(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4854058号
(P4854058)
(45)発行日　平成24年1月11日(2012.1.11)　(24)登録日　平成23年11月4日(2011.11.4)

(51)Int.Cl.　F I
G06F 11/30 (2006.01)　G06F 11/30 305D

請求項の数 4　(全 13 頁)

(21)出願番号　特願2004-104088 (P2004-104088)
(22)出願日　平成16年3月31日 (2004.3.31)
(65)公開番号　特開2005-292980 (P2005-292980A)
(43)公開日　平成17年10月20日 (2005.10.20)
審査請求日　平成19年3月28日 (2007.3.28)

(73)特許権者　393031586
株式会社国際電気通信基礎技術研究所
京都府相楽郡精華町光台二丁目２番地２
(74)代理人　100099933
弁理士　清水　敏
(72)発明者　葦苅　豊
京都府相楽郡精華町光台二丁目２番地２
株式会社国際電気通信基礎技術研究所内
(72)発明者　中村　哲
京都府相楽郡精華町光台二丁目２番地２
株式会社国際電気通信基礎技術研究所内

審査官　多胡　滋

最終頁に続く

(54)【発明の名称】汎用入力データ検査装置及びプログラム

(57)【特許請求の範囲】
【請求項１】
プログラムのオプションに対して入力されるデータの妥当性を検査するための汎用入力データ検査装置であって、当該汎用入力データ検査装置は、前記プログラムで使用される、あるオプション値に対するオプション制約情報を複数個記憶するための記憶手段とともに用いられ、
前記複数個の前記オプション制約情報は、いずれも前記あるオプション値に関するものであり、
前記汎用入力データ検証装置は、
前記記憶手段に記憶された前記複数個のオプション制約情報のうちからオプション制約情報の指定を受けると、指定されたオプション制約情報を前記記憶手段から読出して、前記指定されたオプション制約情報に記述された複数の制約情報をそれぞれ記憶するための複数の制約情報記憶手段と、
前記複数の制約情報記憶手段に対応して設けられ、前記あるオプション値が充足すべき所定の制約条件をそれぞれ規定している複数のプログラムルーチンをそれぞれ記憶するための複数の制約条件規定手段とを含み、
前記複数のプログラムルーチンの各々は、対応の前記制約情報記憶手段に記憶される前記制約情報とオプション値との比較をすることにより、前記所定の制約条件が当該オプション値により充足されているか否かを判定する機能を持ち、
前記汎用入力データ検査装置はさらに、入力される前記あるオプション値により、前記

複数のプログラムルーチンにより規定されている前記制約条件が全て充足されているか否かを、前記複数の制約条件規定手段に記憶されている前記複数のプログラムルーチンを実行することによって検査するための検査手段とを含む、汎用入力データ検査装置。

【請求項２】

前記汎用入力データ検査装置はさらに、前記複数の制約情報のうちの一つを新たな制約情報で更新すべき指定を受けたことに応答して、当該指定された制約情報を記憶した前記制約情報記憶手段の記憶内容を、前記新たな制約情報で更新するための制約情報更新手段を含む、請求項１に記載の汎用入力データ検査装置。

【請求項３】

前記検査手段は、入力される前記あるオプション値により、前記複数のプログラムルーチンにより規定されている前記複数の制約条件が全て充足されているか否かを検査し、前記複数の制約条件のうち、前記入力されるオプション値によって充足されないものがあれば、他の制約条件についての検査を省略し、データが妥当でないことを示す検査結果を出力するための手段を含む、請求項１又は請求項２のいずれかに記載の汎用入力データ検査装置。

【請求項４】

コンピュータにより実行されると、当該コンピュータを請求項１～請求項３のいずれかに記載の汎用入力データ検査装置として動作させる、コンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【０００１】

この発明は、コンピュータにおいてプログラムに引数を渡すための装置及び方法に関し、特に、オプション値の整合性のチェックを行なうためのプログラマの負担を減少させる装置及びコンピュータプログラムに関する。

【背景技術】

【０００２】

コンピュータにおいてプログラムを起動する際に、プログラムの動作の条件を変えることが頻繁に行なわれる。例えば、ある入力ファイルの内容を所定の方式で変換して他のファイルに出力する場合を考える。このとき、入力ファイルと出力ファイルとが固定されていると使いづらいので、プログラムの起動時に、入力ファイル名と出力ファイル名とを引数としてプログラムに渡すことがよく行なわれる。

【０００３】

従来、このようにプログラムにオプション値やパラメータ値（以下単に「オプション値」と呼ぶ。）等を渡す際には、所定のキーワードで引数を受取るように予めプログラムを設計しておき、起動時にいわゆるコマンド入力画面で、このプログラムを指定する文字列と、キーワード及び対応する引数の値とを続けてタイプする。コンピュータのオペレーティング・システム（ＯＳ）又はその上で動作するシェルと呼ばれるプログラムにより、この引数が、プログラム中で指定されたキーワードに対応する変数中に設定され、プログラムがその値に基づいて起動される。

【０００４】

一方、最近のコンピュータのオペレーションの主流は、ＧＵＩ（グラフィカル・ユーザ・インタフェース）を用いたものになっている。プログラムのオプション値をＧＵＩにより設定する場合、当該プログラムの中で、ＧＵＩを設計し実装する必要がある。そしてＧＵＩにより設定されたオプション値をプログラム中で使用する。

【０００５】

このようにプログラムに与えられるオプション値については、その値が妥当なものか否かをチェックする必要がある。例えば数値を設定すべきオプション値に数値以外の値を代入すれば、その結果は予測できないものになる。したがってコマンドベースにしろ、ＧＵＩベースにしろ、入力されたオプション値の整合性をプログラム中で必ずチェックする必要性がある。

【発明の開示】
【発明が解決しようとする課題】
【０００６】
こうしたオプション値をチェックする場合、通常はプログラム中にチェックのためのルーチンを記述する。しかしそうした場合には、そのルーチンを作成したり、ルーチン自体の正当性をチェックしたりするために時間がかかるという問題がある。また、個々のプログラマがルーチンを作成するので、その品質は個々のプログラマの技量に大きく依存する。
【０００７】
そうした問題を解決するための一つの方策として、オプション値ごとに妥当性チェックのための専用のプログラムを熟練したプログラマが書き、各アプリケーションでは、そのオプション値のチェックをする必要が生じるたびにそれを呼出す形で処理する、という方法がある。このようにすると、プログラマの技量により品質が左右されることはあまりなく、またプログラムごとにルーチンを書く必要もなくなる。
【０００８】
しかしそれでも、オプション値ごとに専用のプログラムを書かねばならず、また新しいプログラムを作成するたびに、かならずその正当性をチェックしなければならないという問題がある。そのため、システム全体の品質と生産性を高めることが困難であった。
【０００９】
それゆえに本発明の目的は、オプション値のチェックを行なう際に、システム全体の品質と生産性とを高めることが可能なプログラム開発環境を提供することである。
【００１０】
本発明の他の目的は、オプション値ごとに専用のプログラムを書くことなくオプション値のチェックをできるようにし、システム全体の品質と生産性とを高めることが可能なプログラム開発環境を提供することである。
【００１１】
本発明の他の目的は、複数種類のオプションに対して共通のプログラムを用いてオプション値のチェックをできるようにし、システム全体の品質と生産性とを高めることが可能なプログラム開発環境を提供することである。
【課題を解決するための手段】
【００１２】
本発明に係る汎用入力データ検査装置は、プログラムのオプションに対して入力されるデータの妥当性を検査するための汎用入力データ検査装置である。当該汎用入力データ検査装置は、プログラムで使用されるオプション値に対するオプション制約情報を記憶するための記憶手段とともに用いられる。汎用入力データ検証装置は、オプション制約情報の指定を受けると、指定されたオプション制約情報を記憶手段から読出して、指定されたオプション制約情報に記述された複数の制約情報をそれぞれ記憶するための複数の制約情報記憶手段と、複数の制約情報記憶手段に対応して設けられ、オプション値が充足すべき所定の制約条件を、対応の制約情報記憶手段に記憶される値との関係で規定するための複数の制約条件規定手段と、入力されるオプション値により、複数の制約規定手段により規定される制約条件が全て充足されているか否かを検査するための検査手段とを含む。
【００１３】
オプション制約情報の指定を受けると、当該オプション制約情報を制約情報記憶手段に記憶する。これら制約情報記憶手段に記憶された値との関係で制約条件規定手段により制約条件が規定される。検査手段により、制約条件規定手段の規定する制約条件を、入力されるオプション値が充足するか否かを判定する。オプション制約情報を変えると、制約条件も制約情報に伴い自動的に変わる。そのため、一つの汎用入力データ検査装置で、複数種類のオプション値に関する入力データの妥当性を検査することができる。
【００１４】
好ましくは、汎用入力データ検査装置はさらに、複数の制約情報のうちの一つを新たな

制約情報で更新すべき指定を受けたことに応答して、当該指定された制約情報を記憶した制約情報記憶手段の記憶内容を、新たな制約情報で更新するための制約情報更新手段を含む。

【００１５】

制約情報を、オプション制約情報で指定されたものから別の新たなものに変更することができる。その結果、プログラムの実行中でも動的にオプション値の許容範囲を変更することができる。

【００１６】

さらに好ましくは、検査手段は、複数の制約条件のうち、入力されるオプション値によって充足されないものがあれば、他の制約条件についての検査を省略し、データが妥当でないことを示す検査結果を出力するための手段を含む。

【００１７】

充足されない制約条件があれば残りの検査を省略するので、処理速度を早くできる。

【００１８】

本発明の第２の局面に係るコンピュータプログラムは、コンピュータにより実行されると、当該コンピュータを上記したいずれかの汎用入力データ検査装置として動作させる。

【発明を実施するための最良の形態】

【００１９】

本発明は、複数種類のオプション値について共通に使用できるオプション値チェックのための装置として、コンピュータ上で動作するオプション値チェックライブラリを提供する。このオプション値チェックライブラリは、オプション値チェックのためのオブジェクトクラスからなる。このオブジェクトクラスは「オプション」というクラス名を持つ。

【００２０】

また、以下に説明するシステムの各プログラムは複数個のモジュール（これらの各々もまたプログラムである。）の組合せにより構成される。各モジュールに対応して、それぞれが使用するオプション及びその値に関する情報をもつモジュール定義ファイルが予め準備される。各プログラムで、モジュールごとにオプションクラスのオブジェクト（以下「オプションクラスオブジェクト」と呼ぶ。）を生成し、対応するモジュール定義ファイルのフルパスを指定してオブジェクトを初期化することにより、各オプションクラスオブジェクトは、指定されたモジュールで使用されるオプション値をチェックすることができるように初期化される。

【００２１】

図１に本発明の一実施の形態に係るオプション値チェックライブラリを採用した応用システム２０のブロック図を示す。図１に示す応用システム２０は、一つのコンピュータハードウェアにより実現してもよいし、互いにネットワーク接続された複数のコンピュータハードウェアにより実現してもよい。

【００２２】

図１を参照して、応用システム２０は、モニタ２２と、キーボード及びマウス等のポインティングデバイスを含む入力装置２４と、システム２０の機能の中核を実現するためのアプリケーション２６Ａ～２６Ｃ等（以下、これらを代表して「アプリケーション２６」と呼ぶ。）と、アプリケーション２６等から共通に使用される複数のオプションクラスオブジェクト２８Ａ～２８Ｍ（これらを代表して、オプションクラスオブジェクト２８と呼ぶ。）と、オプションクラスオブジェクト２８がオプション値チェックのために参照するオプション情報４２を含むモジュール定義情報ファイル４０Ａ～４０Ｍ（これらを代表してモジュール定義情報ファイル４０と呼ぶ。）とを含む。

【００２３】

応用システム２０はさらに、アプリケーション２６を起動する際のオプション値を所定の記法にしたがって指定するように予め作成されたオプションファイル３０を含む。アプリケーション２６を起動する際に、コマンドベースの起動パラメータでこのオプションファイル３０を指定することにより、オプションファイル３０中のオプション値の妥当性が

オプションクラスオブジェクト２８によりチェックされる。オプションファイル３０の構成については図４を参照して後述する。

【００２４】

オプションクラスオブジェクト２８は、いずれも前述したオプションクラスのインスタンスであるが、それぞれモジュールごとに、使用されるオプションの値をチェックするために生成されたものである。

【００２５】

モジュール定義情報ファイル４０は、このシステムのプログラムで使用されるモジュールに関する情報を記述したファイルである。モジュール定義情報ファイル４０は、その中に当該モジュールで使用するオプションに関する情報を記述したオプション／パラメータエリア６２を含む。モジュール定義情報ファイル４０の詳細については図２及び図３を参照して後述する。

【００２６】

アプリケーション２６は、使用するモジュールの数だけオプションクラスオブジェクト２８を生成する。さらに各オプションクラスオブジェクト２８に対応するモジュール定義情報ファイル４０のフルパスをそれぞれのオプションクラスオブジェクト２８に与える。オプションクラスオブジェクト２８は、与えられたフルパスにより指定されるモジュール定義情報ファイル４０からオプション／パラメータエリア６２を読み込み、自己を初期化する。初期化が終了すると、アプリケーション２６が入力装置２４から受けたオプションファイルを指定するパラメータで特定されるオプションファイル３０からオプション値を読出し、各モジュールに対応のオプションクラスオブジェクト２８に与えると、各オプションクラスオブジェクト２８はオプション値の妥当性をチェックして結果をアプリケーション２６に返す。アプリケーション２６はオプション値がいずれも妥当であればその値を使用して処理を開始する。妥当でないものが含まれていれば、その旨のメッセージを表示して処理を中止する。以下、オプション値をチェックする処理に必要な構成について説明する。

【００２７】

なお、本実施の形態では、モジュール定義情報ファイル４０及びオプションファイル３０については予めハードディスク等の記憶装置に準備することを想定している。またオプションクラスは、アプリケーション２６の設計時に同時に設計され、アプリケーション２６にその定義情報が組込まれる。

【００２８】

モジュール定義情報ファイル４０は、対応するモジュールに関する情報と、当該モジュールで使用されるオプション値をチェックするための情報とを統一した書式に従って記述したものである。図２に、モジュール定義情報ファイル４０のファイル形式を示す。図２を参照して、モジュール定義情報ファイル４０は、対応するモジュールのファイルシステム上の物理的な名称と記憶場所とをフルパス名によって指定する物理的モジュール名称フィールド５０と、モジュール名称をＧＵＩ表示する際に使用される表示用のモジュール名称のテキストを格納する表示用モジュール名称フィールド５２と、モジュールの機能をＧＵＩ画面上に表示する際のモジュール説明テキストを格納する説明テキストフィールド５４と、当該モジュールの作成者名を格納する作成者名フィールド５６とを含む。作成者名は、ＧＵＩ上での表示と、モジュールへのアクセスのセキュリティ管理のために使用される。

【００２９】

モジュール定義情報ファイル４０はさらに、モジュールのバージョン情報を格納するバージョン情報フィールド５８と、セキュリティ管理のレベル等、セキュリティ制約情報を記憶するセキュリティ制約情報フィールド６０と、このモジュール定義情報ファイル４０に対応するモジュールのオプション値をチェックするためにオプションクラスオブジェクト２８が参照するオプション／パラメータエリア６２とを含む。オプション／パラメータエリア６２は、各々が一つのオプション値の妥当性をチェックするために必要な制約情報

を含む、複数のオプション／パラメータ情報７０を含む。

【００３０】

図３に、オプション／パラメータ情報７０の構成を示す。図３を参照して、オプション／パラメータ情報７０は、モジュールが受け付けるオプション名のテキストを格納するオプション名フィールド８０と、オプションのデフォルト値を記憶するデフォルト値フィールド８２と、オプションの値としてシステムが内部的にとりうる値（有効値）が列挙されたもの（例えば「Toshi1 ｜ Toshi2 ｜ Toshi3」等）を格納する有効値フィールド８４と、オプション値が数値の場合に、オプション値の有効範囲を設定する情報を格納するための有効範囲フィールド８６とを含む。

【００３１】

有効範囲フィールド８６は、オプション値の最小値フィールド１００及び最大値フィールド１０４と、それら最小値又は最大値を有効範囲に含むか否かを示す情報をそれぞれ格納するフィールド１０２及び１０６とを含む。最小値及び最大値は、他のオプション値を参照することもできる。

【００３２】

オプション／パラメータ情報７０はさらに、オプション値の型を示す型フィールド８８と、モジュール起動時にこのオプション値を「，」等の区切り文字で区切って複数回繰返して指定可能な場合の、その回数の範囲を示す複数指定回数フィールド９０と、このオプション値の設定が必須か否かを示す情報を格納する必須指定フィールド９２とを含む。

【００３３】

オプション値の型としては、文字列、整数、実数、ＹＥＳ／ＮＯ、ファイル指定等が挙げられる。複数指定回数フィールド９０には、例えば繰返しを２回以上５回以下の範囲で許す場合、「２－５」という形式で指定する。

【００３４】

オプションクラスは、プロパティとして図３に示したのと同様、オプション値に対する制約情報をプロパティとして記憶することができる。

【００３５】

本実施の形態に係る応用システム２０は、コンピュータ上で動作するプログラムにより実現される。図１に示す応用システム２０のうち、アプリケーション２６について、オプション値のチェックに関し説明する。アプリケーション２６をコマンドベースで起動する際、本システムでは二通りの起動の仕方がある。一つはコマンドラインから直接オプション値を指定する仕方である。この場合、オプション値はプログラム名の後に、「-オプション（名称）＝オプション値」の形で、複数のオプションを「，」で区切りながら入力する。第２の方法は、予めオプションを指定するためのファイル（オプションファイル）を作成しておき、コマンドラインでは、オプションとして「-config=<オプションファイル名>」の形で指定する仕方である。ファイルを指定する場合には、一つだけ指定が許される。

【００３６】

図４に、オプションファイル３０の例を示す。図４を参照して、このオプションファイル３０は、ＸＭＬ形式で記載されたものである。このファイルによりオプションが指定されることを示す開始タグ<options>と終了タグ</options>の間に、まず開始タグ<module>と終了タグ</module>とにはさまれて、このオプションファイルが関係するモジュール名が記載され、その後に当該モジュールで使用されるオプション名及びその値が、開始タグ<option>及び終了タグ</option>の間に記載されている。これらオプション値の指定を複数のモジュールに対して行なうことができる。

【００３７】

最後に、モジュール共通のオプション値を開始タグ<common>及び</common>の間に記載することができる。

【００３８】

オプションクラスは、初期化処理において、上記したオプション／パラメータエリア６

２の内容を対応のプロパティ記憶領域に格納することで、以後のオプション値チェックを行なう。オプションクラスの主な機能には、次のようなものがある。

【００３９】

（１）　初期化
（２）　オプション値の取得及び妥当性のチェック
（３）　デフォルト値の設定
（４）　有効値の設定
（５）　有効範囲の設定
（６）　型の設定
（７）　複数指定回数の設定
（８）　必須の設定
（９）　オプション値の設定

これらの機能は、いずれも、オプション値が充足すべき制約条件を、プロパティに格納された制約情報との関係で規定するメソッドにより実現される。これらプロパティ及びメソッドは、システムの設計時にオプションクラスとして定義され、プログラムに組込まれる。入力されるオプション値により、これら複数のメソッドにより規定される制約条件が全て充足されているか否かを検査することにより、オプション値の妥当性を検査できる。

【００４０】

（１）の初期化処理は、前述したとおり、モジュール定義情報ファイル４０のフルパスを上位プログラムから受け取り、モジュール定義情報ファイル４０の内容を読み込んでオプション値を格納するオブジェクトを生成したり、オプション値チェックのために必要な情報を設定したりする処理である。

【００４１】

（２）のオプション値の取得及び妥当性のチェックは、上位プログラムから、入力されたオプション値を格納したオプションファイルのフルパスを渡されたときに、当該オプションファイル内からオプション値を抽出し、初期化処理で設定されたチェックを行ないその結果を返す処理である。あるオプション値についてデフォルト値が指定されている状態で、当該オプション値の入力がない場合には、デフォルト値がそのオプション値として設定される。

【００４２】

（３）～（８）は、オプション値チェックのための制約情報であるオプションクラスのプロパティを、上位プログラムから設定するための機能である。すなわち、初期化処理でオプション／パラメータエリア６２からプロパティに転記した制約情報に代え、新たな制約情報をプロパティに設定することができる。これらは、新たな制約情報を引数として、（３）～（８）に該当するメソッドを呼出すことにより実現できる。この機能により、初期化処理で設定された情報を、上位プログラムから動的に更新することができる。

【００４３】

（９）のオプション値の設定処理は、オプション値の取得が終わり、その妥当性が確認された後に、入力されたオプション値をそれぞれのオブジェクトに格納する処理である。

【００４４】

これら処理のうち、（１）、（２）及び（９）の処理について、それらを実現するメソッドの制御構造について説明する。（３）～（８）については、オブジェクトのプロパティを変更する処理であって、当業者であれば容易に実現できるので、ここではその詳細な説明は省略する。

【００４５】

図５は、初期化処理のフローチャートである。この処理では、モジュール定義情報ファイル４０のフルパスが引数として与えられる。図５を参照して、ステップ１６０で、モジュール定義情報ファイル４０を開き、その内容をメモリに読み込んでモジュール定義情報ファイル４０を閉じる。ステップ１６２で、読込んだ情報のうち、モジュール定義情報をオプションクラスオブジェクトの所定領域に書込む。ステップ１６４で、モジュール定義

情報ファイル４０中のオプション／パラメータエリア６２にある全てのオプションについて、そのオプション情報をオプションクラスオブジェクトの所定領域に書込む。

【００４６】

以上でオプションクラスの初期化が終了する。

【００４７】

図６は、オプション値の取得及び妥当性チェックを行なうメソッドのフローチャートである。ステップ１８０において、オプション値が格納されているオプションファイル３０から、その記述内容を取得する。取得に失敗すればエラー終了する。

【００４８】

ステップ１８２において、取得した入力データの中から入力されたオプション値をタグに基づいて取得する。

【００４９】

ステップ１８４では、オプションクラスオブジェクトの「必須」プロパティにしたがって必須チェックを行なう。必須プロパティが「必須」（ＹＥＳ）となっているにもかかわらず入力がされていない場合、エラーとなる。それ以外の場合にはエラーなしとする。なお、オプション値が複数でもよい場合には、オプション値が一つでも入力されていればエラーなしとする。

【００５０】

ステップ１８６では、入力されたオプション値の型と、オプションクラスオブジェクトの「型」プロパティとの比較を行ない、入力されたオプション値の型と「型」プロパティとの間に矛盾がないか否かを判定する。矛盾があればエラーとする。矛盾がなければステップ１８８に進む。

【００５１】

ステップ１８８では、型チェックがＯＫであると判定されたので、入力されたオプション値を返却用変数に設定する処理が行なわれる。

【００５２】

ステップ１９０では、「有効値」に値が設定されている場合のみ、入力されたオプション値が有効値のいずれかと一致するか否かを判定する。一致しなければエラーとする。

【００５３】

ステップ１９２では、オプション値の型が数値の場合で、かつ有効範囲プロパティが設定されている場合のみ、入力されたオプション値が有効範囲内か否かを判定する処理が行なわれる。有効範囲内になければエラーとする。

【００５４】

ステップ１９４では、オプション値が複数回指定可能な場合、入力されたオプション値の個数が、複数指定回数可能として指定された範囲に含まれるか否かを判定する。範囲外であればエラーとする。

【００５５】

以上の処理で誤りが一つもなければ、上位プログラムでは入力されたオプション値を利用できる。さもなければ再度ユーザからの入力を受取る。また、エラーが一つでも見つかればその時点で検査を終了し、オプション値が妥当でないことを示す検査結果を戻り値として返す。

【００５６】

なお、図６の説明では、オプションクラスオブジェクトの各属性には、正しい情報が設定されていることを前提として説明した。仮に設定情報に誤りがあれば、入力されたオプション値のいかんにかかわらずエラーとする。

【００５７】

図７は、オプション値の設定処理のフローチャートである。この処理では、既にチェックがなされ正しいと判定されたオプション値が、「オプション名＝オプション値」（「＝」は半角）を半角空白で連結した形でオプションクラスオブジェクトに渡される。

【００５８】

ステップ２００では、与えられたオプション値を半角空白で分割する。ステップ２０２では、分割した文字列を半角「＝」で分割する。半角「＝」がない場合、又は複数存在する場合にはエラー終了する。半角「＝」の後がない場合、オプション値を空とする。

【００５９】

ステップ２０４で、分割で得られたオプション名により設定対象のオプションを特定し、そのオプション値を取得する。設定対象データが存在しない場合にはエラー終了する。

【００６０】

ステップ２０６では、ステップ２０４で読み出されたオプション値が、ステップ２０２で分割により得られたオプション値と等しいか否かを判定する。等しければステップ２０８に進み、さもなければステップ２１０に進む。

【００６１】

ステップ２１０では、オプション値に対応するオブジェクトの対応するプロパティにオプション値を保持させる。ステップ２１２で、オプション値設定リストに当該オプションのオプション名を追加する。この後ステップ２０８に進む。

【００６２】

ステップ２０８では、ステップ２００において分割されたオプション値入力のうち、設定対象のオプション値がまだ残っているか否かを判定する。もし残っていればステップ２０２に戻り、次のオプション値についてステップ２０２以下の処理を繰返す。残っていなければ処理を終了する。

【００６３】

［コンピュータによる実現］

この実施の形態のシステムは、上記したようにコンピュータハードウェアと、そのコンピュータハードウェアにより実行されるプログラムと、コンピュータハードウェアに格納されるデータとにより実現される。図８はこのコンピュータシステム３３０の外観を示し、図９はコンピュータシステム３３０の内部構成を示す。

【００６４】

図８を参照して、このコンピュータシステム３３０は、ＦＤ（フレキシブルディスク）ドライブ３５２及びＣＤ－ＲＯＭ（コンパクトディスク読出専用メモリ）ドライブ３５０を有するコンピュータ３４０と、キーボード３４６と、マウス３４８と、モニタ３４２とを含む。

【００６５】

図９を参照して、コンピュータ３４０は、ＦＤドライブ３５２及びＣＤ－ＲＯＭドライブ３５０に加えて、ＣＰＵ（中央処理装置）３５６と、ＣＰＵ３５６、ＦＤドライブ３５２及びＣＤ－ＲＯＭドライブ３５０に接続されたバス３６６と、ブートアッププログラム等を記憶する読出専用メモリ（ＲＯＭ）３５８と、バス３６６に接続され、プログラム命令、システムプログラム、及び作業データ等を記憶するランダムアクセスメモリ（ＲＡＭ）３６０とを含む。コンピュータシステム３３０はさらに、プリンタ３４４を含んでいる。コンピュータ３４０はさらに、ローカルエリアネットワーク（ＬＡＮ）への接続を提供するネットワークアダプタボード３６８を含む。

【００６６】

コンピュータシステム３３０に汎用データチェック装置としての動作を行なわせるためのコンピュータプログラムは、ＣＤ－ＲＯＭドライブ３５０又はＦＤドライブ３５２に挿入されるＣＤ－ＲＯＭ３６２又はＦＤ３６４に記憶され、さらにハードディスク３５４に転送される。又は、プログラムはネットワークを通じてコンピュータ３４０に送信されハードディスク３５４に記憶されてもよい。プログラムは実行の際にＲＡＭ３６０にロードされる。ＣＤ－ＲＯＭ３６２から、ＦＤ３６４から、又はネットワークを介して、直接にＲＡＭ３６０にプログラムをロードしてもよい。

【００６７】

図５～図７に示したような制御構造を有する、オプションクラスのメソッドは、それぞれコンピュータ３４０を制御する複数の命令を含む。これら命令の実行に必要な基本的機

能のいくつかはコンピュータ３４０上で動作するＯＳ又はサードパーティのプログラム、若しくはコンピュータ３４０にインストールされる各種ツールキットのプログラムにより提供される。従って、このプログラムはこの実施の形態のシステム及び方法を実現するのに必要な機能全てを必ずしも含まなくてよい。このプログラムは、命令のうち、所望の結果が得られるように制御されたやり方で適切な機能又は「ツール」を呼出すことにより、上記した汎用データチェック装置を実現することができる命令のみを含んでいればよい。コンピュータシステム３３０自体の動作は周知であるので、ここでは繰返さない。

【００６８】

以上のように本実施の形態に係る汎用データチェック装置は、モジュールのオプション値の制約に関する記述を統一した記法でモジュール定義情報ファイルに記述しておくことで、いずれのプログラムからも利用することができる。また、複数のモジュールのどのオプション値に対しても共通のオプションクラスを用いることで値のチェックができるので、オプション値の妥当性に関するチェックルーチンを各プログラム内に書く必要がなくなるだけでなく、オプションごとに別々のルーチンを作成する必要もない。したがってプログラマの負担を大きく軽減でき、かつ、プログラマの技量に影響されず、均質で信頼性の高いシステムを構築できる。

【００６９】

今回開示された実施の形態は単に例示であって、本発明が上記した実施の形態のみに制限されるわけではない。本発明の範囲は、発明の詳細な説明の記載を参酌した上で、特許請求の範囲の各請求項によって示され、そこに記載された文言と均等の意味及び範囲内でのすべての変更を含む。

【図面の簡単な説明】

【００７０】

【図１】本発明の一実施の形態に係る応用システム２０のブロック図である。

【図２】モジュール定義情報ファイル４０の構造を模式的に示す図である。

【図３】オプション／パラメータ情報７０の構造を模式的に示す図である。

【図４】オプション値を定めたオプションファイル３０の一例を示す図である。

【図５】本発明の一実施の形態に係る汎用データチェック装置を実現するオプションクラスの初期化メソッドの制御構造を示すフローチャートである。

【図６】オプションクラスのオプション値取得及び妥当性チェックのためのメソッドの制御構造を示すフローチャートである。

【図７】オプションクラスのオプション値設定のためのメソッドの制御構造を示すフローチャートである。

【図８】図１に示す応用システム２０を実現するコンピュータハードウェアの外観図である。

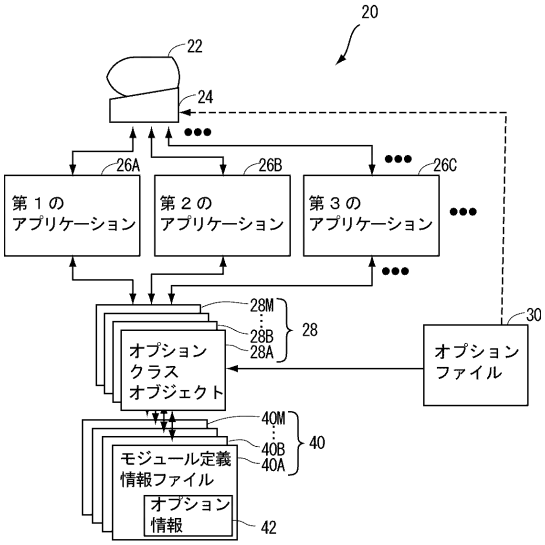
【図９】図８に示すコンピュータシステムのブロック図である。

【符号の説明】

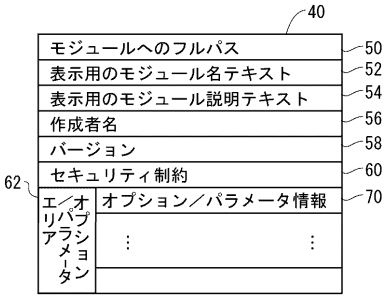
【００７１】

２０　応用システム、２２　モニタ、２４　入力装置、２６，２６Ａ～２６Ｃ　アプリケーション、２８，２８Ａ～２８Ｍ　オプションクラスオブジェクト、３０　オプションファイル、４０，４０Ａ～４０Ｍ　モジュール定義情報ファイル、５０　物理的モジュール名称フィールド、５２　表示用モジュール名称フィールド、５４　説明テキストフィールド、５６　作成者名フィールド、５８　バージョン情報フィールド、６０　セキュリティ制約情報フィールド、６２　オプション／パラメータエリア、７０　オプション／パラメータ情報、８０　オプション名フィールド、８２　デフォルト値フィールド、８４　有効値フィールド、８６　有効範囲フィールド、８８　型フィールド、９０　複数指定回数フィールド、９２　必須指定フィールド

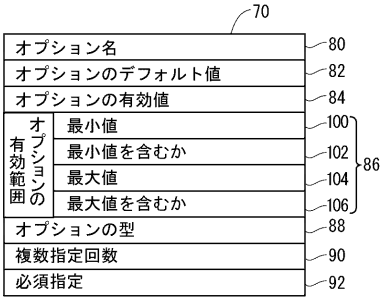
【図１】



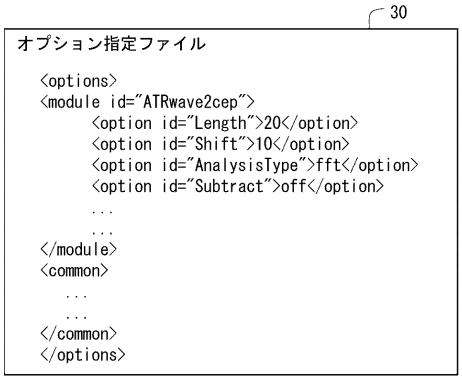
【図２】



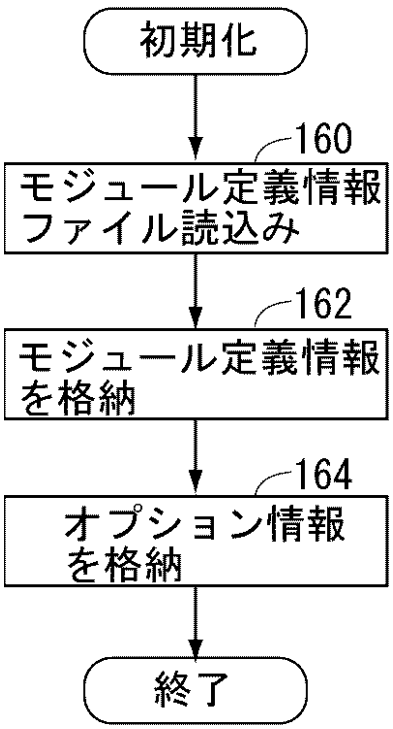
【図３】



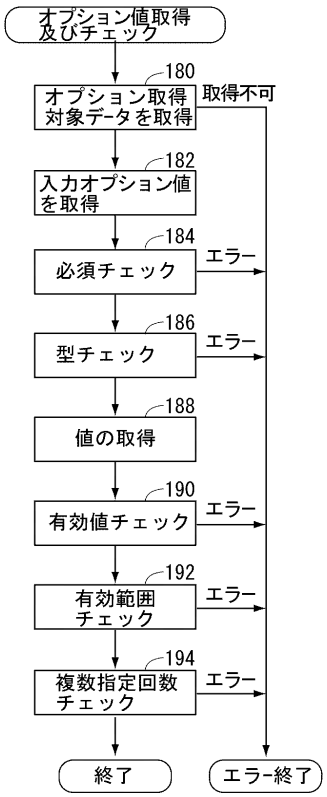
【図４】



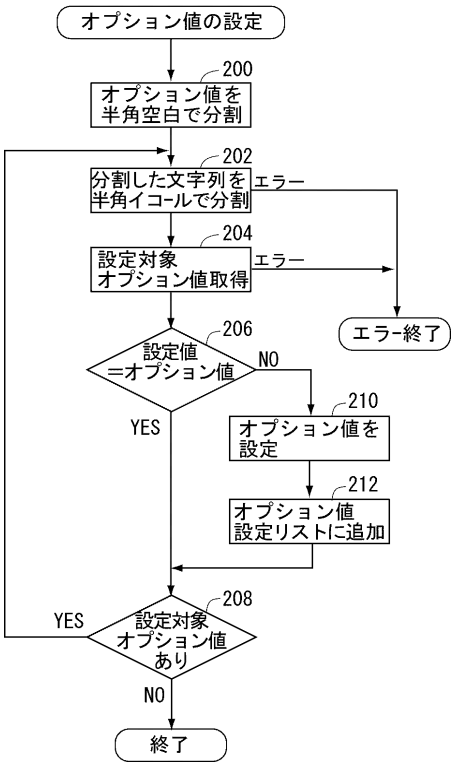
【図５】



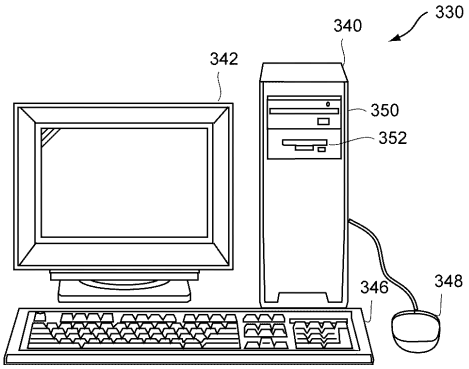
【図６】



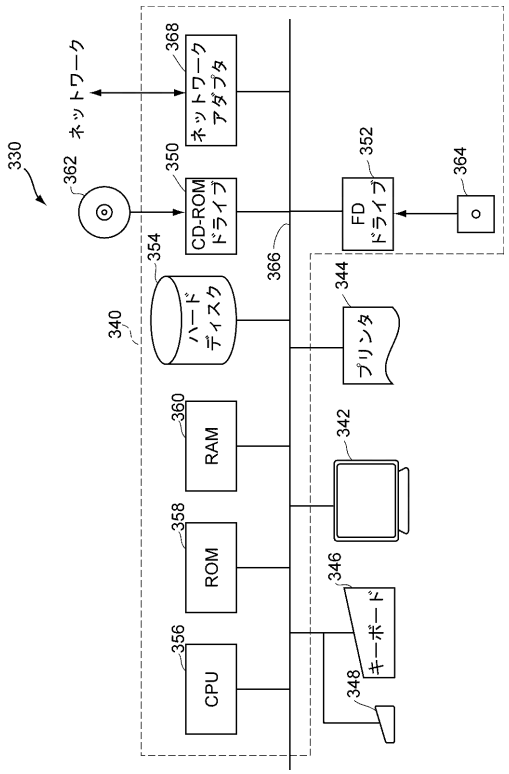
【図７】


オプション値の設定
200
オプション値を
半角空白で分割
202
分割した文字列を
半角イコールで分割
エラー
204
設定対象
オプション値取得
エラー
エラー終了
206
設定値
=オプション値
NO
YES
210
オプション値を
設定
212
オプション値
設定リストに追加
208
YES
設定対象
オプション値
あり
NO
終了

【図８】


330
340
342
350
352
346
348

【図９】


330
ネットワーク
362
368
ネットワーク
アダプタ
350
CD-ROM
ドライブ
352
FD
ドライブ
364
354
ハード
ディスク
340
366
344
プリンタ
360
RAM
342
358
ROM
346
キーボード
356
CPU
348

フロントページの続き

(56)参考文献 特開平０５－０１１９８９（ＪＰ，Ａ）
特開昭６２－１５０４３２（ＪＰ，Ａ）
高橋信頼，アーキテクチャと機能　画面遷移やＤＢアクセスを隠ぺいし，業務ロジックの開発に集中する，日経オープンシステム，日本，日経ＢＰ社，２００２年　2月１５日，第107号，pp.110-120
リソース・ファイルとプロパティ・ファイルの取り扱い，Ｊａｖａ　ＷＯＲＬＤ，日本，（株）ＩＤＧジャパン，２００１年　1月　1日，第5巻、第1号，pp.172-176

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１１／３０