

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5587905号
(P5587905)

(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 12/00 (2006.01)

G 0 6 F 12/00 5 1 0 A

請求項の数 88 (全 23 頁)

(21) 出願番号 特願2011-539630 (P2011-539630)
 (86) (22) 出願日 平成21年12月1日 (2009.12.1)
 (65) 公表番号 特表2012-510686 (P2012-510686A)
 (43) 公表日 平成24年5月10日 (2012.5.10)
 (86) 国際出願番号 PCT/US2009/066206
 (87) 国際公開番号 W02010/065507
 (87) 国際公開日 平成22年6月10日 (2010.6.10)
 審査請求日 平成24年11月27日 (2012.11.27)
 (31) 優先権主張番号 61/119,148
 (32) 優先日 平成20年12月2日 (2008.12.2)
 (33) 優先権主張国 米国 (US)

(73) 特許権者 509123208
 アビニシオ テクノロジー エルエルシー
 アメリカ合衆国 02421 マサチュー
 セッツ州 レキシントン スプリング ス
 トリート 201
 (74) 代理人 100079108
 弁理士 稲葉 良幸
 (74) 代理人 100109346
 弁理士 大貫 敏史
 (72) 発明者 ウィンターズ クリストファー ジェイ
 アメリカ合衆国 02421 マサチュー
 セッツ州 レキシントン ヴァレー スト
 リート 69

最終頁に続く

(54) 【発明の名称】 データメンテナンスシステム

(57) 【特許請求の範囲】

【請求項 1】

少なくとも一つのプロセッサ及び一つのストレージシステムを含むデータ管理システムにおけるデータメンテナンス方法であって、

テーブルを格納するストレージシステムにアクセスすることであって、前記テーブルは、一つ以上のレコードを含み、各レコードは、データを含む少なくとも一つのフィールドを含み、前記ストレージシステムは、前記テーブルに関連付けられた一つ以上の記述子も格納し、前記記述子の少なくともいくつかは、それぞれ、一つのレコードに対する一つのデータソースを特定するデータを含み、前記記述子の少なくともいくつかは、それぞれ、前記一つのレコードの一つ以上のフィールドのそれぞれに対する一つのデータソースを特定するデータを含む、ストレージシステムにアクセスすること；及び、

前記記述子に従って、かつ、データの一つ以上のソース要素を提供する第1のデータソースの身元に少なくとも部分的に基づいて、前記第1のデータソースからの前記データのソース要素の受信に応じて、前記テーブルの前記レコードの一つにおける少なくとも一つのフィールドの変更が許可されているか否かを判別すること；

を含む方法。

【請求項 2】

さらに、少なくとも一つの記述子に従って、前記ストレージシステム内のレコードのフィールドにおけるデータを置換するために前記ソース要素のいずれかが使用されるべきか否かを判別することを含む請求項1に記載の方法。

【請求項 3】

さらに、少なくとも一つの記述子に従って、対応するソース要素が削除されていることに
 応答して前記ストレージシステム内のレコードのフィールドにおけるデータが削除され
 るべきか否かを判別することを含む請求項 1 に記載の方法。

【請求項 4】

さらに、前記第 1 のデータソースによって提供される前記ソース要素と異なる一つ以上
 のデータのソース要素を提供する第 2 のデータソースにアクセスすることを含む請求項 1
 に記載の方法。

【請求項 5】

前記第 1 のデータソースは前記レコードに対する前記データソースであり、前記第 2 の
 データソースは前記レコードの前記フィールドの少なくとも一つにおけるデータに対する
 前記データソースである、請求項 4 に記載の方法。

10

【請求項 6】

前記記述子の少なくとも一つは一つのレコードに対する一つのデータソースを特定し、
 前記記述子の少なくとも他の一つは前記レコードのフィールドにおけるデータに対する他
 の一つのデータソースを特定する請求項 1 に記載の方法。

【請求項 7】

前記判別することはユーザ入力を必要とする請求項 1 に記載の方法。

【請求項 8】

前記第 1 のデータソースは、ユーザの直接入力を受け取るユーザインターフェースを含
 む請求項 1 に記載の方法。

20

【請求項 9】

記述子は所与のレコードの変更を制御するレコード所有者を特定する請求項 1 に記載の
方法。

【請求項 10】

前記レコード所有者は前記所与のレコードの削除を制御する請求項 9 に記載の方法。

【請求項 11】

前記レコード所有者は前記所与のレコードを生成した請求項 9 に記載の方法。

【請求項 12】

記述子は所与のレコードの所与のフィールドに格納されたデータの変更を制御するフ
 ィールド所有者を特定する請求項 1 に記載の方法。

30

【請求項 13】

前記フィールド所有者は前記所与のフィールド内に格納されたデータの更新を制御する
請求項 12 に記載の方法。

【請求項 14】

さらに、ユーザインターフェースをユーザに提供し且つ前記ユーザから前記ストレ
 ジシステムのデータに関係する決定を受け取ることを含む請求項 1 に記載の方法。

【請求項 15】

データをメンテナンスするシステムであって、

テーブルを格納するストレージシステムであって、前記テーブルは、一つ以上のレコー
 ドを含み、各レコードは、データを含む少なくとも一つのフィールドを含み、前記ストレ
 ジシステムは、前記テーブルに関連付けられた一つ以上の記述子も格納し、前記記述子
 の少なくともいくつかは、それぞれ、一つのレコードに対する一つのデータソースを特定
 するデータを含み、前記記述子の少なくともいくつかは、それぞれ、前記一つのレコー
 ドの一つ以上のフィールドのそれぞれに対する一つのデータソースを特定するデータを含む
、ストレージシステム；

40

データの一つ以上のソース要素を提供する第 1 のデータソース；及び、

前記記述子に従って、かつ、データの一つ以上のソース要素を提供する第 1 のデータソ
ースの身元に少なくとも部分的に基づいて、前記第 1 のデータソースからの前記データの
ソース要素の受信に応じて、前記テーブルの前記レコードの一つにおける少なくとも一つ

50

のフィールドの変更が許可されているか否かを判別するように構成されたデータマネージャ；

を含むシステム。

【請求項 1 6】

前記データマネージャは、さらに、少なくとも一つの記述子に従って、前記ソース要素のいずれかが前記ストレージシステム内のレコードのフィールドにおけるデータを置換するために使用されるべきであるか否かを判別するように構成されている、請求項 1 5 に記載のシステム。

【請求項 1 7】

前記データマネージャは、さらに、少なくとも一つの記述子に従って、前記ストレージシステム内のレコードのフィールドにおけるデータが、対応するソース要素が削除されていることに応答して、削除されるべきか否かを判別するように構成されている、請求項 1 5 に記載のシステム。

10

【請求項 1 8】

さらに、前記第 1 のデータソースによって提供される前記ソース要素と異なるデータの一つ以上のソース要素を提供する第 2 のデータソースを含む請求項 1 5 に記載のシステム。

【請求項 1 9】

前記第 1 のデータソースは前記レコードに対する前記データソースであり、前記第 2 のデータソースは前記レコードの前記フィールドの少なくとも一つにおけるデータに対する前記データソースである請求項 1 8 に記載のシステム。

20

【請求項 2 0】

前記記述子の少なくとも一つは一つのレコードに対する一つのデータソースを特定し、前記記述子の少なくとも他の一つは前記レコードのフィールドにおけるデータに対する他の一つのデータソースを特定する請求項 1 5 に記載のシステム。

【請求項 2 1】

前記判別することはユーザ入力を必要とする請求項 1 5 に記載のシステム。

【請求項 2 2】

前記第 1 のデータソースはユーザの直接入力を受け取るユーザインターフェースを含む請求項 1 5 に記載のシステム。

30

【請求項 2 3】

記述子は所与のレコードの変更を制御するレコード所有者を特定する請求項 1 5 に記載のシステム。

【請求項 2 4】

前記レコード所有者は前記所与のレコードの削除を制御する請求項 2 3 に記載のシステム。

【請求項 2 5】

前記レコード所有者は前記所与のレコードを生成した請求項 2 3 に記載のシステム。

【請求項 2 6】

記述子は所与のレコードの所与のフィールドに格納されたデータの変更を制御するフィールド所有者を特定する請求項 1 5 に記載のシステム。

40

【請求項 2 7】

前記フィールド所有者は前記所与のフィールド内に格納された前記データの更新を制御する請求項 2 6 に記載のシステム。

【請求項 2 8】

さらに、前記ストレージシステムのデータに関するユーザの決定を前記データマネージャに提供するユーザインターフェースを含む請求項 1 5 に記載のシステム。

【請求項 2 9】

データ管理システムにおいてデータのメンテナンスを行うシステムであって、テーブルを格納するストレージシステムにアクセスする手段であって、前記テーブルは

50

、一つ以上のレコードを含み、各レコードは、データを含む少なくとも一つのフィールドを含み、前記ストレージシステムは、前記テーブルに関連付けられた一つ以上の記述子も格納し、前記記述子の少なくともいくつかは、それぞれ、一つのレコードに対する一つのデータソースを特定するデータを含み、前記記述子の少なくともいくつかは、それぞれ、前記一つのレコードの一つ以上のフィールドのそれぞれに対する一つのデータソースを特定するデータを含む、ストレージシステムにアクセスする手段；及び、

前記記述子に従って、かつ、データの一つ以上のソース要素を提供する第1のデータソースの身元に少なくとも部分的に基づいて、前記第1のデータソースからの前記データのソース要素の受信に応じて、前記テーブルの前記レコードの一つにおける少なくとも一つのフィールドの変更が許可されているか否かを判別する手段；

10

を含むシステム。

【請求項30】

データ管理システムにおけるデータのメンテナンスを行うコンピュータプログラムを格納するコンピュータ読み出し可能媒体であって、前記コンピュータプログラムはコンピュータに、

テーブルを格納するストレージシステムにアクセスすることであって、前記テーブルは、一つ以上のレコードを含み、各レコードは、データを含む少なくとも一つのフィールドを含み、前記ストレージシステムは、前記テーブルに関連付けられた一つ以上の記述子も格納し、前記記述子の少なくともいくつかは、それぞれ、一つのレコードに対する一つのデータソースを特定するデータを含み、前記記述子の少なくともいくつかは、それぞれ、前記一つのレコードの一つ以上のフィールドのそれぞれに対する一つのデータソースを特定するデータを含む、ストレージシステムにアクセスすること；及び、

20

前記記述子に従って、かつ、データの一つ以上のソース要素を提供する第1のデータソースの身元に少なくとも部分的に基づいて、前記第1のデータソースからの前記データのソース要素の受信に応じて、前記テーブルの前記レコードの一つにおける少なくとも一つのフィールドの変更が許可されているか否かを判別すること；

を実行させるための命令を含むコンピュータ読み出し可能媒体。

【請求項31】

さらに、少なくとも一つの記述子に従って、前記ストレージシステム内のレコードのフィールドにおけるデータを置換するために前記ソース要素のいずれかが使用されるべきか否かを判別する手段を含む請求項29に記載のシステム。

30

【請求項32】

さらに、少なくとも一つの記述子に従って、対応するソース要素が削除されていることに応答して前記ストレージシステム内のレコードのフィールドにおけるデータが削除されるべきか否かを判別する手段を含む請求項29に記載のシステム。

【請求項33】

さらに、前記第1のデータソースによって提供される前記ソース要素と異なる一つ以上のデータのソース要素を提供する第2のデータソースにアクセスする手段を含む請求項29に記載のシステム。

【請求項34】

40

前記第1のデータソースは前記レコードに対する前記データソースであり、前記第2のデータソースは前記レコードの前記フィールドの少なくとも一つにおけるデータに対する前記データソースである、請求項33に記載のシステム。

【請求項35】

前記記述子の少なくとも一つは一つのレコードに対する一つのデータソースを特定し、前記記述子の少なくとも他の一つは前記レコードのフィールドにおけるデータに対する他の一つのデータソースを特定する請求項29に記載のシステム。

【請求項36】

前記判別することはユーザ入力を必要とする請求項29に記載のシステム。

【請求項37】

50

前記第 1 のデータソースは、ユーザの直接入力を受け取るユーザインターフェースを含む請求項 2 9 に記載のシステム。

【請求項 3 8】

記述子は所与のレコードの変更を制御するレコード所有者を特定する請求項 2 9 に記載のシステム。

【請求項 3 9】

前記レコード所有者は前記所与のレコードの削除を制御する請求項 3 8 に記載のシステム。

【請求項 4 0】

前記レコード所有者は前記所与のレコードを生成した請求項 3 8 に記載のシステム。

【請求項 4 1】

記述子は所与のレコードの所与のフィールドに格納されたデータ要素の変更を制御するフィールド所有者を特定する請求項 2 9 に記載のシステム。

【請求項 4 2】

前記フィールド所有者は前記所与のフィールド内に格納されたデータ要素の更新を制御する請求項 4 1 に記載のシステム。

【請求項 4 3】

さらに、ユーザインターフェースをユーザに提供し且つ前記ユーザから前記ストレージシステムのデータ要素に関する決定を受け取る手段を含む請求項 2 9 に記載のシステム。

【請求項 4 4】

さらに、コンピュータに、少なくとも一つの記述子に従って、前記ストレージシステム内のレコードのフィールドにおけるデータを置換するために前記ソース要素のいずれかが使用されるべきかを判別することを実行させるための命令を含む請求項 3 0 に記載のコンピュータ読み出し可能媒体。

【請求項 4 5】

さらに、コンピュータに、少なくとも一つの記述子に従って、対応するソース要素が削除されていることに応答して前記ストレージシステム内のレコードのフィールドにおけるデータが削除されるべきかを判別することを実行させるための命令を含む請求項 3 0 に記載のコンピュータ読み出し可能媒体。

【請求項 4 6】

さらに、コンピュータに、前記第 1 のデータソースによって提供される前記ソース要素と異なる一つ以上のデータのソース要素を提供する第 2 のデータソースにアクセスすることを実行させるための命令を含む請求項 3 0 に記載のコンピュータ読み出し可能媒体。

【請求項 4 7】

前記第 1 のデータソースは前記レコードに対する前記データソースであり、前記第 2 のデータソースは前記レコードの前記フィールドの少なくとも一つにおけるデータに対する前記データソースである、請求項 4 6 に記載のコンピュータ読み出し可能媒体。

【請求項 4 8】

前記記述子の少なくとも一つは一つのレコードに対する一つのデータソースを特定し、前記記述子の少なくとも他の一つは前記レコードのフィールドにおけるデータに対する他の一つのデータソースを特定する請求項 3 0 に記載のコンピュータ読み出し可能媒体。

【請求項 4 9】

前記判別することはユーザ入力を必要とする請求項 3 0 に記載のコンピュータ読み出し可能媒体。

【請求項 5 0】

前記第 1 のデータソースは、ユーザの直接入力を受け取るユーザインターフェースを含む請求項 3 0 に記載のコンピュータ読み出し可能媒体。

【請求項 5 1】

記述子は所与のレコードの変更を制御するレコード所有者を特定する請求項 3 0 に記載

10

20

30

40

50

のコンピュータ読み出し可能媒体。

【請求項 5 2】

前記レコード所有者は前記所与のレコードの削除を制御する請求項 5 1 に記載のコンピュータ読み出し可能媒体。

【請求項 5 3】

前記レコード所有者は前記所与のレコードを生成した請求項 5 1 に記載のコンピュータ読み出し可能媒体。

【請求項 5 4】

記述子は所与のレコードの所与のフィールドに格納されたデータの変更を制御するフィールド所有者を特定する請求項 3 0 に記載のコンピュータ読み出し可能媒体。

【請求項 5 5】

前記フィールド所有者は前記所与のフィールド内に格納されたデータの更新を制御する請求項 5 4 に記載のコンピュータ読み出し可能媒体。

【請求項 5 6】

さらに、ユーザインターフェースをユーザに提供し且つ前記ユーザから前記ストレージシステムのデータに関する決定を受け取る命令を含む請求項 3 0 に記載のコンピュータ読み出し可能媒体。

【請求項 5 7】

前記テーブルは、前記一つ以上の記述子を格納する、請求項 1 に記載の方法。

【請求項 5 8】

前記フィールドの変更が許可されているか否かを判別することは、一つ以上の規則を処理することに基づく、請求項 1 に記載の方法。

【請求項 5 9】

前記テーブルは、前記一つ以上の記述子を格納する、請求項 1 5 に記載のシステム。

【請求項 6 0】

前記フィールドの変更が許可されているか否かを判別することは、一つ以上の規則を処理することに基づく、請求項 1 5 に記載のシステム。

【請求項 6 1】

前記テーブルは、前記一つ以上の記述子を格納する、請求項 2 9 に記載のシステム。

【請求項 6 2】

前記フィールドの変更が許可されているか否かを判別することは、一つ以上の規則を処理することに基づく、請求項 2 9 に記載のシステム。

【請求項 6 3】

前記テーブルは、前記一つ以上の記述子を格納する、請求項 3 0 に記載のコンピュータ読み出し可能媒体。

【請求項 6 4】

前記フィールドの変更が許可されているか否かを判別することは、一つ以上の規則を処理することに基づく、請求項 3 0 に記載のコンピュータ読み出し可能媒体。

【請求項 6 5】

前記規則の少なくとも一つはユーザ入力なしに自動的に取り扱われる請求項 5 8 に記載の方法。

【請求項 6 6】

さらに、前記規則を含むメンテナンス計画にアクセスすることを含む請求項 5 8 に記載の方法。

【請求項 6 7】

前記メンテナンス計画は、どのソース要素が前記ストレージシステム内のどのフィールドに対応するかを示すマッピングを含む請求項 6 6 に記載の方法。

【請求項 6 8】

前記規則の少なくとも一つはユーザ入力なしに自動的に取り扱われる請求項 6 0 に記載のシステム。

10

20

30

40

50

【請求項 6 9】

前記データマネージャは前記規則を含むメンテナンス計画に従って作動する請求項 6 0 に記載のシステム。

【請求項 7 0】

前記メンテナンス計画はどのソース要素が前記ストレージシステム内のどのフィールドに対応するかを示すマッピングを含む請求項 6 9 に記載のシステム。

【請求項 7 1】

前記規則の少なくとも一つはレコードのフィールドにおけるデータを第 1 のフォーマットから第 2 のフォーマットに変換する請求項 5 8 に記載の方法。

【請求項 7 2】

前記規則の少なくとも一つはレコードのフィールドにおけるデータが同フィールドのコンテキストにおいて有効な形式であるか否かを判別する請求項 5 8 に記載の方法。

【請求項 7 3】

前記規則の少なくとも一つはフィールドの前記データソースの構成の変更を取り扱う請求項 5 8 に記載の方法。

【請求項 7 4】

前記規則の少なくとも一つはレコードのフィールドにおけるデータを第 1 のフォーマットから第 2 のフォーマットに変換する請求項 6 0 に記載のシステム。

【請求項 7 5】

前記規則の少なくとも一つはレコードのフィールドにおけるデータが同フィールドのコンテキストにおいて有効な形式であるか否かを判別する請求項 6 0 に記載のシステム。

【請求項 7 6】

前記規則の少なくとも一つはフィールドの前記データソースの構成の変更を取り扱う請求項 6 0 に記載のシステム。

【請求項 7 7】

前記規則の少なくとも一つはユーザ入力なしに自動的に取り扱われる請求項 6 2 に記載のシステム。

【請求項 7 8】

前記規則を処理する手段は、前記規則を含むメンテナンス計画に従って作動する請求項 6 2 に記載のシステム。

【請求項 7 9】

前記メンテナンス計画は、どのソース要素が前記ストレージシステム内のどのフィールドに対応するかを示すマッピングを含む請求項 7 8 に記載のシステム。

【請求項 8 0】

前記規則の少なくとも一つはレコードのフィールドにおけるデータを第 1 のフォーマットから第 2 のフォーマットに変換する請求項 6 2 に記載のシステム。

【請求項 8 1】

前記規則の少なくとも一つはレコードのフィールドにおけるデータが同フィールドのコンテキストにおいて有効な形式であるか否かを判別する請求項 6 2 に記載のシステム。

【請求項 8 2】

前記規則の少なくとも一つはフィールドの前記データソースの構成の変更を取り扱う請求項 6 2 に記載のシステム。

【請求項 8 3】

前記規則の少なくとも一つはユーザ入力なしに自動的に取り扱われる請求項 6 4 に記載のコンピュータ読み出し可能媒体。

【請求項 8 4】

さらに、前記規則を含むメンテナンス計画にアクセスする命令を含む請求項 6 4 に記載のコンピュータ読み出し可能媒体。

【請求項 8 5】

前記メンテナンス計画は、どのソース要素が前記ストレージシステム内のどのフィール

10

20

30

40

50

ドに対応するかを示すマッピングを含む請求項 8 4 に記載のコンピュータ読み出し可能媒体。

【請求項 8 6】

前記規則の少なくとも一つはレコードのフィールドにおけるデータを第 1 のフォーマットから第 2 のフォーマットに変換する請求項 6 4 に記載のコンピュータ読み出し可能媒体。

【請求項 8 7】

前記規則の少なくとも一つはレコードのフィールドにおけるデータが同フィールドのコンテキストにおいて有効な形式であるか否かを判別する請求項 6 4 に記載のコンピュータ読み出し可能媒体。

10

【請求項 8 8】

前記規則の少なくとも一つはフィールドの前記データソースの構成の変更を取り扱う請求項 6 4 に記載のコンピュータ読み出し可能媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明はデータメンテナンスシステムに関する。

【背景技術】

【0002】

近年のデータメンテナンスシステムは多数のソースからデータを受取ることができる。データメンテナンスシステムは、単純な受動的ストレージ機構として作用するのではなく、多数のソースからのデータに対する統合整理収納庫 (consolidated repository) をメンテナンスすることが要求され、且つ、システムのユーザに対してデータ全体に対するアクセスの統一見地及びデータ全体に対する単一のアクセスポイントを提供する。しかしながら、多数のデータソースを統合化することは幾つかの課題を呈示する。異なるソースのそれぞれは、データベースのような論理的なデータ収集のセグメント (segments of a logical data

20

collection) を含むことがあり、或いは新しく更新されたデータ又は期限切れデータののような同一データの時間的に異なるバージョンを含むことがある。これらのソースのすべては、データストレージシステムの構成又はフォーマットとは異なる構成又はフォーマットでデータを格納する場合がある。

30

【発明の概要】

【0003】

< 本願に関連する出願 >

この出願は、2008 年 12 月 2 日出願の米国特許出願第 61 / 119, 148 号の優先権を主張するものであり、本願に引用して組み込まれる。

概して、少なくとも一つのプロセッサ及び一つのストレージシステムを含むデータ管理システムにおけるデータメンテナンス方法は、データの一つ以上のソース要素を提供する第 1 のデータソースをアクセスすること；それぞれが少なくとも一つのデータ要素を含むデータのサブセットであって一つ以上のデータのサブセットを格納する格納システムをアクセスし、且つ、一つのサブセットに対する一つのデータソースを特定するとともに一つ以上の前記サブセットのそれぞれに対するデータソースを特定する一つ以上の記述子を格納すること；及び、前記記述子に従って、前記ストレージシステム内の要素の変更が許可されているか否かを判別する規則を処理すること；を含む。

40

【0004】

他の概要において、データメンテナンス用システムは、それぞれが少なくとも一つのデータ要素を含むデータのサブセットであって一つ以上のデータのサブセットと、それぞれが一つのサブセットに対する一つのデータソースを特定するとともに前記サブセットの一つ以上の要素に対するデータソースを特定する一つ以上の記述子と、を格納するストレージシステム；データの一つ以上のソース要素を提供する第 1 のデータソース；及び、記述

50

子に従って、前記ストレージシステム内の要素の変更が許可されているか否かを判別する規則を処理するように構成されたデータマネージャ；を含む。

【 0 0 0 5 】

他の概要において、データ管理システムにおいてデータメンテナンス用のシステムは、データの一つ以上のソース要素を提供する第1のデータソースをアクセスする手段；それぞれが少なくとも一つのデータ要素を含むデータのサブセットであって一つ以上のデータのサブセットを格納する格納システムをアクセスし、且つ、一つのサブセットに対する一つのデータソースを特定するとともに一つ以上の前記サブセットのそれぞれの要素に対するデータソースを特定する一つ以上の記述子を格納する手段；及び、前記記述子に従って、前記ストレージシステム内の要素の変更が許可されているか否かを判別する規則を処理する手段；を含む。

10

【 0 0 0 6 】

他の概要において、コンピュータ読み出し可能媒体に格納される、データ管理システムにおいてデータのメンテナンスを行うコンピュータプログラムは、コンピュータに、データの一つ以上のソース要素を提供する第1のデータソースをアクセスさせ；それぞれが少なくとも一つのデータ要素を含むデータのサブセットであって一つ以上のデータのサブセットを格納する格納システムをアクセスさせ、且つ、一つのサブセットに対する一つのデータソースを特定するとともに前記サブセットのそれぞれの要素に対するデータソースを特定する一つ以上の記述子を格納させ；更に、前記記述子に従って、前記ストレージシステム内の要素の変更が許可されているか否かを判別する規則を処理させる；命令を含む。

20

【 0 0 0 7 】

複数の実施態様は以下の一つ以上の特徴を含むことができる。

【 0 0 0 8 】

規則の少なくとも一つは、少なくとも一つの記述子に従って、前記ストレージシステム内の要素を置換するためにソース要素のいずれかが使用されるべきか否かを判別する。

【 0 0 0 9 】

規則の少なくとも一つは、少なくとも一つの記述子に従って、前記ストレージシステム内のある要素に対応するソース要素が削除されていることに応じて同要素が削除されるべきか否かを判別する。

【 0 0 1 0 】

さらに、第1のデータソースによって提供されるソース要素と異なる一つ以上のデータのソース要素を提供する第2のデータソースがアクセスされる。

30

【 0 0 1 1 】

第1のソースデータは前記サブセットに対する前記データソースであり、第2のデータソースは前記サブセットの少なくとも一つの前記要素に対するデータソースである。

【 0 0 1 2 】

記述子の少なくとも一つは一つのサブセットに対する一つのデータソースを特定し、記述子の少なくとも他の一つはそのサブセットの一つの要素に対する他の一つのデータソースを特定する。

【 0 0 1 3 】

前記規則の少なくとも一つはユーザ入力を必要とする。

40

【 0 0 1 4 】

前記規則の少なくとも一つはユーザ入力なしに自動的に取扱われる。

【 0 0 1 5 】

前記規則を含むメンテナンス計画がアクセスされる。

【 0 0 1 6 】

前記メンテナンス計画はどのソース要素が前記ストレージシステム内のどの要素に対応するかを示すマッピングを含む。

【 0 0 1 7 】

前記第1のデータソースはユーザの直接的入力を受け取るユーザインターフェースを含

50

む。

【 0 0 1 8 】

前記データはレコードを含み、その各レコードは多数のフィールド内にてデータ要素を格納する。

【 0 0 1 9 】

記述子は所与のレコードの変更を制御するレコード所有者を特定する。

【 0 0 2 0 】

前記レコード所有者は前記所与のレコードの削除を制御する。

【 0 0 2 1 】

前記レコード所有者は前記所与のレコードを生成した者である。

10

【 0 0 2 2 】

記述子は、所与のレコードの所与のフィールド内に格納されたデータ要素への変更を制御するフィールド所有者を特定する。

【 0 0 2 3 】

前記フィールド所有者は前記所与のフィールドに格納された前記データ要素への更新を制御する。

【 0 0 2 4 】

ユーザインターフェースは、ユーザに提供され、且つ、そのユーザからの決定であって前記ストレージシステムのデータ要素に関係する決定、を受け取る。

【 0 0 2 5 】

20

実施態様は以下の一つ以上の利点を有する。

【 0 0 2 6 】

一つのデータメンテナンスシステムは多数のデータソースを取扱うことができる。多数のデータソースは、統一したデータ収集に寄与するデータに関して潜在的に重複し、且つ、構造において潜在的に種々の形をとる。前記データメンテナンスシステムは、前記システム内において要素に寄与する前記ソースの追跡を行うことができる。前記システムは、時間とともに異なるバージョンの要素に寄与する異なる複数のソースを管理する（取り扱う）ことができる。前記システムは、前記ストレージシステム内の前記要素のバージョンを修正するのにどのソース内容が用いられるべきかについての決定を行うことができる。さらに、要素がその権限のあるソースから削除されていれば、そのシステムは、この削除はデータの実質的変更を表すと決定することができる。

30

【 0 0 2 7 】

本発明の他の特徴及び利点は以下の説明及び特許請求の範囲の請求項から明らかになる。

【図面の簡単な説明】

【 0 0 2 8 】

【図 1】データメンテナンスシステムの概観図である。

【図 2】データの目的地（デスティネーション）の幾つかのコンテンツを示す線図（ダイアグラム、diagram）である。

【図 3】データの処理に関する構成要素の線図である。

40

【図 4】データの処理に関する構成要素の他の線図である。

【図 5】データメンテナンス計画を生成するシステムの線図である。

【図 6 A】データマネージャの作動のフローチャートである。

【図 6 B】データマネージャの一部分の作動のフローチャートである。

【図 6 C】データマネージャの他の部分の作動のフローチャートである。

【発明を実施するための形態】

【 0 0 2 9 】

概観

図 1 はデータメンテナンスシステム 10 の一例のブロック線図（ダイアグラム）であって、システム 10 の要素間の相互関係を示す。データマネージャ 100 は、データの目的

50

地（データデスティネーション）120内にロードすべきデータを提供する目的で多数の入力を受け取る。たとえば、データメンテナンスシステムは、メタデータの収納庫（保管所、repository）を実現するために用いられることができる。メタデータは、データ処理システムによって用いられて、種々のデータ処理プログラム、種々のデータソース、及び、これらのプログラムを用いるソースからのデータの種々の処理結果を記述する。この場合、データメンテナンスシステムは、種々のソースから情報を抽出してメタデータを形成することができ、且つ、データ処理システムの統一的ビュー及び種々の操作データ（オペレーショナルデータ）の処理の履歴を維持することができる。

【0030】

データマネージャ100への一つの入力は一エクストラクタ102によって到着する。このエクストラクタは、一つ以上のデータソース130、130a、130bからデータ132を獲得することができる。エクストラクタは一つ以上のそのようなソースをマネージャ100に提供する。

10

【0031】

他の入力データメンテナンス計画(plan)110である。殆どの場合、各データソース130は対応するメンテナンス計画を有する。そのメンテナンス計画はその特定のソースについて唯一であってもよく、或いは、幾つかのソースの間で共有されることができる。その計画は、ソースデータ要素のデスティネーションデータ要素へのマッピング112を含む。さらに、その計画は、変換規則114、検証(validation)規則116、及び、データの異なる処理の態様を取り扱う配置(disposition)規則118を含む。それらの規則は、たとえばソースにおける要素値のフォーマット中、一つの要素の一つのソースへの追加中、及び、要素のソースからの削除中等の特定のソースの処理中に発生し得る状況について取り組む(address)。さらに、規則は修正されるソースに依存してデータデスティネーションにおける要素の修正を許可又は拒否する。多数のメンテナンス計画110a、110b、110cは、長い時間にわたって多数回使用されることを目的として永続記憶装置(persistent storage)150の中に配置される。

20

【0032】

他の入力ユーザインターフェース140によって到着する。幾つかの状況において、マネージャ100は自立的に機能し、且つ、ユーザインターフェース入力を必要としない。他の状況において、ユーザ142は、たとえば、ボタンをクリックすることによりマネージャ動作を開始する。幾つかの状況において、また、マネージャ100は、意志決定点を超えて進行するためにユーザ入力を要求し、或いは、ユーザに提供する情報を有することができる。このような種々の場合において、ユーザインターフェース140はユーザ142とマネージャ100との間の導管(conduit)の機能を果たす。

30

【0033】

他の入力データデスティネーション120から到着する。データデスティネーション120はデータ構造の所与のタイプに対する所与のフォーマットに従ってデータ要素を格納する。たとえば、幾つかの実施態様においては、データ要素は多数のフィールドよりなるレコード122に格納された値に対応する。所与のレコード（たとえば、テーブルに格納されたレコードに対するテーブル中のセルのようなもの）に対する各フィールド値（即ち、「プロパティ」値）は、関連付けられたフィールド所有者を有することができる。このフィールド所有者は、そのフィールドに格納される要素に対する変更のタイプを決定するデータソースを特定する。たとえば、フィールド所有者からのデータは、そのフィールドに格納された要素を変更或いは削除する際に用いられることが許容される。さらに、各レコード（たとえば、テーブルに格納されたレコードに対するテーブル中の行）は、そのレコードに対する変更のタイプを決定するデータソースを識別するレコード所有者であって関連付けられたレコード所有者を有することができる。たとえば、レコードのレコード所有者はそのレコードを生成するデータソースであり得、フィールドのフィールド所有者はそのフィールドに対する初期値を供給したデータソースであり得る。レコード所有者はレコード及びそれに関連するフィールドの削除を制御することができる。この例では、レ

40

50

コードがレコード所有者として特定されたデータソースから削除されると、それに対応するレコードがデータデスティネーションから削除され得るか、さもなければ、レコード所有者はいつレコードを削除することを許されるかを決定できる。同様に、フィールド所有者はフィールド中の値の更新を制御することができる。幾つかの実施形態において、データ要素は、バージョン化データオブジェクト（たとえば、そこにおいて、データオブジェクトは、データオブジェクトの異なるバージョンを表わす多数のレコードに対応する。）内に格納されたデータに対応する。一旦、マネージャ 100 がデータデスティネーション 120 内のどの要素が修正のためにアクセスされるべきかを特定すると、マネージャ 100 は、各レコードのレコード所有者及び（随意的に）各要素のフィールド所有者を入力として受け取る。所与のレコードが所与のレコード所有者に関連付けられることができ、且つ、この所与のレコードの特定のフィールドが所与のフィールド所有者に関連付けられることができるので、この所与のレコードの所与のフィールド内に格納された所与の要素はレコード所有者として特定された一つのデータソース及びフィールド所有者として特定された他のデータソースを有する。

10

【0034】

その入力を受け取ると、マネージャ 100 はメンテナンス計画 110 をデータソース 130 から抽出されたデータ 132 に適用する。メンテナンス計画 110 はマッピング 112 を含み、マッピング 112 は、データソースから到着しているどのデータの要素がデータデスティネーション 120 内のどの要素に対応するかを示す。マネージャ 100 がどのデータがデータデスティネーション 120 内に置かれるべきかを決定すると、マネージャ 100 は変換規則、検証規則及び配置規則を各要素に適用する。配置規則は各要素のレコード所有者及びフィールド所有者を考慮して適用される。もし要素のソースがその要素のレコード所有者またはフィールド所有者に対応していなければ、特別の規則が一連の行動を決定するであろう。たとえば、フィールド所有者が、そのフィールドに変更または更新を提供しているデータソースと同じデータソースでなければ、配置規則は次に取るべき一連の行動を指示することができる。

20

【0035】

処理中、マネージャ 100 は、出力を状況メッセージの形式でユーザインターフェース 140 に提供し、且つ、所与のデータ要素の進め方をユーザが決定しなければならない場合にはさらに入力を受取ることができる。

30

【0036】

一旦、マネージャ 100 が規則をデータ要素に適用すると、マネージャ 100 はその要素をデータデスティネーション 120 内に置くことを目的としてその要素をロード 104 に提供する。

【0037】

データストレージ（データ格納）

データは多数のタイプのソースのいずれかからも到着する。しばしば、データは外部ソースから到着する。たとえば、データは外部のデータベースまたはスプレッドシートファイル内に格納されている。代替として、データは、データストレージシステム（たとえば、データデスティネーション 120 を実現するデータストレージシステム）によって提供されるユーザインターフェースを介してユーザによって手動で入力され得る。データメンテナンスシステムは、これらの列挙されたものを含む多種多様のタイプのソースから到着するデータを維持（メンテナンス）することができる。

40

【0038】

データは時間と共に変化する可能性が高い。従って、最新のデータストレージシステムを維持する（メンテナンス）ことの一部として、このシステムは各対応するソースにおけるデータの最新バージョンを用いて定期的に更新される。

【0039】

データデスティネーション 120 を実現するデータストレージシステムの一つの例は、データの集合体をデータベーステーブルの形式にて格納する。図 2 はデータベーステーブ

50

ルの一つの代表例を示す。このシステムは、別々のデータのサブセットに対応するレコード 2 2 2 a、2 2 2 b、2 2 2 c のようなレコードを格納する。さらに、そのレコードは、データのサブセットの各々の異なる特徴に対応する一つ以上のフィールド 2 2 4 a、2 2 4 b、2 2 4 c を含む。ストレージシステム内に含まれる一つの集合体におけるレコードの総ては同じソースから来ているかもしれないし、或いは、幾つかのレコードがある一つのソースから来ており且つ他のレコードは他のソースから来ているかもしれない。同様に、一つのレコード内のフィールドの総ては同じソースから来ているかもしれないし、或いは、異なるソースが同じレコードに対する異なるフィールドに寄与しているかもしれない。

【 0 0 4 0 】

多数のデータソースから来ているデータをメンテナンスする機能として、データメンテナンスシステムは各要素のデータソースを特定する。そのシステムは、この情報を用いて特定のデータソースが所与の要素の変更（たとえば、所与の要素の更新、修正、削除）に用いられるべきかを判定し、或いは、その代わりに、特定のデータソースがこの目的に不適切であるかを判定する。このことは、ユーザインターフェース 1 4 0 によるデータの直接的な編集を許可することを含むことができるが、その一方、所有者として特定されたデータソース（たとえば、そのシステムの外部のデータソースまたはユーザが最終決定権を持つ権限を有していないデータソース）を有する他のデータ要素に対するユーザインターフェースを用いた編集について警告すること又はその編集を禁止することを含むことができる。さらに、ユーザが外部にてメンテナンスされたデータを修正することを希望するのであれば、そのシステムは適切な外部のデータソースへの経路変更要求のためにそのデータソース情報を用いることができる。このソース情報を用いる幾つかの技術は、所有者形式の記述子のセットをメンテナンスすることを伴う。

【 0 0 4 1 】

たとえば、幾つかの実施形態において、各フィールドは関連したフィールド所有者 2 2 6 a、2 2 6 b、2 2 6 c を有する。そのフィールド所有者は、そのフィールド内に格納された要素を変更または削除する際に使用することが許可されるデータソースを特定する。幾つかの実施形態において、フィールド所有者は、そのフィールドに格納された要素に対する変更または更新を支配するデータソースである。たとえば、要素に内在しているデータが変化したかもしれないし、或いは、既に存在していたデータに加えて追加データがその要素に供給されているかもしれない。さらに、各レコードは関連したレコード所有者 2 2 8 を有する。そのレコード所有者は、そのレコードに対する変更を制御するデータソースを特定する。たとえば、いくつかの実施形態において、レコード所有者は、レコードが削除されることが許容されるかを制御する。所有者情報は、関連したデータレコードと一緒に、ストレージシステム内に格納され得る。マネージャがデータデスティネーション内のどの要素が修正のためにアクセスされるべきかを特定すると、そのマネージャは各要素のレコード所有者及び各要素のフィールド所有者を入力として受け取る。所与のレコードの所与のフィールド内に格納された所与の要素は、その所与のフィールドのフィールド所有者及びその所与のレコードの異なるレコード所有者に関連付けられ得る。また、レコード所有者及びフィールド所有者は同一のデータソースであることもできる。要素は、不明確な（undefined）フィールド所有者または不明確な（undefined）レコード所有者を有することもできる。

【 0 0 4 2 】

幾つかの実施態様は、各データ要素についての情報をさらに含むことができる。たとえば、データメンテナンスシステムは各要素の時間的な状態についての情報を保持することができる。そのシステムは、データデスティネーション内の要素がデータソースによって最後に修正された時刻を記録することができる。同様に、そのシステムは元のデータソースの要素がそのデータソース内にて最後に修正された時刻を記録することができる。

【 0 0 4 3 】

データマッピング

データメンテナンスシステムにデータを提供する各ソースに対して、ソースにおけるどのデータ要素がデスティネーションにおけるどのデータ要素に対応するかを特定するメカニズムがある。データデスティネーションを実現するデータストレージシステムは、所与のソースがデータを格納する配列とは通常異なる特別の配列にてデータを格納する。さらに、上述したように、完全なデータのセットは、しばしば多数のデータソースから来ている。データセットの所与の特徴を記述するいくつかのデータは一つのソース内に存在し、その同一のデータセットの異なる特徴を記述するいくつかのデータは別のソース内に存在し得る。これらの異なる各ソースのそれぞれは、それら自身のデータ配列の方法を有する。

【 0 0 4 4 】

データマネージャ 1 0 0 は、図 3 に示されているように、データソース内の各データの要素をデータデスティネーション 1 2 0 内の対応する要素にマッピングすることができる。これを達成するために、各データソース 3 3 0 は、それに関連したメンテナンス計画の構成要素として、関連付けられ且つ格納されたマッピング 1 1 2 を有する。マッピング 1 1 2 もメンテナンス計画規則 3 1 0 も、データマネージャ 1 0 0 の実行部 3 0 0 によって使用される。ソース要素 3 3 4 a、3 3 4 b がデータデスティネーションに含まれるように準備されると、格納されたマッピング 1 1 2 は、各要素がそのデータデスティネーション内のどこに配置されるべきであるかについての指図（案内）をマネージャ 1 0 0 に提供する。同様に、要素がより古い要素の新しいバージョンであれば、マッピング 1 1 2 は、どの古い要素が削除されて新しいバージョンによって置換されるのかについての指図（案内）を提供する。

【 0 0 4 5 】

一般に、マッピング 1 1 2 は、データソース 3 3 0 内のソースレコード 3 3 2、3 3 2 a、3 3 2 b と、データデスティネーション内のこれらに対応するレコード 3 2 2、3 2 2 a、3 2 2 b と、を特定する。しばしば、ソースレコード 3 3 2 内の各要素は、データデスティネーション内の対応する目標レコード 3 2 2 内に置かれる。代替として、データソース 3 3 0 内のデータの配列に依存してソースレコード 3 3 2 a 内の要素の一部は一つの目標レコード 3 2 2 a 内に置かれ、ソースレコード 3 3 2 a の他の要素の一部は別の目標レコード 3 2 2 b 内に置かれる。さらに、ソースレコード 3 3 2 の要素のすべてが目標レコード 3 2 2 にマップされる必要はない。

【 0 0 4 6 】

また、目標レコードは多数のデータソースからの要素を含むことがある。ある要素 3 2 4 a、3 2 4 b の起源は一つのソースであり、また、少なくとも一つの他の要素 3 2 4 c の起源は別のソースであることができる。一般に、データデスティネーションは多数のソースを特定する記述子を有する。レコード所有者 3 2 8 は、目標レコードが始めてデータデスティネーション内に生成されたときに目標レコードに含まれていた元々のデータ要素を提供したデータソースを示すことができる。レコードは不明確なレコード所有者を有する場合がある。幾つかの実施態様において、データソースがそのレコード所有者であると考慮されるか否かは配置規則によって制御される。また、これらの元々のデータ要素のフィールド所有者 3 2 6 a、3 2 6 b は、その元々のデータソースに対応することができる。仮に、要素 3 2 4 c の起源が別のソースであるならば、対応するフィールド所有者 3 2 6 c はその別のソースを示すことができる。随意に、別のソースに関連するマッピングは、その要素を含むレコード自身がデータデスティネーション内に存在せず、むしろその要素のサブセットのみがデータデスティネーションに存在するであろうことを示すことができる。さらに、要素 3 2 4 a、3 2 4 b、3 2 4 c の一つ以上は、レコードを一意的に特定することができ、そのレコード所有者に等しいフィールド所有者が割当てられるべきである。

【 0 0 4 7 】

データマネージャ 1 0 0 は、所有者情報を用いて所与の要素がこの所与の要素を変更しようとするインポートデータソースによって（たとえば、所有者情報をそのインポ

10

20

30

40

50

ーティングデータソースの身元（アイデンティティ、identity）と比較することによって）追加、削除または更新されるべきか否かを評価する。従って、これらの評価を行うために、所有者情報及びインポートデータソースの身元は入力として実行部 300 によって受け取られる。

【0048】

データ規則及び処理

データソース 330 に関連したメンテナンス計画 110 の他の構成要素は、規則 310 のセットである。これらの規則は、処理中のデータを変更して、そのデータをデータデスティネーションにロードするのに適切な状態にする。図 4 に示したように、各規則は、規則が影響するデータ要素の特徴により、変換規則 114、検証規則（validation rules）116 及び配置規則 118 のセットに属する。データマネージャ 100 は、入ってくるデータに実行部 300 を通して指示を与える。その実行部 300 は、各規則のセットに従って逐次的に作動を行う。変換規則実行部 404 は、入ってくるデータ 402 を入力として取り込み、変換済データを出力として提供する。次に、検証規則実行部 406 は、変換済データを入力として取り込み、検証済データを出力として提供するとともに、無効データに遭遇した場合には、エラー通知をユーザインターフェース 140 に提供する。最後に、配置規則実行部 408 は、検証済データ、そのデータのソースと関連するマッピング 112、及び、入ってくるデータ要素に関係するレコード及びフィールド所有者記述子 412 を入力として取り込む。実行部 404、406、408 の夫々は、処理すべき対応する規則がない場合にはいかなる動作をも実行することなくデータが通過できるという点で随意（optional）なものである。また、実行部 404、406、408 の順序は他の実施態様では異ならせることができる。たとえば、状況によっては、ユーザの決定が配置規則 118 の一つによって要求されていれば、配置規則実行部 408 は、ユーザインターフェース 140 に出力を提供し且つそのユーザインターフェースから入力を受け取ることができる。この実行部 408 は、データが最終的にデータデスティネーション 120 に含まれるように完全に処理されたデータ 410 を出力する。

【0049】

変換規則

要素を配列する際の上述の相違に加えて、所与のデータソースは特別なフォーマットにて個々の要素を格納することができ、その一方、データデスティネーションは同一の要素を異なるフォーマットにて格納することができる。データマネージャ 100 は、処理を行いながら、この要素を第 1 のフォーマットから第 2 のフォーマットへ自動的に変換することができる。たとえば、データソースは総て大文字の文字列を格納することができ、一方、データデスティネーションは理論上総て小文字の文字列を格納する。データマネージャ 100 は、変換規則を適用してこれらの自動的変換を実行する。変換規則は、一つの特定のソースからの一つの特定の要素に適用でき、或いは、様々なソースからの様々な要素にも適用できる。

【0050】

検証規則

このメカニズムは、また、データ要素が、データデスティネーションに含ませるのに適切であるか否かを判別する。あるデータは、意味があるためには、あるフォーマットであって或いはある範囲内にあることを必要とする。たとえば、暦の日付を表わすデータ要素は、月、日、年に対応する構成要素を有しなければならない。また、月は 1 - 12 の範囲内、日はたとえば 1 - 31 の範囲内でなければならない。データマネージャ 100 は、検証規則を処理中の各要素に適用し、その要素がそのコンテキスト（前後関係、文脈）において有効な形式であるか否かを判別する。検証規則は、一つの特定のソースからの一つの特定の要素に適用でき、或いは、様々なソースからの様々な要素にも適用できる。

【0051】

配置規則及びソース管理

時間に伴って、データソースの構成は大いに（significant fashion）変化し得る。当

然のことながら、要素は時間と共に変化することが予期されている。さらに、要素は追加及び削除される。各ケースは、データソースのコンテキストに依り、データソース内での異なった成り行きに対応する。データソースの要素がデータストレージシステム内に含まれる要素に対して単純に変化した場合には、その要素に関連した現実の情報もまた変化した可能性が高く、従って、その要素は単純にそれに応じて更新されている。しかしながら、多数のデータソースが同一のデータのセットに寄与しているときは、この特定のデータソースはデータデスティネーション内で現在表わされているよりも古いバージョンのデータを表わすかもしれない。

【 0 0 5 2 】

さらに、データ要素が関係するデータがもはや存在せず、よって、その要素が意図的に削除されているとの理由から、そのデータ要素が見つからない場合がある。或いは、そのデータ要素が異なるデータソースによって寄与された、もしくは、データ要素が異なるデータソースに移送されてしまったかも知れない。または、そのデータ要素はこのデータソースに依然として存在するが、なにか別の身元 (identity) の下にある場合もある。

【 0 0 5 3 】

同様に、データ要素の追加は、要素の削除に関係したシナリオの一つと同等であるシナリオを表わし得る。その追加された要素はデータについての新しい情報であることもあり、或いは、その追加された要素は新しい場所または新しい身元 (identity) の下で今現在存在する現存のデータの一部であることもある。

【 0 0 5 4 】

データマネージャ 100 は配置規則を用いてデータソースの構成におけるこれらのタイプの総ての変更を取り扱うことができる。各規則は特定の変更を取り扱う。ある規則は予め定められた動作、たとえばデータストレージシステムから一つの要素を削除する動作、を自動的に開始することができる。或いは、ある規則は、ユーザによるさらなる意思決定を促すことができる。その際、配置規則は自動的に取り扱われ得るシナリオとユーザからの入力を受け取るシナリオとを区別する。

【 0 0 5 5 】

また、配置規則は、多数のデータソース間の優先衝突 (priority conflict) を取り扱うことができる。特に、ある配置規則は、適切な一連の動作を決定するために所与の要素の所有者を頼りにする。所有権情報が配置規則により用いられているときには、要素の所有権に従って、配置規則によって許される場合にデータデスティネーション内の特定の要素を更新または削除するためにデータソースのみが使用される。たとえば、幾つかの場合において、要素が更新を実行する際に使用されるべきであれば、データソースはその要素の所有者として定義されなければならない。そのデータソースが所有者でなければ、データマネージャ 100 はその要素に関連付けられた配置規則を適用して正しい一連の動作を取る。その規則は、その要素が更新されるべきでないことを示してもよい。代替として、その規則は、このデータソースにはその要素の新しい所有者を割当てべきであることを指示し、その結果、その更新が進行する。他のタイプの規則は、フィールド所有者が不明確であるならば、たとえそのデータソースがそのフィールド所有者でない場合であっても、そのデータソースを用いてそのフィールドを更新することを指示する。

【 0 0 5 6 】

また、配置規則は要素所有権の変更を開始することができる。たとえば、データソースが所定の時間後に要素の修正にもはや使用されないであろう場合、配置規則は、その要素の所有者が他のデータソースに設定されるべきである、或いは、他のデータソースが所有権を主張するようにその要素の所有者を不明確のままにしておくことを指示することができる。

【 0 0 5 7 】

配置規則は、一つ特定のソースからの一つ特定の要素に適用でき、或いは、様々なソースからの様々な要素にも適用できる。

【 0 0 5 8 】

メンテナンス計画の設立

データストレージシステムは所与のデータソースに関連した所与のメンテナンス計画を生成するメカニズムを有する。図5はこの機能性を提供するシステムの部分の一例を示す。ユーザ542はユーザインターフェース540にアクセスして計画を生成する。このユーザはデータマネージャ100を操作するユーザ142と同一であることができる。同様に、そのユーザはインターフェース540は、データマネージャ100と相互に作用するように用いられるユーザインターフェース140の構成要素でもあり得る。

【0059】

ユーザインターフェース540は、変換規則114、検証規則116及び配置規則118、並びにマッピング112を定義するために使用される。これらは、共に、メンテナンス計画110を備え、そのメンテナンス計画110は、存在しているメンテナンス計画110a、110b、110cと共に永続記憶装置(persistent storage)150内に置かれる。メンテナンス計画110は、テンプレートストレージ550内のデータストレージシステムに提供されている幾つかのメンテナンス計画テンプレート510a、510b、510cの一つに随意に基づくことができる。これらのテンプレートはデータソースファイルのフォーマットの共通の形式に対応する。代替として、メンテナンス計画110はユーザインターフェース540内で新たに生成されることができる。

【0060】

作動例

データマネージャの作動シーケンスの可能性のある一つが、図6A - 図6Cに示されている。図6Aを参照すると、ステップ601では、処理されるべきデータソースが特定(識別、identify)される。ステップ603では、マネージャは、このソースが既存のメンテナンス計画を有するか否かを判定する。もし、既存のメンテナンス計画がなければ、ステップ605にてマネージャは計画が生成されるまで動作を遅延する。もしメンテナンス計画があれば、ステップ607にてマネージャはそのメンテナンス計画を永続記憶装置から獲得する。ステップ609において、マネージャは、データソース内のデータを獲得するのに使用するエクストラクタ(抽出子、extractor)を特定する。このエクストラクタは、メンテナンス計画により、または、他のメカニズムにより、規定される。ステップ611において、マネージャはそのエクストラクタからデータソースを受け取る。ステップ613において、マネージャはそのメンテナンス計画からのマッピングを用いてデータソース内のどのレコードがデータデスティネーション内のどのレコードに関係するかを決定する。ステップ615において、データソース及びメンテナンス計画に従って、データデスティネーション内において各関連するレコードが追加または更新される。ステップ615は、一つのレコードに対応する一つの反復を伴いながら、多数の反復として発生し得る。ステップ617では、データソース及びメンテナンス計画によって命令されるのであれば、データデスティネーション内のレコードは削除のために特定される。このステップもまた多数回の反復される形式で発生し得る。次に、ステップ619において、たとえば、ステップ615において、変更されたデータがロードされていなければ(たとえば、操作されているレコードがキャッシュドコピーまたは一時的コピーであれば)、マネージャは、その変更されたデータをデータデスティネーション内にロードするために使用されるべきローダを随意に特定する。ローダは、メンテナンス計画または他のメカニズムによって規定される。最後に、ステップ621において、追加、更新または削除されたレコードがデータデスティネーションにおいて更新される。

【0061】

図6Bは、ステップ615の反復615aについての詳細な例、即ち、データデスティネーションにおいてレコードを追加する或いはレコードを更新する処理を示す。ステップ631において、マネージャは、データソースがデータデスティネーション内のレコードに対応する可能性のあるレコードを有するか否かを判別するためのマッピングを用いてレコードを探すためにデータソースをチェックする。ステップ633において、マネージャは検証規則及び変換規則をそのレコードの内容に適用する。ステップ635では、マネー

10

20

30

40

50

ジャは、メンテナンス計画内のマッピングに基づいて、データデスティネーションがこのソースレコードに対応するレコードを有するか否かを判別する。データデスティネーションに対応するレコードを有しないときには、マネージャはステップ637にて新しいレコードを生成する。幾つかの場合には、マネージャはステップ637にてレコード所有者として設定されるこのデータソースとともに新しいレコードを生成する。幾つかの場合において、そのレコード所有者は不明確となるか、または、別のデータソースがレコード所有者として設定されるであろう。マッピングは、レコードは既に存在すべきであるが、レコード所有者として異なるソースを有し、従って、マネージャはその異なるソースをレコード所有者のままにしておく選択を有することを指示する場合もある。ステップ639において、マネージャは、フィールド所有者として設定されたデータソースとともに、そのデータソースレコード内の第1のフィールドに基いてそのレコードの第1のフィールドを生成する。ステップ641では、マネージャはソースレコード内に次のフィールドがあるか否かを調べる。これは新しく生成されたデスティネーションレコードであるので、ステップ643においてマネージャはデータデスティネーションには対応するフィールドはないと決定する。このレコードに対して総てのフィールドが生成されるまで、作動シーケンスはステップ639に戻る。また、フィールドが存在すべきであるが、データがマネージャによって未だ獲得されていないことをマッピングが示すならば、マネージャは、プレースホルダ (placeholder) フィールドを生成することができる。

【0062】

ステップ635において、データデスティネーション内に対応するレコードがあるとマネージャは判別したときには、マネージャはステップ641にてソースレコード内のフィールドを処理のために特定する。ステップ643において、マネージャはデスティネーションレコード内に対応するフィールドがあるか否かを判別する。対応するレコードがない場合、マネージャはあたかもレコードが新しい場合と類似の手続きを行い、即ち、そのフィールド所有者としてのこのデータソースを伴う新しいフィールドを生成し、そのソースフィールドの内容を含む。ステップ643において、マネージャが、データデスティネーション内に対応するフィールドがあると判別する場合、そのマネージャはステップ645において関連するレコード所有者及びフィールド所有者を回収する。ステップ647において、マネージャは、このデータソースがフィールド所有者であるか否か、または、そのフィールド所有者が現時点において不明確であるか否かを確かめる。もしそうであれば、ステップ649にて、マネージャはそのフィールドの内容をそのソースからのデータに置換し、ステップ641に戻る。しかしながら、そのデータソースがそのフィールド所有者でなければ、マネージャの幾つかの実行は、ステップ651にて、そのデータソースがそのレコード所有者か否かを確かめる。もしそうであれば、ステップ653にて、マネージャはフィールド所有者をこのデータソースに更新する。この状況において、レコード所有者はこのフィールドの所有者を推定上支配し(引き継ぎ)、それに応じて、このフィールド所有者を自分自身のデータをもって更新する。データソースは、また、不明確なレコード所有者を有するレコードの所有権を主張していることもある。次に、マネージャはステップ649にて、フィールド内容を置換する。しかしながら、このデータソースが、レコード所有者でもフィールド所有者でもなければ、マネージャはステップ655にて配置規則を適用してどのような操作を更に行うかを決定する。加えて、幾つかの実施態様において、マネージャはレコード所有者を用いずにフィールドの内容を更新する。このような実施態様において、データソースがフィールド所有者でなければ、マネージャは、ステップ651を飛ばしてステップ655にて配置規則を適用する。配置規則の適用は、ユーザインターフェースを介してユーザの決定を尋ねることを含む他の多くのサブステップを要求することができる。多数のデータソースの間の優先衝突を取扱うために、配置規則は、また、既存のフィールド所有者とフィールドを更新するときの現在のデータソースと間の相対的優先度を比較するために使用され得る。ステップ657において、マネージャは配置規則に従って、適切な動作を行う。もし、その規則が、このデータソースがフィールドの修正をおこなうことを許可するのであれば、マネージャはステップ653にてフィールド所

10

20

30

40

50

有者を更新し、ステップ 6 4 9 にてフィールドの内容を置換する。このデータソースに対して定義された特定の配置規則、または、マネージャの構成に依存して、幾つかの実施態様は、フィールド所有者を変更することを差し控え、あるいは代替として、そのレコード所有者を更新する。もし、配置規則が、このデータソースがフィールドの修正をおこなうことを許可しないのであれば、ステップ 6 5 9 においてマネージャはフィールドを更新する際の動作を行わずにステップ 6 4 1 に戻る。ステップ 6 5 9 は、エラーメッセージまたはエラーログエントリを随意に含むことができる。

【 0 0 6 3 】

図 6 C は、ステップ 6 1 7 の反復 6 1 7 a についての詳細な例、即ち、データデスティネーションにおけるレコードまたはフィールドの削除の処理を示す。ステップ 6 7 1 において、マネージャは、データデスティネーションが、データソースから潜在的に行方不明となっているレコードまたはフィールドに対応するレコードまたはフィールドを有するかどうかを判別するためのマッピングを用いてレコードを探すためにデータデスティネーションをチェックする。ステップ 6 7 3 において、マネージャは、データデスティネーション内に存在する所与のレコードがデータソースにも存在するか否かを判別する。もし、そうでなければ、ステップ 6 7 5 において、マネージャはレコード所有者を確認する。もし、このデータソースがレコード所有者であれば、ステップ 6 7 7 にて、そのレコードはデータデスティネーションにおける削除の候補となる。たとえば、これは、レコードはデータソース内に存在していたが、最近削除されたことを示し、従って、データデスティネーションはそれに応じて更新されるべきであることを示す。もし、このデータソースがレコード所有者でなければ、ステップ 6 7 9 にてレコードは削除されない。たとえば、これは、レコードがこのレコードの所有者である他のデータソース内に依然として存在していることを示す。

【 0 0 6 4 】

ステップ 6 7 3 において、マネージャが、データデスティネーション内のレコードがデータソース内において対応するレコードを有していたと判別した場合、マネージャはステップ 6 8 1 においてデスティネーションレコード内のフィールドを特定し且つアクセスする。ステップ 6 8 3 においては、マネージャは、データソース内に対応するフィールドがあるか否かを判別する。もし、そうであれば、マネージャはステップ 6 8 5 にてデスティネーションフィールド上の動作を延期する。しかしながら、マネージャが、デスティネーションフィールドに対応するソースフィールドがないと判別した場合、マネージャはステップ 6 8 7 にてフィールド所有者を確認する。もし、このデータソースがデスティネーション要素のフィールド所有者として指定されていれば、そのフィールドはデータソース内に存在していないけれども、マネージャはステップ 6 8 9 にてフィールド所有者としてのこのデータソースを削除する。フィールド所有者のセットがない場合、システムは、後に、このフィールドの内容を削除でき、或いは、他のデータソースが後にこのフィールドの所有権を主張するとともにそのフィールドの内容を更新することができる。このデータソースがそのフィールド所有者でなければ、マネージャはステップ 6 8 5 にてそのフィールド上の動作を延期する。

【 0 0 6 5 】

典型的な実施例

上述したデータメンテナンスのアプローチは、コンピュータ上で実行されるソフトウェアを用いて実現され得る。たとえば、そのソフトウェアは、一つ以上のプログラムされたまたはプログラム可能なコンピュータシステム（分散型、クライアント/サーバ型、または、グリッド型等の種々アーキテクチャ型である）上で実行される一つ以上のコンピュータプログラムにおける手続を形成する。その各コンピュータシステムは、少なくとも一つのプロセッサ、少なくとも一つの記憶システム（揮発性メモリ、及び不揮発性メモリ、及びまたは記憶素子を含む。）、少なくとも一つの入力装置またはポート、及び、少なくとも一つの出力装置またはポートを含む。そのソフトウェアは、たとえば、計算グラフの設計及び構成に関係した他のサービスを提供するより大きなプログラムの一つ以上のモジュ

ールを形成できる。グラフのノード及び要素は、コンピュータ読み出し可能媒体に格納されたデータ構造、または、データ貯蔵庫（保管所）に格納されたデータモデルに合致する他の系統的なデータとして実現され得る。

【0066】

そのソフトウェアは、汎用または専用のプログラマブルコンピュータによって読み出し可能なCD-ROM等の記憶媒体上に提供され、或いは、そのソフトウェアが実行されるコンピュータにネットワークの通信媒体を用いて（伝播信号に符号化されて）配信される。すべての機能は、専用のコンピュータ上で、または、コプロセッサのような専用ハードウェアを用いて実行され得る。ソフトウェアは、ソフトウェアによって特定された異なる計算の部分が異なるコンピュータによって実行されるという分散方式によって実行され得る。好ましくは、上述の各コンピュータプログラムは、汎用または専用プログラマブルコンピュータによって読み出し可能な記憶媒体またはデバイス（たとえば、固体メモリまたは媒体、もしくは磁氣的または光学的媒体）に格納され或いはダウンロードされ、記憶媒体またはデバイスがコンピュータシステムによって読み出されて上述した手順を実行するときに、コンピュータを構築及び動作させる。また、本発明のシステムは、コンピュータプログラムとともに構築されたコンピュータ読み出し可能記憶媒体として実現されることが考えられる。ここで、そのように構築された記憶媒体は、上述の機能を実行するために特別且つ予め定義された様式でコンピュータシステムを作動させる。

10

【0067】

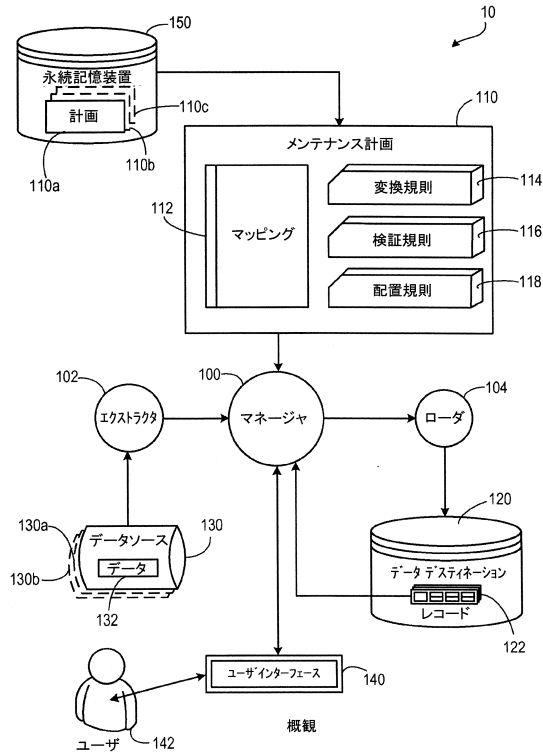
多くの本発明の実施例を記載した。しかしながら、本発明の精神及び範囲から逸脱することなく多種の変更がなされ得ることが理解されよう。たとえば上述のいくつかのステップは、順序が独立であり、従って上述の順序と異なる順序で実行され得る。

20

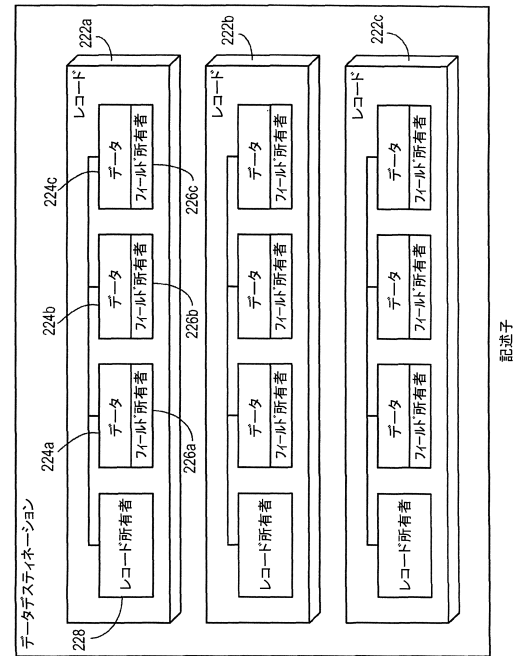
【0068】

上述の記述は説明のためであり、添付された本発明の特許請求の範囲によって定義される発明の範囲を限定する意図はないことが理解されるべきである。たとえば、上述の機能ステップの多くは、全体の処理に実質的に影響を及ぼすことなく異なる順序で実行され得る。他の実施例は特許請求の範囲の請求項の範囲内にある。

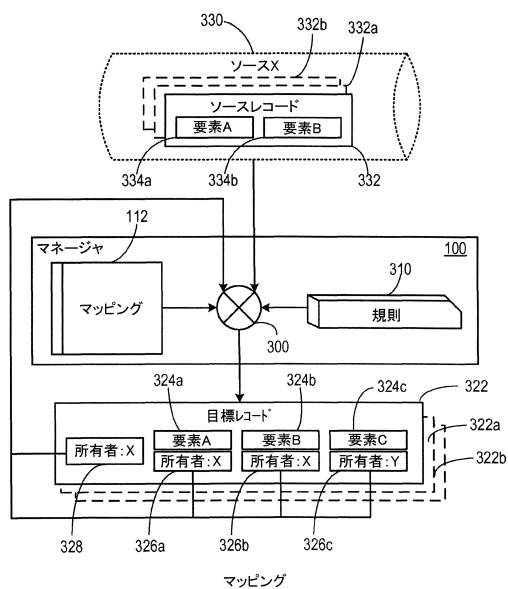
【図 1】



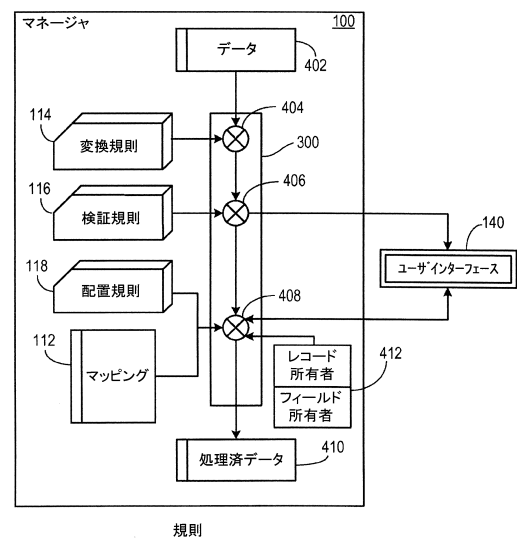
【図 2】



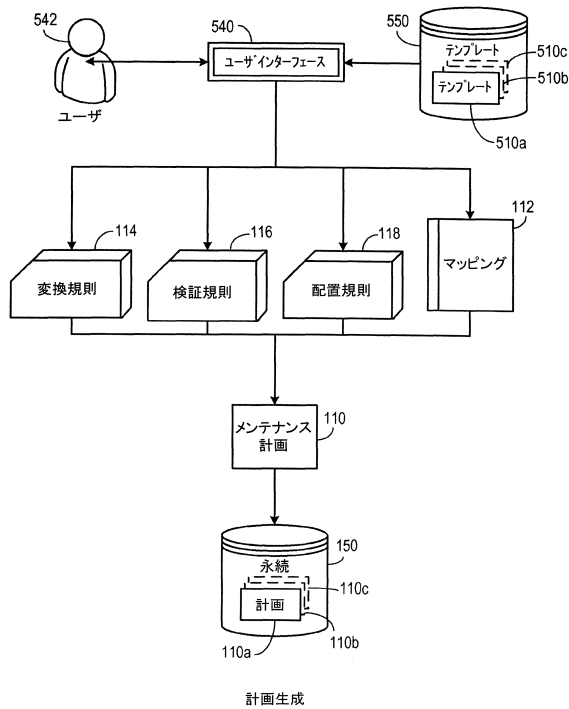
【図 3】



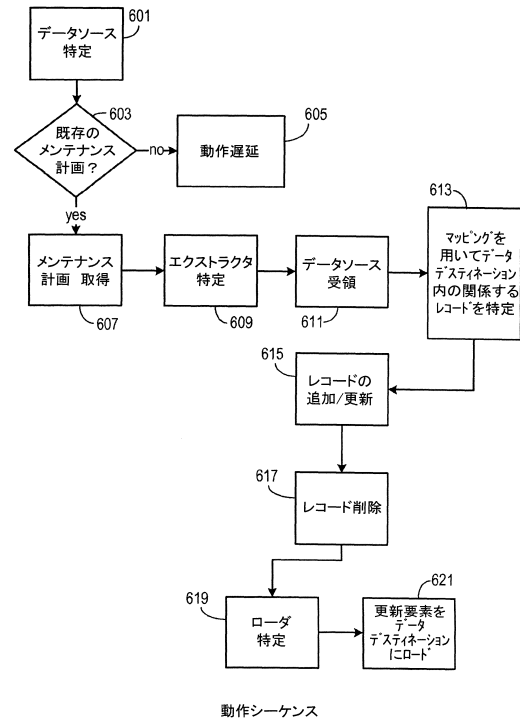
【図 4】



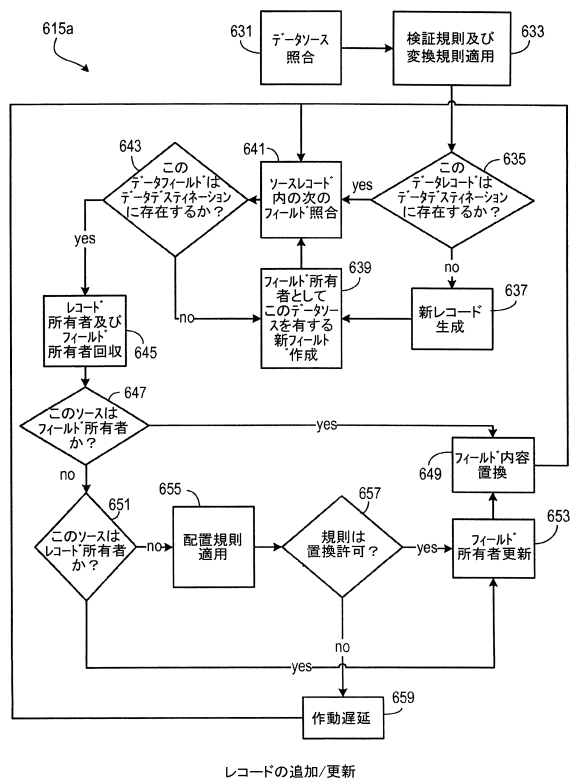
【図 5】



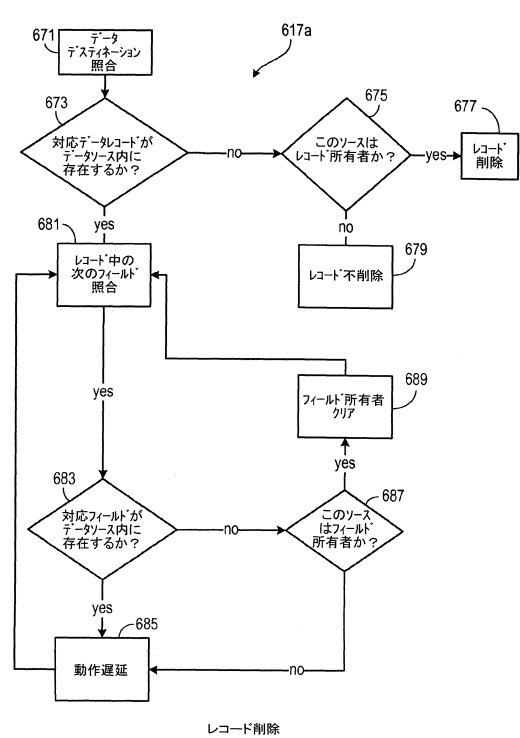
【図 6 A】



【図 6 B】



【図 6 C】



フロントページの続き

(72)発明者 イェラカリス アンソニー

アメリカ合衆国 02460 マサチューセッツ州 ニュートン ワイルドウッド アベニュー
67

(72)発明者 グールド ジョエル

アメリカ合衆国 01890 マサチューセッツ州 ウィンチェスター ソーンベリー ロード
50

審査官 田川 泰宏

(56)参考文献 特開2008-165447(JP,A)

国際公開第2007/083371(WO,A1)

特開昭62-219068(JP,A)

特開2006-285571(JP,A)

特開昭61-196339(JP,A)

国際公開第00/029952(WO,A2)

国際公開第2006/132740(WO,A2)

(58)調査した分野(Int.Cl., DB名)

G06F 12/00、17/30