能ユーザモジュールの有限状態機械の下位にあってもよく、上記さらなる有限状態機械は、上記機能ユーザモジュールの上記有限状態機械がリアルタイム動作状態にある間に起動されてもよい。この機能ユーザモジュールの構成によって、複雑にふるまうアプリケーションであっても、容易かつ信頼性を有する状態で作成することができる。

【0033】

さらなる実施形態によれば、上記機能モジュールは、初期化状態から前動作状態までの間の遷移状態においてデータ処理システムのリソースを占有し、前動作状態から初期化状態までの間の遷移状態において占有していたリソースを解放する。さらに、上記機能モジュールは、試動作状態からリアルタイム動作状態までの間の遷移状態において他の機能モジュールのリソースを確保し、リアルタイム動作状態から試動作状態までの間の遷移状態において、占有した上記リソースを返却する。この手順はランタイムシステムの信頼性の高い動作を確保するだけではなく、効果的なリソースの管理をも可能とする。

【0034】

さらなる実施形態によれば、上記機能モジュールは、さらなる要素として、インターフェース、インターフェース指定子、パラメータ領域、データ領域、データ領域指定子、時間コンテキスト記述領域、設定領域および参照カウンタを備える。したがって、上記参照カウンタは上記機能モジュールの寿命をモニターし、インターフェースがログオンしたときにカウンタ値を1増数させ、インターフェースがログオフしたときに上記カウンタ値を1減数させるように構成される。上記カウンタ値が0に達したとき、上記機能モジュールは自身を自ら削除してもよい。このようにして、上記機能モジュールはモジュールシステムの形態で容易に生成される。

【0035】

図1は、機能モジュールを備える、制御プログラムの記述に基づいてリアルタイムでシステム制御プログラムを実行するデータ処理システム用ランタイムシステムの構成を概略的に示す図である。ここで、図1は、システム制御プログラムを実行中の特定の段階におけるランタイムシステムを示している。上記ランタイムシステムは、システム領域とユーザ領域とに分けられていてもよい。図1におけるユーザ領域では、5つの機能ユーザモジュール1がアクティブであり、システム領域では、3つの機能システムモジュール2、および1つの管理モジュール3がアクティブである。全てのモジュールは、各モジュールにアクセスするのに用いられる通信層を介して、互いに接続されている。データ処理システムが統一されたメモリ領域を形成している場合、モジュール間で直接的にアクセスする。対照的に、データ処理システムが、分散的に構成され、複数のネットワーク接続されたコンピュータ部品を備える場合、通信層4はインターフェースを介して機能モジュールに接続されたメッセージルーターとして構成される。しかしながら、いくつかのモジュールがデータ処理システムの同一メモリ領域に位置し、他のモジュールがデータ処理システムの別のメモリ領域に位置するのであれば、通信層4は直接的なアクセス方法とメッセージルーターを介するアクセス方法とを組み合わせてもよい。

【0036】

したがって、機能ユーザモジュールおよび機能システムモジュールは、同一構成を有する。各機能モジュールは一式の要素からなり、そのうち3つの要素、すなわち有限状態機械、モジュール記述および機能モジュールインターフェースが必須であり、他の要素を含むかは自由である。したがって、機能モジュールの全ての要素は標準化されている。

【0037】

図2は、3つの必須要素およびさらにオプション要素を備える、機能モジュールが構成されうる例を示している。各機能モジュールは、機能モジュールのふるまいをイメージ化する、状態および遷移状態からなる有限状態機械10を備えている。ここで、遷移状態は

、外部から引き起こされてもよい。有限状態機械10は、初期化状態、前動作状態、試動作状態およびリアルタイム動作状態、ならびに初期化状態から前動作状態までの間の遷移状態、前動作状態から試動作状態までの間の遷移状態、および試動作状態からリアルタイム動作状態までの間の遷移状態を含む。

【0038】

機能モジュールの有限状態機械10は、リアルタイム動作状態で起動され得るさらなる有限状態機械の上位モジュールであってもよい。したがって、機能モジュールの有限状態機械10は、初期状態、前動作状態、および試動作状態、ならびに初期状態から前動作状態までの間の遷移状態および前動作状態から試動作状態までの間の遷移状態が非リアルタイムコンテキストで行われるが、リアルタイム動作状態および試動作状態からリアルタイム動作状態までの間の遷移状態はリアルタイムコンテキストで行われる。

10

【0039】

機能モジュールの一連のイベントを決定する機能モジュールの有限状態機械10とは別に、機能モジュールは、さらなる必須要素としてモジュール記述11を備える。モジュール記述11は、モジュールのプロパティを決定し、付加的に識別子、特にランタイムシステムにおいて機能モジュール固有であるモジュール識別子を含んでいる。したがって、モジュールのプロパティを、クラス識別子を用いてモジュール記述内において特徴付けることができる。このモジュール記述はクラス記述を参照するが、クラス記述は制御プログラムの記述の一部であっても、モジュール自体に埋め込まれていてもよい。とりわけ、クラス記述は機能モジュールの全ての要素、特に後述の要素、すなわち、そのインターフェースがモジュール内に実装されている要素、そのインターフェースの指示子を設定可能な要素、そのデータ領域がモジュールに備えられている要素、そのデータ領域の指定子が設けられている要素、そのパラメータをモジュール内にセット可能な要素、およびモジュールが動作する時間コンテキストの要素を示す。さらなる識別子として、モジュール記述はさらに、機能モジュールが他の機能モジュールから派生していることを示す遺伝的識別子をも含んでいてもよい。

20

【0040】

この機能モジュールは、有限状態機械10およびモジュール記述11の他に、有限状態機械10およびモジュール記述11にアクセスするための機能モジュールインターフェース12を必ず備えている。機能モジュールは、この必須要素である機能モジュールインターフェース12の他に、さらにインターフェース13をオプションの要素として備えていてもよい。これらの各インターフェース13は、外部からアドレス指定され、および／または外部よりアクセス可能な、機能モジュールの1つ以上の要素を示す。ここで、各インターフェース13は関連する一意なインターフェース識別子を介して識別されてもよい。これらのインターフェース識別子によって、その機能モジュールと通信可能かどうかを他の機能モジュールが判断することができる。したがって、機能モジュールの全てのインターフェースは、少なくとも2つの基本サービス、すなわち、インターフェース要求機能およびモジュール寿命機能を提供する基本インターフェース構成に基づいている。これら2つの基本サービスは、有限状態機械10およびモジュール記述11にアクセスするための機能モジュールインターフェース12によってサポートされている。

30

40

【0041】

機能モジュールにおいて設けることができる他のインターフェースとしては、特に、周期的に機能モジュールにアクセスするためのインターフェースと上記機能モジュールが周期的なアクセスを受けるために他の機能モジュールがログインする際に介する対応するインターフェースと、挙げられる。さらに、処理データを要求および／または公開するインターフェースも通常設けられる。これら2つのインターフェースは、機能モジュールの入出力を更新するのに用いられる。さらなるインターフェースは、後述する機能モジュールのオプション要素、すなわち、データ領域、パラメータ領域、時間コンテキスト記述領域および参照カウンタにアクセスすることを可能とする。

【0042】

50

さらに、機能モジュールの一般的なインターフェースとして、リアルタイムインターフェースを設けることができる。機能モジュールにリアルタイムタスクが割り当てられた場合、リアルタイムタスクのサイクル時間、優先順位およびその他の情報を、リアルタイムインターフェースを経由して取り出すことができる。

【0043】

サービスが正確に判定され、一元的に交換および変更可能であり、および／または他のインターフェースに移すことができるこれらの一般的なインターフェースの他に、機能モジュールは、さらに、別のインターフェースの同一構成を有する機能モジュールとのみ通信可能な個々のインターフェースを備えていてもよい。

【0044】

機能モジュールは、インターフェース13の他に、さらなるオプション要素として、インターフェース13の相手役として動作するインターフェース指定子14を備えている。上記インターフェース指定子14を介して、機能モジュールは、インターフェース指定子によって特徴付けられた他の機能モジュールのインターフェースにアクセスしてもよい。これに関し、全ての機能モジュールは、ランタイムシステムにおける管理モジュール3の管理モジュールインターフェースに向けられたインターフェース指定子を備え、このことによって上記管理モジュールに機能モジュールをログオンしたり、上記管理モジュールから機能モジュールをログオフしたりすることができる。インターフェース指定子は、非リアルタイムコンテキスト、すなわち、初期状態、前動作状態、試動作状態、およびこれらの間の遷移状態における、機能モジュールの起動中に設定される。

【0045】

さらなるオプション要素として、機能モジュールはパラメータ領域15を有している。パラメータ領域15において、特定パラメータのデータタイプはパラメータ識別子および実際のパラメータ値によって示されている。このため、パラメータ識別子およびパラメータ値は、外部よりアクセス可能とするために設けられた上述のパラメータ領域インターフェースを介して、外部よりアクセスすることができる。

【0046】

機能モジュールのさらなるオプション要素として、同じく外部よりアクセス可能とするために設けられたデータ領域インターフェースを介して外部よりアクセス可能なデータ領域16が挙げられる。データ領域16の構造は、モジュール記述11によって決定される。機能ユーザモジュールにおいて、データ領域16は処理画像を読み込んだり読み出したりするのに用いられる。ここでは、データ領域は、データ領域の構成を詳細に記述する記号的なデータを含んでいてもよい。

【0047】

他の機能モジュールのデータ領域をアクセスするために、機能モジュールはオプション要素としてデータ領域指定子17を備えていてもよい。データ領域指定子17は所望のデータ領域を参照するが、データ領域指定子17が参照することで、対応するデータ領域インターフェースを介してデータ領域へのアクセスが可能となる。インターフェース指定子17と同様に、上記データ領域指定子17は、非リアルタイムコンテキストにおいて機能モジュール内に設定される。

【0048】

さらなるオプション要素として、機能モジュールは時間コンテキスト記述領域18を有していてもよい。上記時間コンテキスト記述領域18は、特に、複数の時間コンテキストにおいて機能モジュールがアクティブであることが可能な場合、すなわち、機能モジュールが複数の時間非依存のアプリケーションをサポートしていたり、閉ループアプリケーションが異なるサイクル時間で閉制御ループの処理を行ったりする場合に設けられる。機能モジュールがこのような時間コンテキスト記述領域18を有する場合、この要素はモジュール記述11における機能モジュールのプロパティとして示される。

【0049】

さらなるオプション要素として、機能モジュールは、機能モジュールの寿命をモニター

10

20

30

40

50

する参照カウンタ19を備えている。これに関し、参照カウンタ19は、機能モジュールのインターフェースが発行された場合、すなわち機能モジュールがさらなる機能モジュールまたは管理モジュールにログオンした場合に、参照カウンタ19が1増数するように構成されることが好ましい。続いて、機能モジュールを介してインターフェースをログオフするとき、参照カウンタ19は1減数される。参照カウンタ19が減数されたときに0の値となった場合、機能モジュールは自主的に自身を削除する。これは通常、管理モジュール3からログオフした後に行われる。しかしながら、参照カウンタ19は、他の機能モジュールのインターフェース指定子が機能モジュールのインターフェースを指している限りは、該機能モジュールを削除しないように構成されている。これら他の機能モジュールのインターフェース指定子が有効になるとすぐに、参照カウンタ19は、機能モジュールの削除を無効にする。

10

【0050】

構成領域20も機能モジュールのオプション要素となりえる。この構成領域20は、機能モジュールに紐付け可能な他の機能モジュールタイプを判定する構成識別子を備えている。モジュール記述11にも記憶されているこれらの構成識別子に基づいて、システム制御プログラムを実行するのにどの機能モジュールを組み合わせることができるかを、容易に判断することが可能になる。

【0051】

上記記載のとおり、ランタイムシステムの全ての機能モジュール、すなわち、ユーザ領域における機能ユーザモジュールおよびシステム領域における機能システムモジュールは、3つの必須要素とさらにオプション要素とを備えた同一構成を有している。これに関し、個々の機能モジュールは様々なツール、および／または、C/C++、I/Oコンフィギュレーション、PLC制御、シミュレーションまたはIEC61131-3など、様々なコンピュータ言語を用いて生成してもよい。さらに、任意の所望のコンピュータ言語を用いて生成したアプリケーションを、図1に示されるリアルタイム・ランタイムシステムにおいて用いられる上述の標準化された機能モジュールに変換する、付加的なエンジニアリングツールを設けてもよい。そして、エンジニアリングツールは、各コンピュータ言語にて生成されたアプリケーションを、上述の形式的な構成の機能モジュールに基づいてパッケージングしてもよく、絶対必須要素である有限状態機械、モジュール記述、並びに、有限状態機械およびモジュール記述にアクセスするための機能モジュールインターフェースに加えてさらにオプションの機能モジュールを付加してもよい。

20

30

【0052】

機能モジュールは、非常にシンプルに構成されてもよく、非常に複雑なに構成されてもよい。ユーザ領域の機能モジュールは、装置の機能、および／または、供給ユニット、アーバー、ツール変更機、もしくは他の機械電子工学ユニットなどの機器集合体を開ループ・閉ループ制御する役割を担う。モジュール記述にて特定されたプロパティは、上記機械電子工学ユニットのふるまいを具象化するアルゴリズムを提供する。したがって、機能ユーザモジュールが簡素な機能を単に制御することも、非常に複雑な処理の流れを制御することもあり得る。これを受け、機能ユーザモジュールは、互いに組み込まれ、機能モジュール有限状態機械に制約され、機能モジュール有限状態機械の状態に応じて起動される有限状態機械をいくつか備えていてもよい。したがって、例えば機能ユーザモジュールは、2つの接続されたドライブを制御するべく、相互に依存している下位の有限状態機械を2つ備えていてもよい。

40

【0053】

このような機能ユーザモジュールの他に、ランタイムシステムは機能システムモジュールを備えている。この機能システムモジュールは、その構成が機能ユーザモジュールの構成と対応しており、リアルタイムでシステム制御プログラムを実行するべくランタイムシステムの基本サービスを提供するものである。SPS/NC用の基本サービスのうちの1つに、ランタイムシステムにおけるアプリケーションがリアルタイムで行われていることを保障するためのリアルタイムモニタリングがある。そして、リアルタイム機能システム

50

モジュールは付加的に、他の機能モジュールからアクセス可能なリアルタイムインターフェースを機能モジュール要素として設けることで、各機能モジュールによって実行されるアプリケーションがリアルタイムで実行されることを補償し、および／または、機能モジュールにそのようなリアルタイムアプリケーションを生成させることができるようにする。さらなる機能システムモジュールは入出力用および／またはフィールドバスシステム用のドライバである。さらに、機能モジュールがログオンし、機能モジュールが接続されるバス4を制御する機能システムモジュールを設けてもよい。さらなるシステムモジュールとして、機能モニタリングモジュール、機能文書化モジュール、または機能管理モジュールを、モジュール記述内の様々な識別子として設けてもよい。

【0054】

通常、機能システムモジュールはランタイムシステムを起動した際に自動的にロードされるが、一方で機能ユーザモジュールは管理モジュールまたは他の機能モジュールによって、より後の時点で起動されてもよい。上記記載の通り、機能モジュールは原則的に個々に独立して開発されていてもよいが、全ての機能モジュールが上記記載の標準化された構成を有している。

【0055】

機能モジュールの上記標準化された構成に加えて、機能モジュールによって用いられるインターフェース、パラメータ領域、データ領域、構成領域の識別子について統一されたデータタイプシステムおよび統一された管理が必要である。ランタイムシステムのフレームワークにおける機能モジュール間のやりとりを可能にするには、識別子は明確に定義されていなければならない。ここでは、各識別子を長い乱数によって生成することができる。この数値範囲の大きさによって、同一の識別子が生成されないことが保障される。しかしながら、たとえば128ビットの長さを有する乱数のような長い乱数の識別子は、複雑且つ時間を要する処理を必要とし、リアルタイムでシステム制御プログラムを実行するランタイムシステムのフレームワークでは特に障害となり得る。

【0056】

しかしながら、リソースおよび時間を節約した処理を可能とするべく、より短い数字の組み合わせで動作するには、アクティブに識別子を管理する必要がある。例えばたかだか32ビットの長さを有する数値識別子は、は一元管理される領域（例えば前半の16ビット）と局所管理される領域（例えば、後半の16ビット）とに分けられる。一元管理されている領域からは、各機能モジュールの開発者は、機能モジュールのプログラミングの際に機能モジュールに設けられる識別子として用いられるべき16ビットの長さを有する数値を1つ以上割り当てられる。そして32ビット長のさらなる16ビットの数値は、機能モジュールが生成されるときに個々に決定される数値であり、その機能モジュールにおいては一意でなければならない。

【0057】

機能モジュールによって用いられるデータタイプのデータタイプ識別値を決定する際に、一般的に長い乱数、例えば128ビット長の乱数を用いることができる。このデータタイプ識別値は実際のところランタイムシステムおよび／または機能モジュールによってリアルタイムコンテキストで評価されているのではなく、評価に十分な時間が残るようにシステムの起動中に非リアルタイムコンテキストで行われる。このデータタイプ識別子によって、全ての機能モジュールによって統一されたデータタイプシステムが用いられることを保障できる。これにより、データタイプは、常に明確に記述され、機能モジュールと外部との間で信頼性のあるデータ交換を保障することができる。データタイプ識別子は、機能モジュールがデータをデータ名とは独立して同一に解釈することを確実にする。機能モジュールを生成することができる様々なコンピュータ言語では、異なるデータタイプで同一のデータ名を頻繁に用いる結果、データ名のみでのデータタイプの記述が明確でなくなる。さらに、データタイプはデータ名を変えずに頻繁に変更される。

【0058】

図1に示すとおり、ランタイムシステムは機能モジュールに加えて中央管理モジュール

10

20

30

40

50

3を備えている。管理モジュール3はその構成において、標準化された機能モジュールと異なり、特別な管理モジュールインターフェースを備えている。管理モジュール3は、制御プログラムの記述に基づいてリアルタイムでシステム制御プログラムを実行するランタイムシステムのフレームワーク内において機能モジュールの動的な管理を可能とする。ランタイムシステムを起動するとき、管理モジュールは最初に起動されるモジュールとなり、機能モジュールに基づいてリアルタイムでシステム制御プログラムの実行を決定する制御プログラムの記述にアクセスする。制御プログラムの記述は、機能モジュールのエンティティを機械制御プログラムのフレームワーク内で実行するための記述データを含んでいる。エンティティの記述は、機能モジュールの正確な設定、すなわち、用いられるパラメータ、インターフェース、インターフェース指定子、データ領域、データ領域指定子、時間コンテキストなどを決定する。

10

【0059】

続いて、管理モジュール3は、機能モジュールの各エンティティを、起動直後にエンティティの記述に基づいて実行するか、またはそれより後の時点で、制御プログラムの記述のさらなる規定に基づいて生成する。したがって、エンティティの生成は管理モジュールによってモジュールクラス識別子を用いて実行されてもよい。その目的のために、管理モジュールは、自身の管理インターフェースを介して、モジュールクラスを生成することが可能なモジュールクラスシステムモジュールにアクセスする。他のシステムモジュールと同様に、上記モジュールクラスシステムモジュールはランタイムシステムの起動時に自動的に初期化され、他のシステムモジュールと同様に、初期状態から前動作状態までの間の遷移状態において、自身のモジュール識別子を用いて管理モジュールにログオンする。エンティティを生成するべく、続いて管理モジュールは、モジュールクラスシステムモジュールからクラス識別子に関連付けられているモジュールクラスの生成要素を選択し、機能モジュールのエンティティ設定を開始する。

20

【0060】

図3は機能モジュールの有限状態機械、および管理モジュールによって管理されているランタイムシステムのフレームワーク内における上記有限状態機械の遷移状態を示している。機能モジュールエンティティは、初期状態101において生成される。機能モジュールエンティティの生成の完了後、機能モジュールはすぐに初期状態101から前動作状態104へと切り替わる。この遷移状態102では、機能モジュールは自身のモジュール識別子を用いて、管理モジュールにログオンする。これを受け、機能モジュールは管理モジュールからパラメータ120を受け取る。さらに、初期化状態から前動作状態までの間の遷移状態102では、機能モジュールは他の機能モジュールから完全に隔てられ、これら他の機能モジュールと通信することができない。初期状態101から前動作状態104までの間の遷移状態102では、機能モジュールには、リアルタイム動作に必要なリソース、特にデータ処理システムのメモリ領域が、必要に応じて管理モジュールによって割り当てられる。遷移状態後の前動作状態104では、機能モジュールは完全に構成されており、通常、大部分がパラメータ化されている。機能モジュールは、管理モジュールにログオンしているが、他の機能モジュールとのやりとりはない。

30

【0061】

前動作状態104から試動作状態107までの間の遷移状態105では、機能モジュールは他の機能モジュールとやりとりを行ってもよい。機能モジュールは、やりとりを行うべき機能モジュールのモジュール識別子を含んでいる関連初期設定データに基づいて、必要な通信を決定する。したがって、機能モジュール間での通信は、管理モジュールインターフェースを用いてモジュール識別子に基づいて機能モジュールを識別する管理モジュールによって生成される。これに関し、前動作状態104から試動作状態107までの間の遷移状態105は、管理モジュールによって引き起こされてもよい。前動作状態104から試動作状態107までの間の遷移状態105では、他のパラメータ121が機能モジュールに転送されてもよい。このことで、モジュール記述が例えば遺伝的識別子を含んでいる場合、下位の機能モジュールからのパラメータは派生した機能モジュールに転送されて

40

50

もよい。

【0062】

試動作状態107では、機能モジュールはその機能が処理対象となるまで待機する。初期状態101、前動作状態104、試動作状態107、ならびに初期化状態から前動作状態までの間の遷移状態102および前動作状態から試動作状態までの間の遷移状態105は、それぞれ、非リアルタイムコンテキストで（すなわち、機能モジュールに割り当てられたアプリケーションが機械制御のフレームワーク内にて用いられる前に）、発生する。

【0063】

リアルタイムコンテキストでアプリケーションを実行するために、機能モジュールは、制御プログラムの記述に基づいて、管理モジュールまたは他の機能モジュールによってリアルタイム動作状態110に遷移する。試動作状態107からリアルタイム動作状態110までの間の遷移状態108では、パラメータ領域およびデータ領域インターフェース指定子を適切に設定することで、機能モジュールが他の機能モジュールからリソースを要求することが可能である。また、機能モジュールは、他の機能モジュールにログオンし、各インターフェース指定子を、例えばこの機能モジュールによって周期的にアクセス可能にするように設定してもよい。続いて、リアルタイム動作状態110において、機能モジュールは、ダウンストリームで接続される実行可能な有限状態機械を起動するアプリケーションを実行する。

【0064】

アプリケーションを終了する際、機能モジュールはリアルタイム動作状態110から試動作状態107へと自身をリセットする。この遷移状態109は他の機能モジュールまたは管理モジュールによって引き起こされてもよい。機能モジュールは試動作状態からリアルタイム動作状態までの間の遷移状態108において実行した全ての動作を無効にするとともに、他の機能モジュールから要求した全てのリソースを開放し、および／または、これらの他の機能モジュールからログオフする。

【0065】

機能モジュールが、例えば、制御プログラムの記述によって試動作状態107の間でより長い期間使用されていない場合、上記機能モジュールは、試動作状態107から前動作状態104に戻る。この遷移状態106は管理モジュールによって引き起こされてもよい。機能モジュールは、その後、前動作状態104から試動作状態107までの間の遷移状態105において行った全ての動作を無効にし、要求した全てのリソースを開放する。

【0066】

その後、制御プログラムの記述のフレームワーク内において、機能モジュールが必要でなくなった場合またはランタイムシステムが完全に終了した場合、機能モジュールは前動作状態104から初期状態102に戻る。この遷移状態103の間で、機能モジュールは初期化状態から前動作状態までの間の遷移状態102において行った動作を全て無効化し、特に、要求したシステムリソースを開放する。機能モジュールは管理モジュールからログオフし、機能モジュールの参照カウンタが0を示している場合、機能モジュール自体を削除する。

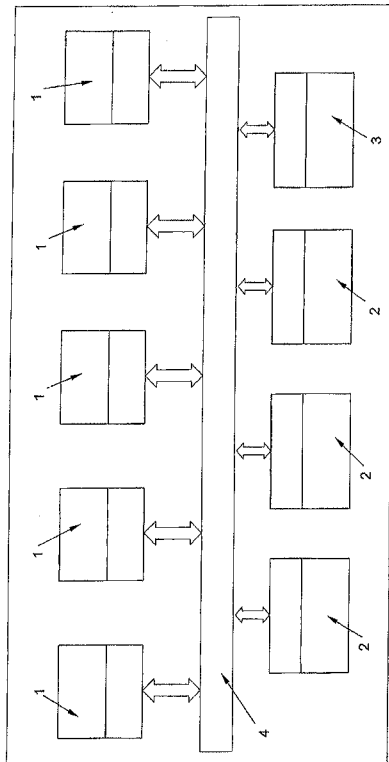
【0067】

制御プログラムの記述に基づいて、リアルタイムでシステム制御プログラムを実行するデータ処理システム用のランタイムシステムによって、システム制御プログラムを実行するための機能モジュールは、ランタイムシステムにおいて動的に管理されてもよい。すなわち、機能モジュールは、実行中に生成され、シャットダウンおよび電源オフ中に、起動および実行が行われてもよい。ランタイムシステムが起動されたとき、個々の機能モジュールが既に完全にロードされ、実行可能な状態である必要はない。全ての機能モジュールは同じ時間コンテキストで動作し、互換性のあるデータタイプおよび統一のメモリ領域を有することで個々の機能モジュールをリンカーで接続する必要もない。全ての機能モジュールは同じ時間コンテキストで動作し、互換性のあるデータタイプおよび統一のメモリ領域を有することで個々の機能モジュールをリンカーで接続する必要もない。

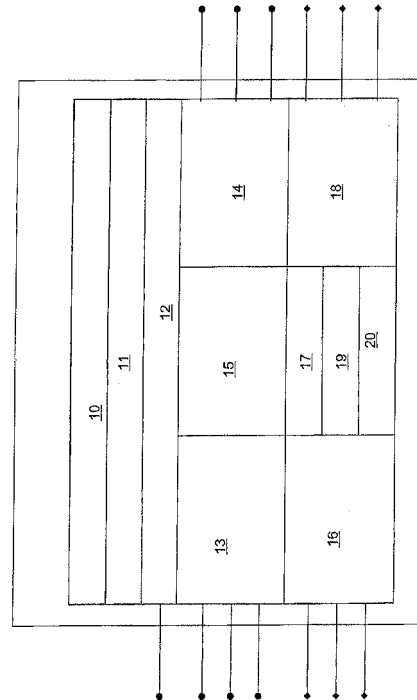
【0068】

上記記載は本発明の実施形態を対象としているが、本発明の他のさらなる実施形態をも、本発明の範囲を逸脱することなく実現可能である。本発明の請求の範囲は、続く請求項によって判断されるものである。

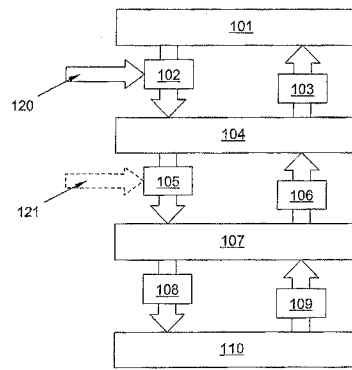
【図1】



【図2】



【図 3】



フロントページの続き

(72)発明者 ヤン ラムケマン

ドイツ連邦共和国, 3 3 3 3 5 ギュータースロー, レーブフーンヴェーク 2

審査官 鈴木 修治

(56)参考文献 特開2001-184221 (JP, A)

特開2009-217503 (JP, A)

特開2001-229172 (JP, A)

特表2005-537550 (JP, A)

特開2000-148702 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G 0 6 F 9 / 4 6 - 9 / 5 4