

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3907463号
(P3907463)

(45) 発行日 平成19年4月18日(2007.4.18)

(24) 登録日 平成19年1月26日(2007.1.26)

(51) Int. Cl.

F I

G 0 6 F 17/50 (2006.01)

G 0 6 F 17/50 6 1 4 A

G 0 6 F 9/54 (2006.01)

G 0 6 F 9/06 6 4 0 C

G 0 6 F 9/445 (2006.01)

G 0 6 F 9/06 6 5 0 D

請求項の数 7 (全 22 頁)

(21) 出願番号 特願2001-370946 (P2001-370946)
 (22) 出願日 平成13年12月5日(2001.12.5)
 (65) 公開番号 特開2002-259469 (P2002-259469A)
 (43) 公開日 平成14年9月13日(2002.9.13)
 審査請求日 平成16年5月24日(2004.5.24)
 (31) 優先権主張番号 特願2000-395662 (P2000-395662)
 (32) 優先日 平成12年12月26日(2000.12.26)
 (33) 優先権主張国 日本国(JP)

(73) 特許権者 000005223
 富士通株式会社
 神奈川県川崎市中原区上小田中4丁目1番
 1号
 (74) 代理人 100092152
 弁理士 服部 毅麿
 (72) 発明者 藤枝 誠
 神奈川県川崎市中原区上小田中4丁目1番
 1号 富士通株式会社内

審査官 木方 庸輔

(56) 参考文献 特開平08-235231(JP, A)

最終頁に続く

(54) 【発明の名称】 CADデータを管理するためのプログラムを記録したコンピュータ読み取り可能な記録媒体およびプログラム

(57) 【特許請求の範囲】

【請求項1】

CADデータを管理するためのプログラムを記録したコンピュータ読み取り可能な記録媒体であって、

前記コンピュータを、

所定の種別のCADデータを処理するCADデータ処理手段、

前記CADデータ処理手段に依存しない形式でCADデータに対する処理を特定する処理要求の集合があらかじめ定義されており、前記CADデータ処理手段で処理可能な種別のCADデータを指定した操作入力に応答して、操作入力に対応する前記処理要求を出力する処理要求出力手段、

前記処理要求を前記CADデータ処理手段が解釈可能な命令に変換する関数があらかじめ定義されており、前記処理要求出力手段から前記処理要求が出力されると、前記関数を実行して、前記CADデータ処理手段に前記処理要求によって特定される処理を実行させる連携手段、

として機能させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項2】

複数の種別のCADデータを管理するためのプログラムを記録したコンピュータ読み取り可能な記録媒体であって、

前記コンピュータを、

それぞれが所定の種別のCADデータを処理する複数のCADデータ処理手段、

前記 C A D データ処理手段に依存しない形式で C A D データに対する処理を特定する処理要求の集合があらかじめ定義されており、C A D データを指定した操作入力に応答して、操作入力に対応する前記処理要求を出力する処理要求出力手段、

前記処理要求を前記 C A D データ処理手段が解釈可能な命令に変換する関数がそれぞれの前記処理要求出力手段についてあらかじめ定義されており、前記処理要求出力手段から前記処理要求が出力されると、前記関数を実行して、指定された C A D データを処理可能な前記 C A D データ処理手段に前記処理要求によって特定される処理を実行させる連携手段、

として機能させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項 3】

前記処理要求出力手段は、前記複数の種別の C A D データの構成を表示し、表示された構成の少なくとも一部を選択する操作入力に基づいて、指定された C A D データを判断することを特徴とする請求項 2 記載の記録媒体。

【請求項 4】

前記コンピュータを、更に、

前記処理要求出力手段から出力される可能性のある前記処理要求に対応する前記関数をメモリにロードするロード手段、

前記ロード手段がロードした前記関数のメモリ内の位置を指し示すポインタを、ポインタテーブルに登録するポインタ登録手段として機能させ、

前記連携手段は、前記ポインタ登録手段により前記ポインタテーブルに登録された前記ポインタに基づいて、前記処理要求出力手段から出力された前記処理要求に対応する前記関数が登録されている場所を判断することを特徴とする請求項 2 記載の記録媒体。

【請求項 5】

前記ロード手段は、前記複数の C A D データ処理手段毎に、それぞれに対応する前記関数がまとめられたモジュールをロードすることを特徴とする請求項 4 記載の記録媒体。

【請求項 6】

C A D データを管理するためのプログラムであって、

コンピュータを、

所定の種別の C A D データを処理する C A D データ処理手段、

前記 C A D データ処理手段に依存しない形式で C A D データに対する処理を特定する処理要求の集合があらかじめ定義されており、前記 C A D データ処理手段で処理可能な種別の C A D データを指定した操作入力に応答して、操作入力に対応する前記処理要求を出力する処理要求出力手段、

前記処理要求を前記 C A D データ処理手段が解釈可能な命令に変換する関数があらかじめ定義されており、前記処理要求出力手段から前記処理要求が出力されると、前記関数を実行して、前記 C A D データ処理手段に前記処理要求によって特定される処理を実行させる連携手段、

として機能させるためのプログラム。

【請求項 7】

複数の種別の C A D データを管理するためのプログラムであって、

コンピュータを、

それぞれが所定の種別の C A D データを処理する複数の C A D データ処理手段、

前記 C A D データ処理手段に依存しない形式で C A D データに対する処理を特定する処理要求の集合があらかじめ定義されており、C A D データを指定した操作入力に応答して、操作入力に対応する前記処理要求を出力する処理要求出力手段、

前記処理要求を前記 C A D データ処理手段が解釈可能な命令に変換する関数がそれぞれの前記処理要求出力手段についてあらかじめ定義されており、前記処理要求出力手段から前記処理要求が出力されると、前記関数を実行して、指定された C A D データを処理可能な前記 C A D データ処理手段に前記処理要求によって特定される処理を実行させる連携手段、

10

20

30

40

50

として機能させるためのプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明はC A Dデータを管理するためのプログラムを記録したコンピュータ読み取り可能な記録媒体およびC A Dデータを管理するためのプログラムに関し、特に種類の異なるC A Dデータを処理するためのプログラムを記録した記録媒体および種類の異なるC A Dデータを処理するためのプログラムに関する。

【0002】

【従来の技術】

機械図面作成の分野では、3次元C A D (Computer-Aided Design)の導入が進んでいる。現在、市場には様々な種類の3次元C A Dアプリケーションソフトウェア（以下、「C A Dアプリケーション」という）が存在している。各C A Dアプリケーションでは、それぞれの特徴を出すために独自の機能を有している。C A Dアプリケーションで取り扱うデータのデータ構造は、そのC A Dアプリケーションの有している機能に適合したデータ構造となる。したがって、C A Dアプリケーションは、その種類ごとに個別のデータ構造のC A Dデータを生成する。しかも、C A Dアプリケーションはそれぞれ個別の機能が随時追加されるため、種別の異なるC A Dアプリケーション同士で互いにデータの互換性を保つのは困難である。

【0003】

一般に、種別の異なるC A Dデータを管理するために、各C A Dデータに対して、そのデータのC A D種別を表す属性項目を持たせている。C A Dデータにアクセスする際には、そのC A Dデータの属性項目を参照してC A D種別を判断し、種別に応じたC A Dアプリケーションを起動する。そして、起動されたC A DアプリケーションがC A Dデータのロード処理を行い、そのC A Dデータの内容が表示される。

【0004】

【発明が解決しようとする課題】

機械部品を製造するメーカーなどでは、部品納入先が指定するC A Dデータで機械図面を納める必要がある。そのため、部品納入先が複数あれば、部品メーカー内では、複数のC A Dアプリケーションが使用される。また、多くの会社では、事業体ごとにそれぞれの設計業務に合ったC A Dアプリケーションを導入している。そのため、社内には、データ構造の異なるC A Dデータが、幾種類も存在する。全社のシステムを考えた場合、これら異なるC A Dデータを統合的に管理したいという要求がある。

【0005】

しかし、3次元C A Dデータの多くはアセンブリ情報などのファイル同士の関連情報を内部的に持っている。そのため、複数のC A Dデータを統合的に管理するシステムを作るには、各C A Dデータに対応するC A Dアプリケーション固有のA P I (Application Programming Interface)を使用して、各C A Dデータをロードできるようにしなければならない。このように、C A Dデータの種別ごとにC A Dアプリケーション固有のA P Iを用いた統合管理用のアプリケーションを作成するのには、多大な手間を要する。

【0006】

そのため、種別の異なるC A Dデータを統合して管理できるシステムを、容易に構築できることが望まれている。

本発明はこのような点に鑑みてなされたものであり、任意の種別のC A Dデータの管理機能を容易に追加することができる、C A Dデータを管理するためのプログラムを記録したコンピュータ読み取り可能な記録媒体を提供することを目的とする。

【0007】

また、本発明の別の目的は、任意の種別のC A Dデータの管理機能を容易に追加することができる、C A Dデータを管理するためのプログラムを提供することである。

【0008】

10

20

30

40

50

【課題を解決するための手段】

本発明では上記課題を解決するために、図1に示す機能をコンピュータに実行させるための、CADデータを管理するためのプログラムを記録したコンピュータ読み取り可能な記録媒体が提供される。本発明の記録媒体に記録されたプログラムを実行するコンピュータは、次の構成を有している。CADデータ処理手段1a、1b、1cは、それぞれ所定の種別のCADデータを処理する。処理要求出力手段6は、CADデータ処理手段1a、1b、1cに依存しない形式でCADデータに対する処理を特定する処理要求の集合があらかじめ定義されており、CADデータ処理手段1a、1b、1cで処理可能な種別のCADデータを指定した操作入力に応答して、操作入力に対応する処理要求を出力する。連携手段7は、処理要求をCADデータ処理手段1a、1b、1cが解釈可能な命令に変換する関数があらかじめ定義されており、処理要求出力手段6から処理要求が出力されると、関数を実行して、CADデータ処理手段1a、1b、1cに処理要求によって特定される処理を実行させる。

10

【0009】

これにより、処理要求出力手段6に対して操作入力が行われると、処理要求出力手段6から操作入力に対応する処理要求が出力される。すると、連携手段7により、処理要求に対応する関数が実行され、CADデータ処理手段1a、1b、1cが解釈可能な命令に変換されて、処理が実行される。

【0010】

このように、処理要求に応じた関数を、連携手段7に定義するだけで、CADデータ処理手段1a、1b、1cに対して連携処理を実行させることが可能となる。その結果、任意の種別のCADデータの管理機能の追加が容易となる。

20

【0011】

また、上記課題を解決するために、CADデータを管理するためのプログラムであって、コンピュータを、所定の種別のCADデータを処理するCADデータ処理手段、前記CADデータ処理手段に依存しない形式でCADデータに対する処理を特定する処理要求の集合があらかじめ定義されており、前記CADデータ処理手段で処理可能な種別のCADデータを指定した操作入力に応答して、操作入力に対応する前記処理要求を出力する処理要求出力手段、前記処理要求を前記CADデータ処理手段が解釈可能な命令に変換する関数があらかじめ定義されており、前記処理要求出力手段から前記処理要求が出力されると、前記関数を実行して、前記CADデータ処理手段に前記処理要求によって特定される処理を実行させる連携手段、として機能させるためのプログラムが提供される。

30

【0012】

このようなプログラムをコンピュータに実行させれば、処理要求出力手段に対して操作入力が行われると、処理要求出力手段から操作入力に対応する処理要求が出力される。すると、連携手段により、処理要求に対応する関数が実行され、CADデータ処理手段が解釈可能な命令に変換されて、処理が実行される。

【0013】**【発明の実施の形態】**

以下、本発明の実施の形態を図面を参照して説明する。

40

図1は、本発明の原理構成図である。図1には、本発明に係るプログラムを記録した記録媒体を実行するコンピュータ（以下、CADデータ管理装置という）で実現される機能が示されている。このCADデータ管理装置は、複数の種別のCADデータを管理するものであり、以下の要素で構成される。

【0014】

CADデータ処理手段1a、1b、1cは、それぞれ異なる種別のCADデータの生成、編集等を行う機能を有している。CADデータ処理手段1a、1b、1cは、たとえば、3次元CADアプリケーションソフトウェア（CADアプリケーション）で提供されるCAD機能である。各CADデータ処理手段1a、1b、1cそれぞれに対応して、CADインタフェースモジュール2a、2b、2cが用意されている。CADインタフェースモ

50

ジュール 2 a , 2 b , 2 c には、処理要求出力手段 6 から出力される可能性のある処理要求に応じた複数の関数が定義されている。

【 0 0 1 5 】

ロード手段 3 は、C A D インタフェースモジュール 2 a , 2 b , 2 c をメモリにロードし、連携手段 7 による連携処理を可能な状態にする。ポインタ登録手段 4 は、ロード手段 3 がロードした関数のメモリ内の位置を指し示すポインタをポインタテーブル 5 に登録する。処理要求出力手段 6 は、C A D データを指定した操作入力に応答して、指定された C A D データを処理可能な C A D データ処理手段に対する処理要求を出力する。連携手段 7 は、処理要求出力手段 6 から出力された処理要求に対応する関数があらかじめ定義されており、関数に定義された処理を、指定された C A D データを処理可能な C A D データ処理手段と連携して実行する。

10

【 0 0 1 6 】

このように、処理要求出力手段 6 から出力された処理要求に対応する関数を、連携手段 7 が C A D データ処理手段 1 a , 1 b , 1 c と連携して処理するようにした。そのため、処理要求出力手段 6 に対して操作入力を行う作業者は、指定した C A D データの種別等を意識せずに、C A D データを処理可能な C A D データ処理部に対して処理を行わせることができる。つまり、統一されたユーザインタフェースを用いて、複数種の C A D データに対する処理要求を出すことが可能となる。

【 0 0 1 7 】

しかも、連携手段 7 にロードさせる関数を、たとえばインタフェースモジュールの形で用意するだけで、新たに追加された C A D データ処理手段にも対応させることができる。用意すべき関数は、処理要求出力手段 6 において出力される可能性のある処理要求に対応付けて用意すればよい。すなわち、新たな C A D データ処理部に対応する C A D インタフェースモジュールであっても、処理要求出力手段 6 において出力する処理要求に合わせた処理を定義すればよい。これは、各 C A D データ処理手段それぞれにおいて定義されている A P I を理解して、それぞれの A P I に合わせたアプリケーションを作成するよりも、容易な作業である。

20

【 0 0 1 8 】

以下に、本発明の実施の形態について詳細に説明する。

図 2 は、本実施の形態のシステム構成例を示すブロック図である。本実施の形態では、複数のクライアントコンピュータ 1 0 , 1 0 a とサーバコンピュータ 3 0 とがネットワーク 2 4 を介して接続されている。ネットワーク 2 4 は、たとえばイントラネットやインターネットである。

30

【 0 0 1 9 】

図 3 は、クライアントコンピュータの構成例を示す図である。クライアントコンピュータ 1 0 は、C P U 1 1 によって装置全体が制御されている。C P U 1 1 には、バス 1 7 を介して R A M 1 2 、ハードディスク装置 (H D D) 1 3 、グラフィック処理装置 1 4 、入力インタフェース 1 5 、および通信インタフェース 1 6 が接続されている。

【 0 0 2 0 】

R A M 1 2 は、C P U 1 1 に実行させる O S (Operating System) や C A D プログラムの少なくとも一部が一時的に格納される。また、R A M 1 2 には、C P U 1 1 による処理に必要な各種データが格納される。H D D 1 3 は、O S や C A D のプログラムが格納される。また、H D D 1 3 には、C A D データが格納される。

40

【 0 0 2 1 】

グラフィック処理装置 1 4 には、モニタ 2 1 が接続されている。グラフィック処理装置 1 4 は、C P U 1 1 からの命令に従って、C A D 図面等の画像をモニタ 2 1 の画面に表示させる。入力インタフェース 1 5 には、キーボード 2 2 とマウス 2 3 とが接続されている。入力インタフェース 1 5 は、キーボード 2 2 やマウス 2 3 から送られてくる信号を、バス 1 7 を介して C P U 1 1 に送信する。

【 0 0 2 2 】

50

通信インタフェース 16 は、ネットワーク 24 に接続されている。ネットワーク 24 は、たとえばインターネットのような広域ネットワークである。通信インタフェース 16 は、ネットワーク 24 を介して、他のコンピュータとの間でデータの送受信を行う。

【0023】

なお、サーバコンピュータ 30 も図 3 に示したクライアントコンピュータと同様の構成である。ただし、サーバコンピュータ 30 の HDD には、3 次元 CAD の CAD データが格納され、CAD データのデータベース (DB) が構築される。すなわち、サーバコンピュータ 30 は、CAD データのデータベースサーバとして機能する。

【0024】

以上のようなハードウェア構成によって、本実施の形態の処理機能を実現することができる。

10

図 4 は、本実施の形態の機能ブロック図である。図 4 に示すように、本実施の形態では、DB 40 に CAD データ 41 が格納されている。CAD データ 41 は、特定の CAD アプリケーションによって生成されている。そして、各 CAD データ 41 は、そのデータを生成した CAD データ処理部 71 ~ 73 に応じたデータ構造を有している。したがって、異なる CAD データ処理部 71 ~ 73 から生成された CAD データ同士は、異なるデータ構造を有している。たとえば、図中「CAD # 1」と示した CAD データは、「CAD # 1」と示した CAD データ処理部 71 に応じたデータ構造である。

【0025】

インタフェース (I/F) モジュール保持部 50 は、各 CAD データ処理部 71 ~ 73 に対応する CAD インタフェースモジュール 51 を保持している。CAD インタフェースモジュール 51 には、CAD データ管理部 60 からの要求を各 CAD データ処理部 71 ~ 73 に伝えるための関数が定義されている。

20

【0026】

CAD データ管理部 60 は、DB 40 に格納されている CAD データ 41 の管理を行う。また、CAD データ管理部 60 には、表示装置 81 や入力装置 82 が接続されている。そして、CAD データ管理部 60 は、表示装置 81 や入力装置 82 を介して、DB 40 を管理するための操作環境をユーザに提供している。

【0027】

CAD データ管理部 60 は、インタフェースモジュールロード部 61、ポインタ登録部 62、データベース管理部 63、および GUI 64 で構成される。

30

インタフェースモジュールロード部 61 は、CAD データ管理部 60 が起動された際に、インタフェースモジュール保持部 50 から RAM 12 に、インタフェースモジュール 51 をロードする。どのようなインタフェースモジュール 51 をロードすべきかについては、たとえば、レジストリに登録されている。ここでレジストリとは、コンピュータ上で動作する OS やアプリケーションの動作に必要な情報が格納されたデータベースである。本実施の形態では、CAD インタフェースモジュールが OS にインストールされる際に、インストールされた CAD インタフェースモジュールに関する情報がレジストリに設定されているものとする。

【0028】

40

さらに、インタフェースモジュールロード部 61 は、環境設定ファイル (たとえば、拡張子が「ini」のファイル) からインタフェースモジュールの名前や格納場所等を取得し、該当するインタフェースモジュール 51 を CAD インタフェースモジュール保持部 50 から取得する。インタフェースモジュールロード部 61 は、取得したインタフェースモジュール 51 を RAM 12 に格納する。

【0029】

ポインタ登録部 62 は、インタフェースモジュールロード部 61 のロード処理を監視し、インタフェースモジュール 51 の各関数が格納された RAM 12 内の場所 (アドレス) を取得する。ポインタ登録部 62 は、ロードしたインタフェースモジュール 51 の各関数の関数名とその関数が格納されたアドレスとの組を生成する。ポインタ登録部 62 は、生成

50

した関数名とアドレスとの組に基づいて、関数テーブル12bを生成し、生成した関数テーブル12bをRAM12に格納する。さらにポインタ登録部62は、ロードされたインタフェースモジュールの関数テーブル12bが格納されたRAM12内のアドレスを取得する。ポインタ登録部62は、取得した関数テーブルのアドレスとロードされたインタフェースモジュールの種別との組を、インタフェーステーブル12aに登録する。

【0030】

データベース管理部63は、DB40に格納されたCADデータの構成を解析し、その内容をGUI64に送る。また、データベース管理部63は、GUI64からの処理要求に応答して、DB40内のファイル操作を行う。たとえば、部品の形状を表すデータが1つのファイルに格納されていた場合に、そのファイルの削除やコピーの生成等を行うことができる。

10

【0031】

GUI64は、データベース管理部63からのデータに基づいて、DB40内のCADデータの構成を示す画像を生成し、表示装置81に送信する。またGUI64は、入力装置82からの入力信号に応じたコマンドを生成する。GUI64は、DB40内のファイル进行操作するコマンドを生成した場合には、そのコマンドをデータベース管理部63に送る。また、GUI64は、CADデータ処理部71~73に対する処理を要求するコマンドを生成した場合には、そのコマンドを渡すべきCADデータ処理部71~73に対応するCADインタフェース91~93を起動する。そして、GUI64は、起動したCADインタフェース91~93に対してコマンドを渡す。

20

【0032】

具体的には、GUI64は、処理すべきCADデータの属性種別に対応するCADデータ処理部を判断する。次に、GUI64は、そのCADデータ処理部に対応する関数テーブルへのポインタを、インタフェーステーブル12aより取得する。さらに、GUI64は、取得したポインタで指し示された関数テーブル12bを参照し、コマンドに含まれる関数へのポインタを取得する。そして、CADインタフェース91~93は、取得したポインタで指し示された場所に格納された関数の実行要求を出す。その結果、CADインタフェース91~93が起動され、実行要求で指定された関数が実行される。

【0033】

CADデータ処理部71~73は、操作入力に応答して、3次元CADのCADデータを生成する。各CADデータ処理部71~73は、それぞれ異なる機能を有している。また、各CADデータ処理部71~73は、それぞれ異なるデータ構造のCADデータを生成する。また、CADデータ処理部71~73は、CADインタフェース91~93からの要求に応じて、CADインタフェース91~93との連携処理を行う。

30

【0034】

表示装置81は、CADデータ管理部60等から送られる画像情報に基づいて、画像を表示する。表示装置81は、たとえば図3に示すモニタ21である。また、入力装置82は、CADにより図面を作成する作業からの操作入力を受け付け、CADデータ管理部60に入力する。入力装置82は、たとえば図3に示すキーボード22やマウス23である。

40

【0035】

CADインタフェース91~93は、それぞれ個別のCADデータ処理部71~73に対応して設けられている。CADインタフェース91~93は、インタフェースモジュールロード部61がRAM12にロードしたCADインタフェースモジュール内に定義された関数を実行するプロセスである。CADインタフェース91~93は、GUI64から渡されたコマンドに応じた関数を実行する。具体的には、GUI64から関数の場所を示すポインタがCADインタフェース91~93に渡される。そして、ポインタで示された場所に格納された関数を、CADインタフェース91~93が実行する。

【0036】

なお、各CADインタフェース91~93による関数の実行は、対応関係にあるCADデ

50

ータ処理部 7 1 ~ 7 3 との間で連携して行われる。連携処理の際には、各 C A D インタフェース 9 1 ~ 9 3 から C A D データ処理部 7 1 ~ 7 3 に対して、処理要求が出力される。どのような処理要求を出力するかは、関数内に定義されている。C A D データ処理部 7 1 ~ 7 3 が、各 C A D インタフェース 9 1 ~ 9 3 からの処理要求に応じた処理を実行することで、連携処理が行われる。

【 0 0 3 7 】

なお、図 4 に示した構成要素の中で、D B 4 0 は、たとえば図 2 に示したサーバコンピュータ 3 0 が有する機能である。また、D B 4 0 以外の構成要素は、たとえばクライアントコンピュータ 1 0 , 1 0 a が有する機能である。このように、D B 4 0 をサーバコンピュータ 3 0 に設けることで、C A D データを複数のクライアントコンピュータで共用することができ

10

【 0 0 3 8 】

次に、本実施の形態を実行する際に R A M 1 2 に格納されるデータについて説明する。図 5 は、R A M のメモリマップの例を示す図である。この図のメモリマップ 1 2 c に示すように、R A M 1 2 には、複数の C A D インタフェースモジュール 5 1 a , 5 1 b、インタフェーステーブル 1 2 a、複数の関数テーブル 1 2 b a , 1 2 b b が格納されている。図 5 では、アドレスを 1 6 進数で表している。図の例では、C A D # 1 の C A D インタフェースモジュール 5 1 a が、「0x0002000」番地から「0x0013FFF」番地までのメモリ空間に格納されている。C A D # 2 の C A D インタフェースモジュール 5 1 b が、「0x0014000」番地から「0x0019FFF」番地までのメモリ空間に格納されている。

20

【 0 0 3 9 】

なお、各 C A D インタフェースモジュール 5 1 a , 5 1 b は、それぞれが関数の集合体である。関数とは、所定の要求に対応する処理を実行し、応答を返すまでの処理手順が定義されたサブルーチンプログラムである。C A D インタフェースモジュール 5 1 a , 5 1 b は、たとえば、ダイナミックリンクライブラリ (D L L) の形式のファイルであらかじめ H D D 等に格納されている。また、本実施の形態では、各関数をメソッドと呼ぶこととする。

【 0 0 4 0 】

図の例では、各 C A D インタフェースモジュール 5 1 a , 5 1 b に対して、ロードメソッドやセーブメソッドが用意されている。ロードメソッドは、C A D データを C A D データ処理部にロード処理を行わせるための関数である。セーブメソッドは、C A D データ処理部に、編集された C A D データのセーブ処理を行わせるための関数である。

30

【 0 0 4 1 】

インタフェーステーブル 1 2 a は、「0x0120000」番地から「0x0120FFF」番地までのメモリ空間に格納されている。「C A D # 1」の関数テーブル 1 2 b a が、「0x0121000」番地から「0x0121FFF」番地までのメモリ空間に格納されている。「C A D # 2」の関数テーブル 1 2 b b が、「0x0122000」番地から「0x0122FFF」番地までのメモリ空間に格納されている。

【 0 0 4 2 】

R A M 1 2 に格納されたインタフェーステーブル 1 2 a と関数テーブル 1 2 b a , 1 2 b b とは、たとえば以下のようなデータ構成になる。

40

図 6 は、インタフェーステーブルの一例を示す図である。インタフェーステーブル 1 2 a には、I / F 種別とポインタとの欄が設けられている。I / F 種別の欄には、ロードされた C A D インタフェースモジュールが対応している C A D データ処理部の名称が登録されている。ポインタの欄には、I / F 種別の欄に登録された名称に関連づけて、ロードされた C A D インタフェースモジュールの関数テーブルの格納先アドレスが登録されている。この格納先アドレスは、関数テーブルが格納された領域の先頭アドレスである。

【 0 0 4 3 】

図 6 の例では、I / F 種別が「C A D # 1」の C A D インタフェースモジュールの関数テーブルが、「0x0121000」番地から始まるアドレスに格納されていることを示している。

50

また、I / F 種別が「C A D # 2」のC A Dインタフェースモジュールの関数テーブルが、「0x0122000」番地から始まるアドレスに格納されていることを示している。

【0044】

図7は、関数テーブルの一例を示す図である。関数テーブル12ba, 12bbには、メソッド（関数の名称）の欄とポインタの欄とが設けられている。メソッドの欄には、関数の種別が登録されている。図7の例では、ロードやセーブ等のメソッドが登録されている。ポインタの欄には、各関数が格納された領域の先頭アドレスを指し示すポインタが登録されている。図7に示す「C A D # 1」に対応する関数テーブル12baでは、ロードメソッドへのポインタとして「0x0002000」が登録されている。セーブメソッドへのポインタとして「0x0004000」が登録されている。

10

【0045】

以上のような構成のシステムにおいて、本実施の形態の処理が実行される。まず、C A Dインタフェース91～93のロード手順について説明する。

図8は、ロード処理手順を示すフローチャートである。以下、図8のフローチャートをステップ番号に沿って説明する。なお、本実施の形態では、C A Dデータ管理部60の機能がクライアントコンピュータで起動された際に、以下のロード処理が実行されるものとする。

【0046】

〔ステップS11〕インタフェースモジュールロード部61は、コンピュータのHDDに格納されているレジストリ情報を取得する。

20

〔ステップS12〕インタフェースモジュールロード部61は、取得したレジストリ情報の内容を解析し、ロードすべきC A Dインタフェースモジュールの一覧を取得する。

【0047】

〔ステップS13〕インタフェースモジュールロード部61は、各C A Dインタフェースモジュールに関する情報を、環境設定ファイル（たとえば「ini」の拡張子が付けられたテキストファイル）より取得する。環境設定ファイルは、たとえばC A Dインタフェースモジュール毎に設けられている。このステップで取得される情報は、C A Dインタフェースモジュールのファイル名（たとえば、拡張子が「dll」であるダイナミックリンクライブラリのファイル名）や、種別に関する情報である。ここで、種別とは、どのC A Dデータ処理部（C A Dアプリケーション）に対応するインタフェースなのかを示す情報である。

30

【0048】

〔ステップS14〕インタフェースモジュールロード部61は、インタフェースモジュール保持部50より、1つのC A DインタフェースモジュールをRAM12にロードする。

〔ステップS15〕インタフェースモジュールロード部61は、ロードしたC A Dインタフェースモジュールの関数テーブルを登録すべき記憶領域をRAM12内に確保する。そして、インタフェースモジュールロード部61は、確保した記憶領域の先頭のアドレス（ポインタ）を、インタフェーステーブルに登録する。

【0049】

〔ステップS16〕インタフェースモジュールロード部61は、ロードしたC A Dインタフェースモジュールに含まれるメソッド（関数）が格納されたアドレス（ポインタ）を、関数テーブルに登録する。

40

〔ステップS17〕インタフェースモジュールロード部61は、登録すべきC A Dインタフェースモジュールが他にあるか否かを判断する。インタフェースモジュールロード部61は、登録すべきC A Dインタフェースモジュールがある場合には、処理をステップS13に進め、登録すべきC A Dインタフェースモジュールがない場合には、ロード処理を終了する。

【0050】

このような処理をC A Dデータ管理部60の起動時に実行することにより、C A Dデータ管理部60から各種C A Dデータ処理部へのコマンドの処理時間が高速化される。

50

【 0 0 5 1 】

次に、C A D データ管理部 6 0 による C A D データ処理部へのコマンド出力方法について説明する。本実施の形態では、C A D データ管理部 6 0 の G U I 6 4 が、入力装置 8 2 からの入力内容を解析して、コマンドを生成する。作業者は、画面に表示されるアセンブリ構成から任意のデータを選択することで、処理対象のオブジェクト（部品データやアセンブリデータ）を指定することが出来る。

【 0 0 5 2 】

図 9 は、アセンブリの構成を表示する画面例を示す図である。この表示画面 1 0 0 の上部には複数のメニューが設けられている。ファイルメニュー 1 0 1 は、データのロード等を行うためのメニューである。編集メニュー 1 0 2 は、ファイルのコピー等の操作を行うためのメニューである。表示メニュー 1 0 3 は、画面の表示方法等を設定するためのメニューである。オブジェクトメニュー 1 0 4 は、オブジェクトに対する専有（他の作業者による更新等の操作を禁止すること）の設定や専有解除の設定等を行うためのメニューである。ここで、オブジェクトとは、C A D データで定義される製品の構成要素（ユニット、サブユニット、部品）を示す。ツールメニュー 1 0 5 は、メニューバーのカスタマイズ等を行うためのメニューである。ヘルプメニュー 1 0 6 は、オンラインマニュアルの表示等のヘルプ機能を起動するためのメニューである。

10

【 0 0 5 3 】

メニューバーの左下には、データ構成表示部 1 0 7 が設けられている。データ構成表示部 1 0 7 には、D B に格納されているデータの構成が表示される。図 9 の例では「A s s y」という名のアセンブリ 1 0 7 a の下位構造として、「S u b A s s y」という名のサブアセンブリ 1 0 7 b が存在していることが表示されている。

20

【 0 0 5 4 】

データ構成表示部 1 0 7 の右側には、データ表示部 1 0 8 が設けられている。データ表示部 1 0 8 には、データ構成表示部 1 0 7 で選択されたアセンブリを構成するデータが表示される。図 9 の例では、データ構成表示部 1 0 7 においてサブアセンブリ 1 0 7 b が選択されており、データ表示部 1 0 8 には、サブアセンブリ 1 0 7 b を構成する部品データ 1 0 8 a , 1 0 8 b が表示されている。

【 0 0 5 5 】

図 9 に示したような C A D データの構成表示処理は、図 4 に示すデータベース管理部 6 3 と G U I 6 4 とによって行われる。すなわち、データベース管理部 6 3 が D B 4 0 内の C A D データ 4 1 の内容を解析する。解析の結果判明した C A D データの木構造に基づいて、G U I 6 4 が図 9 に示したような画像を表示装置 8 1 表示させる。

30

【 0 0 5 6 】

作業者は、表示された C A D データ中の 1 以上のオブジェクトを選択し、データのロード指令を入力することで、クライアントコンピュータに対して、選択したオブジェクトのデータをロードさせることができる。データのロード指令の入力は、たとえば C A D データ管理部 6 0 の G U I 6 4 が提供する表示画面 1 0 0 内のメニューを選択することで行われる。

【 0 0 5 7 】

図 1 0 は、C A D データをロードする際の表示画面例を示す図である。この例では、マウス等のポインティングデバイスを用いてマウスカーソル 1 1 1 を移動させ、所望のコマンドを選択することで、ロード指令を入力することができる。

40

【 0 0 5 8 】

メニューバーのファイルメニュー 1 0 1 を選択すると、開くコマンド 1 0 1 a や終了コマンド 1 0 1 b を含むプルダウンメニューが表示される。開くコマンド 1 0 1 a は、C A D データをロードするためのコマンドである。終了コマンド 1 0 1 b は、C A D データ管理部 6 0 の処理を終了させるためのコマンドである。

【 0 0 5 9 】

作業者が開くコマンド 1 0 1 a を選択すると、ロード種別を指定するためのサブメニュー

50

(ポップアップメニュー)が開く。このサブメニューには、全ロードコマンド101c、部分ロードコマンド101d、および再ロードコマンド101eが含まれている。

【0060】

全ロードコマンド101cは、選択されているアセンブリが属するCADデータの全てをコンピュータにロードさせるためのコマンドである。たとえば、サブアセンブリを選択した状態で全ロードコマンド101cが指定されると、選択されたサブアセンブリを含む製品の全てのCADデータ(最上位のアセンブリデータ、最上位のアセンブリデータ配下の各サブアセンブリデータや部品データ)がロードされる。

【0061】

部分ロードコマンド101dは、製品の一部のデータをコンピュータにロードさせるためのコマンドである。たとえば、サブアセンブリを選択して部分ロードコマンド101dが指定されると、選択されたサブアセンブリデータおよびそのサブアセンブリの下位構造のデータがロードされる。

10

【0062】

再ロードコマンド101eは、参照目的で開いたデータに関する最新版のデータをロードし直すためのコマンドである。たとえば、設計責任者がある部品データをロードし、その部品形状を画面に表示させている最中にも、その部品の作業者が、部品データに変更を加え改訂版を保存する場合がある。このとき、設計責任者が再ロードコマンド101eを指令することで、改訂版の部品データを画面に表示させることができる。

【0063】

20

全ロードコマンド101cが選択されると、さらにサブメニュー(ポップアップメニュー)が表示される。サブメニューには、最新の状態ロードコマンド101f、最新のマスタロードコマンド101g、および登録時の状態ロードコマンド101hが含まれる。

【0064】

最新の状態ロードコマンド101fが指定されると、ロードすべきオブジェクトに関する最新の版数のデータがロードされる。最新のマスタロードコマンド101gが指定されると、ロードすべきオブジェクトにおける承認されたデータの中で、最新の版数のデータがロードされる。登録時の状態ロードコマンド101hが指定されると、ロードすべきオブジェクトのアセンブリデータに含まれる登録時情報に示されている版数のデータがロードされる。

30

【0065】

なお、部分ロードコマンド101dが選択された場合にも、全ロードコマンド101cが選択された場合と同様のサブメニューが表示される。

以上のような画面により、作業者が所望のロードコマンドを選択することで、ロード指令を入力することが出来る。ロード指令が入力されると、GUI64によって、そのロード指令に応じたロードコマンドが生成される。そして、インタフェース呼び出し処理を実行される。

【0066】

図11は、インタフェース呼び出し処理手順を示すフローチャートである。以下に、図11の処理をステップ番号に沿って説明する。

40

[ステップS21] GUI64は、指定オブジェクトをDB40から取り出す。指定オブジェクトとは、画面上で作業者が指定したオブジェクトである。

[ステップS22] GUI64は、取り出した指定オブジェクトに含まれる情報の中から、アプリケーション種別属性を取り出す。アプリケーション種別属性とは、そのオブジェクトがどのCADデータ処理部(CADアプリケーション)で取り扱うことができるのかを示す情報である。

【0067】

作業により指定されたコマンドに、指定オブジェクトおよびそのオブジェクトのアプリケーション種別属性をパラメータとして付加した情報が、インタフェースモジュールへの処理要求となる。

50

【ステップS23】GUI64は、インタフェーステーブル12aを参照し、処理要求に含まれるアプリケーション種別属性に対応するCADアプリケーションのポイントを取得する。このポイントは、CADアプリケーションに対応する関数テーブルの場所（先頭アドレス）を示している。

【0068】

【ステップS24】GUI64は、処理要求を解析し、処理すべき関数を決定する。処理要求が1つの関数で実行される場合には、その関数が処理すべき関数として決定される。処理要求が複数の関数で実行される場合には、まだ実行していない関数の中で最も先に実行すべき関数が、処理すべき関数として決定される。たとえば、ある関数（第1の関数）の戻り値を利用して、他の関数（第2の関数）に対して処理を実行させる場合がある。この場合、第1の関数が、先に実行すべき関数となる。

10

【0069】

【ステップS25】GUI64は、ステップS23で取得したポイントで指し示された場所の関数テーブルの中から、ステップS24で決定された処理すべき関数を検索する。そして、処理すべき関数へのポイントを取得する。

【ステップS26】GUI64は、ステップS25で取得したポイントに基づいて関数を呼び出す。関数が呼び出されると、その関数を実行するためのCADインタフェース（プロセス）がOSによって生成される。そして、生成されたCADインタフェースが関数の処理を、CADデータ処理部と連携して実行する。関数の戻り値は、GUI64に返される。

20

【0070】

【ステップS27】GUI64は、他に処理すべき関数があるか否かを判断し、処理すべき関数があればステップS24に処理を進め、処理すべき関数が無ければ処理を終了する。

【0071】

以上のようにして、CADデータ管理部60に対して入力されたコマンドを、CADインタフェースを介して、CADデータ管理部60とCADデータ処理部とが連携して実行することができる。

【0072】

ところで、GUI64で入力可能なコマンドには、全てのCADデータ処理部71～73で共通のコマンドと、各CADデータ処理部71～73それぞれ固有のコマンドとがある。共通のコマンドについては、そのコマンドによる関数呼び出しの形式があらかじめ決められている。一方、固有のコマンドについては、GUI64において、予約外のコマンドを処理するためのモジュール（関数）を呼び出すように定義されている。予約外のコマンドのパラメータとして、その処理名を受け渡す。呼び出された関数は、パラメータとして受け取った処理名に応じた処理を実行する。これにより、CADデータ処理部71～73それぞれに固有の処理が実行される。

30

【0073】

以上のように本実施の形態では、CADデータ管理部60から出力される関数への処理要求の種類は、あらかじめ決められている。したがって、CADインタフェースを開発する者は、CADデータ処理部の機能に応じて、各処理要求に応じた関数の処理内容を定義しておけばよい。

40

【0074】

関数の処理をCADインタフェースで実行させることにより、CADデータ管理部60とCADインタフェースとの間での取り決めが少なくてすむ。その結果、CADインタフェースの開発が容易となる。

【0075】

なお、上記実施の形態では、CADインタフェースが独立のプロセスとして起動されている。CADデータ管理部60と各CADインタフェースとの間では、プロセス間のメッセージ交換が行われる。それぞれのCADインタフェースが独立したプロセスで実行される

50

ことで、種別の異なる複数のC A Dデータを指定した処理要求を、C A Dデータごとに並列で処理することが可能となる。

【0076】

[応用例]

本実施の形態では、次のような応用例が考えられる。

[C A Dインタフェースモジュール問い合わせ方式]

上記の実施の形態では、C A Dデータに対して、そのC A Dデータの属性種別があらかじめ定義されているものとしている。これには、新たに登録するC A Dデータに対して、そのC A Dデータの属性種別を設定する必要がある。しかし、各インタフェースに対して、そのインタフェースモジュールが処理可能なデータか否かを判断するメソッド（関数）を用意することでも、選択されたC A Dデータに応じたC A Dインタフェースの選択が可能となる。

10

【0077】

たとえば、インタフェースモジュールの先頭に、「IsMydata」というメソッド（関数）を定義する。このメソッドは、そのインタフェースモジュールが処理可能なデータか否かを判断する関数である。そして、コマンドの入力が行われる、G U I 6 4が各インタフェースモジュールの先頭にある「IsMydata」というメソッドに対して、選択されたオブジェクトを処理可能かどうかの問い合わせを行う。G U I 6 4は、処理可能であるという応答を返してきたC A Dインタフェースに対して、コマンドの処理要求を渡す。

【0078】

20

図12は、C A Dインタフェースモジュール問い合わせの処理手順を示すフローチャートである。図12のフローチャートを、ステップ番号に沿って説明する。

【0079】

[ステップS31] G U I 6 4は、指定オブジェクトをD B 4 0から取り出す。指定オブジェクトとは、画面上で作業者が指定したオブジェクトである。

[ステップS32] G U I 6 4は、取り出した指定オブジェクトに含まれる情報の中から、アプリケーション種別属性を取り出す。

【0080】

[ステップS33] G U I 6 4は、インタフェーステーブルおよび各関数テーブルを参照し、各C A Dインタフェースモジュールの先頭に記憶されている、「IsMydata」メソッドの位置を取得する。そして、指定オブジェクトの処理が可能かどうかを、「IsMydata」メソッドに問い合わせる。これにより、各C A Dインタフェースが起動され、それぞれのC A Dインタフェースにおいて「IsMydata」メソッドが実行される。実行結果は、G U I 6 4に返される。たとえば、指定オブジェクトを処理できる場合には「1」のデータが返され、指定オブジェクトを処理できない場合には「0」のデータが返されるようにする。

30

[ステップS34] G U I 6 4は、処理可能と応答したC A Dインタフェースの1つを、処理要求の出力先として決定する。

【0081】

以上のように、C A Dインタフェースモジュールの先頭に置かれたメソッドを用いることで、各C A Dデータを処理可能なC A Dインタフェースを見つけることが可能となる。

40

【0082】

[C A Dドライバ方式]

上記の実施の形態では、C A Dインタフェースモジュールを「ダイナミックインクライブラリ（DLL）」形式のファイルで格納しているが、デバイスドライバの形式でC A DインタフェースモジュールをO Sに組み込むこともできる。

【0083】

たとえば、各クライアントコンピュータに対して、各C A Dデータ処理部（C A Dアプリケーション）に対応したインタフェースモジュールを、デバイスドライバとしてロードする。このとき、インタフェースモジュールに対しては、「Unknown Drive」としてドライバを割り当てる。C A Dデータ管理部60のG U I 6 4は、各C A Dデータ処理部に対す

50

る処理要求を、そのC A Dデータ処理部に対応するドライブに対して依頼する。これにより、各C A Dデータ処理部に対応する処理を実行することが可能となる。

【0084】

[サーバ処理方式]

C A Dデータ処理部ごとに処理を実行するサーバを定義しておくことで、各C A Dデータ処理部に対応する処理を実行することも可能である。この場合、C A Dデータ管理部60のG U I 64は、各C A Dデータ処理部に対する処理要求を、そのC A D種別に対応したサーバに依頼する。サーバでは、依頼された処理要求に応じた処理を実行し、処理結果をクライアントコンピュータのG U Iに返す。

【0085】

[コマンドメニューの動的切り替え方式]

上記実施の形態のC A Dデータ管理部60は、複数の種別のC A Dデータの構成を解析し、その構成を画面に表示することができる。そして、所望のC A Dデータのオブジェクトを選択し、コマンドメニューからコマンドを指定することで、選択したオブジェクトを処理可能なC A Dデータ処理部に対して、指定したコマンドに応じた処理要求が出される。

【0086】

しかし、コマンドメニューの中には、一部のC A Dデータ処理部しか対応していないコマンドもある。そこで、選択されたオブジェクトの種別に対応するC A Dデータ処理部が対応していないコマンドを、画面上で選択できないようにすることもできる。

【0087】

たとえば、各C A Dデータ処理部で対応しているコマンドの一覧を、各C A Dデータ処理部のC A Dインタフェースモジュールの先頭に定義する。

図13は、対応コマンド一覧の例を示す図である。この例では、C A Dデータ処理部ごとに、対応コマンド一覧201、202が設けられている。たとえば、「C A D # 1」のC A Dデータ処理部の対応コマンド一覧201には、「全ロード」、「部分ロード」、「再ロード」、「セーブ」等のコマンドに対応する機能を有していることが示されている。「C A D # 2」のC A Dデータ処理部の対応コマンド一覧202には、「全ロード」、「再ロード」、「セーブ」等のコマンドに対応する機能を有していることが示されている。この例では、「C A D # 2」のC A Dデータ処理部は、部分ロードに対応する機能を有していない。そのため、「C A D # 2」のC A Dデータ処理部に対応するC A Dデータが選択された場合には、C A Dデータ管理部60のG U I 64において、部分ロードのコマンドを選択できなうようにする。

【0088】

選択できないコマンドは、プルダウンメニューやポップアップメニューに表示されないようにしてもよい。また、選択できないコマンドの文字の色を背景の色に近づけることで、そのコマンドが選択できないことを、画面上で明示してもよい。

【0089】

図14は、コマンドメニュー表示処理のフローチャートである。以下に、図14の処理を、ステップ番号に沿って説明する。

[ステップS41] G U I 64は、指定オブジェクトをD B 40から取り出す。[ステップS42] G U I 64は、D B 40から、指定オブジェクトのアプリケーション種別属性の情報を取り出す。

【0090】

[ステップS43] G U I 64は、作業者の操作入力に応じたポップアップメニューに含まれる表示対象コマンドのリストを取得する。

[ステップS44] G U I 64は、指定オブジェクトに対応するC A Dインタフェースモジュールの対応コマンドテーブルを参照し、ステップS43で取得した各表示対象コマンドが実行可能か否かを判断する。対応コマンドテーブルに含まれるコマンドのみが、そのC A Dインタフェースモジュールにおいて実行可能なコマンドである。

[ステップS45] G U I 64は、ステップS44の処理で、実行可能なコマンドとされ

10

20

30

40

50

たコマンドの表示のみを選択可能として、ポップアップメニューを画面に表示する。

【 0 0 9 1 】

このようにして、選択されている C A D データの種別に応じて、ポップアップメニューの内容を動的に切り替えることが可能となる。

なお、各 C A D データ処理部で対応している機能のメニュー一覧を取り出すためのメソッド（関数）を、C A D インタフェースモジュールの先頭に定義してもよい。この場合、メニューを表示する際には、G U I 6 4 は、メニュー一覧を取り出すためのメソッドを呼び出して、戻り値としてメニュー一覧を得る。そして、G U I 6 4 は、この戻り値に基づいて、ポップアップメニューの内容を動的に切り替える。

【 0 0 9 2 】

また、C A D データ管理部 6 0 において標準で表示されるように設定されているメニュー中のコマンドに関しては、個別に C A D インタフェースに問い合わせるようにしてもよい。たとえば、各 C A D インタフェースには、コマンドに対応する処理の可否に関する問い合わせに回答するメソッド（関数）を定義しておく。そして、画面上にコマンドメニューを表示する際には、G U I 6 4 が、メニューに含まれるコマンドに対応する処理の可否を、選択されたオブジェクトに対応する C A D インタフェースに問い合わせる。そして、G U I 6 4 は、処理可能という回答があったコマンドのみを選択可能な状態して、ポップアップメニュー等のコマンドメニューを表示する。

【 0 0 9 3 】

[C A D 種別に応じたメッセージ表示方式]

上記の実施の形態によれば、複数の属性種別の C A D データを、C A D データ管理部 6 0 の G U I 6 4 を介して処理することができる。ところが、同じものの呼び名が、C A D データ処理部（C A D アプリケーション）毎に異なる場合がある。そこで、C A D データ管理部 6 0 において、C A D データの種別属性ごとにメッセージファイルを用意する。そして、G U I 6 4 は、C A D データの処理を行った際には、その C A D データの種別属性に対応するメッセージファイルからメッセージを取得し、そのメッセージを表示する。

【 0 0 9 4 】

これにより、C A D データ管理部 6 0 内において実行する処理であっても、処理対象となるオブジェクト（部品やアセンブリ）の種別属性に応じたメッセージを表示させることができる。

【 0 0 9 5 】

[オブジェクト処理メニュー分離方式]

C A D データ管理部 6 0 の機能には、C A D データ処理部に関係しない機能が含まれる。たとえば、サーバへのアクセスに関する環境設定等の処理は、C A D データ処理部に依存せずに行うことができる。そこで、コマンドメニュー内のコマンドが、C A D データ処理部に関係があるか否かを識別する識別子を、メニュー定義ファイルとして用意しておいてもよい。そして、G U I 6 4 は、指定されたコマンドが C A D データ処理部に関係のない処理を要求するコマンドであれば、C A D データ処理部に対応する C A D インタフェースモジュールとは別に用意した、固定のモジュールを呼び出して処理を実行させる。

【 0 0 9 6 】

[記録媒体に記録されたプログラムでの実現]

なお、上記の処理機能は、コンピュータによって実現することができる。その場合、コンピュータに実行させるべき機能の処理内容は、コンピュータで読み取り可能な記録媒体に記録されたプログラムに記述されており、このプログラムをコンピュータで実行することにより、上記処理がコンピュータで実現される。コンピュータで読み取り可能な記録媒体としては、磁気記録装置や半導体メモリ等がある。市場へ流通させる場合には、C D - R O M (Compact Disk Read Only Memory) やフレキシブルディスク等の可搬型記録媒体にプログラムを格納して流通させることができる。また、ネットワークを介して接続されたコンピュータの記憶装置にプログラムを格納しておき、ネットワークを介して接続された他のコンピュータにプログラムを転送することもできる。コンピュータで実行する際には、

10

20

30

40

50

コンピュータ内のハードディスク装置等にプログラムを格納しておき、メインメモリにロードして実行することができる。

【0097】

(付記1) CADデータを管理するためのプログラムを記録したコンピュータ読み取り可能な記録媒体であって、

前記コンピュータを、

所定の種別のCADデータを処理するCADデータ処理手段と、

前記CADデータ処理手段で処理可能な種別のCADデータを指定した操作入力にตอบสนองして、前記CADデータ処理手段に対する処理要求を出力する処理要求出力手段と、

前記処理要求出力手段から出力された前記処理要求に対応する関数があらかじめ定義されており、前記関数に定義された処理を前記CADデータ処理手段と連携して実行する連携手段と、

10

として機能させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【0098】

(付記2) 複数の種別のCADデータを管理するためのプログラムを記録したコンピュータ読み取り可能な記録媒体であって、

前記コンピュータを、

それぞれが所定の種別のCADデータを処理する複数のCADデータ処理手段と、

CADデータを指定した操作入力にตอบสนองして、前記複数のCADデータ処理手段の中の、

前記指定されたCADデータを処理可能なCADデータ処理手段に対する処理要求を出力する処理要求出力手段と、

20

前記処理要求出力手段から出力された前記処理要求に対応する関数があらかじめ定義されており、前記関数に定義された処理を、前記指定されたCADデータを処理可能なCADデータ処理手段と連携して実行する連携手段と、

として機能させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【0099】

(付記3) 前記処理要求出力手段は、前記複数の種別のCADデータの構成を表示し、表示された構成の少なくとも一部を選択する操作入力に基づいて、指定されたCADデータを判断することを特徴とする付記2記載の記録媒体。

【0100】

30

(付記4) 前記処理要求出力手段から出力される可能性のある処理要求に対応する関数をメモリにロードするロード手段と、

前記ロード手段がロードした前記関数のメモリ内の位置を指し示すポインタを、ポインタテーブルに登録するポインタ登録手段とをさらに有し、

前記連携手段は、前記ポインタ登録手段により前記ポインタテーブルに登録された前記ポインタに基づいて、前記処理要求出力手段から出力された前記処理要求に対応する関数が登録されている場所を判断することを特徴とする付記2記載の記録媒体。

【0101】

(付記5) 前記ロード手段は、前記複数のCADデータ処理手段毎に、それぞれに対応する関数がまとめられたモジュールをロードすることを特徴とする付記4記載の記録媒体。

40

【0102】

(付記6) 前記複数のCADデータ処理手段毎の前記各モジュールの中には、それぞれのモジュールに含まれる関数で対応可能なCADデータの種別を応答する関数が含まれていることを特徴とする付記5記載の記録媒体。

【0103】

(付記7) 前記処理要求出力手段は、前記指定されたCADデータを処理可能なCADデータ処理手段に対して、実行可能な処理内容を問い合わせる処理要求を出力し、出力した処理要求に対する応答内容に応じて、画面上で選択可能なメニューの種別を切り替えることを特徴とする付記2記載の記録媒体。

50

【 0 1 0 4 】

(付 記 8) C A D データを管理するための C A D データ管理方法であって、所定の種別の C A D データを処理する C A D データ処理機能で処理可能な種別の C A D データを指定した操作入力に応答して、前記 C A D データ処理機能に対する処理要求を出力し、出力された前記処理要求に対応する関数があらかじめ定義されており、前記関数に定義された処理を前記 C A D データ処理機能と連携して実行すること、を特徴とする C A D データ管理方法。

【 0 1 0 5 】

(付 記 9) 複数の種別の C A D データを管理する C A D データ管理方法であって、それぞれが所定の種別の C A D データを処理する複数の C A D データ処理機能が用意されており、C A D データを指定した操作入力に応答して、前記複数の C A D データ処理機能の中の、前記指定された C A D データを処理可能な C A D データ処理機能に対する処理要求を出力し、出力された前記処理要求に対応する関数があらかじめ定義されており、前記関数に定義された処理を、前記指定された C A D データを処理可能な C A D データ処理手段と連携して実行すること、を特徴とする C A D データ管理方法。

10

【 0 1 0 6 】

(付 記 1 0) C A D データを管理する C A D データ管理装置であって、所定の種別の C A D データを処理する C A D データ処理手段と、前記 C A D データ処理手段で処理可能な種別の C A D データを指定した操作入力に応答して、前記 C A D データ処理手段に対する処理要求を出力する処理要求出力手段と、前記処理要求出力手段から出力される前記処理要求に対応する関数があらかじめ定義されており、前記関数に定義された処理を前記 C A D データ処理手段と連携して実行する連携手段と、を有することを特徴とする C A D データ管理装置。

20

【 0 1 0 7 】

(付 記 1 1) 複数の種別の C A D データを管理する C A D データ管理装置であって、それぞれが所定の種別の C A D データを処理する複数の C A D データ処理手段と、C A D データを指定した操作入力に応答して、前記複数の C A D データ処理手段の中の、前記指定された C A D データを処理可能な C A D データ処理手段に対する処理要求を出力する処理要求出力手段と、前記処理要求出力手段から出力された前記処理要求に対応する関数があらかじめ定義されており、前記関数に定義された処理を、前記指定された C A D データを処理可能な C A D データ処理手段と連携して実行する連携手段と、を有することを特徴とする C A D データ管理装置。

30

【 0 1 0 8 】

(付 記 1 2) C A D データを管理するためのプログラムであって、コンピュータを、所定の種別の C A D データを処理する C A D データ処理手段と、前記 C A D データ処理手段で処理可能な種別の C A D データを指定した操作入力に応答して、前記 C A D データ処理手段に対する処理要求を出力する処理要求出力手段と、前記処理要求出力手段から出力された前記処理要求に対応する関数があらかじめ定義されており、前記関数に定義された処理を前記 C A D データ処理手段と連携して実行する連携手段と、として機能させるためのプログラム。

40

【 0 1 0 9 】

(付 記 1 3) 複数の種別の C A D データを管理するためのプログラムであって、コンピュータを、それぞれが所定の種別の C A D データを処理する複数の C A D データ処理手段と、

50

C A Dデータを指定した操作入力にตอบสนองして、前記複数のC A Dデータ処理手段の中の、前記指定されたC A Dデータを処理可能なC A Dデータ処理手段に対する処理要求を出力する処理要求出力手段と、

前記処理要求出力手段から出力された前記処理要求に対応する関数があらかじめ定義されており、前記関数に定義された処理を、前記指定されたC A Dデータを処理可能なC A Dデータ処理手段と連携して実行する連携手段と、

として機能させるためのプログラム。

【 0 1 1 0 】

【 発明の効果 】

以上説明したように本発明では、処理要求出力手段から出力された処理要求に対応する関数を実行することで、連携手段がC A Dデータ処理手段に依存しない形式の処理要求をC A Dデータ処理手段が解釈可能な命令に変換するようにしたため、処理要求出力手段が出力する処理要求に応じた関数を連携手段に定義しておくだけで、C A Dデータの種別に応じたC A Dデータ処理手段に処理を行わせることができる。そのため、任意の種別のC A Dデータの管理機能を容易に追加することができる。

10

【 図面の簡単な説明 】

【 図 1 】 本発明の原理構成図である。

【 図 2 】 本実施の形態のシステム構成例を示すブロック図である。

【 図 3 】 クライアントコンピュータの構成例を示す図である。

【 図 4 】 本実施の形態の機能ブロック図である。

20

【 図 5 】 R A Mのメモリマップの例を示す図である。

【 図 6 】 インタフェーステーブルの一例を示す図である。

【 図 7 】 関数テーブルの一例を示す図である。

【 図 8 】 ロード処理手順を示すフローチャートである。

【 図 9 】 アセンブリの構成を表示する画面例を示す図である。

【 図 1 0 】 C A Dデータをロードする際の表示画面例を示す図である。

【 図 1 1 】 インタフェース呼び出し処理手順を示すフローチャートである。

【 図 1 2 】 C A Dインタフェースモジュール問い合わせの処理手順を示すフローチャートである。

【 図 1 3 】 対応コマンド一覧の例を示す図である。

30

【 図 1 4 】 コマンドメニュー表示処理のフローチャートである。

【 符号の説明 】

1 a , 1 b , 1 c C A Dデータ処理手段

2 a , 2 b , 2 c インタフェース (I / F) モジュール

3 ロード手段

4 ポインタ登録手段

5 ポインタテーブル

6 処理要求出力手段

7 連携手段

1 0 , 1 0 a クライアントコンピュータ

40

1 1 C P U

1 2 R A M

1 3 H D D

1 4 グラフィック処理装置

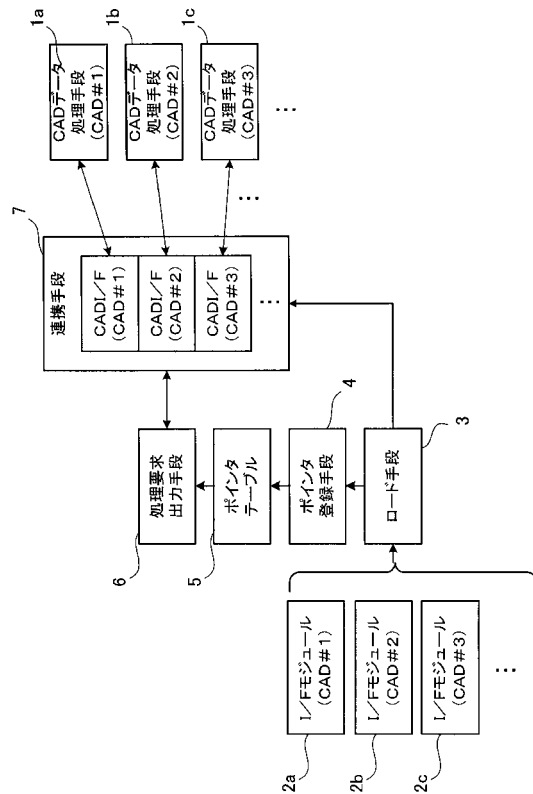
1 5 入力インタフェース

1 6 通信インタフェース

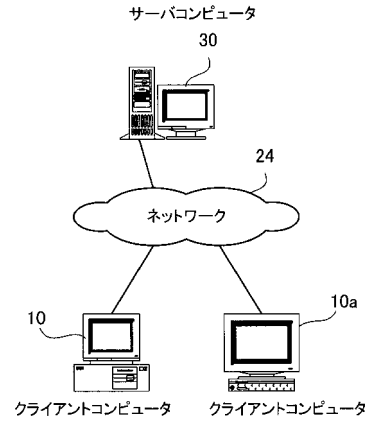
2 4 ネットワーク

3 0 サーバコンピュータ

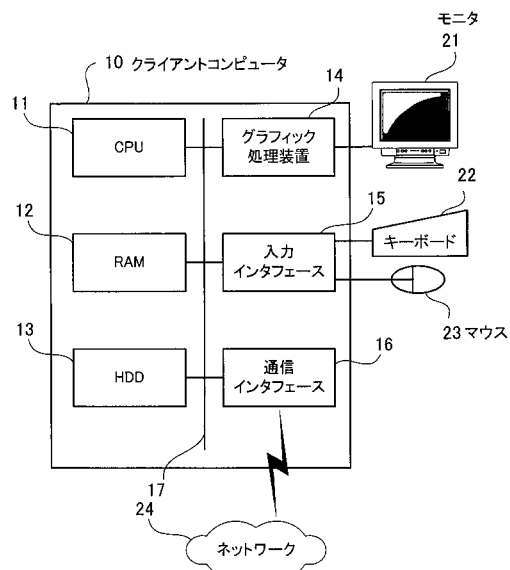
【図 1】



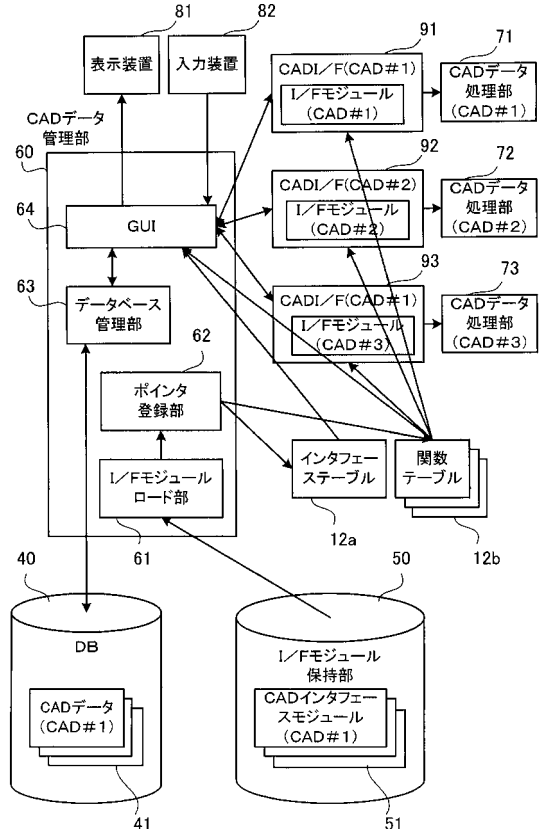
【図 2】



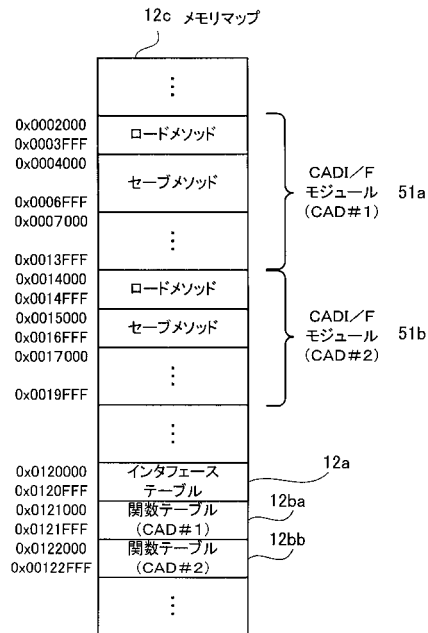
【図 3】



【図 4】



【図5】

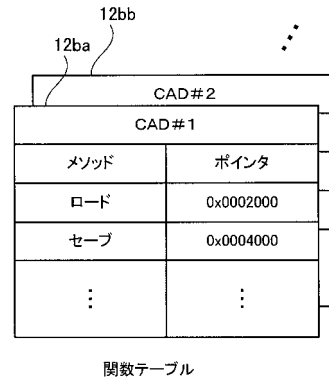


【図6】

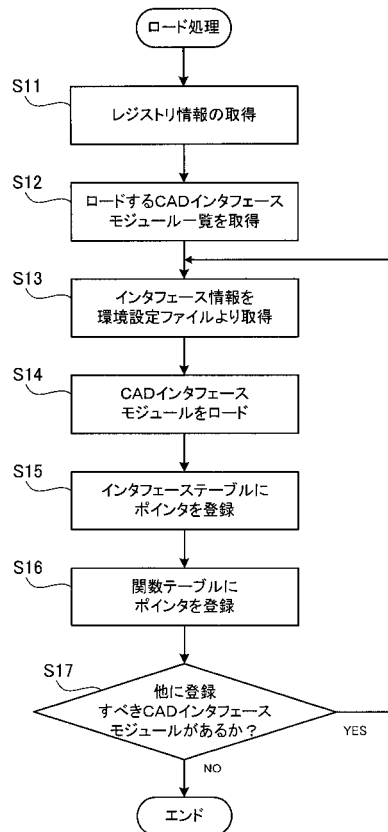
12a インタフェーステーブル

I/F種別	ポインタ
CAD #1	0x0121000
CAD #2	0x0122000
...	...

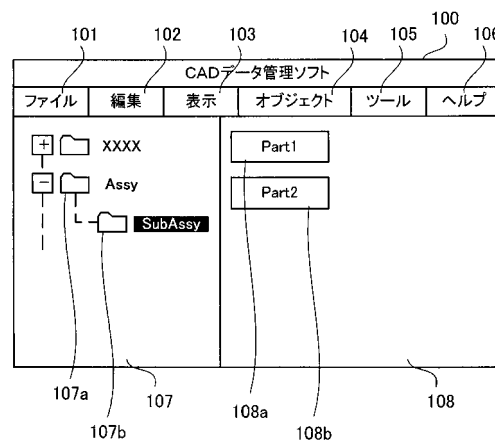
【図7】



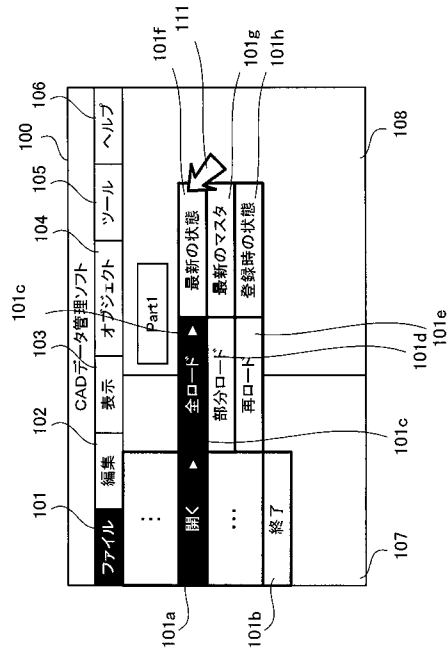
【図8】



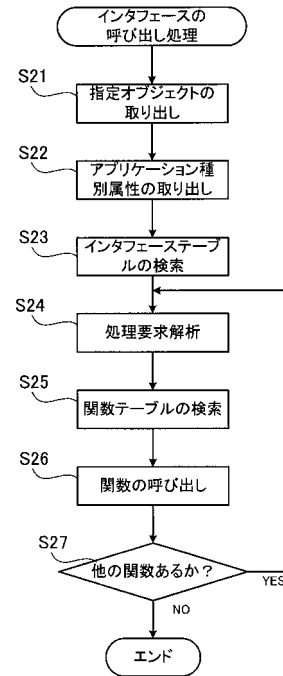
【図9】



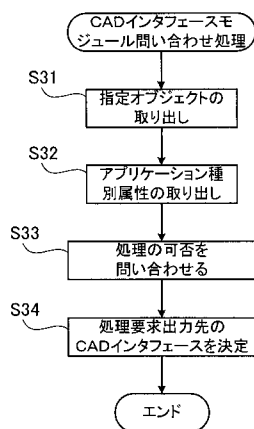
【図10】



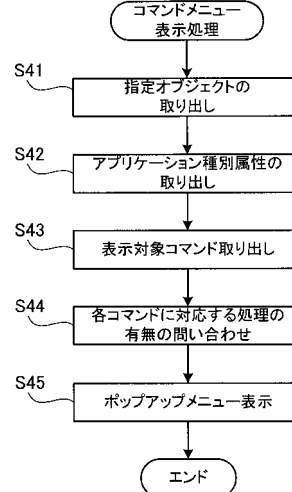
【図11】



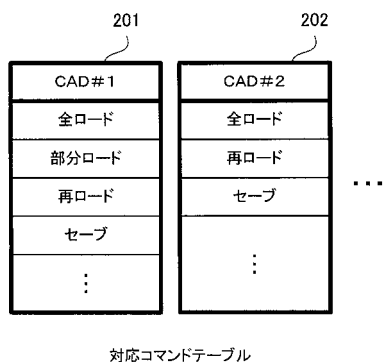
【図12】



【図14】



【図13】



フロントページの続き

(58)調査した分野(Int.Cl. , D B 名)

G06F 17/50

G06F 9/445

G06F 9/54