(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4073740号
(P4073740)
(45)発行日　平成20年4月9日(2008.4.9)　(24)登録日　平成20年2月1日(2008.2.1)

(51)Int.Cl.　F I
G06F 3/14 (2006.01)　G06F 3/14 310A

請求項の数 10　(全 11 頁)

(21)出願番号	特願2002-259693(P2002-259693)
(22)出願日	平成14年9月5日(2002.9.5)
(65)公開番号	特開2003-196078(P2003-196078A)
(43)公開日	平成15年7月11日(2003.7.11)
審査請求日	平成17年9月5日(2005.9.5)
(31)優先権主張番号	09/948,781
(32)優先日	平成13年9月10日(2001.9.10)
(33)優先権主張国	米国(US)

(73)特許権者　398038580
ヒューレット・パッカード・カンパニー
HEWLETT-PACKARD COMPANY
アメリカ合衆国カリフォルニア州パロアルト　ハノーバー・ストリート　3000
(74)代理人　100081721
弁理士　岡田　次生
(74)代理人　100105393
弁理士　伏見　直哉
(74)代理人　100111969
弁理士　平野　ゆかり
(72)発明者　ミンジャン・バオ
アメリカ合衆国80547コロラド州ティムナス、メイン・ストリート　4329
最終頁に続く

(54)【発明の名称】複式データ表現方法

(57)【特許請求の範囲】
【請求項１】
アプリケーションのための複式データ表現方法であって、
オブジェクトについての未処理データセットを検索するステップであって、該未処理データは、機械読み取り可能なデータであり、ソフトウェア・アプリケーション用の物理オブジェクトを記述する、ステップと、
前記未処理データセットをデータ構造のインスタンスに格納するステップであって、前記データ構造は、複数のインスタンスを生成するためにデータの受け取りが可能な複数のフィールドを含み、前記未処理データセットを編成して論理的にグループ化する、ステップと、
前記未処理データセットを対応する表示データセットに変換するステップであって、前記対応する表示データセットは、前記未処理データセットよりも記述的であり、人間に読み取り可能なストリングを含む、ステップと、
前記対応する表示データセットを、前記未処理データセットと同じ、前記データ構造のインスタンスに格納するステップであって、該表示データは、前記未処理データが記述する物理オブジェクトに直接関係する、ステップと、を含み、
これによって、前記物理オブジェクトの前記未処理データセットおよび前記対応する表示データセットが、前記アプリケーションを通して整合的にかつ一貫して維持される、方法。
【請求項２】

前記データ構造における前記対応する表示データセットへの前記アプリケーションのフロントエンドからのアクセスを提供するステップをさらに含む、請求項１に記載の方法。
【請求項３】
前記検索するステップは、機械読み取り可能フォーマットの前記未処理データセットを検索するステップを含む、請求項１に記載の方法。
【請求項４】
前記格納するステップは、前記未処理データセットを前記アプリケーションのバックエンドに格納するステップを含む、請求項１に記載の方法。
【請求項５】
前記変換するステップは、前記未処理データセットを、人間読み取り可能フォーマットの前記対応する表示データセットに変換するステップを含む、請求項１に記載の方法。
【請求項６】
前記変換するステップは、コンパイルされたプログラミング言語を使用して前記未処理データセットを変換するステップを含む、請求項１に記載の方法。
【請求項７】
アプリケーションに複式データ表現（ＤＤＲ）を提供するユーザインタフェースであって、
データ構造を格納する、前記アプリケーションのバックエンドであって、前記データ構造は、複数のインスタンスを生成するためにデータの受け取りが可能な複数のフィールドを含み、
前記データ構造の各インスタンスは、オブジェクトについての未処理データセットであって、該未処理データは、機械読み取り可能なデータであり、ソフトウェア・アプリケーション用の物理オブジェクトを記述する、未処理データセットと、
該未処理データセットから変換される対応する表示データセットであって、該表示データは、人間に読み取り可能なストリングを含む表示データセットと、を含み、
前記データ構造は、前記未処理データセットおよび前記対応する表示データセットを編成し、論理的にグループ化して、同じインスタンスに格納する、バックエンドと、
前記対応する表示データセットをユーザに対して表示し、該表示データは、前記未処理データが記述する物理オブジェクトに直接関係する、前記アプリケーションのフロントエンドと、
を備える、ユーザインタフェース。
【請求項８】
前記物理オブジェクトの前記未処理データセットおよび前記対応する表示データセットは、前記アプリケーションを通して整合的かつ一貫して維持される、請求項７に記載のユーザインタフェース。
【請求項９】
前記物理オブジェクトの前記未処理データセットおよび前記対応する表示データセットは、前記物理オブジェクトの共通属性に従ってグループ化される、請求項７に記載のユーザインタフェース。
【請求項１０】
前記未処理データセットは、コンパイルされたプログラミング言語を使用して前記対応する表示データセットに変換される、請求項７に記載のユーザインタフェース。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
技術分野は、コンピュータソフトウェアアプリケーションに関し、特に、ソフトウェアアプリケーションの複式データ表現に関する。
【０００２】
【従来の技術】
ユーザは通常、ユーザインタフェース、すなわちアプリケーションのフロントエンドを通

してコンピュータソフトウェアと対話する。バックエンドとは、アプリケーションの、ユーザから隠されユーザインタフェースに示されない部分を指す。ユーザがユーザインタフェースにコマンドを入力すると、フロントエンドは通常、ユーザの命令を記録し、その情報をバックエンドに渡す。次いでバックエンドが、ユーザの命令に従ってすべての動作を実行し、ユーザインタフェースに表示すべきデータをフロントエンドに供給する。

【０００３】

【発明が解決しようとする課題】

現在、アプリケーションは通常、限られた記述性を有する単純な機械読み取り可能データである未処理データ(raw data)のみを格納する。より記述的でユーザを意図した表示データは、必要とされるとき常にアプリケーションのバックエンドから導出される。アプリケーションは、アプリケーション中の異なる場所から複数回、同じ未処理データセットを検索し、表示することを含みうるため、同じデータセットについて導出プロセスを繰り返し実行する必要があることがあり、アプリケーションを通しての冗長コーディングにつながる。アプリケーション中のコードが重複すると、アプリケーションサイズが増大し、より多くの作業がコードの維持に関わることになる。したがって、アプリケーションパフォーマンスが減速しうる。さらに、同じ未処理データセットから導出される表示データのフォーマットは一貫していないことがある。

【０００４】

【課題を解決するための手段】

アプリケーションの複式データ表現（ＤＤＲ）方法は、オブジェクトの未処理データセットを検索するステップと、複数のフィールドを含みうるデータ構造のインスタンスに未処理データセットを格納するステップと、を含む。データ構造は通常、未処理データセットを格納する前に、未処理データセットを編成し論理的にグループ化する。本方法は、未処理データセットを、未処理データセットよりも記述的な、対応する表示データセットに変換するステップと、対応する表示データセットを、未処理データセットと同じ、データ構造のインスタンスに格納するステップと、をさらに含む。本方法では、オブジェクトの未処理データセットおよび対応する表示データセットを、アプリケーションを通して整合的に(coherently)、かつ一貫して維持することが可能であり、これによってユーザへのより容易なデータの表示およびデータの計算が提供される。

【０００５】

未処理データおよび対応する表示データをグループ化し、同じデータ構造に格納することにより、ＤＤＲ方法で作成される、アプリケーションのフロントエンドおよびバックエンドの間でのインタフェースはより少なく、したがってより簡素化されたアプリケーションになる。さらに、本方法は、ソフトウェアアプリケーションを通して表示データを一貫したフォーマットで維持し提示するため、よりユーザフレンドリなインタフェースが作成される。

【０００６】

複式データ表現方法の好ましい実施形態について、同じ符号が同じ要素を指す添付図面を参照して詳細に説明する。

【０００７】

【発明の実施の形態】

コンピュータソフトウェアアプリケーションは通常、２種類のデータ、すなわち未処理データおよび表示データを有する。未処理データは、ユーザによる閲覧を意図しない単純な機械読み取り可能データである。たとえば、寸法測定の場合の未処理データは、長さ０．７５、幅３．５、および高さ５でありうる。表示データは通常、ユーザにとってそのデータが読みやすく理解しやすいように、単位またはフォーマットを未処理データに追加する。たとえば、上記大きさ測定の場合の表示データは、長さ０．７５ｃｍ、幅３．５ｃｍ、および高さ５ｍでありうる。重量測定の場合、表示データはグラム、キログラム、およびポンド等の単位を有しうる。

【０００８】

複式データ表現（ＤＤＲ）は、データを、機械読み取り可能（未処理データ）でありかつ人間読み取り可能（表示データ）であるように、複式形式で表現する方法である。ＤＤＲ方法は、整合的に都合よく、かつ一貫して、ソフトウェアアプリケーションを通して物理オブジェクトの属性を２つのフォーマット（未処理データおよび表示データ）で表現、維持することにより、ユーザへのより容易なデータの表示およびデータの計算を提供する。

【０００９】

オブジェクトに割り当てられた番号またはシンボルを使用する計算を含むコンピュータソフトウェアアプリケーションでは、複数の物理オブジェクトを論理的に定義しグループ化することができる。たとえば、あるオブジェクトについての長さ、幅、および高さの測定値は、別のオブジェクトの測定値とは別個にグループ化することができる。ＤＤＲ方法は、１つのオブジェクトについての未処理データおよび対応する表示データ双方を同じデータ構造にグループ化することができる。データ構造は、コンピュータにおいてデータを記録として編成する各種方法を指す。たとえば、プログラミング言語において、単純なデータ型は文字型または整数型でありうる。データ構造は、データを論理的に編成するために、異なる単純なデータ型を表す、異なるフィールドを含むものと定義することができる。例として、データ構造の第１のフィールドは、文字型であり、第２のフィールドは整数型でありうる。フィールドをデータで埋めることで、インスタンスを生成することができ、データ構造の各インスタンスは、同じ型の異なるデータ値のセットを含むことができる。上記例から、データ構造の第１のインスタンスは、１つのオブジェクトについてのデータを含み、第２のインスタンスは、別のオブジェクトについてのデータを含むことができる。

【００１０】

ＤＤＲは、未処理データおよび対応する表示データを論理的にグループ化し、データ構造の同じインスタンスに格納することによって達成することができる。ＤＤＲ方法は、Ｃまたは他の種類のプログラミング言語等コンピュータプログラミング言語を使用して実施することができる。好ましくは、ＤＤＲ方法は、データ構造を名付ける方法であり、データ構造および型定義の使用をサポートする大部分のコンパイルされた言語と併せて使用することができる。

【００１１】

図１は、未処理データ１１０および対応する表示データ１２０の双方を格納可能なデータ構造１３０を有する例示的なソフトウェアアプリケーション１６０を示す。データ構造１３０は、ソフトウェアエンジニアによりアプリケーション１６０の一部として設計する必要がありうる。データ構造１３０は通常、アプリケーション１６０のバックエンド１４０に配置される。

【００１２】

ＤＤＲ方法は、データベース、ハードウェアオブジェクト、または他の種類のデータ記憶装置１７０からオブジェクトについての未処理データ１１０を検索した後、未処理データ１１０の共通の属性に従って未処理データ１１０を論理グループにグループ化し、未処理データ１１０をデータ構造１３０のインスタンスに格納する。次に、ＤＤＲ方法は、未処理データ１１０を処理して、人間が読み取ることのできる文字列を有する対応する表示データ１２０に変換する。対応する表示データ１２０は、未処理データ１１０としてデータ構造１３０の同じインスタンスに格納することができる。最後に、ＤＤＲ方法は、データ構造１３０における表示データ１２０へのアクセスをアプリケーションのフロントエンド１５０に提供し、アプリケーションのフロントエンド１５０で表示データ１２０を人間であるユーザ１９０に提示することができる。

【００１３】

表１～表３は、データ構造１３０の異なるフィールドからのインスタンス生成の例を提供する。表１を参照すると、“Geometrical_Block”データ構造１３０において、未処理データフィールドは、整数型と定義される、length、width、およびheightというより小さなフィールドを含む。表示データフィールドは、文字列型と定義される、surface_areaお

よびvolumeというより小さなフィールドを含む。

【００１４】

【表１】

データ構造: Geometrical_Block
未処理データ:
int Length;
int Width;
int Height;
表示データ:
String Surface_Area
String Volume

【００１５】

表２および表３を参照すると、データ構造１３０のフィールドにデータを入力した後、インスタンス１（立方体１）およびインスタンス２（立方体２）を生成することができる。length、width、heightについての数は任意のものである。定義されているオブジェクトが立方体である場合、volumeおよびsurface_areaは、式：volume = WxLxHおよびsurface_area = WxLx6から導出される。

【００１６】

【表２】

インスタンス１：立方体１
未処理データ
Length: 3
Width: 3
Height: 3
表示データ:
Surface Area: 54 cm^2
Volume: 27 cm^3

【００１７】

【表３】

インスタンス２：立方体２
未処理データ
Length: 4
Width: 4
Height: 4
表示データ
Surface Area 96 cm^2
Volume: 64 cm^3

【００１８】

未処理データ１１０および表示データ１２０をグループ化して同じデータ構造１３０に格納することにより、単一ソフトウェア関数を使用して、表示データを導出するプロセスをカプセル化することができるため、データがユーザ１９０により要求される都度、表示データ１２０をバックエンド１４０から導出する必要をなくすことができる。その結果、ＤＤＲ方法では、アプリケーション１６０のフロントエンド１５０およびバックエンド１４０の間で作成されるインタフェースがより少なく、フロントエンド１５０が、データ１２

０にアクセスするために知る、あるいは実行する必要がある関数の数が低減される。したがって、アプリケーション１６０は、より簡素化され、それに伴いアプリケーションサイズが低減し、アプリケーションパフォーマンスが向上する。アプリケーション１６０に関わるコーディングがより少ないと、より容易なコードメンテナンスを保証することができる。さらに、未処理データ１１０および対応する表示データ１２０をグループ化し同じデータ構造１３０に格納することは、ソフトウェアアプリケーション１６０を通して一貫したフォーマットで表示データ１２０を維持し提供する助けとなり、よりユーザフレンドリなインタフェースが作成される。

【００１９】

アプリケーションにおける未処理データ１１０を使用してデータ構造１３０を埋めるソフトウェア関数、未処理データ１１０を処理して対応する表示データ１２０に変換するソフトウェア関数、および対応する表示データ１２０を未処理データ１１０と同じ、データ構造１３０のインスタンスに格納するソフトウェア関数を書くことができる。Ｃ言語等プログラミング言語を使用して、未処理データ１１０を処理して表示データ１２０に変換することができる。データの変換および処理は、たとえば、“Tools for Data Manipulation and Visualization”と称する１９９９年１２月２８日付けでＡｌｍｅｉｄａ他に付与された米国特許第６，００８，８０８号に記載されており、これを参照により本明細書に援用する。Almeida他は、複雑なデータの編成、管理、およびナビゲートのためのデータ処理および視覚化ツールを開示している。

【００２０】

表４は、異なる幾何学的な物理オブジェクトを表現し、ユーザ１９０に表示する必要のあるソフトウェアアプリケーション１６０の別の例を提供する。幾何学的なオブジェクトは、高さ、幅、長さ、色、密度、材料、およびテクスチャ等属性を有しうる。データ構造１３０は、アプリケーション１６０のバックエンド１４０に格納し、このバックエンド１４０において動作することができる。異なる幾何学的なオブジェクトは、データ構造１３０の１つのインスタンスが１つの幾何学的なオブジェクトに関連するデータ構造１３０において、論理的に表現することができる。

【００２１】

【表４】

```
Shape_Data_Type:
  Raw_data:
    Int height, width, length, weight;
    Color_t color;
    Density_t density;
    Material_t material;
    Texture_t texture;
  Display_data:
    String Dimension;  /*format: “WxLxH cm”*/
    String Color;    /*format: “Mauve:xxxrgb” or ”Lilac:xxxrgb”, etc...
*/
    String Volume  /*format: “V cc”*/
    String Weight  /*format: “W kg”*/
```

【００２２】

表４に示すように、各幾何学的オブジェクトは、未処理データ１１０および対応する表示データ１２０を含むことができる。未処理データ１１０によって、アプリケーション１６０のバックエンド１４０においてデータの計算および他の処理を迅速かつ容易に行うことができる。未処理データ１１０を処理して、ユーザインタフェース１５０上でユーザ１９０に表示される、対応する表示データ１２０を生成することができる。対応する表示デー

タ１２０によって、人間読み取り可能フォーマットでユーザ１９０にオブジェクトの属性を容易に表示することができる。双方の種類のデータは、整合性(coherency)および容易なアクセスのために、データ構造１３０の同じインスタンスに格納することができる。”get_shape_information”等の関数が、埋められた”Shape_Data_Type”構造を提供することができ、したがって、アプリケーション１６０での１つの場所だけで、すなわち”get_shape_information”関数だけで、未処理データ１１０を表示データ１２０に変換することができる。

【００２３】

図２は、複式データ表現の例示的な方法を示すフローチャートである。まず、ステップ２１０において、データ構造１３０をソフトウェアアプリケーション１６０の一部になるように設計することができる。次いで、ステップ２２０において、本方法は、オブジェクトについての未処理データ１１０を検索し収集する。次に、ステップ２３０において、物理オブジェクトを格納可能なデータ構造１３０のインスタンス内に未処理データ１１０を格納することができる。次いで、ステップ２４０において、未処理データ１１０を処理して、対応する表示データ１２０に変換することができ、ステップ２５０において、表示データが表示のためにデータ構造１３０の同じインスタンスに格納される。最後に、ステップ２６０において、本方法は、データ構造１３０における表示データ１２０へのアクセスをアプリケーション１６０のフロントエンド１５０に提供し、それによってユーザ１９０が表示データ１２０にアクセスすることができる。

【００２４】

図３は、例示的なＤＤＲ方法と併せて使用しうるコンピュータ３００の例示的なハードウェアコンポーネントを示す。コンピュータ３００は、インターネットまたは他の種類のコンピュータネットワークまたは電話網等、ネットワーク３１８との接続を備える。コンピュータ３００は通常、メモリ３０２、補助記憶装置３１２、プロセッサ３１４、入力装置３１６、表示装置３１０、および出力装置３０８を備える。

【００２５】

メモリ３０２は、ランダムアクセスメモリ（ＲＡＭ）または同様の種類のメモリを備えることができる。コンピュータ３００は、ウェブブラウザ３０６によりネットワーク３１８に接続することが可能である。ウェブブラウザ３０６は、ワールドワイドウェブ（ＷＷＷ）を介して他のコンピュータに接続し、コンピュータ３００に表示される情報を他のコンピュータから受信する。コンピュータ３００に表示される情報は通常、ＨＴＭＬまたはＸＭＬ等専門言語を使用して構築されるページに編成されている。補助記憶装置３１２は、ハードディスクドライブ、フレキシブルディスクドライブ、ＣＤ－ＲＯＭドライブ、または他の種類の不揮発性データ記憶装置を含むことができ、また様々なデータベースまたは他の資源に対応しうる。プロセッサ３１４は、メモリ３０２、補助記憶装置３１２に格納されている情報、またはインターネットや他のネットワーク３１８から受信する情報を実行することができる。入力装置３１６は、キーボード、キーパッド、カーソル制御装置、タッチスクリーン（おそらくスタイラスを備える）、またはマイクロホン等、データをコンピュータ３００に入力するための任意の装置を含みうる。表示装置３１０は、たとえば、コンピュータモニタ、フラットスクリーンディスプレイ、またはディスプレイパネル等、視覚的イメージを提示するための任意の種類の装置を含みうる。出力装置３０８は、プリンタ等ハードコピーフォーマットでデータを提示するための任意の種類の装置、およびスピーカを含む他の種類出力装置、またはデータをオーディオ形態で提供する任意の装置を含みうる。コンピュータ３００は、複数の入力装置、出力装置、および表示装置を備えることが可能である。

【００２６】

コンピュータ３００は、様々なコンポーネントを備えて図示されるが、当業者は、コンピュータ３００はさらなる、または異なるコンポーネントを包含しうることを理解しよう。さらに、本発明と一致する実施の態様は、メモリに格納されるものとして説明したが、当業者は、これら態様は、他の種類のコンピュータプログラム製品、またはハードディスク

、フレキシブルディスク、またはＣＤ－ＲＯＭを含む補助記憶装置等コンピュータ読み取り可能媒体、インターネットまたは他のネットワークからの搬送波、または他の形態のＲＡＭまたはＲＯＭにも格納／読み出し可能であることを理解しよう。コンピュータ読み取り可能媒体は、特定の方法を実行するようにコンピュータ３００を制御する命令を含むことができる。

【００２７】

複式データ表現の方法および装置について例示的な実施形態と併せて説明したが、当業者は、これら教示を鑑みて多くの変更が可能であり、本出願はそのあらゆる変形を網羅するものであることを理解しよう。

【００２８】

以下に本発明の態様を例示する。

【００２９】

１．アプリケーション（１６０）のための複式データ表現（ＤＤＲ）方法であって、オブジェクトについての未処理データ（１１０）セットを検索するステップ（２２０）と、前記未処理データ（１１０）セットをデータ構造（１３０）のインスタンスに格納するステップ（２３０）であって、前記データ構造（１３０）は、複数のインスタンスを生成するためにデータの受け取りが可能な複数のフィールドを含み、また前記未処理データ（１１０）セットを編成して論理的にグループ化する、ステップ（２３０）と、前記未処理データ（１１０）セットを対応する表示データ（１２０）セットに変換するステップ（２４０）であって、前記対応する表示データ（１２０）セットは、前記未処理データ（１１０）セットよりも記述的である、ステップ（２４０）と、前記対応する表示データ（１２０）セットを、前記未処理データ（１１０）セットと同じ、前記データ構造（１３０）のインスタンスに格納するステップ（２５０）と、を含み、これによって、前記オブジェクトの前記未処理データ（１１０）セットおよび前記対応する表示データ（１２０）セットが、前記アプリケーション（１６０）を通して整合的にかつ一貫して維持される、方法。

【００３０】

２．前記データ構造（１３０）における前記対応する表示データ（１２０）セットへの前記アプリケーション（１６０）のフロントエンド（１５０）からのアクセスを提供するステップ（２６０）をさらに含む、上記１記載の方法。

【００３１】

３．前記検索するステップは、機械読み取り可能フォーマットの前記未処理データ（１１０）セットを検索するステップ（２２０）を含む、上記１記載の方法。

【００３２】

４．前記格納するステップは、前記未処理データ（１１０）セットを前記アプリケーション（１６０）のバックエンド（１４０）に格納するステップ（２３０）を含む、上記１記載の方法。

【００３３】

５．前記変換するステップは、前記未処理データ（１１０）セットを、人間読み取り可能フォーマットの前記対応する表示データ（１２０）セットに変換するステップ（２４０）を含む、上記１記載の方法。

【００３４】

６．前記変換するステップは、コンパイルされたプログラミング言語を使用して前記未処理データ（１１０）セットを変換するステップ（２４０）を含む、上記１記載の方法。

【００３５】

７．アプリケーション（１６０）に複式データ表現（ＤＤＲ）を提供するユーザインタフェースであって、データ構造（１３０）を格納する、前記アプリケーション（１６０）のバックエンド（１４０）であって、前記データ構造（１３０）は、複数のインスタンスを生成するためにデータの受け取りが可能な複数のフィールドを含み、前記データ構造（１３０）の各インスタンスは、オブジェクトについての未処理データ（１１０）セット、および該未処理データ（１１０）セットから変換される対応する表示データ（１２０）セッ

トを含み、前記データ構造（１３０）は、前記未処理データ（１１０）セットおよび前記対応する表示データ（１２０）セットを編成し、論理的にグループ化して、同じインスタンスに格納する、バックエンド（１４０）と、前記対応する表示データ（１２０）セットをユーザ（１９０）に対して表示する、前記アプリケーション（１６０）のフロントエンド（１５０）と、を備える、ユーザインタフェース。

【００３６】

８．前記オブジェクトの前記未処理データ（１１０）セットおよび前記対応する表示データ（１２０）セットは、前記アプリケーション（１６０）を通して整合的かつ一貫して維持される、上記７記載のユーザインタフェース。

【００３７】

９．前記オブジェクトの前記未処理データ（１１０）セットおよび前記対応する表示データ（１２０）セットは、前記オブジェクトの共通属性に従ってグループ化される、上記７記載のユーザインタフェース。

【００３８】

１０．前記未処理データ（１１０）セットは、コンパイルされたプログラミング言語を使用して前記対応する表示データ（１２０）セットに変換される、上記７記載のユーザインタフェース。

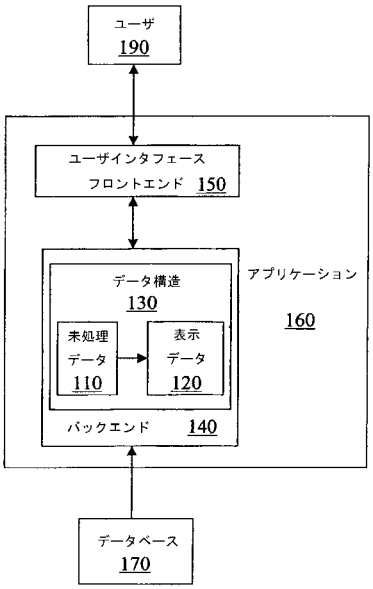
【図面の簡単な説明】

【図１】未処理データおよび表示データを格納することが可能なデータ構造を有する例示的なソフトウェアアプリケーションを示す。

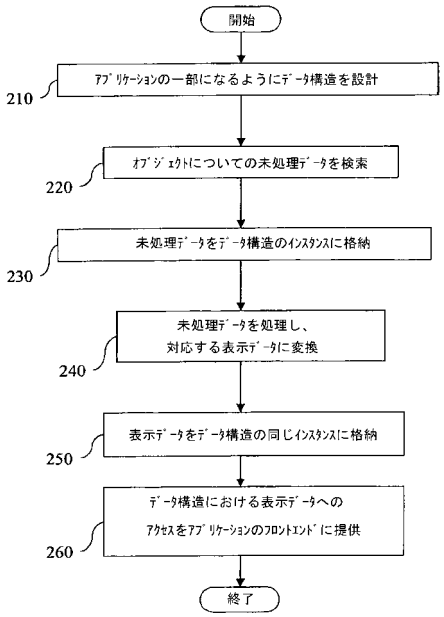
【図２】例示的な複式データ表現方法を示すフローチャートである。

【図３】図２の例示的な複式データ表現方法と併せて使用することができるコンピュータの例示的なハードウェアコンポーネントを示す。

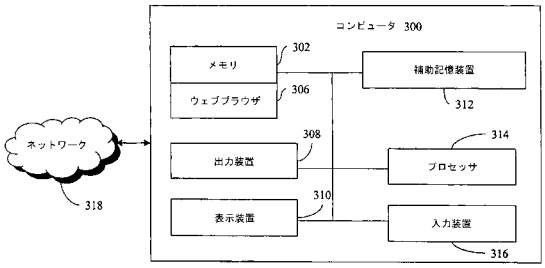
【図１】



【図２】



【図３】


コンピュータ 300
メモリ
302
ウェブブラウザ
306
補助記憶装置
312
ネットワーク
318
出力装置
308
プロセッサ
314
表示装置
310
入力装置
316

フロントページの続き

(72)発明者　シェリル・ハーンドン
アメリカ合衆国８０５２５コロラド州フォート・コリンズ、アムハースト・ストリート　２３１３

審査官　日下　善之

(56)参考文献　特開平１０－１２４３５６（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F　3/14