

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2006-12173

(P2006-12173A)

(43) 公開日 平成18年1月12日(2006.1.12)

(51) Int.C1.

GO6F 17/30

(2006.01)

F 1

GO6F 17/30

GO6F 17/30

330B

180D

テーマコード(参考)

5B075

審査請求 未請求 請求項の数 25 O L (全 32 頁)

(21) 出願番号 特願2005-183544 (P2005-183544)
 (22) 出願日 平成17年6月23日 (2005.6.23)
 (31) 優先権主張番号 10/877,238
 (32) 優先日 平成16年6月25日 (2004.6.25)
 (33) 優先権主張国 米国(US)

(71) 出願人 390009531
 インターナショナル・ビジネス・マシーンズ・コーポレーション
 INTERNATIONAL BUSINESS MACHINES CORPORATION
 アメリカ合衆国10504 ニューヨーク
 州 アーモンク ニューオーチャード
 ロード
 (74) 代理人 100086243
 弁理士 坂口 博
 (74) 代理人 100091568
 弁理士 市位 嘉宏
 (74) 代理人 100108501
 弁理士 上野 剛史

最終頁に続く

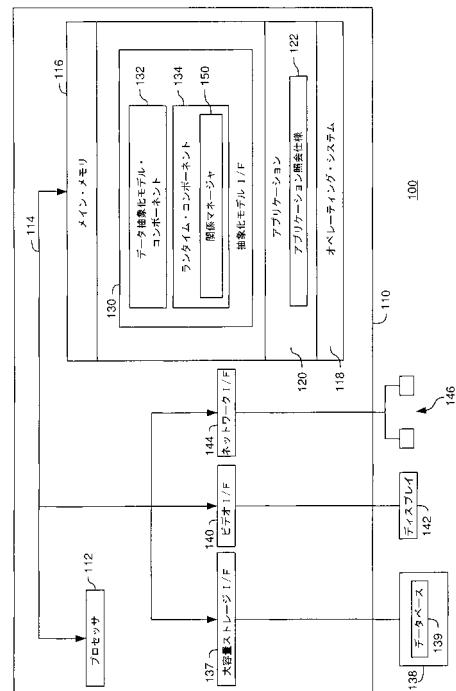
(54) 【発明の名称】データ抽象化モデルにおける関係管理

(57) 【要約】

【課題】 データベース内のデータを抽象的に記述するデータ抽象化モデルにおける関係を管理するための方法、システム、及び製品を提供すること。

【解決手段】 1つの実施形態は、この関係に基づいてデータベース内のデータの論理分岐をリンクする方法を提供する。本方法は、データベース内のデータを抽象的に記述するデータ抽象化モデルにアクセスすることを含む。データ抽象化モデルは、複数の論理分岐を有する論理ツリー構造を定める。異なる論理分岐の論理フィールド間の関係が特定され、特定された関係を抽象的に記述する論理リンクが生成される。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

データ抽象化モデルにおいて、データベース内の物理フィールドに対応する論理フィールドの間の関係を管理する方法であって、

幾つかが共通名を共有する論理フィールドを含む前記データ抽象化モデルの論理分岐間のリンクを伴う構造を提供するステップであって、前記リンクが、多数の論理フィールドによって共有される共通名に対する参照を含む抽象照会を実行するときに、前記物理フィールドを含むデータ構造の適切な連結を可能にするステップ、
を有する方法。

【請求項 2】

各々の論理分岐が前記データ抽象化モデルのカテゴリー仕様によって定められる、請求項 1 に記載の方法。

【請求項 3】

対応するカテゴリー仕様を持つ少なくとも 1 つの生成された論理リンクを含むステップを有する、請求項 2 に記載の方法。

【請求項 4】

前記データ抽象化モデルが複数の論理フィールド仕様を含み、前記論理フィールド仕様の各々が特定の論理フィールドを定める方法であって、前記複数の論理フィールド仕様のうちの対応する論理フィールド仕様を持つ少なくとも 1 つの生成された論理リンクを含むステップをさらに有する、請求項 2 に記載の方法。

【請求項 5】

各々の論理リンクが、前記データベースにおけるデータの物理表現の、異なるデータ構造間のパスを定める、請求項 1 に記載の方法。

【請求項 6】

前記物理表現が関係(リレーションナル)表現である、請求項 5 に記載の方法。

【請求項 7】

各々のデータ構造が前記関係表現の表である、請求項 6 に記載の方法。

【請求項 8】

各々のデータ構造が前記関係表現の表における列である、請求項 6 に記載の方法。

【請求項 9】

複数の論理分岐の各々について、前記データベースにおけるデータの物理表現の、対応するデータ構造を特定するステップと、

前記特定されたデータ構造が互いに関連するかどうかを判断するステップと、
を有する、請求項 1 に記載の方法。

【請求項 10】

データ抽象化モデルによって論理的に表現された物理データについて問い合わせを行う照会を生成する方法であって、

各々が前記データ抽象化モデルの論理フィールド仕様に対応する 1 つ又は複数の結果フィールドを有する、データベース内の物理データに対する抽象照会を受け取るステップと、

前記抽象照会に含まれる論理フィールド間の関係を定める関連論理リンクを、前記データ抽象化モデルが含むかどうかを判断するステップと、

前記データ抽象化モデルが関連論理リンクを含む場合に、

前記関連論理リンクを取得するステップと、

前記抽象照会を、前記物理データに対して実行することが可能な実行可能照会に変換するステップであって、前記変換が、前記データ抽象化モデルと前記取得した関連論理リンクとを用いて行われるステップと、
を有する方法。

【請求項 11】

前記データ抽象化モデルが関連論理リンクを含むかどうかを判断するステップが、前記

10

20

30

40

50

抽象照会の各々の結果フィールドについて、対応する論理フィールド仕様が論理リンクを含むかどうかを判断するステップを有する、請求項 10 に記載の方法。

【請求項 12】

前記データ抽象化モデルが前記関連論理リンクを含むかどうかを判断するステップが、前記抽象照会の前記結果フィールドの 1 つ又は複数が、論理リンクを含むカテゴリー仕様に含まれるかどうかを判断するステップを有する、請求項 10 に記載の方法。

【請求項 13】

各々の論理リンクが、前記データベースにおける前記物理データの関係表現の、異なるデータ構造間のパスを定める、請求項 10 に記載の方法。

【請求項 14】

前記実行可能照会が S Q L 照会である、請求項 13 に記載の方法。

【請求項 15】

前記取得した関連論理リンクが、前記 S Q L 照会において、前記異なるデータ構造をリンクする J O I N ステートメントとして表現される、請求項 14 に記載の方法。

【請求項 16】

データベース内のデータを抽象的に記述しており、その各々が前記データの特定のセットの論理表現を定めるデータ抽象化モデル、の論理フィールドをリンクさせる方法であって、

前記データ抽象化モデルについての論理ツリー構造を生成するステップであって、前記論理ツリー構造が複数の論理分岐を持ち、その各々が 1 つ又は複数の論理フィールドを含むステップと、

異なる論理分岐間の関係を特定するステップと、

前記特定された関係を抽象的に記述する論理リンクを生成するステップと、

前記生成された論理リンクを前記データ抽象化モデルと関連付けるステップと、
を有する方法。

【請求項 17】

プロセッサによって実行されるとき、データ抽象化モデルによって論理的に表現された物理データについて問い合わせを行う照会の生成プロセスを実施するプログラム、を含むコンピュータ可読媒体であって、前記プロセスが、

データベース内の物理データに対する抽象照会を受け取るステップであって、前記抽象照会の論理フィールドが前記データ抽象化モデルによって定められるステップと、

前記抽象照会に含まれる論理フィールドと関連付けられた論理リンクを前記データ抽象化モデルが含むかどうかを判断するステップであって、前記論理リンクが関連論理フィールドと他の論理フィールドとの間の関係を定めるステップと、

前記データ抽象化モデルが関連論理リンクを含む場合に、

前記関連論理リンクを取得するステップと、

前記抽象照会を、前記物理データに対して実行することが可能な実行可能照会に変換するステップであって、前記変換が、前記データ抽象化モデルと前記取得した関連論理リンクとを用いて行われるステップと、

を有するコンピュータ可読媒体。

【請求項 18】

前記データ抽象化モデルが前記関連論理リンクを含むかどうかを判断するステップが、前記抽象照会の各々の結果フィールドについて、対応する論理フィールド仕様が論理リンクを含むかどうかを判断するステップを有する、請求項 17 に記載のコンピュータ可読媒体。

【請求項 19】

前記データ抽象化モデルが前記連論理リンクを含むかどうかを判断するステップが、前記抽象照会の前記結果フィールドの 1 つ又は複数が、論理リンクを含むカテゴリー仕様に含まれるかどうかを判断するステップを含む、請求項 17 に記載のコンピュータ可読媒体。

10

20

30

40

50

【請求項 20】

各々の論理リンクが、前記データベースにおける前記物理データの関係表現の、異なるデータ構造間のパスを定める、請求項17に記載のコンピュータ可読媒体。

【請求項 21】

前記実行可能照会がSQL照会である、請求項20に記載のコンピュータ可読媒体。

【請求項 22】

前記取得した関連論理リンクが、前記SQL照会において、前記異なるデータ構造をリンクするJOINステートメントとして表現される、請求項21に記載のコンピュータ可読媒体。

【請求項 23】

プロセッサによって実行されるとき、データ抽象化モデルによって論理的に表現された物理データについて問い合わせを行う照会の生成プロセスを実施するプログラム、を含むコンピュータ可読媒体であって、前記プロセスが、

前記データ抽象化モデルを表現する論理ツリー構造をユーザが検索することを可能にするグラフィカル・ユーザ・インターフェースを表示するステップであって、前記論理ツリー構造が複数の論理分岐を有し、少なくとも2つの論理分岐がそれらの間の関係を定める論理リンクによってリンクされ、かつ、各々の論理分岐が前記データ抽象化モデルの1つ又は複数の論理フィールドを含み、各々の論理フィールドが前記物理データの特定のセットの論理表現を定めるステップ、

を有するコンピュータ可読媒体。

10

20

【請求項 24】

前記プロセスが、

抽象照会について照会条件を定めるために1つ又は複数の論理フィールドを選択するステップと、

前記抽象照会についての結果フィールドとして1つ又は複数の論理フィールドを選択するステップと、

のうちの少なくとも一方のために、前記グラフィカル・ユーザ・インターフェースを用いる前記論理ツリー構造のユーザによる検索を追跡するステップをさらに含む、請求項23に記載のコンピュータ可読媒体。

【請求項 25】

30

前記プロセスが、

追跡されたユーザによる検索に基づいて、リンクされた論理分岐内を前記ユーザが検索するかどうかを判断するステップと、

リンクされた論理分岐内を前記ユーザが検索する場合に、

前記データ抽象化モデルから対応する論理リンクを取得するステップと、

前記抽象照会を、前記物理データに対して実行することが可能な実行可能照会に変換するステップであって、前記変換が、前記データ抽象化モデルと前記取得した対応する論理リンクとを用いて行われるステップと、

をさらに有する、請求項24に記載のコンピュータ可読媒体。

【発明の詳細な説明】

40

【技術分野】**【0001】**

本発明は、一般に、データ抽象化モデルを管理することに関し、より具体的には、データベースのデータを抽象的に記述するデータ抽象化モデルにおける関係を管理することに関する。

【背景技術】**【0002】**

データベースは、電子化された情報の保管・検索システムである。最も一般的な形式のデータベースは、関係データベース、すなわち、多くの異なる方法で再構成し、アクセスできるようにデータを定めた表データベースである。分散データベースは、ネットワーク

50

内の異なる箇所の間で分散又は複製することができるデータベースである。オブジェクト指向プログラミング・データベースは、オブジェクト・クラス及びサブクラスで定義されたデータと一致するデータベースである。

【0003】

特定のアーキテクチャにかかわらず、DBMSは、要求エンティティ（例えば、アプリケーション、オペレーティング・システム、又はエンドユーザー）についての様々な異なるタイプの動作をサポートするように構成することができる。こうした動作は、DBMSによって保管され、管理されている情報を検索し、追加し、修正し、削除するように設定することができる。標準的なデータベース・アクセス方法は、構造化照会言語（SQL）のような高水準の照会言語を用いて、こうした動作をサポートするものである。「照会」という用語は、保管されたデータベースのデータを処理するための動作を実行させるコマンドの組を指す。例えば、SQLは、4つのタイプの照会動作、すなわち、SELECT、INSERT、UPDATE、及びDELETEをサポートする。SELECT動作はデータベースからデータを検索し、INSERT動作は新たなデータをデータベースに追加し、UPDATE動作はデータベースのデータを修正し、DELETE動作はデータベースからデータを除去する。

【0004】

2002年2月26日に出願され、発明の名称を「APPLICATION PORT ABILITY AND EXTENSIBILITY THROUGH DATABASE SCHEMA AND QUERY ABSTRACTION」とする特許文献1（‘075出願）において、物理データを抽象的に閲覧するためのフレームワークが開示された。‘075出願のフレームワークは、物理データの論理表現を要求エンティティ（すなわち、エンドユーザー又はアプリケーション）に与えるものであった。換言すれば、‘075出願のフレームワークは、基礎物理データ構造を論理的に記述するデータ抽象化モデルを要求エンティティに与えるものであった。このようにして、要求エンティティは、アクセスされることになる基礎物理データから分離される。したがって、物理データを変更しても、その物理データにアクセスするアプリケーションを変更する必要はない。さらに、このフレームワークに基づく抽象照会は、物理データの構造を考慮せずに構築することができる。

【0005】

【特許文献1】米国特許出願第10/083,075号明細書

【発明の開示】

【発明が解決しようとする課題】

【0006】

‘075出願のフレームワークによるデータ抽象化モデルを用いると、抽象照会を、データベース内の物理データに対して実行することが可能な実行可能照会に変換することができる。しかしながら、特定の場合においては、所与の抽象照会が、データ抽象化モデルによって決まる2つ以上の実行可能照会に変換されることがある。例えば、病院内のユーザが、所与の血液検査を受けた患者のうち200より大きな対応する結果値を有する者の氏名を決定しようとする場合を想定する。この目的のために、ユーザは、以下の抽象照会を指定することができる。

FIND NAME

WHERE RESULT > 200

しかしながら、データ抽象化モデルを扱うときには、NAMEという用語は、基礎データベース内の2つ以上の物理パラメータ（例えば、医療環境においては患者又は医師の氏名、製造環境においては従業員又は顧客の氏名、など）を指すことがある。

【0007】

結果として、さらなる情報がなければ、上記の抽象照会は、（ユーザが意図したように）検査を受けた患者ではなく、特定の結果を有する検査を管理する医師の氏名を戻す実行可能照会に変換されることがある。さらに、対応するデータ抽象化モデルによって、20

10

20

30

40

50

0より大きな対応する結果値を伴う所与の血液検査を指示した病院の医師の氏名を検索する抽象照会について、別の実行可能照会を生成することも可能になると想定する。しかしながら、両方の実行可能照会に対して戻される結果の組は異なる。換言すれば、抽象照会を、ユーザが期待する照会結果をもたらす実行可能照会に変換するためには、その抽象照会の生成時におけるユーザの意図が考慮されなければならない。

【0008】

この問題に対応するための1つの手法は、所与の抽象照会について生成することができるすべての実行可能照会を生成し、そこから対応する必要な実行可能照会を選択するようにユーザを促すことを可能にすることである。しかしながら、このことは、全ての実行可能照会が基礎物理データ構造に関して表現されるため、基礎物理データからの分離を損なうことになるであろう。さらに、この手法は、ユーザを混乱させる可能性があり、ユーザが、基礎物理データ構造をより深く理解することが必要な場合がある。

【0009】

したがって、抽象照会の少なくとも2つの異なる解釈が可能であるときに、該抽象照会を実行可能照会に変換するための効率的な技術についての必要性が存在する。

【課題を解決するための手段】

【0010】

本発明は、一般に、データ抽象化モデルを管理するための方法、システム、及び製品に向けられ、より具体的には、データベース内のデータを抽象的に記述するデータ抽象化モデルにおける関係を管理するための方法、システム、及び製品に向けられる。

【0011】

1つの実施形態は、データベース内のデータの論理分岐をリンクする方法を提供する。この方法は、一般に、幾つかが共通名を共有する論理フィールドを含むデータ抽象化モデルの論理分岐間にリンクを持つ構造を構成することを含み、該リンクは、多数の論理フィールドによって共有される共通名に対する参照を含む抽象照会を実行するときに、物理フィールドを含むデータ構造の適切な連結を可能にする。

【0012】

別の実施形態は、データ抽象化モデルによって論理的に表現された物理データについて問い合わせを行う照会を生成する方法を提供する。この方法は、1つ又は複数の結果フィールドを有する、データベース内の物理データに対する抽象照会を受け取ることを含む。各々の結果フィールドは、データ抽象化モデルの論理フィールド仕様に対応する。次いで、データ抽象化モデルが抽象照会と関連付けられた論理リンクを含むかどうかが判断される。データ抽象化モデルが関連論理リンクを含む場合には、該関連論理リンクが取得され、抽象照会が実行可能照会に変換される。変換は、データ抽象化モデルと、取得した関連論理リンクとを用いて行われる。実行可能照会は、物理データに対して実行することが可能である。

【0013】

さらに別の実施形態は、データベース内のデータを抽象的に記述するデータ抽象化モデルの論理フィールドをリンクする方法を提供する。各々の論理フィールドは、データの特定のセットの論理表現を定める。本方法は、データ抽象化モデルについて、複数の論理分岐を有する論理ツリー構造を生成することを含む。各々の論理分岐は、1つ又は複数の論理フィールドを含む。次いで、異なる論理分岐間の関係が特定され、論理リンクが生成される。論理リンクは、特定された関係を抽象的に記述し、データ抽象化モデルと関連付けられる。

【0014】

さらに別の実施形態は、プロセッサによって実行されるとき、データ抽象化モデルによって論理的に表現された物理データについて問い合わせを行う照会を生成するプロセスを実施するプログラムを含むコンピュータ可読媒体を提供する。このプロセスは、データベース内の物理データに対する抽象照会を受け取ることを含む。抽象照会は、各々がデータ抽象化モデルの論理フィールド仕様に対応する1つ又は複数の結果フィールドを有する。

次いで、データ抽象化モデルが、抽象照会と関連付けられた論理リンクを含むかどうかが判断される。データ抽象化モデルが関連論理リンクを含む場合には、該関連論理リンクが取得される。次いで、抽象照会は、物理データに対して実行することが可能な実行可能照会に変換される。変換は、データ抽象化モデルと、取得した関連論理リンクとを用いて行われる。

【0015】

さらに別の実施形態は、プロセッサによって実行されるとき、データ抽象化モデルによって論理的に表現された物理データについて問い合わせを行う照会を生成するプロセスを実施するプログラムを含むコンピュータ可読媒体を提供する。このプロセスは、データ抽象化モデルを表現する論理ツリー構造内をユーザがナビゲートすることを可能にするグラフィカル・ユーザー・インターフェースを表示することを含む。論理ツリー構造は、複数の論理分岐を有する。少なくとも2つの論理分岐が、論理リンクによってリンクされる。各々の論理分岐は、データ抽象化モデルの1つ又は複数の論理フィールドを含む。各々の論理フィールドは、物理データの特定のセットの論理表現を定める。

【発明を実施するための最良の形態】

【0016】

本発明の上述した特徴、利点、及び目的が達成され、詳細に理解することができるよう 10 に、上記に簡潔に要約した本発明を、添付図面に示す本発明の実施形態を参照して、より具体的に説明する。

しかしながら、添付図面は、本発明の典型的な実施形態のみを示すものであり、したがって、本発明は他の等しく有効な実施形態を許容するものであり、その範囲を限定するものと考えるべきではない。

【0017】

本発明の実施形態は、一般に、データの論理分岐を他のデータの論理分岐と関連付ける方法を提供する。その際、表を相互にどのように結合するかに関する情報を、（例えば、 20 こうした情報を得るようユーザを直接促すことなく）自動的に収集することができる。幾つかの実施形態については、特定された関係に基づいて、関連論理分岐間に論理リンクを生成することができる。次いで、これらの論理リンクを用いて、異なって解釈されることがある論理フィールド名を伴う抽象照会を解釈することができる。換言すれば、論理リンクを用いて、その照会を生成するユーザの意図を自動的に判断することができる。

【0018】

1つの態様によれば、データ抽象化モデルは、データベース内のデータを抽象的に記述する。データ抽象化モデルは、複数の論理分岐を有する論理ツリー構造で表現することができる。各々の論理分岐は、1つ又は複数の論理フィールドを含む。各々の論理フィールドは、データベースにおけるデータの特定のセットの論理表現を定める。データ抽象化モデルは、各々の論理フィールドに、データの特定のセットの論理表現を定める論理フィールド仕様を与える。

【0019】

1つの実施形態においては、データベース内のデータの論理分岐はリンクされる。この目的のために、複数の論理分岐を有する対応する論理ツリー構造を定めるデータ抽象化モデルにアクセスする。次いで、複数の論理分岐のうちの異なる論理分岐の論理フィールド間の関係が特定される。1つの態様によれば、対応する論理ツリー構造の図形表現を生成し、グラフィカル・ユーザー・インターフェースを用いてユーザに対して表示することができる。このように、ユーザは、論理フィールド間の関係を特定するために、グラフィカル・ユーザー・インターフェースを介して論理ツリー構造にアクセスすることができる。特定された関係に基づいて、論理リンクが生成される。論理リンクは、特定された関係を抽象的に記述する。生成された論理リンクは、データ抽象化モデルと関連付けられる。

【0020】

1つの実施形態においては、1つ又は複数のデータベース内のデータに対して抽象照会が生成される。抽象照会についての1つ又は複数の結果フィールドとして、対応するデータ

10

20

30

40

50

タ抽象化モデルの複数の論理フィールドから少なくとも1つの論理フィールドが選択される。1つ又は複数の結果フィールドは、グラフィカル・ユーザー・インターフェースを用いて選択することができる。1つの態様によれば、グラフィカル・ユーザー・インターフェースは、ユーザがデータ抽象化モデルを表現する論理ツリー構造内をナビゲートすることが可能になるように構成することができる。論理ツリー構造は、少なくとも2つが論理リンクによってリンクされる複数の論理分岐を有することができる。1つ又は複数の結果フィールドの選択に際して、対応するデータ抽象化モデルが、抽象照会と関連付けられた論理リンクを含むかどうかが判断される。データ抽象化モデルが関連論理リンクを含む場合には、その関連論理リンクが取得される。このように、抽象照会が実行可能照会に変換されるときは、データ抽象化モデルと取得した関連論理リンクとを用いて、変換を行うことができる。実行可能照会は、1つ又は複数のデータベース内のデータに対して実行することができる。1つ又は複数のデータベースに対して実行可能照会を実行すると、各々の結果フィールドについてデータが戻される。

【0021】

以下では、本発明の実施形態が参照される。しかしながら、本発明は説明された特定の実施形態に限定されないものと理解されるべきである。それどころか、異なる実施形態に関連するか否かにかかわらず、本発明を実装し、実施するために、以下の特徴及び要素のいずれの組み合わせも考えられる。さらに、種々の実施形態において、本発明は、従来技術に勝る多くの利点を提供する。しかしながら、本発明の実施形態は他の可能性ある解決方法及び／又は従来技術に勝る利点を達成することができるが、所与の実施形態によって特定の利点が達成されるか否かは、本発明を限定するものではない。したがって、以下の態様、特徴、実施形態、及び利点は、単なる例示であって、明示されない限り添付の特許請求の範囲の要素又は限定とはみなされない。

【0022】

本発明の1つの実施形態は、例えば図1に示され、以下に説明されるコンピュータ・システム110のようなコンピュータ・システムで用いるためのプログラム製品として実装される。プログラム製品の1つ又は複数のプログラムは、(本明細書で説明される方法を含む)実施形態の機能を定め、様々な信号伝達媒体に収容することができる。例示的な信号伝達媒体は、(i)書き込み不可能なストレージ媒体(例えば、CD-ROMドライブによって読み出し可能なCD-ROMディスクといった、コンピュータ内部の読み取り専用メモリ・デバイス)上に永久的に保管された情報、(ii)書き込み可能なストレージ媒体(例えば、ディスクケット・ドライブ又はハードディスク・ドライブ内部のフロッピー(商標)ディスク)に保管された可変情報、又は(iii)コンピュータ・ネットワーク又は電話ネットワークなどを介して、無線通信を含む通信媒体によってコンピュータに伝達される情報を含むが、これらに限定されるものではない。最後の実施形態は、特に、インターネット及び他のネットワークからダウンロードされる情報を含む。こうした信号伝達媒体は、本発明の機能を指示するコンピュータ可読命令を運ぶときは、本発明の実施形態を表す。

【0023】

一般に、本発明の実施形態を実施するために実行されるルーチンは、オペレーティング・システム又は特定のアプリケーションの一部、コンポーネント、プログラム、モジュール、オブジェクト、又は命令のシーケンスとすることができます。本発明のソフトウェアは、典型的には、ネイティブ・コンピュータによって、機械読み出し可能なフォーマット、すなわち実行可能な命令に翻訳される多数の命令から構成される。また、プログラムは、プログラムにローカルに常駐するか又はメモリ内若しくはストレージ・デバイス上に見出される変数及びデータ構造から構成される。さらに、以下に説明される種々のプログラムは、本発明の特定の実施形態においてプログラムが実装されるアプリケーションに基づいて、特定することができる。しかしながら、以下の特別な用語のいずれも単に便宜的に用いられるものであることが認識されるべきであり、したがって、本発明は、こうした用語によって特定される及び／又は示唆される特定のアプリケーションのいずれかにのみ使用

されるものと限定されるべきではない。

【0024】

例示的なコンピューティング環境

ここで図1を参照すると、コンピューティング環境100が示される。一般に、分散環境100は、コンピュータ・システム110と、複数のネットワーク装置146とを含む。コンピュータ・システム110は、クライアント・コンピュータ、サーバ・コンピュータ、ポータブル・コンピュータ、組み込みコントローラ、PCベースのサーバ、ミニコンピュータ、ミッドレンジ・コンピュータ、メインフレーム・コンピュータ、並びに、本発明の方法、装置、及び製品をサポートするのに適した他のコンピュータを含むいずれかのタイプのコンピュータ、コンピュータ・システム、又は他のプログラム可能な電子装置を表すことができる。1つの実施形態においては、コンピュータ・システム110は、ニューヨーク州アーモンク所在の International Business Machines から入手可能なe Serverコンピュータである。

10

【0025】

例として、コンピュータ・システム110はネットワーク・システムを構成する。しかしながら、コンピュータ・システム110は、独立型の装置とすることもできる。いずれにしても、図1はコンピュータ・システムについての単なる一構成であることが理解される。本発明の実施形態は、コンピュータ・システム110が複雑なマルチユーザー装置であるか、シングルユーザー・ワークステーションであるか、又は、専用の不揮発性ストレージを持たないネットワーク機器であるかにかかわらず、同等の構成のいずれにも適用することができる。

20

【0026】

本発明の実施形態は、通信ネットワークを通じてリンクされる遠隔処理装置によってタスクが行われる分散コンピューティング環境において実施することもできる。分散コンピューティング環境においては、プログラム・モジュールは、ローカル・メモリ・ストレージ・デバイスと遠隔メモリ・ストレージ・デバイスの両方に配置することができる。この観点から、コンピュータ・システム110、及び/又は、1つ又は複数のネットワーク装置146は、処理をほとんど又はまったく行わないシン・クライアントとすることができます。

30

【0027】

コンピュータ・システム110は、例えば、直接アクセス・ストレージ・デバイス138に作動状態で接続される大容量ストレージ・インターフェース137、ディスプレイ142に作動状態で接続されるビデオ・インターフェース140、及び複数のネットワーク装置146に作動状態で接続されるネットワーク・インターフェース144によって示されるような、多数のオペレータ又は周辺システムを含むことができる。ディスプレイ142は、可視情報を出力するためのビデオ出力装置のいずれかとすることができます。

【0028】

コンピュータ・システム110は、メイン・メモリ116からバス114を介して命令及びデータを取得する少なくとも1つのプロセッサ112を備えるものとして示される。プロセッサ112は、本発明の方法をサポートするのに適したいずれかのプロセッサとすることができます。メイン・メモリ116は、必要なプログラム及びデータ構造を保持するのに十分な大きさを持ついずれかのメモリである。メイン・メモリ116は、ランダム・アクセス・メモリ、不揮発性メモリ、又はバックアップ・メモリ（例えば、プログラムマップル・メモリ、フラッシュ・メモリ、読み出し専用メモリなど）を含むメモリ・デバイスの1つ、又はそれらの組み合わせとすることができます。さらに、メモリ116は、例えば、仮想メモリとして用いられるか、大容量ストレージ・デバイス（例えば直接アクセス・ストレージ・デバイス138）上に保管されるか、又は、バス114を介してコンピュータ・システム110に連結される別のコンピュータ上に保管される、いずれかのストレージ容量といった、コンピュータ・システム110の他の場所に物理的に配置されたメモリを含むものと考えることができる。

40

50

【0029】

メモリ116は、オペレーティング・システム118を伴って構成されるものとして示される。オペレーティング・システム118は、コンピュータ・システム110の動作を管理するために用いられるソフトウェアである。オペレーティング・システム118の例には、IBM OS/400(商標)、UNIX(商標)、Microsoft Windows(商標)、及び同種のものが含まれる。

【0030】

メモリ116は、1つ又は複数のアプリケーション120と、抽象化モデル・インターフェース130とをさら含む。アプリケーション120及び抽象化モデル・インターフェース130は、様々な時点でコンピュータ・システム110の様々なメモリ及びストレージ・デバイスに常駐する複数の命令を有するソフトウェアである。アプリケーション120及び抽象化モデル・インターフェース130は、コンピュータ・システム110の1つ又は複数のプロセッサ112によって読み出されて実行されたときは、本発明の様々な様様を具体化するステップ又は要素を実行するのに必要なステップをコンピュータ・システム110に行わせる。アプリケーション120(より一般的には、オペレーティング・システム118を含むいすれかの要求エンティティ)は、(ストレージ138内に示される)データベース139に対して照会を発行するように設定される。データベース139は、データの特定の物理表現にかかわらず、データのいすれかのセットを代表するものである。データの物理表現は、データの構成スキーマを定める。例として、データベース139は、(SQL照会によってアクセス可能な)関係スキーマに従って、又は(XML照会によってアクセス可能な)XMLスキーマに従って、構成することができる。しかしながら、本発明は、特定のスキーマに限定されるものではなく、現在は知られていないスキーマへの拡張も考えられる。ここで用いられるとき、「スキーマ」という用語は、一般に、データの特定の配列を指す。

10

20

30

40

【0031】

アプリケーション120によって発行される照会は、各々のアプリケーションに含まれるアプリケーション照会仕様122に従って定められる。アプリケーション120によって発行される照会は、あらかじめ定める(すなわち、アプリケーション120の一部としてハードコーディングする)か、又は、入力(例えば、ユーザー入力)に応答して生成することができる。どちらの場合においても、照会(ここでは「抽象照会」と呼ぶ)は、抽象化モデル・インターフェース130によって定められた論理フィールドを用いて合成される。論理フィールドは、個々のデータ項目として、又は、例えばデータベース表の形態のデータ構造として、データの抽象ビューを定める。具体的には、抽象照会に用いられる論理フィールドは、抽象化モデル・インターフェース130のデータ抽象化モデル・コンポーネント132によって定められる。1つの実施形態においては、データ抽象化モデル・コンポーネント132は、関係マネージャ150によって生成され、管理される論理リンクを含む。論理リンクは、データベース139におけるデータの論理分岐間の関係を記述する。データ抽象化モデル・コンポーネント132及び関係マネージャ150の動作及び対話は、図2及び図3を参照して、以下にさらに説明される。

【0032】

例として、関係マネージャ150は、ランタイム・コンポーネント134の一部として示される。ランタイム・コンポーネント134は、抽象照会を、データベース139に含まれるデータの物理表現と一致する形式を有する具象照会に変換する。具体的には、ランタイム・コンポーネント134は、抽象照会を具象照会に変換する際に、データ抽象化モデル・コンポーネント132に含まれる論理リンクを用いることができる。具象照会は、ランタイム・コンポーネント134がデータベース139に対して実行することができる。ランタイム・コンポーネント134の動作は、図2を参照して以下にさらに説明される。

【0033】

例示的な照会実行ランタイム環境

50

ここで図2を参照すると、照会実行ランタイムにおける、ランタイム・コンポーネント134と、アプリケーション120と、データ抽象化モデル132と、関係マネージャ150との対話を説明する関係図が示される。データ抽象化モデル132が、データベース（例えばデータベース139）内のデータ構造に対応する論理フィールドを定め、それによりデータベース139内のデータの抽象ビューすなわち論理ビューを提供することから、データ抽象化モデル132は、ここでは「論理表現」とも呼ばれる。データ構造は、データベース表の形式又はデータベース表の列の形式をとった配列のような、データの物理配列である。より具体的には、各々の論理フィールドは、データベース139内のデータの特定のセットの論理表現を定める。多数のデータベース表を有する関係データベース環境においては、特定の論理フィールドを有する特定の論理表現を各々のデータベース表に与えることができる。この場合は、全ての特定の論理表現が一体となって、データ抽象化モデル132を構成する。データの物理エンティティが、データの物理表現に従ってデータベース139内に配列される。データの物理エンティティ（同じ意味で、物理データ・エンティティとも呼ばれる）は、基礎物理表現におけるデータ項目である。したがって、物理データ・エンティティは、データベース表又はデータベース表の列に含まれるデータ、すなわちデータそのものである。例として、2つの物理表現、すなわち、XMLデータ表現214₁及び関係データ表現214₂が示される。しかしながら、物理表現214_Nは、既知であるか未知であるかにかかわらず、他のいずれかの物理表現が考えられることを示す。1つの実施形態においては、関係データベース環境の場合について上述したように、異なる単一のデータ抽象化モデル132が、別個の物理表現214の各々について与えられる。代替的な実施形態においては、単一のデータ抽象化モデル132は、2つ又はそれ以上の物理表現214についての（関連アクセス方式を持つ）フィールド仕様を含む。フィールド仕様は、論理フィールドの記述であり、一般に、論理フィールドを特定の物理表現の1つ又は複数のデータ構造にマップするマッピング規則を有する。

【0034】

データの論理表現を用いて、アプリケーション照会仕様122は、1つ又は複数の論理フィールドを指定し、結果として得られる照会202を作成する。要求エンティティ（例えばアプリケーション120）は、要求エンティティのアプリケーション照会仕様によって定められる通りに、結果として得られる照会202を発行する。1つの実施形態においては、抽象照会202は、データ選択に用いられる基準と、データ選択基準に基づいて戻されることになる結果フィールドの明示的な仕様との両方を含むことができる。選択基準、及び、抽象照会202の結果フィールド仕様の例が、図3に示される。したがって、抽象照会202は、例として、選択基準304及び結果フィールド仕様306を含む。

【0035】

結果として得られる照会202は、データベース139内の基礎データ構造を直接参照することによってではなく、抽象（すなわち、論理）フィールドに従って合成されることから、ここでは一般に「抽象照会」と呼ばれる。結果として、抽象照会は、用いられる特定の基礎物理データ表現とは独立に定めることができる。実行するために、抽象照会は、データ抽象化モデル132を用いて、データの基礎物理表現と整合する具象照会に変換される。具象照会は、データベース139に対して実行可能である。抽象照会を具象照会に変換する例示的な方法は、図5及び図6を参照して以下に説明される。

【0036】

一般に、データ抽象化モデル132は、情報を論理フィールドの組として公開する。1つの態様によれば、データ抽象化モデル132は、複数の論理分岐を有する論理ツリー構造として表現することができる。各々の論理分岐は、1つ又は複数の論理フィールドを含むことができる。例としての論理ツリー構造の2つの例示的な論理分岐が図9に示される。1つの実施形態においては、データ選択のための基準と、照会動作から戻される結果データの形式とを指定するために、抽象照会の内部で論理フィールドを用いることができる。論理フィールドは、データベース139において用いられている基礎物理表現とは独立に定められ、そのことにより、基礎物理表現に緩やかに結合される抽象照会を形成するこ

10

20

30

40

50

とが可能になる。

【0037】

例として、データ抽象化モデル132は、論理リンク204を含む。論理リンク204は、データ抽象化モデル132を表現する論理ツリー構造の異なる論理分岐間の関係を抽象的に記述する。2つ以上のデータ抽象化モデルが与えられる場合には、論理リンク204は、異なるデータ抽象化モデルの論理分岐間の関係を記述することができる。抽象照会202を具象照会に変換する際には、関係マネージャ150を用いて、関係を決定するために論理リンク204にアクセスすることができる。決定された関係を用いて、データベース139におけるデータの物理表現の異なるデータ構造間のパスを、具象照会について特定することができる。例えば、具象照会はSQL照会であり、データベースの物理表現は関係スキーマであり、データ構造はデータベース表であると仮定する。この例においては、論理リンク204に基づいて、SQL照会についてJOINステートメントを生成し、SQL照会の中の対応するデータベース表間のリンクを指定することができる。このリンクは、対応する照会結果を決定するためにSQL照会に従ってデータベース139内のデータを照会するときに利用される。

【0038】

例示的なデータ抽象化モデル

ここで図3を参照すると、抽象照会202とデータ抽象化モデル132との対話を説明する関係図が示される。1つの実施形態においては、データ抽象化モデル132は、複数のフィールド(field)仕様308₁、308₂、308₃、308₄、308₅、及び308₆（一例として、6つが示される）を備え、これらはまとめてフィールド仕様308と呼ばれる。具体的には、フィールド仕様は、抽象照会の合成に利用可能な各々の論理フィールドについて与えられる。各々のフィールド仕様は、1つ又は複数の属性を含むことができる。例として、フィールド仕様308は、論理フィールド名(name)属性320₁、320₂、320₃、320₄、320₅、及び320₆（まとめて、フィールド名320）と、関連アクセス方法(access method)属性322₁、322₂、322₃、322₄、322₅、及び322₆（まとめて、アクセス方法322）とを含む。各々の属性は値を持つことができる。例えば、論理フィールド名属性320₁は「Patient ID」という値を有し、アクセス方法属性322₁は「Simple」という値を有する。さらに、各々の属性は、1つ又は複数の関連抽象特性を含むことができる。各々の抽象特性は、データ構造の特徴を記述し、関連値を有する。上述のように、データ構造は、論理フィールドに対応するデータの1つ又は複数の物理エンティティによって定められる基礎物理表現の一部を指す。具体的には、抽象特性は、データベース表(table)の名前又はデータベース表の列(column)の名前などのように、データ構造に対応する物理データ・エンティティの位置を抽象的に記述するデータ位置メタデータを表すことになる。例として、アクセス方法属性322₁は、データ位置メタデータ「Table」及び「Column」を含む。さらに、データ位置メタデータ「Table」は、値「Patient info」を有し、データ位置メタデータ「Column」は、値「patient_ID」を有する。したがって、本例における基礎関係データベース・スキーマを想定する場合、データ位置メタデータ「Table」及び「Column」の値は、列「patient_ID」を有する表「Patient info」を指す。

【0039】

1つの実施形態においては、論理フィールドのグループ（すなわち、2つ又はそれ以上）は、カテゴリー(category)の一部とすることができます。したがって、データ抽象化モデル132は、複数のカテゴリー仕様310₁及び310₂（一例として、2つが示される）を含み、これらはまとめてカテゴリー仕様と呼ばれる。1つの実施形態においては、2つ又はそれ以上の論理フィールドの論理グループの各々について、カテゴリー仕様が与えられる。例えば、論理フィールド308₁～308₃及び308₄～308₆は、それぞれ、カテゴリー仕様310₁及び310₂の一部である。カテゴリー仕様は、

10

20

30

40

50

ここでは単に「カテゴリー」とも呼ばれる。カテゴリーは、例えばカテゴリー名 330₁ 及び 330₂（まとめて 1 つ又は複数のカテゴリー名 330）といったカテゴリー名に従って区別される。本例においては、論理フィールド 308₁ ~ 308₃ は「Patient」カテゴリーの一部であり、論理フィールド 308₄ ~ 308₆ は「Tests」カテゴリーの一部である。

【0040】

アクセス方法 322 は、一般に、論理フィールド名をデータベース（例えば図 1 のデータベース 139）内のデータに関連付ける（すなわち、マップする）。対応することになる異なるタイプの論理フィールド数に応じて、あらゆる数のアクセス方法が考えられる。

1 つの実施形態においては、単純（simple）フィールド、フィルタ（filter ed）フィールド、及び合成（composed）フィールドのためのアクセス方法が与えられる。フィールド仕様 308₁、308₂、308₅、及び 308₆ には、それぞれ、単純フィールド・アクセス方法 322₁、322₂、322₅、及び 322₆ が例示される。単純フィールドは、基礎物理表現内の特定のデータ構造に直接マップされる（例えば、所与のデータベース表及び列にマップされたフィールド）。例として、上述のように、単純フィールド・アクセス方法 322₁ は、論理フィールド名 320₁（「Patient ID」）を、「Patient info」と名前を付けた表の中の「patient_ID」と名前を付けた列にマップする。フィールド仕様 308₃ には、フィルタ・フィールド・アクセス方法 322₃ が例示される。フィルタ・フィールドは、関連データ構造を特定し、物理表現内部の項目の特定のサブセットを定めるのに用いられるフィルタを与える。図 3 に例が設けられており、ここでは、フィルタ・フィールド・アクセス方法 322₃ は、論理フィールド名 320₃（「Street」）を、「Patient info」と名前を付けた表の中の「street」と名前を付けた列のデータにマップし、「NY」市内の個人についてのフィルタ（filter）を定める。フィルタ・フィールドの別の実施例は、郵便番号の物理表現にマップし、データをニューヨーク州について定められた郵便番号のデータのみに制限する、ニューヨーク郵便番号フィールドである。フィールド仕様 308₄ には、合成フィールド・アクセス方法 322₄ が例示される。合成フィールド・アクセス方法は、アクセス方法定義の一部として与えられる式（expression）を用いて、1 つ又は複数のデータ構造から論理フィールドを計算する。このようにして、基礎物理データ表現には存在しない情報を計算することができる。図 3 に示される例においては、合成フィールド・アクセス方法 322₄ は、論理フィールド名 320₄（「Normalized Results」）を「Results / 10」にマップする。別の例は、販売価格フィールドに消費税率を乗ずることによって合成される消費税フィールドである。

【0041】

基礎データの所与のいずれかのデータ・タイプ（例えば、日付、十進数など）についてのフォーマットは、様々であると考えられる。したがって、1 つの実施形態においては、フィールド仕様 308 は、基礎データのフォーマットを反映するタイプ属性を含む。しかしながら、別の実施形態においては、フィールド仕様 308 のデータ・フォーマットは関連基礎物理データとは異なり、この場合には、基礎物理データを論理フィールドのフォーマットに換算することが必要である。

【0042】

例として、図 3 に示されるデータ抽象化モデル 132 のフィールド仕様 308 は、図 2 に示される関係データ表現 214₂ で表現されたデータにマップされる論理フィールドを代表するものである。しかしながら、データ抽象化モデル 132 の他の事例では、論理フィールドは、XML などの他の物理表現にマップされる。

【0043】

図 3 に示される抽象照会 202 に対応する例示的な抽象照会が、以下の表 1 に示される。例として、例示的な抽象照会は、XML を用いて定められる。しかしながら、他のあらゆる言語を用いることができるという利点を有する。

10

20

30

40

50

【表1】

表1-抽象照会の例

```

001 <?xml version="1.0"?>
002 <!--Query string representation: (Result)"200"-->
003 <QueryAbstraction>
004   <Selection>
005     <Condition internalID="4">
006       <Condition field="Results" operator="GT" value="200"
007         internalID="1"/>
008   </Selection>
009   <Results>
010     <Field name="Name"/>
011   </Results>
012 </QueryAbstraction>

```

10

【0044】

例として、表1に示される抽象照会は、選択基準を含む選択仕様（行004～008）と結果仕様（行009～011）とを含む。1つの実施形態においては、選択基準（これ以降、「検索基準」とも呼ばれる）は、（論理フィールドについての）フィールド名、比較演算子（=、>、<など）、及び値式（そのフィールドが何と比較されているか）で構成される。1つの実施形態においては、結果仕様は、照会実行の結果として戻されることになる抽象フィールドのリストである。抽象照会における結果仕様は、フィールド名及びソート基準で構成することができる。

20

【0045】

図3に示されるデータ抽象化モデル132に対応する例示的なデータ抽象化モデル（DAM）が以下の表2に示される。例として、例示的なデータ抽象化モデルは、XMLを用いて定められる。しかしながら、他のあらゆる言語を用いることができるという利点を有する。

【表2】

表2-データ抽象化モデルの例

```

001 <?xml version="1.0"?>
002 <DataAbstraction>
003   <Category name="Patient">
004     <Field queryable="Yes" name="Patient ID" displayable="Yes">
005       <AccessMethod>
006         <Simple columnName="patient_ID" tableName="Patientinfo"></Simple>
007       </AccessMethod>
008     </Field>
009     <Field queryable="Yes" name="Name" displayable="Yes">
010       <AccessMethod>
011         <Simple columnName="name" tableName="Patientinfo"></Simple>
012       </AccessMethod>
013     </Field>
014     <Field queryable="Yes" name="Street" displayable="Yes">
015       <AccessMethod>
016         <Filter columnName="street" tableName="Patientinfo">
017           <Filter="Patientinfo.city=NY">
018         </AccessMethod>
019       </Field>
020   </Category>
021   <Category name="Tests">
022     <Field queryable="Yes" name="Normalized Results" displayable="Yes">
023       <AccessMethod>
024         <Composed columnName="results" tableName="Bloodtest">
025           <Composed Expression="Results/10">
026         </AccessMethod>
027       </Field>
028     <Field queryable="Yes" name="Results" displayable="Yes">
029       <AccessMethod>
030         <Simple columnName="results" tableName="Bloodtest"></Simple>
031       </AccessMethod>
032     </Field>
033     <Field queryable="Yes" name="Requester" displayable="Yes">
034       <AccessMethod>
035         <Simple columnName="requester" tableName="Bloodtest"></Simple>
036       </AccessMethod>
037     </Field>
038   </Category>
039 </DataAbstraction>

```

10

20

30

【0046】

例として、行004～008は、図3に示されるDAM132の第1のフィールド仕様308₁に対応し、行009～013は、第2のフィールド仕様308₂に対応する点に留意されたい。

【0047】

1つの態様によれば、データ抽象化モデル132は、論理リンク（例えば、図2の論理リンク204）を含むことができる。より具体的には、データ抽象化モデル132のフィールド仕様308の各々は、関連論理リンクを含むことができる。さらに、データ抽象化モデル132の各々のカテゴリーは、関連論理リンクを含むことができる。論理リンクを含む例示的なデータ抽象化モデルは、図4を参照して、以下に説明される。

【0048】

論理リンクを有する例示的なデータ抽象化モデル

前述のように、論理フィールド間の関係に基づいて生成された論理リンクを、抽象照会における曖昧なフィールド名参照を解決するのに用いることができる。ここで図4を参照すると、例示的な論理リンクを含むデータ抽象化モデル350の1つの実施形態が示される。データ抽象化モデル350は、図3のデータ抽象化モデル132に例示的な論理リンクが追加されたものに対応する。簡略化のために、データ抽象化モデル350においては

40

50

、図3のデータ抽象化モデル132におけるカテゴリー(category)310₁及び310₂のフィールド(field)仕様308₁及び308₄のみが示される。さらに、データ抽象化モデル350は、図3のデータ抽象化モデル132に追加された例示的な関係(relations)選択360を含む。例として、論理リンクは、リンク仕様380₁及びリンク定義370₁から構成される。例として、リンク仕様308₁は、図3のデータ抽象化モデル132のカテゴリー310₁に追加された。リンク定義370₁は、関係選択360に含まれる。より一般には、関係選択は、1つ又は複数のリンク定義を含むことができる。

【0049】

1つの実施形態においては、リンク仕様は、データの異なる論理分岐間の関係を特定し、複数の属性を含むことができる。各々の属性は値を持つことができる。1つの実施形態においては、各々のリンク仕様は、リンク分岐属性、リンク・タイプ属性、及びリンク名属性を含む。リンク分岐属性は、リンク仕様を含む論理分岐にリンクされる論理分岐を特定する値を有する。例えば、カテゴリー「Patient」310₁のリンク仕様380₁は、値「/Tests」を有するリンク分岐属性「to」380₂を含む。値「/Tests」は、カテゴリー「Patient」310₁が、カテゴリー「Tests」310₂によって表現される論理分岐に関連することを示す。リンク・タイプ属性は、関係のタイプを示す値を有する。例えば、リンク仕様380₁は、「has」という値を有するリンクタイプ属性「type」380₃を含む。より具体的には、値「has」は、カテゴリー「Patient」310₁の論理フィールドによって抽象的に記述される各々の患者が、カテゴリー「Tests」310₂の論理フィールドによって抽象的に記述される1つ又は複数の関連検査を有することを示す。換言すれば、所与の実施例においては、両方のカテゴリー310₁及び310₂のいずれかの各論理フィールドは、所与の患者に関する情報を含む。しかしながら、値「has」は単なる例示の目的で示されたものであり、リンク・タイプ属性の他の様々な値が考えられることに留意すべきである。例えば、対応するリンクされた論理分岐の他の論理フィールドの情報を用いて所与の論理フィールドを記述できることを示すために、値「is_a」を実装することができる。リンク名属性は、論理リンクを特定する値を有する。例えば、リンク仕様380₁は、値「Patient_To_Tests」を有するリンク名属性「relation」380₄を含む。換言すれば、リンク仕様380₁を有する論理リンクは、「Patient_To_Tests」という名を持つ。

【0050】

1つの実施形態においては、リンク定義370₁は、リンク仕様380₁によって特定される関係を抽象的に記述する。したがって、リンク定義370₁は、複数の属性を含むことができる。各々の属性は値を持つことができる。さらに、各々の属性は、1つ又は複数の抽象特性を持つことができる。1つの態様によれば、各々のリンク定義は、名前(name)属性と、少なくとも1つのリンクID(linkID)属性と、少なくとも1つのリンク・ポイント(link_point)属性とを含む。名前属性は、論理リンクを特徴付ける値を有する。名前属性値は、一般に、対応するリンク仕様のリンク名属性の値に対応する。例として、リンク定義370₁は、リンク仕様380₁のリンク名属性「relation」380₄の値に対応する値「Patient_To_Tests」を有する名前属性「Name」370₂を含む。各々のリンクID属性は、論理リンクの少なくとも1つのセグメントを一意的に特定する値を有する。例えば、論理リンクは、データベース表2を経由するデータベース表1とデータベース表3との間のパスを記述することができる。この場合においては、論理リンクは、(1)データベース表1とデータベース表2の間の関係を抽象的に記述する第1のセグメントと、(2)データベース表2とデータベース表3の間の関係を抽象的に記述する第2のセグメント、の2つのセグメントを含むことになる。したがって、第1のセグメントは第1のリンクID属性によって一意的に特定することができ、第2のセグメント第2のリンクID属性によって一意的に特定することができる。図4に示される例においては、リンク定義370₁は、値「Patient

10

20

30

40

50

「`ts_To_Tests1`」を有する单一のリンクID属性「`Link_ID`」`3703`を含む。例として、リンクID属性「`Link_ID`」`3703`は、図1のデータベース`139`のようなリンクされた物理データ構造を特定する2つの抽象特性を含む。例として、リンクID属性「`Link_ID`」`3703`は、抽象特性「`source`」`3704`及び「`target`」`3705`を含む。抽象特性「`source`」`3704`は値「`PatientInfo`」を持ち、抽象特性「`target`」`3705`は値「`Bloodtest`」を持つ。換言すれば、論理リンク「`Patient_To_Tests`」は、データベース表「`PatientInfo`」及び「`Bloodtest`」の間の関係を抽象的に記述する。各々のリンク・ポイント属性は、データベース表の列のようなリンクされた物理データ構造のリンク・ポイントを特定する2つの抽象特性を含む。例として、リンク・ポイント属性「`Link_point`」`3706`は、抽象特性「`source`」`3707`及び「`target`」`3708`を含む。抽象特性「`source`」`3707`は値「`patient_ID`」を持ち、抽象特性「`target`」`3708`もまた、値「`patient_ID`」を持つ。換言すれば、データベース表「`PatientInfo`」及び「`Bloodtest`」は、両方の表に含まれるそれぞれの「`patient_ID`」列を介してリンクされる。

10

【0051】

以下の表3及び表4に、例示的なデータベース表「`PatientInfo`」及び「`Bloodtest`」が示される。分かりやすく簡略化するために、表3及び表4に示される例示的なデータベース表について、列の名前とデータ・タイプのみが示される。しかしながら、対応するデータベース表には多数の照会可能なデータが存在すると考えるべきである。

20

【表3】

表3－例示的データベース表「`PatientInfo`」

```
001 patient_ID int
002 name      varchar
003 street    varchar
```

【0052】

【表4】

表4－例示的データベース表「`Bloodtest`」

```
001 test_num  int
002 patient_ID int
003 results   float
004 requester int
```

30

【0053】

表3から分かるように、表「`PatientInfo`」は、例として、各々の患者に関する名前（「`name`」）及び住所（「`street`」）情報を含む。さらに、表「`PatientInfo`」の各々の患者は、識別子「`patient_ID`」によって一意的に特定される。表「`Bloodtest`」は、例として、病院内の所与の医師（「`requester`」）によって要求され、患者（「`patient_ID`」）に対して行われた血液検査の結果（「`results`」）に関する情報を含む。各々の血液検査結果は、識別子「`test_num`」によって一意的に特定される。

40

【0054】

表2のデータ抽象化モデルの全フィールド仕様を含む、図4に示されるデータ抽象化モデル`350`に対応する例示的なデータ抽象化モデル（DAM）が、以下の表5に示される。例として、例示的なデータ抽象化モデルは、XMLを用いて定められる。しかしながら、他のあらゆる言語を用いることができるという利点を有する。

【表5】

表5-データ抽象化モデルの例

```

001 <?xml version="1.0"?>
002 <DataAbstraction>
003   <Category name="Patient">
004     <Field queryable="Yes" name="Patient ID" displayable="Yes">
005       <AccessMethod>
006         <Simple columnName="patient_ID" tableName="Patientinfo"></Simple>
007       </AccessMethod>
008     </Field>
009     <Field queryable="Yes" name="Name" displayable="Yes">
010       <AccessMethod>
011         <Simple columnName="name" tableName="Patientinfo"></Simple>
012       </AccessMethod>
013     </Field>
014     <Field queryable="Yes" name="Street" displayable="Yes">
015       <AccessMethod>
016         <Filter columnName="street" tableName="Patientinfo">
017           </Filter="Patientinfo.city=NY">
018         </AccessMethod>
019     </Field>
020     <Relationship to="/Tests" type="has" relation="Patient_To_Tests"/>
021   </Category>
022   <Category name="Tests">
023     <Field queryable="Yes" name="Normalized Results" displayable="Yes">
024       <AccessMethod>
025         <Composed columnName="results" tableName="Bloodtest">
026           </Composed Expression="Results/10">
027         </AccessMethod>
028     </Field>
029     <Field queryable="Yes" name="Results" displayable="Yes">
030       <AccessMethod>
031         <Simple columnName="results" tableName="Bloodtest"></Simple>
032       </AccessMethod>
033     </Field>
034     <Field queryable="Yes" name="Requester" displayable="Yes">
035       <AccessMethod>
036         <Simple columnName="requester" tableName="Bloodtest"></Simple>
037       </AccessMethod>
038       <Relationship
039         to="/Patient" type="is_a" relation="Requester_To_Patient"/>
040     </Field>
041   </Category>
042   <Relations>
043     <Relation name="Patient_To_Tests">
044       <Link id="Patient_To_Tests1"
045         source="Patientinfo" target="Bloodtest" type="left">
046           <LinkPoint source="patient_ID" target="patient_ID"/>
047         </Link>
048     </Relation>
049     <Relation name="Requester_To_Patient">
050       <Link id="Requester_To_Patient1"
051         source="Bloodtest" target="Employeeinfo" type="left">
052           <LinkPoint source="requester" target="employeenum"/>
053         </Link>
054       <Link id="Requester_To_Patient2"
055         source="Employeeinfo" target="Patientinfo" type="left">
056           <LinkPoint source="patient_ID" target="patient_ID"/>
057         </Link>
058     </Relation>
059   </Relations>
060 </DataAbstraction>

```

10

20

30

40

行 0 4 2 ~ 0 5 9 は、図 4 に示される関係セクション 3 6 0 に対応することに留意すべきである。例として、行 0 2 0 は図 4 に示されるリンク仕様 3 8 0₁ に対応し、行 0 4 3 ~ 0 4 8 はリンク仕様 3 7 0₁ に対応することに留意されたい。さらに、行 0 4 9 ~ 0 5 8 は、行 0 3 8 ~ 0 3 9 のリンク仕様によって指定される論理リンクについての別のリンク定義を含む。具体的には、行 0 4 9 ~ 0 5 8 のリンク定義は、データベース表「B l o o d t e s t」が、表「B l o o d t e s t」に含まれる列「r e q u e s t e r」と、表「E m p l o y e e i n f o」に含まれる列「e m p l o y e e n u m」とを介して、別のデータベース表「E m p l o y e e i n f o」にリンクされる（行 0 5 1 ~ 0 5 2）ことを示す。行 0 4 9 ~ 0 5 8 のリンク定義は、さらに、データベース表「E m p l o y e e i n f o」が、両方の表に含まれるそれぞれの列「p a t i e n t _ I D」を介して、データベース表「P a t i e n t i n f o」にリンクされる（行 0 5 5 ~ 0 5 6）ことを示す。以下の表 6 に、対応する例示的なデータベース表「E m p l o y e e i n f o」が示される。分かりやすく簡略化するために、表 6 に示される例示的なデータベース表について、列の名前とデータ・タイプのみが示される。しかしながら、対応するデータベース表には多数の照会可能なデータが存在すると考えるべきである。10

【表 6】

表 6 一例示的データベース表「E m p l o y e e i n f o」

001 employeenum int
002 patient_ID int

20

【0 0 5 6】

上述のように、表 1 の抽象照会は、照会を実行するために具象照会に変換することができる。抽象照会を具象照会に変換するための例示的な方法が、図 5 及び図 6 を参照して以下に説明される。

【0 0 5 7】

抽象照会の具象照会への変換

ここで図 5 を参照すると、図 1 のランタイム・コンポーネント 1 3 4 の動作に関する 1 つの実施形態を具体化する例示的なランタイム方法 4 0 0 が示される。ランタイム・コンポーネント 1 3 4 が（表 1 に示される抽象照会のような）抽象照会を入力として受け取ると、ステップ 4 0 2 において方法 4 0 0 に入る。ステップ 4 0 4 において、ランタイム・コンポーネント 1 3 4 は抽象照会を読み込んで解析し、個々の選択基準及び所望の結果フィールドを探し出す。ステップ 4 0 6 において、ランタイム・コンポーネント 1 3 4 は、抽象照会に存在する各々の照会選択基準ステートメントを処理し、それによって具象照会のデータ選択部分を構築するために、（ステップ 4 0 6、4 0 8、4 1 0、及び 4 1 2 を含む）ループに入る。1 つの実施形態においては、選択基準は、（論理フィールドについての）フィールド名、比較演算子（=、>、<など）、及び値式（そのフィールドが何と比較されているか）で構成される。ステップ 4 0 8 において、ランタイム・コンポーネント 1 3 4 は、抽象照会の選択基準からのフィールド名を用いて、データ抽象化モデル 1 3 2 のフィールドの定義を検索する。上述のように、フィールド定義は、そのフィールドと関連付けられたデータ構造にアクセスするのに用いられるアクセス方法の定義を含む。次いで、ランタイム・コンポーネント 1 3 4 は、処理されている論理フィールドについて具象照会コントリビューションを構築する（ステップ 4 1 0）。ここで定義されるとき、具象照会コントリビューションは、現在の論理フィールドに基づいてデータ選択を実施するのに用いられる具象照会の一部である。具象照会は、S Q L 及び X M L Q u e r y といった言語で表現された照会であり、所与の物理データ・リポジトリ（例えば、関係データベース又は X M L リポジトリ）のデータと整合する。したがって、具象照会は、図 1 に示されるデータベース 1 3 9 によって表現される物理データ・リポジトリからデータを探し出して取得するために用いられる。次いで、現在のフィールドについて生成された具象照会コントリビューションは、具象照会ステートメントに追加される。次いで、方法 4 0 0 は、ステップ 4 0 6 に戻り、抽象照会の次のフィールドについて処理を開始する。したが304050

つて、ステップ406から入った処理は抽象照会の各々のデータ選択フィールドについて繰り返され、そのことにより、実施されることになる最終的な照会に付加的内容を与える。

【0058】

抽象照会のデータ選択部分を構築した後で、ランタイム・コンポーネント134は、照会実行の結果として戻されることになる情報を特定する。上述のように、1つの実施形態においては、抽象照会は、ここでは結果仕様と呼ばれる結果フィールドのリスト、すなわち照会実行の結果として戻されることになる論理フィールドのリストを定める。抽象照会の結果仕様は、フィールド名及びソート基準から構成することができる。したがって、方法400は、ステップ414においてループ(ステップ414、416、418、及び420によって定められる)に入り、生成されている抽象照会に結果フィールド定義を追加する。ステップ416において、ランタイム・コンポーネント134は、データ抽象化モデル132内で(抽象照会の結果仕様から)結果フィールド名を検索し、次いで、データ抽象化モデル132から結果フィールド定義を取得して、現在の論理結果フィールドについて戻されることになるデータの物理位置を特定する。次いで、ランタイム・コンポーネント134は、論理結果フィールドについて、(戻されることになるデータの物理位置を特定する抽象照会の)抽象照会コントリビューションを(ステップ418において)構築する。次いで、ステップ420において、抽象照会コントリビューションは抽象照会ステートメントに追加される。抽象照会の結果仕様の各々が処理されると、ステップ422において抽象照会が実行される。

10

20

30

40

50

【0059】

ステップ410及び418に従って論理フィールドについて抽象照会コントリビューションを構築するための方法500の1つの実施形態が、図6を参照して説明される。ステップ502において、方法500は、現在の論理フィールドと関連付けられたアクセス方法が単純(simple)アクセス方法であるかどうかを照会する。そうである場合には、物理データ位置情報に基づいて抽象照会コントリビューションが構築され(ステップ504)、次いで上述の方法400に従って処理が続行する。そうでない場合には、処理はステップ506に続き、現在の論理フィールドと関連付けられたアクセス方法がフィルタ(filter)アクセス方法であるかどうかを照会する。そうである場合には、1つ又は複数の所与のデータ構造について、物理データ位置情報に基づいて抽象照会コントリビューションが構築される(ステップ508)。ステップ510において、抽象照会コントリビューションは、1つ又は複数の所与のデータ構造と関連付けられたサブセット・データに対して使用される付加的な論理(フィルタ選択)を用いて拡張される。次いで上述の方法400に従って処理が続行する。

【0060】

アクセス方法がフィルタ・アクセス方法でない場合には、処理はステップ506からステップ512に進み、そこで方法500は、アクセス方法が合成(composed)アクセス方法であるかどうかを照会する。アクセス方法が合成アクセス方法である場合には、ステップ514において、合成されたフィールド式における各々のサブフィールド参照についての物理データ位置を探し出して取得する。ステップ516において、合成されたフィールド式の物理フィールド位置情報は、合成されたフィールド式の論理フィールド参照と置き換えられ、そのことにより、抽象照会コントリビューションが生成される。次いで上述の方法400に従って処理が続行する。

【0061】

アクセス方法が合成アクセス方法でない場合には、処理はステップ512からステップ518に進む。ステップ518は、本発明の実施形態として考えられる他の種類のアクセス方法のいずれかを代表するものである。しかしながら、すべてではない利用可能なアクセス方法を実装する実施形態が考えられることを理解すべきである。例えば、特定の実施形態においては、単純アクセス方法のみが用いられる。別の実施形態においては、単純アクセス方法とフィルタ・アクセス方法のみが用いられる。

【 0 0 6 2 】

上述したように、データ抽象化モデルは、関係マネージャ（例えば、図1の関係マネージャ150）を用いて生成し、管理することが可能な論理リンクを含むことができる。論理リンクは、データ抽象化モデルを表現する論理ツリー構造の異なる論理分岐間の関係を抽象的に記述する。1つの実施形態において、論理リンクを生成するための関係マネージャの動作が、図7を参照して以下に説明される。

【 0 0 6 3 】**関係に基づく論理リンクの生成**

ここで図7を参照すると、論理リンク（例えば、図2の論理リンク204）を生成するための例示的な方法600が示される。1つの実施形態においては、方法600は、関係マネージャ（例えば、図1の関係マネージャ150）によって実施される。方法600はステップ610において開始する。

10

【 0 0 6 4 】

ステップ629において、データ抽象化モデル（例えば、図1のデータ抽象化モデル132）が準備される。例として、データ抽象化モデルを準備することには、メモリ（例えば、図1のメインメモリ116）からデータ抽象化モデルを取得することが含まれる。データ抽象化モデルを準備することには、さらに、実行時にデータ抽象化モデルを生成することが含まれる。ステップ630において、準備されたデータ抽象化モデルについて、論理ツリー構造が生成される。論理ツリー構造は、複数の論理分岐を含み、データ抽象化モデルを表現する。各々の論理分岐は、データ抽象化モデルからの1つ又は複数の論理フィールドを含むことができる。2つの例示的な論理分岐を有する例示的な論理ツリー構造が、図9を参照して後述される。

20

【 0 0 6 5 】

ステップ640において、複数の論理分岐のうちの異なる論理分岐間の関係が特定される。1つの実施形態においては、異なる論理分岐間の関係は、ユーザによって特定される。例えば、ユーザは、基礎物理データ構造間の関係に関するユーザの知識に基づいて、関係を特定することができる。1つの態様によれば、ユーザのこの知識は、ステップ620又はステップ630に先立って基礎物理データ構造に関して行われる分析に基づくものとすることが可能である。

30

【 0 0 6 6 】

異なる論理分岐の論理フィールド間の関係に対して参照が行われることに留意すべきである。しかしながら、他の関係を特定して、論理リンクを生成することができる。例えば、親カテゴリーと子フィールドとの関係、例えば同一の論理分岐内の論理フィールド間の関係を、論理リンクの生成に用いることもできる。したがって、異なる論理分岐間の関係に基づいて論理リンクを生成することは、単に一例として説明されるに過ぎないことが理解される。しかしながら、いずれかの適切な関係に基づいて論理リンクを生成することが、広く考えられる。

【 0 0 6 7 】

ステップ650において、1つ又は複数の論理リンクが、特定された関係に基づいて生成される。1つの実施形態においては、1つ又は複数の論理リンクを生成することには、実行時に1つ又は複数の論理リンクを生成することが含まれる。例えば、ユーザは、適切なユーザー・インターフェースを用いて、各々の論理リンクについてリンク仕様（例えば、図4のリンク仕様380₁）及びリンク定義（例えば、図4のリンク定義370₁）を編集することができる。しかしながら、完全に自動化された生成プロセスを含めて、1つ又は複数の論理リンクの生成に適したいかなる技術も考えられることに留意すべきである。

40

【 0 0 6 8 】

ステップ660において、生成された論理リンクは、データ抽象化モデルと関連付けられる。例えば、生成された論理フィールドをデータ抽象化モデルに含ませて、こうした関連付けを生成することができる。より具体的には、表5を参照して上述したように、ユー

50

ザが適切なユーザー・インターフェースを用いてデータ抽象化モデルを編集し、そこに1つ又は複数の論理リンクを挿入することができる。しかしながら、代替的に、生成された論理リンクに対する1つ又は複数の参照をデータ抽象化モデルに含ませることができるに留意すべきである。したがって、生成された論理リンクは、データ抽象化モデルとは別に、1つ又は複数の永続的データ・オブジェクトとして保管することができる。換言すれば、生成された論理リンクは、すべてにわたって広く考えられる様々な異なる方法で、データ抽象化モデルと関連付けることができる。次いで方法600はステップ670において終了する。

【0069】

上述したように、論理リンクを用いて抽象照会を具象照会に変換することができる。論理リンクを用いた抽象照会の変換は、図8を参照して以下により詳細に説明される。

【0070】

照会変換における論理リンクの使用

ここで図8を参照すると、論理リンク（例えば、図2の論理リンク204）を用いて、抽象照会（例えば、図2の抽象照会202）を具象照会に変換するための例示的な方法700が示される。1つの実施形態においては、方法700は、関係マネージャ（例えば、図1の関係マネージャ150）によって実施される。方法700はステップ710において開始する。

【0071】

ステップ720において、抽象照会が準備される。準備された抽象照会は、1つ又は複数の結果フィールドと、1つ又は複数の選択基準とを含むことができる。1つの実施形態においては、抽象照会を準備することには、メモリ（例えば、図1のメイン・メモリ116又はデータベース139）から抽象照会を取得することが含まれる。別の実施形態においては、抽象照会を準備することには、実行時に抽象照会を生成することが含まれる。例えば、ユーザは、適切なユーザー・インターフェースを用いて、抽象照会についての結果フィールド仕様（例えば、図3の結果フィールド仕様306）として、対応するデータ抽象化モデル（例えば、図1のデータ抽象化モデル132）から論理フィールドを選択することができる。ユーザは、さらに、適切なユーザー・インターフェースを用いて、抽象照会についての所望の選択基準（例えば、図3の選択基準304）を指定することができる。ユーザが所望の選択基準を指定することを可能にする例示的なユーザー・インターフェースが、図12及び図13を参照して後述される。

【0072】

準備された抽象照会の例として、例示的な抽象照会が以下の表7に示される。分かりやすく簡略化するために、例示的な抽象照会は、200より大きな結果値を持つ所との血液検査を指示した病院の医師（要求者）の名前を取得するために、言葉で表現した要求として定められる。

【表7】

表7－抽象照会の例

```

001 FIND
002      /Patient/Tests/Requester/Name
003 WHERE
004      /Patient/Tests/Results>200

```

【0073】

ステップ730において、準備された抽象照会に対応するデータ抽象化モデルが取得される。所との例においては、表5のデータ抽象化モデルが取得される。ステップ740において、取得されたデータ抽象化モデルが抽象照会に関する論理リンクを含むかどうかを判断するために、確認が行われる。より具体的には、上述したように、抽象照会は論理フィールドに従って合成される。換言すれば、結果フィールド仕様及び選択基準は、論理フィールド及び対応する値から合成される。このように、ステップ740において、抽象照

10

20

30

40

50

会を合成する論理フィールドに対応するそれぞれの論理フィールド仕様が1つ又は複数の論理リンクを含むかどうかを判断することができる。さらに、それぞれの論理フィールド仕様が、1つ又は複数の関連論理リンクを有するカテゴリー仕様に含まれるかどうかを判断することができる。

【0074】

所与の例において、表7の抽象照会は、表5のデータ抽象化モデルの行028～032に記述される論理条件フィールド「Results」(行004)を含む検索基準(行003～004)を有する。より具体的には、条件フィールド「Results」は、表5のデータ抽象化モデルに対応する論理ツリー構造を通るパス、すなわち「/Patient/Tests/Results」に関して定められる。このパスを含む例示的な論理ツリー構造が、図11を参照して後述される。1つの実施形態においては、このパスは、ステップ740において表5のデータ抽象化モデルを用いて再構成される。10

【0075】

より具体的には、所与の例において、パスは、論理ツリー構造すなわち表5のデータ抽象化モデルを、「Patient」カテゴリー(表5の行003～021)から「Tests」カテゴリー(表5の行029～041)まで探索する。「Tests」カテゴリーは、抽象照会の条件フィールド「Results」に対応する論理フィールド仕様(表5の行029～033)を含む。「Results」フィールドに到達するために、表5のデータ抽象化モデルの「Patient」カテゴリーから「Tests」カテゴリーまでの探索が、論理リンク「Patient_To_Tests」に基づいて実施される。対応するリンク仕様(例えば、図4のリンク仕様380₁)は、表5の行020に示される。関連リンク定義(例えば、図4のリンク定義370₁)は、表5の行043～048に示される。さらに、表7の抽象照会は、表5のデータ抽象化モデルの行009～013に記述される論理フィールド「Name」を、結果フィールド(表7の行002)として含む。「Name」フィールドについて、表5の行038～039に示されるリンク仕様と、表5の行049～058に示される対応するリンク定義とを、上述した条件フィールドの特定と同様の処理で特定することができる。20

【0076】

ステップ740において、取得されたデータ抽象化モデルが抽象照会に関する論理リンクを含まないと判断された場合には、抽象照会は、ステップ750において、図5及び図6にそれぞれ示される方法400及び500に従って具象照会に変換される。次いで方法700はステップ780において終了する。30

【0077】

しかしながら、取得されたデータ抽象化モデルが関連論理リンクを含む場合には、該関連論理リンクは取得されたデータ抽象化モデルから特定される。次いで、ステップ770において、特定された論理リンクを用いて、抽象照会が具象照会に変換される。1つの実施形態においては、抽象照会は、図5及び図6にそれぞれ示される方法400及び500に従って具象照会に変換される。例えば、抽象照会はSQL照会に変換される。次いで、特定された論理リンクは、表8に関して以下に示されるように、SQL照会についてのJOINステートメントに変換される。次いで方法700はステップ780において終了する。40

【0078】

より具体的には、所与の例において、表7の抽象照会は、上述の特定された論理リンクを用いて、以下の表8に示される具象照会に変換することができる。例として、例示的な具象照会は、SQLを用いて定められる。しかしながら、他のあらゆる言語を用いることができるという利点を有する。

【表8】

表8—例示的な具象SQL照会

```

001 SELECT
002   T4.name
003   FROM
004   Patientinfo as T1
005   JOIN
006   Bloodtest as T2 on T1.patient_ID=T2.patient_ID
007   JOIN
008   Employeeinfo as T3 on T2.requester=T3.Employeeinfo
009   JOIN
010  Patientinfo as T4 on T3.patient_ID=T4.patient_ID
011 WHERE
012   T2.results>200

```

10

【0079】

例として、具象SQL照会は、行005～010にJOINステートメントを含む。JOINステートメントは、行006において、表5のデータ抽象化モデルの行043～048のリンク定義に従ってデータベース表「Patientinfo」(表3)とデータベース表「Bloodtest」(表4)との間の関係を定める。表5のデータ抽象化モデルの行049～058におけるリンク定義に従って、行008におけるJOINステートメントは、データベース表「Bloodtest」(表4)とデータベース表「Employeeinfo」(表6)との間の関係を定め、行010におけるJOINステートメントは、データベース表「Employeeinfo」(表6)とデータベース表「Patientinfo」(表3)との間の関係を定める。

20

【0080】

上述したように、データ抽象化モデル(例えば、図3のデータ抽象化モデル132)は、複数の論理分岐を有する論理ツリー構造によって表現することができる。複数のデータ抽象化モデルが与えられる1つの実施形態においては、各々のデータ抽象化モデルは、対応する論理分岐によって表現することができる。複数のカテゴリー仕様(例えば、図3のカテゴリー仕様310₁及び301₂)を持つ単一のデータ抽象化モデルのみが与えられる別の実施形態においては、各々のカテゴリーは、対応する論理分岐として表現することができる。換言すれば、様々な手法が、1つ又は複数のデータ抽象化モデルを論理ツリー構造として表現するのに適していることに留意すべきである。こうした手法の全てが広く考えられる。データ抽象化モデルを表現する例示的な論理ツリー構造が、図9に示される。

30

【0081】

ここで図9を参照すると、図3(表2)のデータ抽象化モデル132を表現する例示的な論理ツリー構造が示される。例示的な論理ツリー構造は、例として、抽象化モデル132のカテゴリー「Patient」310₁を表す第1の論理分岐800、及び、カテゴリー「Tests」310₂を表す第2の論理分岐850を含む。したがって、第1の論理分岐800は、カテゴリー名「Patient」(すなわち、カテゴリー名330₁)を示す親ノード810を含む。さらに、親ノード810は、「Patient」カテゴリーに含まれる論理フィールドの各々について1つずつ、関連子ノード812、814、及び816を有する。より具体的には、ノード812は論理フィールド「PatientID」308₁に対応し、ノード814は論理フィールド「Name」308₂に対応し、ノード816は論理フィールド「Street」308₃に対応する。同様に、第2の論理分岐850は、カテゴリー名「Tests」(すなわち、カテゴリー名330₂)を示す親ノード820を含む。親ノード820は、「Tests」カテゴリーに含まれる論理フィールドの各々について1つずつ、関連子ノード822、824、及び826を有する。より具体的には、ノード822は論理フィールド「NormalizedResults」308₄に対応し、ノード824は論理フィールド「Results」308₅

40

50

に対応し、ノード 826 は論理フィールド「Requester」3086 に対応する。

【0082】

表5に関して上述したように、第1の論理分岐800と第2の論理分岐850との間に、2つの関係が特定された。特定された関係は、図10に示される。したがって、第1の関係「Patient_has_Tests」830が、親ノード810及び820の間に特定された。例として、関係830は、「has」タイプの関係である。第2の関係「Requester_is_a_Patient」840が、子ノード826と親ノード810との間に特定された。例として、関係840は、「is_a」タイプの関係である。特定された関係830及び840を用いて論理分岐800及び850を接続し、図11に示されるように、リンクされた論理ツリー構造を形成することができる。

10

【0083】

ここで図11を参照すると、例示的なリンクされた論理ツリー構造860が示される。例示的なリンクされた論理ツリー構造860は、図10の特定された関係830及び840を用いて、図9の論理分岐800及び850から生成された。具体的には、図11のリンクされた論理ツリー構造860は、図4(表5)のデータ抽象化モデル350を表現する。

【0084】

例として、リンク論理ツリー構造860においては、特定された関係830に従って、論理分岐850が論理分岐800に追加された。さらに、特定された関係840に従って、子ノード862、864、及び866が「Requester」ノード826に追加された。追加された子ノード862、864、及び866は、それぞれ子ノード812、814、及び816によって与えられる情報に対応する、「Requester」ノード826に関する情報を与える。しかしながら、論理分岐800及び850は、依然として互いに独立にアクセス可能であることに留意すべきである。例えば、ユーザは、依然として、論理分岐850を通じてノード820からノード826にナビゲートすることができる。換言すれば、リンク論理ツリー構造860は、論理分岐800及び850を置き換えない。その代わり、リンク論理ツリー構造は、これらの論理分岐とは別に生成される。

20

【0085】

上述したように、ユーザは、抽象照会についての結果フィールドとしてデータ抽象化モデルから論理フィールドを選択するために、適切なユーザー・インターフェースを用いることができる。ユーザは、さらに、抽象照会について所望の選択基準を指定するために、適切なユーザー・インターフェースを用いることができる。ユーザが所望の選択基準を指定することを可能にする例示的なユーザー・インターフェースが、図12及び図13を参照して以下に説明される。

30

【0086】

ここで図12を参照すると、例示的なユーザー・インターフェース900が示される。例示的なユーザー・インターフェース900は、抽象照会(例えば、図3の抽象照会202)の指定が可能になるように設定される。例として、ユーザー・インターフェース900は、カテゴリー名(例えば、図3のカテゴリー名330₁及び330₂)のリスト912を示す検索基準選択領域910を表示する。各々のカテゴリー名は、図形選択要素と関連付けられる。例として、図形選択要素はチェックボックスとして示される。例えば、カテゴリー名「Patient」はチェックボックス914と関連付けられ、カテゴリー名「Tests」はチェックボックス916と関連付けられる。1つの実施形態においては、コンピュータ・マウス、ライトペン、又はタッチスクリーンの場合には人間の指も含む、ポインティング・デバイスなどの適切な入力装置のいずれかを用いて、対応するチェックボックスを選択することができる。

40

【0087】

1つの態様によれば、チェックボックス914及び916の1つを選択して、関連カテゴリーに含まれる論理フィールドを、検索基準を定める条件フィールドとして指定することができる。この目的のために、リスト912における各々のカテゴリー名は、対応する

50

カテゴリーに含まれる論理フィールドを示す図形要素と関連付けられる。例として、「Patient」カテゴリーはドロップダウン・リスト922と関連付けられ、「Tests」カテゴリーはドロップダウン・リスト924と関連付けられる。しかしながら、ドロップダウン・リストのような図形要素を実装することは、単なる一例として示されるものであり、従って本発明を限定するためのものではないことに留意すべきである。代わりに、ポップアップ・ウィンドウなどといったいずれかの図形要素も考えられる。さらに、各々のカテゴリーの論理フィールドは、対応する論理ツリー構造の論理分岐表現（例えば、図9の論理分岐800及び850）を用いて表示することができる。

【0088】

例として、チェックボックス914を選択し、「Patient」カテゴリー（例えば、図3のカテゴリー3101）から条件フィールドを選択した。1つの実施形態においては、次いで、ドロップダウン・リスト922をドロップダウンし、選択可能な論理フィールド932のリストを表示することができる。例として、選択可能な論理フィールド932のリストは、図11のリンクされた論理ツリー構造860のノード812、814、816、及び820に対応するすべての論理フィールドを含む。選択可能な論理フィールドの1つを選択すると、リンクされた論理ツリー構造860の探索に用いられた対応するパスが追跡される。例として、ユーザが、論理フィールド「Tests」を条件フィールドとして選択したと仮定する。その結果、ユーザが「Patient」カテゴリーの選択後に「Tests」フィールドを選択したときは、パス「/Patient/Tests」が生成されることになる。次いで、ユーザは、プッシュボタン「次へ」952をクリックして検索基準の選択を続行するか、又はプッシュボタン「取消」954をクリックして選択を中止することができる。ここでユーザが、「Tests」フィールドの選択後にプッシュボタン「次へ」952をクリックしたと仮定する。その結果、図13に示されるように、別の選択領域がユーザー・インターフェース900に表示され、検索基準の選択を続行することができる。

【0089】

検索基準選択領域910は、ドロップダウン・リスト922及び924がリンクされた論理分岐に基づいて論理フィールドを表示すべきかどうかを指定するようにユーザに促す表示944をさらに含む。表示944はチェックボックス942と関連付けられる。より具体的には、チェックボックス942が選択された場合には、ドロップダウン・リスト922及び924は、論理リンクを用いてデータ抽象化モデルについて生成された基礎論理ツリー構造に基づく論理フィールドを表示する。しかしながら、チェックボックス942が選択されない場合には、ドロップダウン・リスト922及び924は、論理リンクを用いてデータ抽象化モデルについて生成された基礎論理ツリー構造に基づく論理フィールドを表示する。

【0090】

ここで図13を参照すると、ユーザー・インターフェース900は、例示的な条件フィールド選択領域960を表示する。例として、条件フィールドを選択することができる例示的な条件フィールド選択領域960は、カテゴリーの表示982を含む。選択することができる条件フィールドが、指定されたカテゴリーにリンクされる論理分岐に含まれる場合には、条件フィールド選択領域960は、論理リンクの表示984をさらに含む。例として、表示984は、図10の特定された関係830に対応する論理リンク「Patient_has_Tests」986を記述する。換言すれば、特定された関係830を用いてリンクされた論理ツリー構造860を探索することによって、「Patient」カテゴリーから、ドロップダウン・リスト962に表示されるユーザ選択可能フィールドのすべてに到達する。

【0091】

ドロップダウン・リスト962から選択可能な論理フィールドの1つを選択すると、リンクされた論理ツリー構造860の探索に用いられた対応するパスが追跡される。例として、ユーザが、論理フィールド「Results」を条件フィールドとして選択したと仮

10

20

30

40

50

定する。その結果、ユーザが「Patient」カテゴリーから「Tests」フィールドを選択した後で「Results」フィールドを選択したときは、パス「/Patient/Tests/Results」を生成することができる。次いで、ユーザは、プッシュボタン「次へ」972をクリックして検索基準の選択を続行するか、又はプッシュボタン「取消」974をクリックして選択を中止することができる。ここでユーザが、「Results」フィールドの選択後にプッシュボタン「次へ」972をクリックしたと仮定する。その結果、ユーザが検索基準について対応する値及び演算子を指定することが可能になるように、別の選択領域を表示することができる。ここで、ユーザが、演算子として「>」を指定し、対応する値として「200」を指定したと仮定する。その結果、ユーザは、表7(行004)の抽象照会の選択基準「/Patient/Tests/Results > 200」を生成した。
10

【0092】

関連論理分岐間に論理リンクを生成することによって、異なって解釈される場合がある論理フィールドを持つ照会を生成するときのユーザの意図に関する情報を、自動的に収集することができる。この情報を用いると、こうした照会を適切に解釈することができ、実行時に適切に表を結合することができる。

【0093】

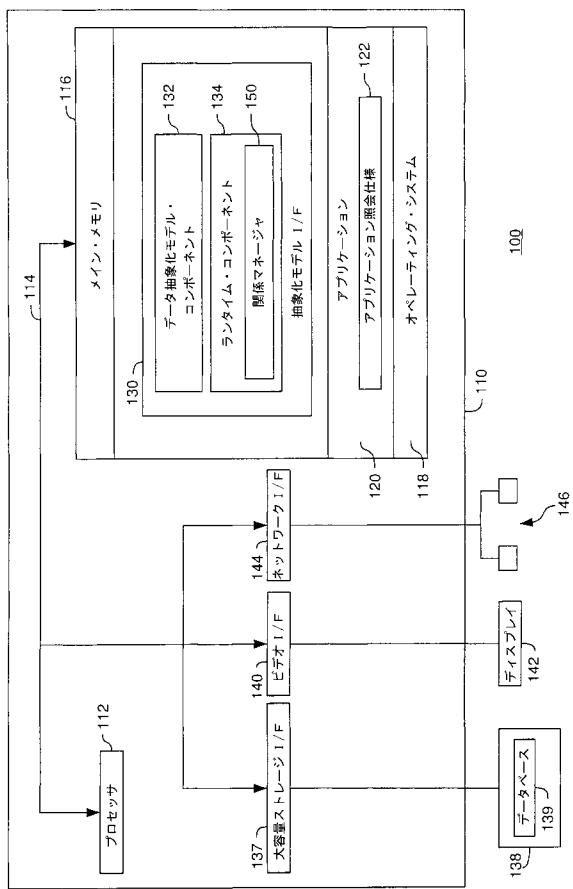
ここでは、特定の値、定義、プログラミング言語、及び例に対するいかなる参照も、単なる説明目的に過ぎないことに留意すべきである。したがって、本発明は、いかなる特定の説明図又は例によっても限定されるものではない。さらに、上記は本発明の実施形態に向けられるものであるが、本発明の他の実施形態及びさらなる実施形態は、その基本的な範囲から逸脱することなく考案することができ、その範囲は、添付の特許請求の範囲によって決定される。
20

【図面の簡単な説明】

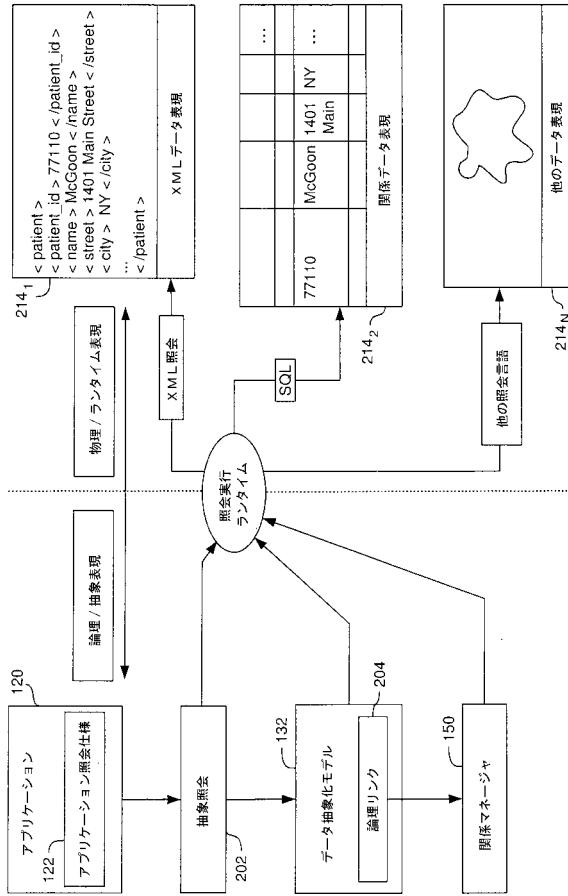
【0094】

- 【図1】本発明によって例として用いられるコンピュータ・システムである。
- 【図2】抽象照会管理のためのソフトウェア・コンポーネントの関係図である。
- 【図3】抽象照会管理のためのソフトウェア・コンポーネントの関係図である。
- 【図4】抽象照会管理のためのソフトウェア・コンポーネントの関係図である。
- 【図5】ランタイム・コンポーネントの動作を示すフロー・チャートである。
30
- 【図6】ランタイム・コンポーネントの動作を示すフロー・チャートである。
- 【図7】1つの実施形態におけるデータ抽象化モデルの関係管理を示すフロー・チャートである。
- 【図8】1つの実施形態における抽象照会を生成するためのデータ抽象化モデルにおいて、関係管理の適用を示すフロー・チャートである。
- 【図9】1つの実施形態におけるリンクされた論理分岐を示すツリー構造である。
- 【図10】1つの実施形態におけるリンクされた論理分岐を示すツリー構造である。
- 【図11】1つの実施形態におけるリンクされた論理分岐を示すツリー構造である。
- 【図12】1つの実施形態における抽象照会を指定するために構成されたユーザー・インターフェースを示すスクリーン・ショットである。
40
- 【図13】1つの実施形態における抽象照会を指定するために構成されたユーザー・インターフェースを示すスクリーン・ショットである。

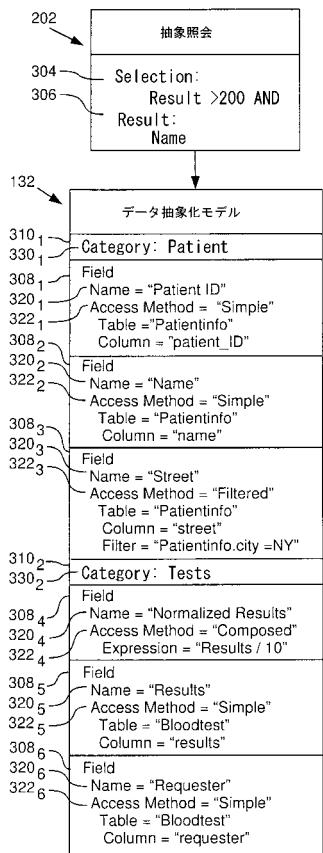
【図1】



【図2】



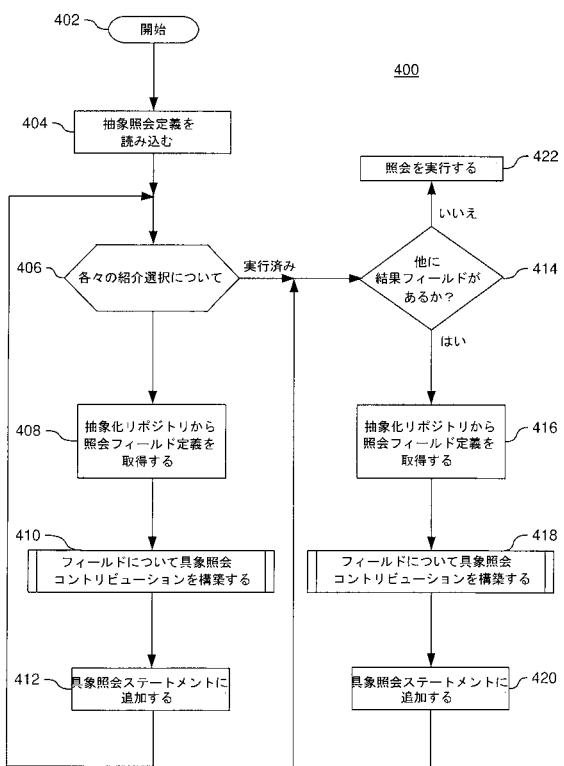
【図3】



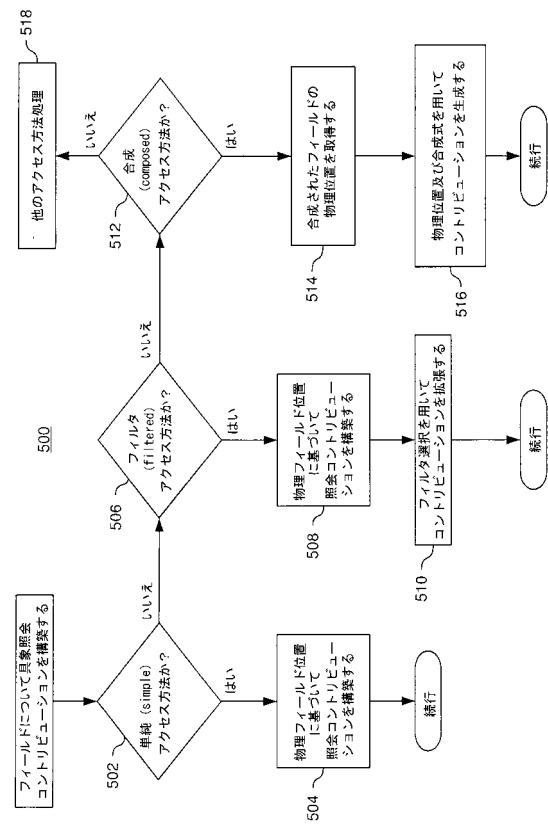
【図4】

データ抽象化モデル	
310 ₁	Category: Patient
308 ₁	Field Name = "Patient ID" Access Method = "Simple" Table = "Patientinfo" Column = "patient_ID"
380 ₁	⋮
380 ₂	Relationship to = "/Tests" type = "has" relation = "Patient_To_Tests"
380 ₃	⋮
310 ₂	Category: Tests
308 ₄	Field Name = "Normalized Results" Access Method = "Composed" Expression = "Results / 10"
360	⋮
370 ₁	Relations
370 ₂	Relation Name = "Patient_To_Tests" Link ID = "Patients_To_Tests1" source = "Patientinfo" target = "Bloodtest"
370 ₃	⋮
370 ₄	Link point source = "patient_ID" target = "patient_ID"
370 ₅	⋮

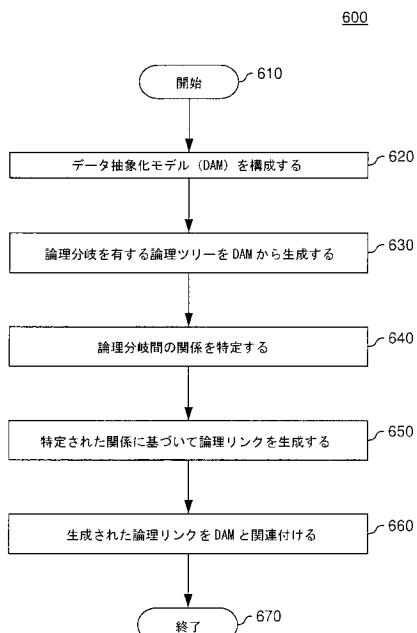
【図5】



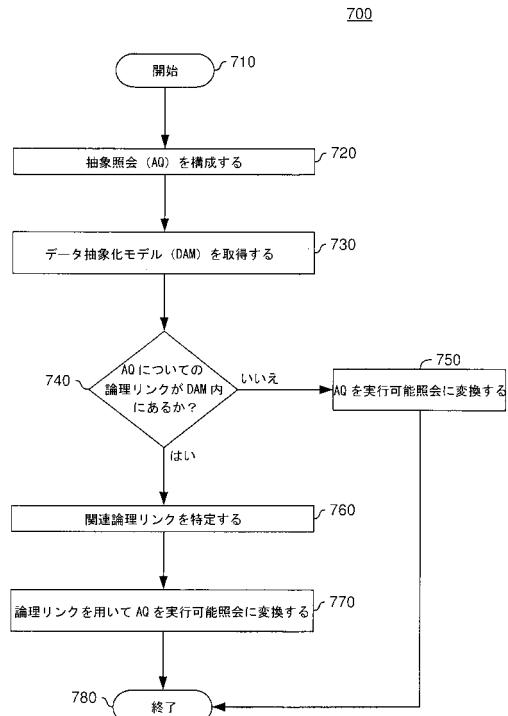
【図6】



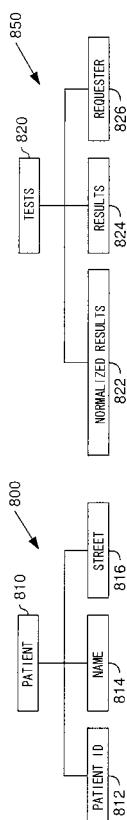
【図7】



【図8】



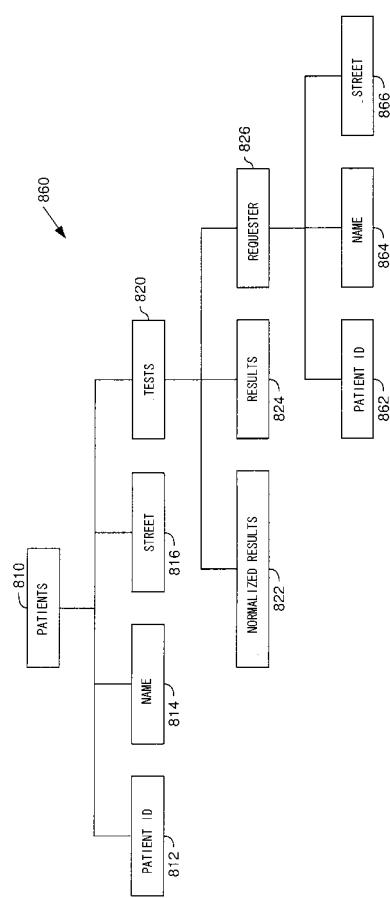
【図 9】



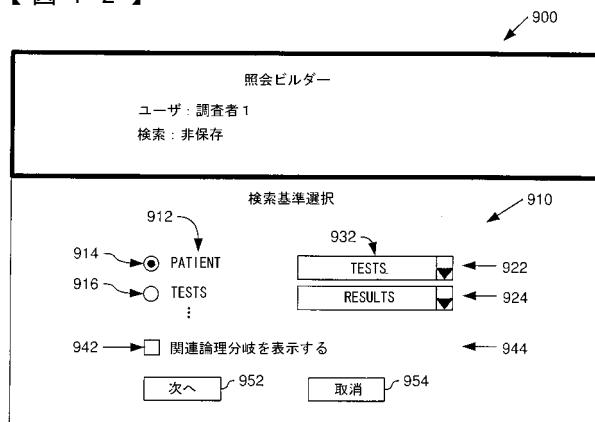
【図 10】



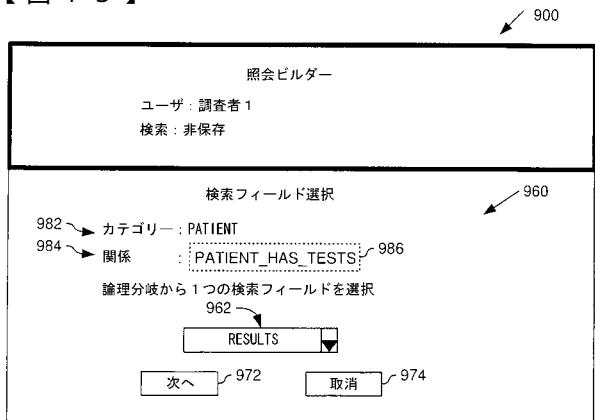
【図 11】



【図 12】



【図13】



フロントページの続き

(72)発明者 リチャード・ディー・デッティンガー
アメリカ合衆国 55901 ミネソタ州 ロチェスター ケンジントン・レーン ノース・ウェ
スト 5305

(72)発明者 ダニエル・ピー・コルツ
アメリカ合衆国 55901 ミネソタ州 ロチェスター フォース・プレイス ノース・ウェス
ト 3833

(72)発明者 リチャード・ジェイ・スティーブンス
アメリカ合衆国 55901 ミネソタ州 ロチェスター アーボー・ドライブ ノース・ウェス
ト 3707

(72)発明者 シャノン・イー・ウェンツェル
アメリカ合衆国 54421 ウィスコンシン州 コルビー カウンティー・ハイウェイ・ノース
F 2389

F ターム(参考) 5B075 NK44 QT06