

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5586076号
(P5586076)

(45) 発行日 平成26年9月10日(2014.9.10)

(24) 登録日 平成26年8月1日(2014.8.1)

(51) Int.Cl.		F I			
G 0 6 F	9/44	(2006.01)	G 0 6 F	9/06	6 2 0 K
G 0 6 F	9/45	(2006.01)	G 0 6 F	9/44	3 2 2 A

請求項の数 4 (全 16 頁)

(21) 出願番号	特願2008-21228 (P2008-21228)	(73) 特許権者	000001443
(22) 出願日	平成20年1月31日 (2008.1.31)		カシオ計算機株式会社
(65) 公開番号	特開2009-181437 (P2009-181437A)		東京都渋谷区本町1丁目6番2号
(43) 公開日	平成21年8月13日 (2009.8.13)	(72) 発明者	今田 雅幸
審査請求日	平成22年10月29日 (2010.10.29)		東京都八王子市石川町2951番地5 カ
審判番号	不服2013-6614 (P2013-6614/J1)		シオ計算機株式会社 八王子技術センター
審判請求日	平成25年4月11日 (2013.4.11)		内
		合議体	
		審判長	仲間 晃
		審判官	山崎 達也
		審判官	小林 大介
		(56) 参考文献	特開2007-034807 (JP, A)
			)
			最終頁に続く

(54) 【発明の名称】 データ処理装置及びプログラム

(57) 【特許請求の範囲】

【請求項1】

ソースコードをコンパイルしてオブジェクトコードに変換し、各種業務項目を処理する業務アプリケーションプログラムを作成するコンパイル手段と、

前記コンパイル手段のコンパイル時に、前記ソースコードに含まれる関数が、当該ソースコード上における特定の業務項目の有無をチェックして当該特定の業務項目への置き換えを行うシステム関数であるか否かを判別する関数判別手段と、

前記コンパイル手段のコンパイル時に、前記関数判別手段による判別結果が前記システム関数である場合、当該システム関数の引数を解析する解析手段と、

前記コンパイル手段のコンパイル時に、前記解析手段により解析された引数に含まれる前記特定の業務項目がソースコードに含まれるか否かを判別する判別手段と、

前記コンパイル手段のコンパイル時に、前記判別手段の判別結果に基づいて、前記ソースコードに含まれるシステム関数を前記解析手段により解析された引数に置き換える置換手段と、

を備えるデータ処理装置。

【請求項2】

前記解析手段は、前記特定の業務項目を含む第1の引数、及び、当該特定の業務項目がソースコードに含まれない場合に置き換えるべき値を含む第2の引数を解析し、

前記置換手段は、前記判別手段の判別結果が前記特定の業務項目を含む場合、前記ソースコードに含まれるシステム関数を前記第1の引数に置き換える請求項1に記載のデータ

10

20

処理装置。

【請求項 3】

前記解析手段は、前記特定の業務項目を含む第 1 の引数、及び、当該特定の業務項目がソースコードに含まれない場合に置き換えるべき値を含む第 2 の引数を解析し、

前記置換手段は、前記判別手段の判別結果が前記特定の業務項目を含まない場合、前記ソースコードに含まれるシステム関数を前記第 2 の引数に置き換える請求項 1 に記載のデータ処理装置。

【請求項 4】

コンピュータを、

ソースコードをコンパイルしてオブジェクトコードに変換し、各種業務項目を処理する業務アプリケーションプログラムを作成するコンパイル手段、

前記コンパイル手段のコンパイル時に、前記ソースコードに含まれる関数が、当該ソースコード上における特定の業務項目の有無をチェックして当該特定の業務項目への置き換えを行うシステム関数であるか否かを判別する関数判別手段、

前記コンパイル手段のコンパイル時に、前記関数判別手段による判別結果が前記システム関数である場合、当該システム関数の引数を解析する解析手段、

前記コンパイル手段のコンパイル時に、前記解析手段により解析された引数に含まれる前記特定の業務項目がソースコードに含まれるか否かを判別する判別手段、

前記コンパイル手段のコンパイル時に、前記判別手段の判別結果に基づいて、前記ソースコードに含まれるシステム関数を前記解析手段により解析された引数に置き換える置換手段、

として機能させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、データ処理装置及びプログラムに関する。

【背景技術】

【0002】

従来から、WS (Work Station) や PC (Personal Computer) 等の業務処理を行うデータ処理装置では、汎用的なプログラム言語を用いて作成されたアプリケーションプログラムにより、上記業務処理を行っている。具体的には、プログラム言語を用いて処理を記述したソースコードを CPU (Central Processing Unit) が実行可能なオブジェクトコードに変換するコンパイル処理を行うことで、業務処理に関するアプリケーションプログラムを作成する。データ処理装置は、上記コンパイル処理により作成されたアプリケーションプログラムと CPU との協働により業務処理を実行する。

【0003】

データ処理装置において、アプリケーションプログラムを作成する有用な技術としては、特許文献 1～3 が知られている。この特許文献 1～3 には、ソースコードをコンパイルしてアプリケーションプログラムを生成するアプリケーション生成装置、コンパイル装置等が開示されている。

【特許文献 1】特開 2007-226358 号公報

【特許文献 2】特開平 10-133883 号公報

【特許文献 3】特開平 11-212802 号公報

【発明の開示】

【発明が解決しようとする課題】

【0004】

特に業務処理に関するアプリケーションプログラムの作成では、業務分野のデータ処理をモデル化し、各種業務処理ごとにプログラムを部品化して再利用する開発手法が知られている。例えば、伝票作成に関するアプリケーションプログラムを作成する場合は、伝票に記載する売上、数量などの項目に関するプログラム部品や、参照すべきデータベースへ

のアクセスに関するプログラム部品を組合せることでソースコードを作成し、コンパイルして、業務処理に関するアプリケーションプログラムを作成している。

【0005】

また、業務処理における項目処理については、他の項目の値を使用して処理を行うものがある。このため、項目に関するプログラム部品を組み合わせるアプリケーションプログラムを作成する際には、各項目の存在をチェックするシステム関数を組み込む必要があった。

【0006】

しかしながら、上記システム関数を組み込んだアプリケーションプログラムについては、実行時において各項目の存在をチェックする処理が行われるため、本来行うべき業務処理の処理速度に影響が生じる虞があった。

10

【0007】

本発明は、プログラム部品を組み合わせる業務処理に関するアプリケーションプログラムを作成する際に、本来行うべき業務処理の処理速度への影響を低減させる技術を提供することである。

【課題を解決するための手段】

【0008】

請求項1、4に記載の発明は、ソースコードをコンパイルしてオブジェクトコードに変換し、各種業務項目を処理する業務アプリケーションプログラムを作成するコンパイル手段と、

20

前記コンパイル手段のコンパイル時に、前記ソースコードに含まれる関数が、当該ソースコード上における特定の業務項目の有無をチェックして当該特定の業務項目への置き換えを行うシステム関数であるか否かを判別する関数判別手段と、

前記コンパイル手段のコンパイル時に、前記関数判別手段による判別結果が前記システム関数である場合、当該システム関数の引数を解析する解析手段と、

前記コンパイル手段のコンパイル時に、前記解析手段により解析された引数に含まれる前記特定の業務項目がソースコードに含まれるか否かを判別する判別手段と、

前記コンパイル手段のコンパイル時に、前記判別手段の判別結果に基づいて、前記ソースコードに含まれるシステム関数を前記解析手段により解析された引数に置き換える置換手段と、

30

を備える。

【0009】

請求項2に記載の発明は、請求項1に記載の発明において、前記解析手段は、前記特定の業務項目を含む第1の引数、及び、当該特定の業務項目がソースコードに含まれない場合に置き換えるべき値を含む第2の引数を解析し、

前記置換手段は、前記判別手段の判別結果が前記特定の業務項目を含む場合、前記ソースコードに含まれるシステム関数を前記第1の引数に置き換える。

【0010】

請求項3に記載の発明は、請求項1に記載の発明において、前記解析手段は、前記特定の業務項目を含む第1の引数、及び、当該特定の業務項目がソースコードに含まれない場合に置き換えるべき値を含む第2の引数を解析し、

40

前記置換手段は、前記判別手段の判別結果が前記特定の業務項目を含まない場合、前記ソースコードに含まれるシステム関数を前記第2の引数に置き換える。

【発明の効果】

【0011】

本発明によれば、プログラム部品を組み合わせる業務処理に関するアプリケーションプログラムを作成する際に、本来行うべき業務処理の処理速度への影響を低減させる。

【発明を実施するための最良の形態】

【0012】

以下、本発明の実施の形態について図を参照して説明するが、この発明の範囲は以下の

50

実施の形態に限定されない。

【0013】

先ず、本実施の形態に係るデータ処理装置の構成について、図1を参照して説明する。図1は、データ処理装置1の機能的構成を模式的に示すブロック図である。

【0014】

図1に示すように、データ処理装置1は、制御装置11、記憶装置12、表示装置13、入力装置14、通信装置15、印刷装置16を有する。上記データ処理装置1の各部はバス17により電氣的に接続されている。データ処理装置1は、WSやPC等の情報機器などであってよい。

【0015】

制御装置11は、CPU、RAM(Random Access Memory)、ROM(Read Only Memory)などを有しており(いずれも図示しない)、データ処理装置1の動作を中央制御する。具体的には、制御装置11は、RAMの作業領域に展開されたROMや記憶装置12に記憶されたプログラムデータとCPUとの協働により各部を統括制御する。

【0016】

記憶装置12は、プログラムデータや各種設定データ等のデータを制御装置11から読み書き可能に記憶する。例えば、記憶装置12は、HDD(Hard Disk Drive)、半導体メモリであってよい。

【0017】

上記記憶装置12が記憶するデータとしては、業務プログラムファイル121、業務スケルトンテンプレート122、項目部品123、印刷部品124、DB処理部品125、業務処理部品126、設定ファイル127、各種参照ファイル128及び業務アプリケーション生成プログラム129などがある。

【0018】

業務プログラムファイル121は、業務処理に関するアプリケーションプログラムである。業務プログラムファイル121は、制御装置11のCPUとの協働により、データ処理装置1における業務アプリケーションの実行処理を実行させる。具体的には、業務プログラムファイル121は、制御装置11においてCPUが実行可能なオブジェクトコードであり、コンパイル処理によりソースコードから生成されたデータである。

【0019】

業務スケルトンテンプレート122は、売上伝票入力処理、請求書処理、顧客情報管理処理などの事務処理業務における処理の雛形をソースコードで定義したファイルである。業務スケルトンテンプレート122は、コンパイル処理により業務プログラムファイル121を生成する際の元となるソースコードである。すなわち、業務プログラムファイル121は、処理の雛形である業務スケルトンテンプレート122に業務処理の詳細を部品化したソースコードである項目部品123、印刷部品124、DB処理部品125、業務処理部品126を挿入したソースコードをコンパイルして生成される。なお、業務スケルトンテンプレート122は、上述した各種処理ごとにソースコードが分割された、各種処理ごとのデータファイルであってよい。

【0020】

項目部品123は、業務スケルトンテンプレート122における各処理の雛形において、各項目における入力や表示を行うための定義や処理に関するソースコードを含むデータファイルである。ここにおける項目とは、例えば販売管理に関する売上伝票や受注伝票をユニークに扱う伝票番号、取引先や得意先を示す得意先コード、商品を示す商品コード、当該商品コードに対応する商品名などである。項目部品123は、上述した各項目ごとに定義や処理に関するソースコードが分割された、各項目ごとのデータファイルであってよい。

【0021】

印刷部品124は、業務スケルトンテンプレート122における各処理の雛形において、伝票や帳票などの印刷を行うための定義情報、書式情報、印字制御などに関するソース

10

20

30

40

50

コードを含むデータファイルである。なお、印刷部品 1 2 4 については、印刷する項目単位に定義や処理に関するソースコードが分割された、各項目ごとのデータファイルであってよい。

【0022】

D B 処理部品 1 2 5 は、業務処理における顧客情報、商品情報、売買記録などの業務情報を格納するデータベース D B に対し、データの更新、登録又は抽出など、当該データベース D B への整合性の取れたデータアクセスを行う処理内容や定義情報に関するソースコードを含むデータファイルである。

【0023】

業務処理部品 1 2 6 は、上述した部品で呼び出させて使用されるサブルーチン的な処理に関するソースコードを含むデータファイルである。業務処理部品 1 2 6 におけるサブルーチン的な処理の例としては、入力可能な項目をリスト表示し、選択することでデータ入力の補助を行う処理、伝票入力などのトランザクション処理の途中で新たにデータベースに対してレコード追加する処理、各部品で共通な処理又は問い合わせ・確認のダイアログを表示する処理などがある。

【0024】

設定ファイル 1 2 7 は、アプリケーションプログラムを作成する際に必要な補助的な情報を管理するためのファイルである。具体的には、設定ファイル 1 2 7 には、その業務アプリケーションプログラムの実行時に参照する各種参照ファイル 1 2 8 を、その実体が格納されたフォルダから実行時に参照可能なフォルダへコピーするためのデータなどが格納されている。

【0025】

例えば、設定ファイル 1 2 7 は、ソースコード上で利用されるシステム関数の名目や、当該システム関数と、ライブラリデータとの対応関係などを示すシステム関数テーブルなどであってよい。データ処理装置 1 では、このシステム関数テーブルを参照することで、コンパイル時におけるシステム関数のチェックを行うことができる。また、データ処理装置 1 では、システム関数テーブルを参照することで、ソースコードをコンパイルする際に、ソースコードに含まれるシステム関数に対応するライブラリデータを、アプリケーションプログラムの実行時に参照可能なフォルダへコピーすることが可能となる。

【0026】

各種参照ファイル 1 2 8 は、上述した業務スケルトンテンプレート 1 2 2 やプログラム部品で組み合わされた業務処理において、画面表示や演算等の処理で参照されるデータファイルである。具体的には、C P U が直接読み出して処理が可能なライブラリデータ（オブジェクトファイル）であり、ソースコードのシステム関数として予め用意されたライブラリデータ、業務処理時の処理に関する実行モジュール、J P E G (Joint Photographic Experts Group) 等の業務処理時の表示画面に関するイメージデータ、制御内容や定義内容を含むテキストデータなどがある。

【0027】

業務アプリケーション生成プログラム 1 2 9 は、ソースコードの作成や、ソースコードのコンパイル等の処理に関するアプリケーションプログラムである。業務アプリケーション生成プログラム 1 2 9 は、制御装置 1 1 の C P U との協働により、データ処理装置 1 におけるソースコード作成処理を実行させる。また、業務アプリケーション生成プログラム 1 2 9 は、制御装置 1 1 の C P U との協働により、データ処理装置 1 におけるコンパイル処理を実行させる。

【0028】

表示装置 1 3 は、制御装置 1 1 から出力された表示制御信号に基づいた画像を表示画面に表示する。表示装置 1 3 は、L C D (Liquid Crystal Display) や C R T (Cathode Ray Tube) などのディスプレイであってよい。

【0029】

入力装置 1 4 は、ユーザからの操作入力を受け付け、当該操作入力に応じた操作信号を

10

20

30

40

50

制御装置 11 へ出力する。例えば、通信装置 15 は、文字入力キー、数字入力キー、その他各種機能に対応付けられたキーを備えたキーボード、マウス等のポインティングデバイスであってよい。

【0030】

通信装置 15 は、制御装置 11 の制御の下、所定の通信プロトコルを用いて、互いに通信可能に接続する他の機器との間でデータ通信を行う。具体的には、通信装置 15 は、無線通信回路及びアンテナや、有線で通信を行うための通信インターフェースを備えた通信回路であり、無線／有線 LAN (Local Area Network)、インターネットなどの通信ネットワークと接続し、当該通信ネットワークと接続する他の情報機器との間でデータ通信を行う。データ処理装置 1 は、この通信装置 15 を介したデータ通信により、業務データ

10

【0031】

印刷装置 16 は、制御装置 11 の制御の下、用紙などに画像形成を行う。具体的には、印刷装置 16 は、電子写真方式やインクジェット方式などのプリンタなどである。

【0032】

次に、データ処理装置 1 における業務アプリケーションの実行についての概要を図 2、図 3 を参照して説明する。図 2 は、業務処理に関するソースコードの作成から業務実行までの処理を示す概念図である。図 3 は、データ処理装置 1 における業務アプリケーション処理の概要を示す概念図である。

【0033】

20

図 2 に示すように、データ処理装置 1 では、業務スケルトンテンプレート 122 に項目部品 123、印刷部品 124、DB 処理部品 125、業務処理部品 126 を組み合わせてソースコードを作成するソースコード作成処理が行われ (ステップ S1)、その作成されたソースコードをコンパイルするコンパイル処理が行われることで (ステップ S2)、業務アプリケーションプログラムであるオブジェクトコード 121a が生成される。上記ステップ S1、S2 は、前述したとおり、制御装置 11 の CPU と業務アプリケーション生成プログラム 129 との協働により、実行される処理である。

【0034】

ステップ S2 により生成されるオブジェクトコード 121a は、制御装置 11 の CPU との協働により実行される業務アプリケーションプログラムである。この業務アプリケーションプログラムは、上述したステップ S2、S3 により、図 3 に示すように、業務処理における全体制御・画面制御・データ制御などの基本処理が定義された業務スケルトンテンプレート 122 に項目部品 123、印刷部品 124、DB 処理部品 125 及び業務処理部品 126 を組み合わせて作成されている。

30

【0035】

具体的には、制御装置 11 が実行する業務アプリケーションプログラムにおいて、表示装置 13 や入力装置 14 で表示又は入力されるデータ項目については、業務スケルトンテンプレート 122 と項目部品 123 とで定義される。また、データベース DB を SQL (Structured Query Language) など検索、更新制御する場合の動作は、業務スケルトンテンプレート 122 と DB 処理部品 125 とで定義される。さらに、データベース DB で検索した結果などのデータ出力を印刷装置 16 から印刷する場合の動作は印刷部品 124 で定義される。また、上述した各処理のサブルーチ的な処理については、業務処理部品 126 で定義される。例えば、業務処理部品 126 には、日付情報を業務スケルトンテンプレート 122 と項目部品 123 との定義に従って和暦で受け付けて、その和暦を元に西暦に変化する日付変換処理などが定義されている。

40

【0036】

次いで、データ処理装置 1 では、ステップ S2 により作成されたオブジェクトコード 121a を制御装置 11 の CPU が順次解析して実行することで、業務アプリケーションの実行処理が行われる (ステップ S3)。このステップ S3 では、オブジェクトコード 121a の元となるソースコードで定義されたシステム関数など、必要に応じてシステム用ラ

50

イブラリデータ128aが参照される。

【0037】

次に、ソースコード作成処理（ステップS1）の詳細について、図4～図7を参照して説明する。図4は、データ処理装置1が実行するソースコード作成処理を示すフローチャートである。図5は、ソースコード作成処理における編集画面を例示する概念図である。図6は、項目値をチェックするシステム関数を例示する概念図である。図7は、ソースコードの記載内容を例示する概念図である。

【0038】

図4に示すように、ソースコード作成処理が開始されると、入力装置14からの操作入力に基づいて、ソースコードを作成するアプリケーション名の設定が行われる（ステップS101）。

10

【0039】

次いで、入力装置14からの操作により選択された業務処理の内容に基づいて、業務スケルトンテンプレート122の選択が行われる（ステップS102）。具体的には、このステップS102では、売上伝票入力処理、請求書処理、顧客情報管理処理などの業務スケルトンテンプレート122として用意された各種業務処理の中から、ユーザが作成を所望する処理が選択され、その選択された業務内容に関するデータファイルが選択される。

【0040】

次いで、入力装置14からの操作入力に基づいて、ステップS102により選択された業務スケルトンのプロパティが設定される（ステップS103）。このステップS103で設定される業務スケルトンのプロパティとは、処理の定義などであってよい。例えば、プロパティによる設定としては、売上伝票入力処理における税率設定などがある。

20

【0041】

次いで、入力装置14からの操作により選択された参照すべきデータベース名などに基づいて、DB処理部品125の選択が行われる（ステップS104）。例えば、ステップS104では、得意先情報が格納された得意先データベース、担当者情報が格納された担当者データベース、売上情報が格納された売上データベースなどの中から、ユーザが利用したいデータベースが選択され、その選択されたデータベースの利用に必要なDB処理部品125が選択される。

【0042】

30

次いで、ステップS102により選択された業務スケルトンテンプレート122の定義内容に基づいて、表示装置13に実際の処理画面、印刷紙面を模した編集画面が表示され、項目や印刷レイアウトの編集を入力装置14からの操作入力に基づいて行う編集作業が開始される（ステップS105）。

【0043】

例えば、ステップS105において、ステップS102により選択された業務スケルトンテンプレート122が売上伝票入力である場合は、図5に示すように、売上伝票入力を行う実際の画面や、売上伝票入力時の印刷紙面と同じレイアウトの編集画面が表示装置13に表示される。

【0044】

40

この売上伝票入力に関する業務スケルトンテンプレート122には、入力項目としていくつかの項目テンプレートが予め用意されている。同様に、業務スケルトンテンプレート122には、印刷紙面や画面などの構成を示すいくつかの印刷テンプレートなど用意されている。編集開始時には、上記テンプレートの中から標準のテンプレートが使用される。例えば、図5の例では、標準の入力項目として、項目K1～K9が使用されている。

【0045】

次いで、編集作業が開始されると、入力装置14からの操作入力に基づいた編集作業内容の判定処理が行われる（ステップS106）。

【0046】

ステップS106において、入力装置14からの操作指示が項目の削除であると判定さ

50

れた場合は、入力装置 1 4 からの操作入力に基づいて、削除すべき項目テンプレートの選択が受け付けられ（ステップ S 1 0 7）、削除選択された項目テンプレートに基づいた項目や、その項目に関してコントロールを行うボタンなどが編集画面上から削除される（ステップ S 1 0 8）。

【0047】

ステップ S 1 0 6 において、入力装置 1 4 からの操作指示が項目処理の編集であると判定された場合は、入力装置 1 4 からの操作指示に基づいて、編集すべき項目テンプレートの選択が受け付けられ（ステップ S 1 0 9）、選択された項目テンプレートの内容を変更するテンプレート編集が行われる（ステップ S 1 1 0）。このテンプレート編集では、例えば、項目のプロパティの変更、編集画面上における項目の配置位置の変更などが行われる。

10

【0048】

ステップ S 1 0 6 において、入力装置 1 4 からの操作指示が印刷処理の編集であると判定された場合は、入力装置 1 4 からの操作指示に基づいて、編集すべき印刷テンプレートの選択が受け付けられ（ステップ S 1 1 1）、選択された印刷テンプレートの内容を変更する印刷レイアウト編集が行われる（ステップ S 1 1 2）。この印刷レイアウト編集では、例えば、印刷領域の大きさやサイズ変更、印刷位置の調整などが行われる。

【0049】

ステップ S 1 0 6 において、入力装置 1 4 からの操作指示が編集作業の終了であると判定された場合は、上述したステップ S 1 0 7～S 1 1 2 で変更した内容の定義ファイル（ソースコード）を保存するか否かが入力装置 1 4 からの保存実行指示の有無などに基づいて判定される（ステップ S 1 1 3）。

20

【0050】

ステップ S 1 1 3 の判定において、定義ファイルを保存すると判定された場合（YES）は、上述したステップ S 1 0 7～S 1 1 2 で変更した内容に基づいて、業務スケルトンテンプレート 1 2 2 に項目部品 1 2 3、印刷部品 1 2 4、業務処理部品 1 2 6 を組み込んだ定義ファイルが保存される（ステップ S 1 1 4）。

【0051】

具体的には、ステップ S 1 1 4 において、上述した編集作業において項目の削除が行われた場合は、当該削除された項目に関する項目部品 1 2 3 が組み込まれることなく、削除された項目の処理や定義を含まないソースコードが作成される。また、編集作業において項目処理の編集が行われた場合は、編集対象となった項目に関する項目部品 1 2 3 の内容が編集内容に基づいて修正される。同様に、編集作業において印刷処理の編集が行われた場合は、編集対象となった印刷テンプレートに関する印刷部品 1 2 4 の内容が編集内容に基づいて修正される。

30

【0052】

ソース生成処理により、項目の削除が行われた場合に生成されるソースコードでは、当該削除された項目に関する項目部品 1 2 3 が組み込まれることがなくなるため、削除された項目を利用する他の項目がある場合にエラーとなってしまう。そこで、業務スケルトンテンプレート 1 2 2、項目部品 1 2 3、印刷部品 1 2 4、DB 処理部品 1 2 5、業務処理部品 1 2 6 などにおいて、項目を利用する処理については、図 6 に例示した項目値の有無をチェックするシステム関数が記載されている。

40

図 6 に示すように、項目値をチェックするシステム関数である「@項目値取得」は、引数 1 で指定された項目名が画面上（ソースコード上）に存在しない場合、引数 2 で指定された所定値を戻す処理を行う関数である。従って、上記システム関数を利用することで、処理に必要な項目が削除された場合であっても、引数 2 の所定値が代入されるために、正常に処理を実行することが可能となる。

【0053】

具体的には、図 7 に示すように、業務スケルトンテンプレート 1 2 2、項目部品 1 2 3、印刷部品 1 2 4、DB 処理部品 1 2 5、業務処理部品 1 2 6 などにおける雛形のソース

50

コードでは、売上単価や売上金額の項目で利用される他の項目について「@項目値取得」の引数1で指定してチェックを行い、存在しない場合の値を引数2で指定している。例えば、売上単価の項目では、取引区分が商品割引である条件で利用される値引単価について、ソースコード上に存在しない場合は0を代入するように設定されている。

【0054】

次に、コンパイル処理（ステップS2）の詳細について、図8、図9を参照して説明する。図8は、データ処理装置1が実行するコンパイル処理を示すフローチャートである。図9は、項目値を取得するシステム関数の置き換えを例示する概念図である。

【0055】

図8に示すように、コンパイル処理が開始されると、コンパイル対象のソースコードが読み出され、当該ソースコード上の関数名が取得される（ステップS201）。 10

【0056】

次いで、前述したシステム関数テーブルが読み出され、ステップS201で取得されたソースコード上の関数名と比較されることで、ソースコード上の各関数のチェックが行われ（ステップS202）、項目値を取得するシステム関数（「@項目値取得」など）であるか否かが判定される（ステップS203）。 20

【0057】

ステップS203において、項目値を取得するシステム関数でないと判定された場合（NO）は、関数の引数が取得されて値を解析する解析処理が行われ（ステップS204）、解析した引数の情報がメモリ展開される（ステップS205）。 20

【0058】

次いで、関数の処理自体について、該当するライブラリデータを用いるライブラリ関数を参照するように設定される（ステップS206）。このステップS206では、ステップS202によりチェックされた関数をどのように処理するかを設定する。システム関数は、予め用意されたシステム関数用のライブラリデータを参照して処理を行っている。ステップS206では、どのライブラリデータを参照して関数の処理を行うかを設定する。

【0059】

次いで、引数の処理自体について、該当するライブラリデータを用いるライブラリ関数及び項目を参照するように設定される（ステップS207）。このステップS207では、引数内に設定している処理について、システム関数用のライブラリデータや項目値を参照して行うように設定される。 30

【0060】

従って、ステップS202でチェックされた関数について、項目値を取得するシステム関数でないと判定された場合（NO）では、既存のコンパイル処理が行われることとなる。

【0061】

ステップS203において、項目値を取得するシステム関数であると判定された場合（YES）は、当該システム関数の引数が解析され（ステップS208）、引数の値が取得される（ステップS209）。具体的には、前述した「@項目値取得」における引数1、引数2が取得される。 40

【0062】

次いで、ステップS209において取得された引数1に該当する項目がソースコード上に存在するか否かがチェックされ（ステップS210）、そのチェック結果に基づいて引数1に該当する項目の存在の有無が判定される（ステップS211）。

【0063】

ステップS211において、引数1に該当する項目がソースコード上に存在すると判定された場合（YES）は、本システム関数の処理が引数1で指定した項目に置き換えられる（ステップS212）。

【0064】

具体的には、図9に示すように、項目値を取得するシステム関数として、引数1として 50

値引単価、引数2として0が指定されており、ソースコード上に値引単価の項目が存在する場合は、引数1で指定した値引単価に置き換えられることとなる。

【0065】

また、ステップS211において、引数1に該当する項目がソースコード上に存在しないと判定された場合（NO）は、引数2で指定された所定値に置き換えられる（ステップS213）。

【0066】

具体的には、図9に示すように、項目値を取得するシステム関数として、引数1として値引単価、引数2として0が指定されており、ソースコード上に値引単価の項目が存在しない場合は、引数2で指定した値である0に置き換えられることとなる。

10

【0067】

次いで、関数内の引数内の情報として、ステップS212、S213で置き換えられた情報がメモリに展開され（ステップS214）、引数の処理自体について、該当するライブラリデータを用いるライブラリ関数及び項目を参照するように設定される（ステップS215）。

【0068】

従って、ステップS202でチェックされた関数について、項目値を取得するシステム関数であると判定された場合（YES）では、当該システム関数を実行した後にコンパイル処理が行われることとなる。

【0069】

20

以上のように、データ処理装置1は、ソースコードをコンパイルしてオブジェクトコードに変換し、各種業務項目を処理する業務アプリケーションプログラムを作成するコンパイル手段（制御装置11、ステップS2）と、コンパイル手段のコンパイル時に、ソースコードに含まれる関数が、当該ソースコード上における特定の業務項目の有無をチェックして当該特定の業務項目への置き換えを行うシステム関数であるか否かを判別する関数判別手段（制御装置11、ステップS203）、コンパイル手段のコンパイル時に、関数判別手段による判別結果が上記システム関数である場合、当該システム関数の引数を解析する解析手段（制御装置11、ステップS208）と、コンパイル手段のコンパイル時に、解析手段により解析された引数に含まれる特定の業務項目がソースコードに含まれるか否かを判別する判別手段（制御装置11、ステップS210、S211）と、コンパイル手段のコンパイル時に、判別手段の判別結果に基づいて、ソースコードに含まれるシステム関数を解析手段により解析された引数に置き換える置換手段（制御装置11、ステップS212、S213）と、を備える。

30

【0070】

従って、データ処理装置1は、ソースコード上において、特定の業務項目の有無をチェックして当該特定の業務項目への置き換えを行うシステム関数がある場合に、当該システム関数を実行した後にコンパイル手段によるコンパイルが行われることとなる。

【0071】

このため、データ処理装置1では、コンパイル処理により作成されたオブジェクトコードを実行する際において（ステップS3）、上記システム関数の処理を実行する必要がなくなるため、本来行うべき業務処理の処理速度への影響を低減させることができる。

40

【0072】

また、データ処理装置1では、図10に示すように、コンパイル処理により作成されたオブジェクトコードを実行する際に、上記システム関数の実行に必要なシステム用ライブラリデータ128aが不要となるため、ライブラリデータの読出時間を短縮することができる。従って、データ処理装置1では、コンパイル処理により作成されたオブジェクトコードの実行時における処理速度を向上させることができる。

【0073】

また、データ処理装置1の解析手段は、特定の業務項目を含む第1の引数、及び、当該特定の業務項目がソースコードに含まれない場合に置き換えるべき値を含む第2の引数を

50

解析する。また、データ処理装置 1 の置換手段は、判別手段の判別結果が特定の業務項目を含む場合、ソースコードに含まれるシステム関数を第 1 の引数に置き換える。また、データ処理装置 1 の置換手段は、判別手段の判別結果が特定の業務項目を含まない場合、ソースコードに含まれるシステム関数を第 2 の引数に置き換える。

【0074】

従って、データ処理装置 1 では、特定の業務項目を含む第 1 の引数、及び、当該特定の業務項目がソースコードに含まれない場合に置き換えるべき値を含む第 2 の引数により、ソースコード上における特定の業務項目の有無をチェックして当該特定の業務項目への置き換えを行うシステム関数について、対応することが可能となる。

【0075】

10

〔変形例〕

次に、業務処理に関するソースコードの作成から業務実行までの処理の変形例について、図 11、図 12 を参照して説明する。図 11 は、業務処理に関するソースコードの作成から業務実行までの処理の変形例を示す概念図である。図 12 は、実行モード設定時の表示画面例を示す概念図である。

【0076】

前述した実施の形態では、項目の存在をチェックするシステム関数の処理をコンパイル処理後のアプリケーションプログラム実行時に行うのではなく、オブジェクトコードを作成する時に行うようにしている。これに対し、変形例は、図 11 に示すように、前述した実施の形態の処理に実行モードを設定する処理（ステップ S 30）を追加したものである。

20

【0077】

ステップ S 30 における実行モードの設定では、前述したシステム関数「@項目値取得」の処理をどのように行うかについてが設定される。具体的には、ステップ S 30 における実行モードの設定は、図 12 に示すような画面が表示装置 13 に表示され、入力装置 14 からの操作入力に基づいて設定される。例えば、図 12 の画面例は、システム関数「@項目値取得」で項目が存在しない場合にエラーとするか否かの設定を行う画面である。

【0078】

ステップ S 30 において設定された情報は、各種参照ファイル 128 と同種の制御情報 128b として保持される。従って、アプリケーションプログラム実行時には、制御情報 128b も参照されることとなる。例えば、項目の非存在を許可しないようなアプリケーションプログラムを実行する場合、アプリケーションプログラムを修正することなく、ステップ S 30 における実行モードの設定で動作を切り替えて使用することができる。

30

【0079】

なお、上述した実施の形態における記述は、一例を示すものであり、これに限定するものではない。上述した実施の形態における構成及び動作に関しては、適宜変更が可能である。

【0080】

例えば、以上の説明では、本発明に係るプログラムのコンピュータ読み取り可能な媒体として ROM を使用した例を開示したが、この例に限定されない。その他のコンピュータ読み取り可能な媒体として、フラッシュメモリ等の不揮発性メモリ、CD-ROM 等の可搬型記録媒体を適用することが可能である。また、本発明に係るプログラムのデータを通信回線を介して提供する媒体として、キャリアウエーブ(搬送波)も本発明に適用される。

40

【図面の簡単な説明】

【0081】

【図 1】本実施の形態に係るデータ処理装置の機能的構成を模式的に示すブロック図である。

【図 2】業務処理に関するソースコードの作成から業務実行までの処理を示す概念図である。

【図 3】データ処理装置における業務アプリケーション処理の概要を示す概念図である。

50

【図４】データ処理装置が実行するソースコード作成処理を示すフローチャートである。

【図５】ソースコード作成処理における編集画面を例示する概念図である。

【図６】項目値をチェックするシステム関数を例示する概念図である。

【図７】ソースコードの記載内容を例示する概念図である。

【図８】データ処理装置が実行するコンパイル処理を示すフローチャートである。

【図９】項目値を取得するシステム関数の置き換えを例示する概念図である。

【図１０】コンパイル処理により作成されたオブジェクトコードの実行を例示する概念図である。

【図１１】業務処理に関するソースコードの作成から業務実行までの処理の変形例を示す概念図である。

10

【図１２】実行モード設定時の表示画面例を示す概念図である。

【符号の説明】

【００８２】

１ データ処理装置

１１ 制御装置

１２ 記憶装置

１３ 表示装置

１４ 入力装置

１５ 通信装置

１６ 印刷装置

20

１７ バス

１２１ 業務プログラムファイル

１２１ａ オブジェクトコード

１２２ 業務スケルトンテンプレート

１２３ 項目部品

１２４ 印刷部品

１２５ ＤＢ処理部品

１２６ 業務処理部品

１２７ 設定ファイル

１２７ａ システム関数テーブル

30

１２８ 各種参照ファイル

１２８ａ システム用ライブラリデータ

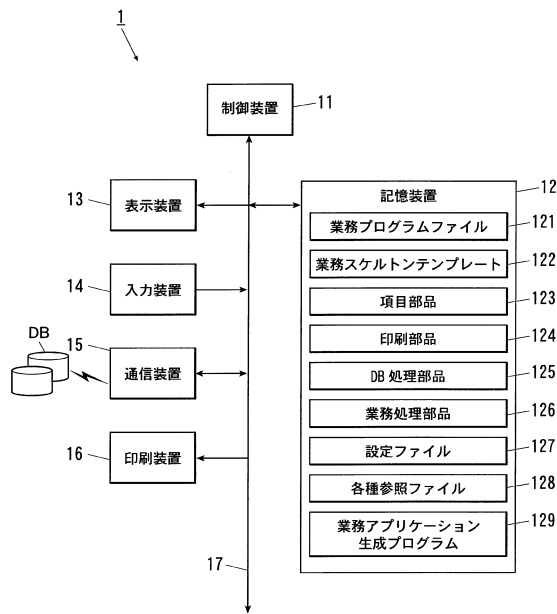
１２８ｂ 制御情報

１２９ 業務アプリケーション生成プログラム

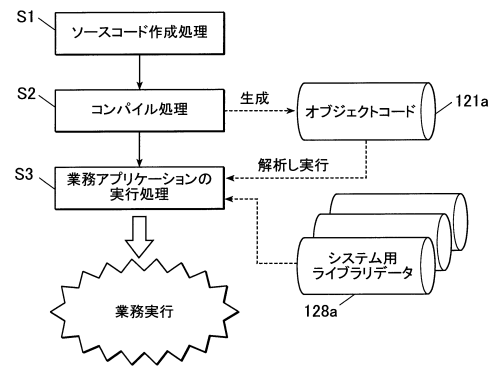
ＤＢ データベース

Ｋ１～Ｋ９ 項目

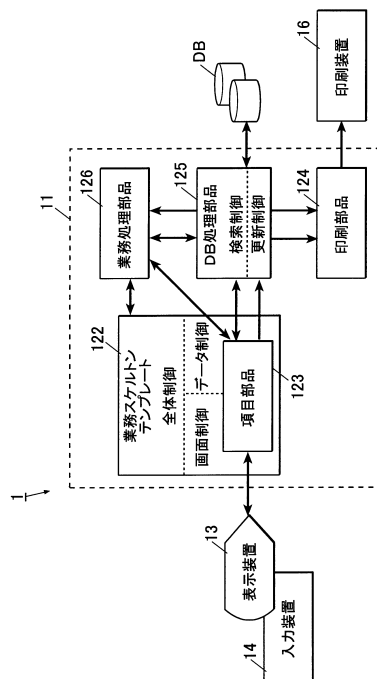
【図 1】



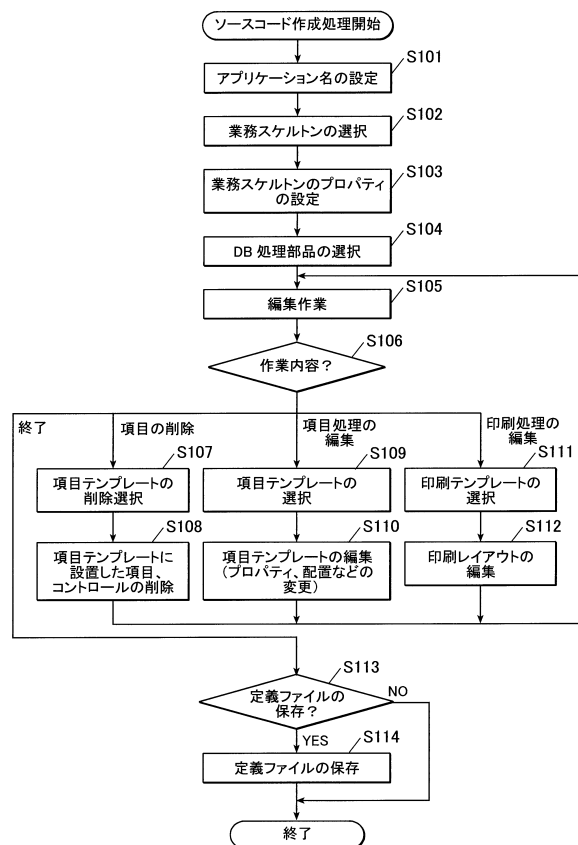
【図 2】



【図 3】



【図 4】



【図 5】

売上伝票入力

伝票番号 K1

得意先コード K2

得意先名 K3 K7 K8

商品コード	商品名	売上単価	数量	売上金額

K4 K5 K6

合計金額 K9

【図 7】

項目	内容	条件
売上単価	④項目値取得(値引単価, 0)	取引区分 = “商品割引”
	⑤項目値取得(見積入力時単価, 0)	処理区分 = “出荷から接続”
	⑥索引(商品MT, 商品コード = ⑥項目値取得(商品コード, “”), 売上単価)	
売上金額	④項目値取得(数量, 0) * ⑥項目値取得(売上単価, 0)	

【图 6】

＜項目値をチェックするシステム関数＞

関数名 : @項目値取得

引数1 :項目名

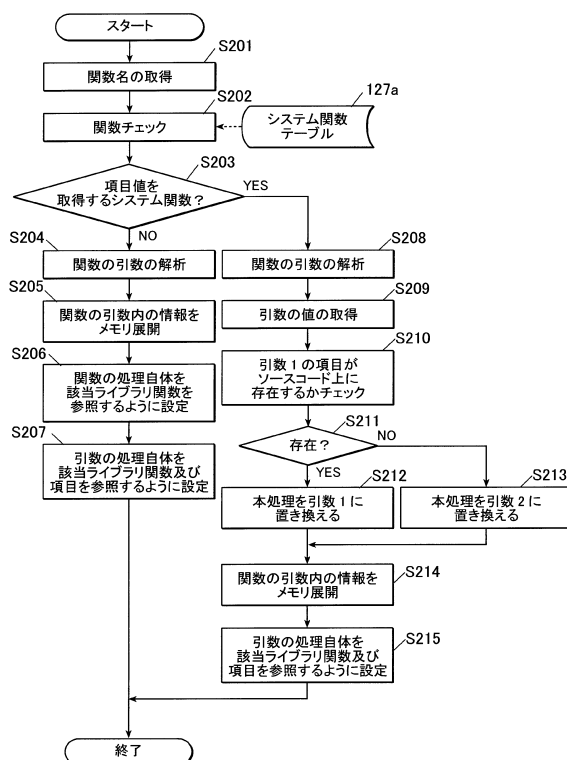
引数2 : 所定値

機能 : 引数1で指定された項目名が画面上(ソースコード上)に存在しない場合、引数2で指定された所定値を戻す。
引数1で指定された項目名が存在する場合、その項目の値を戻す。

使用例：@項目値取得(商品コード, “0000001”)

@項目値取得(数量, 0)

【図 8】



例) @項目値取得(値引単価, 0)

索引数1:值|索引单例
索引数2:0

【図 9】

項目	内容	条件
売上単価	= ④項目値取得(値引単価 0)	取引区分 = “商品割引”

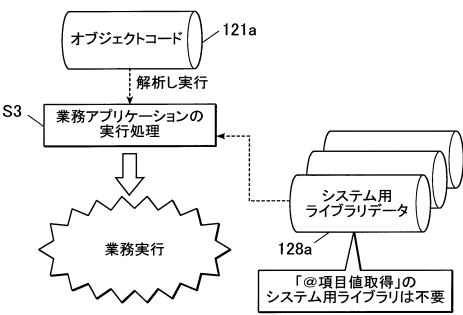
<「値引単価」が存在する場合>

売上単価	= 値引単価	取引区分 = “商品割引”
------	--------	---------------

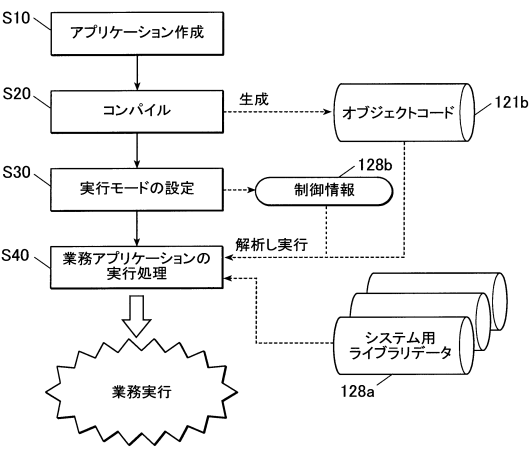
<「値引単価」が存在しない場合>

売上単価	= 0	取引区分 = “商品割引”
------	-----	---------------

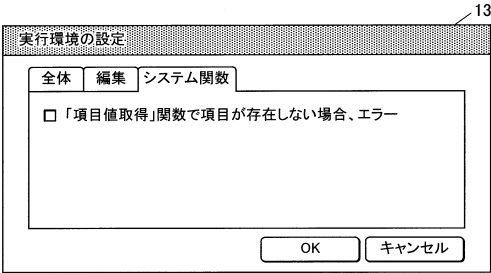
【図 1 0】



【図 1 1】



【図 1 2】



フロントページの続き

(58)調査した分野(Int.Cl., D B名)

G06F9/44, 9/45