

(19)日本国特許庁(JP)

## (12)特許公報(B2)

(11)特許番号

特許第7036597号

(P7036597)

(45)発行日 令和4年3月15日(2022.3.15)

(24)登録日 令和4年3月7日(2022.3.7)

(51)国際特許分類

F I

G 0 6 F 16/182 (2019.01)

G 0 6 F 16/182

G 0 6 F 16/28 (2019.01)

G 0 6 F 16/28

請求項の数 60 (全36頁)

(21)出願番号	特願2017-543965(P2017-543965)	(73)特許権者	509123208
(86)(22)出願日	平成28年2月16日(2016.2.16)		アビニシオ テクノロジー エルエルシー
(65)公表番号	特表2018-514012(P2018-514012 A)		アメリカ合衆国 0 2 4 2 1 マサチュー
(43)公表日	平成30年5月31日(2018.5.31)		セッツ州 レキシントン スプリング ス
(86)国際出願番号	PCT/US2016/018028	(74)代理人	トリート 2 0 1
(87)国際公開番号	WO2016/133880		100079108
(87)国際公開日	平成28年8月25日(2016.8.25)	(74)代理人	弁理士 稲葉 良幸
審査請求日	平成31年2月8日(2019.2.8)	(74)代理人	100109346
(31)優先権主張番号	62/117,588	(74)代理人	弁理士 大貫 敏史
(32)優先日	平成27年2月18日(2015.2.18)	(74)代理人	100117189
(33)優先権主張国・地域又は機関	米国(US)	(74)代理人	弁理士 江口 昭彦
(31)優先権主張番号	14/752,094	(74)代理人	100134120
(32)優先日	平成27年6月26日(2015.6.26)	(72)発明者	弁理士 内藤 和彦
	最終頁に続く		シェヒター, イアン
			アメリカ合衆国, マサチューセッツ州
			最終頁に続く

(54)【発明の名称】 ネットワーク上のデータソースへの照会

## (57)【特許請求の範囲】

## 【請求項 1】

リレーショナルデータベースに適用可能なクエリ言語に従って表現されたクエリに基づいてコンピュータプログラムを実行する、データ処理システムによって実施されるコンピュータ実施方法であって、前記コンピュータプログラムは、前記データ処理システムの有形の非一時的なコンピュータ可読媒体内に記憶された構成データに少なくとも一部基づいて実行され、前記実行することは、

S Q Lクエリを受信し、前記S Q Lクエリは、前記データ処理システムの外部にあるリソースに関連付けられた識別子を含むことと、

前記S Q Lクエリに基づいてコンピュータプログラムを生成することであって、前記生成することは、

前記リソースに関連付けられた少なくとも1つのエントリを有するカタログを識別し、前記少なくとも1つのエントリは、データを受信する目的のために、前記リソースに対する命令を送信するように、コンピュータプログラムを構成するのに使用可能である構成データを表すこと、

前記S Q Lクエリに基づいてクエリプランナへの要求を生成すること、

前記要求を前記クエリプランナへ提供すること、

前記要求に基づいて前記クエリプランナによって生成されたクエリプランを受信し、前記クエリプランは、リレーショナルデータベースを管理するシステムによって実施されるべき1つまたは複数のステップの記述を含むこと、

前記クエリプランに基づいて、前記コンピュータプログラムを生成し、前記コンピュータプログラムの少なくとも一部は、前記カタログの前記少なくとも1つのエントリに基づいて前記リソースと通信するように構成されていること、

を含む、生成することと、

前記識別された前記カタログから前記構成データを受信することと、

前記データ処理システムの実行システム上で、前記構成データに基づいて前記コンピュータプログラムを実行することとを含み、

前記コンピュータプログラムの前記実行は、

1つまたは複数の命令を前記リソースへ伝送し、前記命令は前記SQLクエリによって指定された操作と等価の操作を実施するように前記リソースによって解釈可能であること、および

前記命令に応答して前記リソースからのデータを受信すること、  
を引き起こす、方法。

#### 【請求項2】

前記コンピュータプログラムの部分が、前記リソースがアクセスされる仕方を定義する実行可能命令を含み、前記実行可能命令は前記コンピュータプログラムの前記部分に提供された前記構成データに基づいて動作する、請求項1に記載の方法。

#### 【請求項3】

前記構成データが、前記リソースによって用いられるデータフォーマットに対する変更に基づいて更新されることができカタログ内で指定される、請求項1に記載の方法。

#### 【請求項4】

前記構成データに基づいてパラメータ値を生成することと、

前記パラメータ値を前記コンピュータプログラムの部分へ提供し、前記部分は前記実行システムに前記リソースと通信させる機能を有することと、

を含む、

前記コンピュータプログラムの前記実行が前記パラメータ値に基づく、請求項1に記載の方法。

#### 【請求項5】

前記構成データに基づいてパラメータ値を生成することが、パラメータファイルを、前記コンピュータプログラムの前記部分によって可読のフォーマットで生成するパラメータジェネレータを実行することを含み、

前記パラメータ値を前記コンピュータプログラムの前記部分へ提供することが、前記パラメータファイルを前記コンピュータプログラムの前記部分が利用できるようにすることを含む、請求項4に記載の方法。

#### 【請求項6】

前記SQLクエリが、引き数を含むSELECT文を含み、前記引き数の少なくとも一部分は、前記リソースに関連付けられた前記識別子に対応する、請求項1に記載の方法。

#### 【請求項7】

前記コンピュータプログラムが、前記SQLクエリの操作を表すコンポーネントを含む、請求項1に記載の方法。

#### 【請求項8】

前記コンピュータプログラムがデータフローグラフであり、前記コンピュータプログラムの前記部分が前記データフローグラフの部分グラフである、請求項1に記載の方法。

#### 【請求項9】

前記リソースが、前記リソースによって公開されたアプリケーションプログラミングインターフェース(API)を用いてアクセスされる、請求項1に記載の方法。

#### 【請求項10】

前記SQLクエリを受信したことに応答して前記APIの関数を実行させることを含み、前記APIの前記関数は前記リソースにおいて実行可能であり、前記リソースへ伝送された前記命令が前記APIの前記関数を実行させる、請求項9に記載の方法。

10

20

30

40

50

## 【請求項 1 1】

前記外部リソースから受信された前記データをデータベーステーブルの形式でフォーマットすることを含む、請求項 1 に記載の方法。

## 【請求項 1 2】

前記リソースがリレーショナルデータベース管理システムでない、請求項 1 に記載の方法。

## 【請求項 1 3】

前記命令が、SQLクエリに回答して結果を返さない前記リソースのファシリティへ伝送される、請求項 1 に記載の方法。

## 【請求項 1 4】

前記命令に回答して前記リソースから受信された前記データが、前記SQLクエリによって指定されたデータを含む、請求項 1 に記載の方法。

10

## 【請求項 1 5】

前記命令に回答して前記リソースから受信された前記データ内のレコードおよびフィールドを識別することを含み、前記レコードおよびフィールドは、前記データ処理システムの外部にある前記リソースに関連付けられたレコードフォーマットに基づいて識別される、請求項 1 に記載の方法。

## 【請求項 1 6】

リレーショナルデータベースに適用可能なクエリ言語に従って表現されたクエリに基づいてコンピュータプログラムを実行する能力を有するデータ処理システムであって、前記コンピュータプログラムは、前記データ処理システムの有形の非一時的なコンピュータ可読媒体内に記憶された構成データに少なくとも一部基づいて実行され、前記データ処理システムは、

20

SQLクエリを受信し、前記SQLクエリは、前記データ処理システムの外部にあるリソースに関連付けられた識別子を含むことと、

前記SQLクエリに基づいてコンピュータプログラムを生成することであって、

前記リソースに関連付けられた少なくとも1つのエントリを有するカタログを識別し、前記少なくとも1つのエントリは、データを受信する目的のために、前記リソースに対する命令を送信するように、コンピュータプログラムを構成するのに使用可能である構成データを表すこと、

前記SQLクエリに基づいてクエリプランナへの要求を生成すること、

30

前記要求を前記クエリプランナへ提供すること、

前記要求に基づいて前記クエリプランナによって生成されたクエリプランを受信し、前記クエリプランは、リレーショナルデータベースを管理するシステムによって実施されるべき1つまたは複数のステップの記述を含むこと、

前記クエリプランに基づいて、前記コンピュータプログラムを生成し、前記コンピュータプログラムの少なくとも一部は、前記カタログの前記少なくとも1つのエントリに基づいて前記リソースと通信するように構成されていること、

を含む、生成することと、

前記識別された前記カタログから前記構成データを受信することと、

前記データ処理システムの実行システム上で、前記構成データに基づいて前記コンピュータプログラムを実行することと

40

を含む、演算を実施し、

前記コンピュータプログラムの前記実行は、

1つまたは複数の命令を前記リソースへ伝送し、前記命令は前記SQLクエリによって指定された操作と等価の操作を実施するように前記リソースによって解釈可能であること、および

前記命令に回答して前記リソースからのデータを受信すること、

を含む、データ処理システム。

## 【請求項 1 7】

前記コンピュータプログラムの部分が、前記リソースがアクセスされる仕方を定義する実

50

行可能命令を含み、前記実行可能命令は前記コンピュータプログラムの前記部分に提供された前記構成データに基づいて動作する、請求項 1 6 に記載のシステム。

【請求項 1 8】

前記構成データが、前記リソースによって用いられるデータフォーマットに対する変更に基づいて更新されることができカタログ内で指定される、請求項 1 6 に記載のシステム。

【請求項 1 9】

前記演算は、

前記構成データに基づいてパラメータ値を生成することと、

前記パラメータ値を前記コンピュータプログラムの部分へ提供し、前記部分は前記実行システムに前記リソースと通信させる機能を有することと、

を含み、

前記コンピュータプログラムの前記実行が前記パラメータ値に基づく、請求項 1 6 に記載のシステム。

【請求項 2 0】

前記構成データに基づいてパラメータ値を生成することが、パラメータファイルを、前記コンピュータプログラムの前記部分によって可読のフォーマットで生成するパラメータジェネレータを実行することを含み、

前記パラメータ値を前記コンピュータプログラムの前記部分へ提供することが、前記パラメータファイルを前記コンピュータプログラムの前記部分が利用できるようにすることを含み、請求項 1 9 に記載のシステム。

【請求項 2 1】

前記 S Q L クエリが、引き数を含む S E L E C T 文を含み、前記引き数の少なくとも一部分は、前記リソースに関連付けられた前記識別子に対応する、請求項 1 6 に記載のシステム。

【請求項 2 2】

前記コンピュータプログラムが、前記 S Q L クエリの操作を表すコンポーネントを含む、請求項 1 6 に記載のシステム。

【請求項 2 3】

前記コンピュータプログラムがデータフローグラフであり、前記コンピュータプログラムの前記部分が前記データフローグラフの部分グラフである、請求項 1 6 に記載のシステム。

【請求項 2 4】

前記リソースが、前記リソースによって公開されたアプリケーションプログラミングインターフェース ( A P I ) を用いてアクセスされる、請求項 1 6 に記載のシステム。

【請求項 2 5】

前記演算は、

前記 S Q L クエリを受信したことに応答して前記 A P I の関数を実行させることを含み、前記 A P I の前記関数は前記リソースにおいて実行可能であり、前記リソースへ伝送された前記命令が前記 A P I の前記関数を実行させる、請求項 2 4 に記載のシステム。

【請求項 2 6】

前記演算は、

前記外部リソースから受信された前記データをデータベーステーブルの形式でフォーマットすることを含む、請求項 1 6 に記載のシステム。

【請求項 2 7】

前記リソースがリレーショナルデータベース管理システムでない、請求項 1 6 に記載のシステム。

【請求項 2 8】

前記命令が、S Q L クエリに応答して結果を返さない前記リソースのファシリティへ伝送される、請求項 1 6 に記載のシステム。

【請求項 2 9】

前記命令に応答して前記リソースから受信された前記データが、前記 S Q L クエリによっ

10

20

30

40

50

て指定されたデータを含む、請求項 1\_6 に記載のシステム。

【請求項 3 0】

前記演算は、

前記命令に応答して前記リソースから受信された前記データ内のレコードおよびフィールドを識別することを含み、前記レコードおよびフィールドは、前記データ処理システムの外部にある前記リソースに関連付けられたレコードフォーマットに基づいて識別される、請求項 1\_6 に記載のシステム。

【請求項 3 1】

データ処理システムが、リレーショナルデータベースに適用可能なクエリ言語に従って表現されたクエリに基づいてコンピュータプログラムを実行することを可能にする命令を記憶する非一時的なコンピュータ可読記憶デバイスであって、前記コンピュータプログラムは、前記データ処理システムの有形の非一時的なコンピュータ可読媒体内に記憶された構成データに少なくとも一部基づいて実行され、前記命令は、前記データ処理システムに、SQLクエリを受信し、前記SQLクエリは、前記データ処理システムの外部にあるリソースに関連付けられた識別子を含むことと、

前記SQLクエリに基づいてコンピュータプログラムを生成することであって、

前記リソースに関連付けられた少なくとも1つのエントリを有するカタログを識別し、前記少なくとも1つのエントリは、データを受信する目的のために、前記リソースに対する命令を送信するように、コンピュータプログラムを構成するのに使用可能である構成データを表すこと、

前記SQLクエリに基づいてクエリプランナへの要求を生成すること、

前記要求を前記クエリプランナへ提供すること、

前記要求に基づいて前記クエリプランナによって生成されたクエリプランを受信し、前記クエリプランは、リレーショナルデータベースを管理するシステムによって実施されるべき1つまたは複数のステップの記述を含むこと、

前記クエリプランに基づいて、前記コンピュータプログラムを生成し、前記コンピュータプログラムの少なくとも一部は、前記カタログの前記少なくとも1つのエントリに基づいて前記リソースと通信するように構成されていること、

を含む、生成することと、

前記識別された前記カタログから前記構成データを受信することと、

前記データ処理システムの実行システム上で、前記構成データに基づいて前記コンピュータプログラムを実行することとを含み演算を実施させ、

前記コンピュータプログラムの前記実行は、

1つまたは複数の命令を前記リソースへ伝送し、前記命令は前記SQLクエリによって指定された操作と等価の操作を実施するように前記リソースによって解釈可能であること、および

前記命令に応答して前記リソースからのデータを受信すること、

を含む、非一時的なコンピュータ可読記憶デバイス。

【請求項 3 2】

前記コンピュータプログラムの部分が、前記リソースがアクセスされる仕方を定義する実行可能命令を含み、前記実行可能命令は前記コンピュータプログラムの前記部分に提供された前記構成データに基づいて動作する、請求項 3\_1 に記載の方法。

【請求項 3 3】

前記構成データが、前記リソースによって用いられるデータフォーマットに対する変更に基づいて更新されることができカタログ内で指定される、請求項 3\_1 に記載のコンピュータ可読記憶デバイス。

【請求項 3 4】

前記演算は、

前記構成データに基づいてパラメータ値を生成することと、

前記パラメータ値を前記コンピュータプログラムの部分へ提供し、前記部分は前記実行シ

10

20

30

40

50

システムに前記リソースと通信させる機能を有することと、  
を含み、

前記コンピュータプログラムの前記実行が前記パラメータ値に基づく、請求項 3 1 に記載のコンピュータ可読記憶デバイス。

【請求項 3 5】

前記構成データに基づいてパラメータ値を生成することが、パラメータファイルを、前記コンピュータプログラムの前記部分によって可読のフォーマットで生成するパラメータジェネレータを実行することを含み、

前記パラメータ値を前記コンピュータプログラムの前記部分へ提供することが、前記パラメータファイルを前記コンピュータプログラムの前記部分が利用できるようにすることを  
含む、請求項 3 4 に記載のコンピュータ可読記憶デバイス。

10

【請求項 3 6】

前記 S Q L クエリが、引き数を含む S E L E C T 文を含み、前記引き数の少なくとも一部分は、前記リソースに関連付けられた前記識別子に対応する、請求項 3 1 に記載のコンピュータ可読記憶デバイス。

【請求項 3 7】

前記コンピュータプログラムが、前記 S Q L クエリの操作を表すコンポーネントを含む、請求項 3 1 に記載のコンピュータ可読記憶デバイス。

【請求項 3 8】

前記コンピュータプログラムがデータフローグラフであり、前記コンピュータプログラムの前記部分が前記データフローグラフの部分グラフである、請求項 3 1 に記載のコンピュータ可読記憶デバイス。

20

【請求項 3 9】

前記リソースが、前記リソースによって公開されたアプリケーションプログラミングインターフェース ( A P I ) を用いてアクセスされる、請求項 3 1 に記載のコンピュータ可読記憶デバイス。

【請求項 4 0】

前記演算は、

前記 S Q L クエリを受信したことに応答して前記 A P I の関数を実行させることを含み、前記 A P I の前記関数は前記リソースにおいて実行可能であり、前記リソースへ伝送された前記命令が前記 A P I の前記関数を実行させる、請求項 3 9 に記載のコンピュータ可読記憶デバイス。

30

【請求項 4 1】

前記演算は、

前記外部リソースから受信された前記データをデータベーステーブルの形式でフォーマットすることを含む、請求項 3 1 に記載のコンピュータ可読記憶デバイス。

【請求項 4 2】

前記リソースがリレーショナルデータベース管理システムでない、請求項 3 1 に記載のコンピュータ可読記憶デバイス。

【請求項 4 3】

前記命令が、S Q L クエリに応答して結果を返さない前記リソースのファシリティへ伝送される、請求項 3 1 に記載のコンピュータ可読記憶デバイス。

40

【請求項 4 4】

前記命令に応答して前記リソースから受信された前記データが、前記 S Q L クエリによって指定されたデータを含む、請求項 3 1 に記載のコンピュータ可読記憶デバイス。

【請求項 4 5】

前記演算は、

前記命令に応答して前記リソースから受信された前記データ内のレコードおよびフィールドを識別することを含み、前記レコードおよびフィールドは、前記データ処理システムの外部にある前記リソースに関連付けられたレコードフォーマットに基づいて識別される、

50

請求項 3 1 に記載のコンピュータ可読記憶デバイス。

【請求項 4 6】

リレーショナルデータベースに適用可能なクエリ言語に従って表現されたクエリに基づいてコンピュータプログラムを実行する能力を有するデータ処理システムであって、前記コンピュータプログラムは、前記データ処理システムの有形の非一時的なコンピュータ可読媒体内に記憶された構成データに少なくとも一部基づいて実行され、前記データ処理システムは、

S Q L クエリを受信するための手段であって、前記 S Q L クエリは、前記データ処理システムの外部にあるリソースに関連付けられた識別子を含む、受信するための手段と、

前記 S Q L クエリに基づいてコンピュータプログラムを生成するための手段であって、前記リソースに関連付けられた少なくとも 1 つのエントリを有するカタログを識別することを含み、前記少なくとも 1 つのエントリは、データを受信する目的のために、前記リソースに対する命令を送信するように、コンピュータプログラムを構成するのに使用可能である構成データを表す、生成するための手段と、

前記データ処理システムの実行システム上で、前記構成データに基づいて前記コンピュータプログラムを実行するための手段と、

前記識別された前記カタログから前記構成データを受信する手段とを含み、

前記 S Q L クエリに基づいて前記コンピュータプログラムを生成する手段が、

前記 S Q L クエリに基づいてクエリプランナへの要求を生成する手段、

前記要求を前記クエリプランナへ提供する手段、

前記要求に基づいて前記クエリプランナによって生成されたクエリプランを受信し、前記クエリプランは、リレーショナルデータベースを管理するシステムによって実施されるべき 1 つまたは複数のステップの記述を含む手段、

前記クエリプランに基づいて、前記コンピュータプログラムを生成し、前記コンピュータプログラムの少なくとも一部は、前記カタログの前記少なくとも 1 つのエントリに基づいて前記リソースと通信するように構成されている手段、

を含む、

前記コンピュータプログラムの前記実行は、

1 つまたは複数の命令を前記リソースへ伝送し、前記命令は前記 S Q L クエリによって指定された操作と等価の操作を実施するように前記リソースによって解釈可能であること、および

前記命令に応答して前記リソースからのデータを受信すること、

を引き起こす、データ処理システム。

【請求項 4 7】

前記コンピュータプログラムの部分が、前記リソースがアクセスされる仕方を定義する実行可能命令を含み、前記実行可能命令は前記コンピュータプログラムの前記部分に提供された前記構成データに基づいて動作する、請求項 4 6 に記載のシステム。

【請求項 4 8】

前記構成データが、前記リソースによって用いられるデータフォーマットに対する変更に基づいて更新されることができカタログ内で指定される、請求項 4 6 に記載のシステム。

【請求項 4 9】

前記構成データに基づいてパラメータ値を生成する手段と、

前記パラメータ値を前記コンピュータプログラムの部分へ提供し、前記部分は前記実行システムに前記リソースと通信させる機能を有する手段と、

を含み、

前記コンピュータプログラムの前記実行が前記パラメータ値に基づく、請求項 4 6 に記載のシステム。

【請求項 5 0】

前記構成データに基づいてパラメータ値を生成する手段が、パラメータファイルを、前記コンピュータプログラムの前記部分によって可読のフォーマットで生成するパラメータジ

10

20

30

40

50

エネレータを実行する手段を含み、

前記パラメータ値を前記コンピュータプログラムの前記部分へ提供する手段が、前記パラメータファイルを前記コンピュータプログラムの前記部分が利用できるようにする手段を含む、請求項 4 9 に記載のシステム。

【請求項 5 1】

前記 S Q L クエリが、引き数を含む S E L E C T 文を含み、前記引き数の少なくとも一部分は、前記リソースに関連付けられた前記識別子に対応する、請求項 4 6 に記載のシステム。

【請求項 5 2】

前記コンピュータプログラムが、前記 S Q L クエリの操作を表すコンポーネントを含む、請求項 4 6 に記載のシステム。

10

【請求項 5 3】

前記コンピュータプログラムがデータフローグラフであり、前記コンピュータプログラムの前記部分が前記データフローグラフの部分グラフである、請求項 4 6 に記載のシステム。

【請求項 5 4】

前記リソースが、前記リソースによって公開されたアプリケーションプログラミングインターフェース ( A P I ) を用いてアクセスされる、請求項 4 6 に記載のシステム。

【請求項 5 5】

前記 S Q L クエリを受信したことに応答して前記 A P I の関数を実行させる手段を含み、前記 A P I の前記関数は前記リソースにおいて実行可能であり、前記リソースへ伝送された前記命令が前記 A P I の前記関数を実行させる、請求項 5 4 に記載のシステム。

20

【請求項 5 6】

前記外部リソースから受信された前記データをデータベーステーブルの形式でフォーマットする手段を含む、請求項 4 6 に記載のシステム。

【請求項 5 7】

前記リソースがリレーショナルデータベース管理システムでない、請求項 4 6 に記載のシステム。

【請求項 5 8】

前記命令が、S Q L クエリに応答して結果を返さない前記リソースのファシリティへ伝送される、請求項 4 6 に記載のシステム。

30

【請求項 5 9】

前記命令に応答して前記リソースから受信された前記データが、前記 S Q L クエリによって指定されたデータを含む、請求項 4 6 に記載のシステム。

【請求項 6 0】

前記命令に応答して前記リソースから受信された前記データ内のレコードおよびフィールドを識別する手段を含み、前記レコードおよびフィールドは、前記データ処理システムの外部にある前記リソースに関連付けられたレコードフォーマットに基づいて識別される、請求項 4 6 に記載のシステム。

【発明の詳細な説明】

【技術分野】

40

【0 0 0 1】

優先権の主張

本出願は、2015年2月18日に出願された米国特許出願第62/117,588号に対する優先権を主張する、2015年6月26日に出願された、米国特許出願第14/752,094号に対する優先権を主張する。これらの出願の内容全体は本明細書において参照により組み込まれる。

【0 0 0 2】

本記載は、例えば、S Q L または別の種類のクエリ言語を用いた、ネットワーク上のデータソースへの照会に関する。

【背景技術】

50



## 【 0 0 0 3 】

データソースに対するクエリは、データソースから取得されるべきデータを指定する。クエリはデータソース（例えば、データベース）へ提供されることが可能であり、データソースに関連付けられたデータ処理システム（例えば、データベース管理システム）が、クエリによって指定されたデータを返すことができる。クエリによって指定されたデータソース内のデータを識別するべくクエリを構文解析するために、様々な技法を用いることができる。

## 【 発明の概要 】

## 【 課題を解決するための手段 】

## 【 0 0 0 4 】

態様 1 は、リレーショナルデータベースに適用可能なクエリ言語に従って表現されたクエリに基づいてコンピュータプログラムを実行する、データ処理システムによって実施されるコンピュータ実施方法であって、コンピュータプログラムは、有形の非一時的なコンピュータ可読媒体内に記憶されたデータに少なくとも一部基づいて実行され、実行することは、SQLクエリを受信し、SQLクエリは、データ処理システムの外部にあるリソースに関連付けられた識別子を含むことと、SQLクエリに基づいてコンピュータプログラムを生成することであって、リソースに関連付けられた構成データを識別することを含み、構成データは、リソースと通信する際に用いられる少なくとも1つの値を指定する、生成することと、構成データに基づいてコンピュータプログラムを実行することと、を含み、コンピュータプログラムの実行は、1つまたは複数の命令をリソースへ伝送し、命令はSQLクエリの操作以外の操作を定義すること、および命令に応答してデータをリソースから受信すること、を引き起こす、方法である。

## 【 0 0 0 5 】

態様 2 は、コンピュータプログラムの部分が、リソースがアクセスされる仕方を定義する実行可能命令を含み、実行可能命令は、コンピュータプログラムの部分に提供された構成データに基づいて動作する、態様 1 の方法である。

## 【 0 0 0 6 】

態様 3 は、構成データが、リソースによって用いられるデータフォーマットに対する変更に基づいて更新されることができカタログ内で指定される、態様 1 または 2 の方法である。

## 【 0 0 0 7 】

態様 1 は、構成データに基づいてパラメータ値を生成することと、パラメータ値をコンピュータプログラムの部分へ提供し、部分はリソースと通信する能力を有することと、を含み、コンピュータプログラムの実行がパラメータ値に基づく、態様 1、2、または3のいずれか1つの方法である。

## 【 0 0 0 8 】

態様 5 は、構成データに基づいてパラメータ値を生成することが、パラメータファイルを、コンピュータプログラムの部分によって可読のフォーマットで生成するパラメータジェネレータを実行することを含み、パラメータ値をコンピュータプログラムの部分へ提供することが、パラメータファイルをコンピュータプログラムの部分が利用できるようにすることを含む、態様 4 の方法である。

## 【 0 0 0 9 】

態様 6 は、SQLクエリが、引き数を含むSELECT文を含み、引き数の少なくとも一部分は、リソースに関連付けられた識別子に対応する、態様 1 ~ 5 のいずれか1つの方法である。

## 【 0 0 1 0 】

態様 7 は、コンピュータプログラムが、SQLクエリの操作を表すコンポーネントを含む、態様 1 ~ 6 のいずれか1つの方法である。

## 【 0 0 1 1 】

態様 8 は、コンピュータプログラムがデータフローグラフであり、コンピュータプログラ

10

20

30

40

50

ムの部分がデータフローグラフの部分グラフである、態様 1 ~ 6 のいずれか 1 つの方法である。

【 0 0 1 2 】

態様 9 は、リソースが、リソースによって公開されたアプリケーションプログラミングインターフェース (API) を用いてアクセスされる、態様 1 ~ 8 のいずれか 1 つの方法である。

【 0 0 1 3 】

態様 10 は、SQL クエリを受信したことに応答して API の関数を実行させることを含み、API の関数はリソースにおいて実行可能であり、リソースへ伝送された命令が API の関数を実行させる、態様 9 の方法である。

【 0 0 1 4 】

態様 11 は、外部リソースから受信されたデータをデータベーステーブルの形式でフォーマットすることを含む、態様 1 ~ 10 のいずれか 1 つの方法である。

【 0 0 1 5 】

態様 12 は、リソースがリレーショナルデータベース管理システムでない、態様 1 ~ 11 のいずれか 1 つの方法である。

【 0 0 1 6 】

態様 13 は、命令が、SQL クエリに応答して結果を返さないリソースのファシリティへ伝送される、態様 1 ~ 12 のいずれか 1 つの方法である。

【 0 0 1 7 】

態様 14 は、命令に応答してリソースから受信されたデータが、SQL クエリによって指定されたデータを含む、態様 1 ~ 13 のいずれか 1 つの方法である。

【 0 0 1 8 】

態様 15 は、命令に応答してリソースから受信されたデータ内のレコードおよびフィールドを識別することを含み、レコードおよびフィールドは、データ処理システムの外部にあるリソースに関連付けられたレコードフォーマットに基づいて識別される、態様 1 ~ 14 のいずれか 1 つの方法である。

【 0 0 1 9 】

他の諸態様は、対応する装置、システム、およびコンピュータ可読記憶デバイスを含むことができる。

【 0 0 2 0 】

諸態様は以下の利点のうちの 1 つまたは複数を含むことができる。データ処理システムは、データ処理システムの外部のリソース (例えば、インターネット上で利用可能なもの) を参照し、リレーショナルデータベース管理システムではないクエリを実行することができる。

【 0 0 2 1 】

ネットワーク (例えば、インターネットなどの大規模ネットワーク) を通じて分散したデータは、たとえ、ネットワーク内のデータソースがデータベース言語に応答することができない場合、またはリレーショナルデータベースでない場合であっても、データベースプログラミング言語 (例えば、SQL) を用いることによって効率的にアクセス可能にすることができる。これは、特定のデータソースに特に適応させることができ、それにより、時間とともに変化するデータソースを有するネットワーク内で用いることができる柔軟な探索スキームを提供し得る。その結果、これは、分散ネットワークを、(例えば、データソースの数を拡大もしくは縮小することによって、またはデータソースを交換もしくは変更することによって) より急速に発展させ、その一方で、発展させたネットワークの現在のデータソース内に記憶されたデータへのアクセスを維持することを可能にし得る。

【 0 0 2 2 】

本発明の他の特徴および利点は、以下の説明および請求項から明らかになるであろう。

【図面の簡単な説明】

【 0 0 2 3 】

10

20

30

40

50

【図 1】クエリを処理することができるデータ処理システムを示す図である。

【図 2 A】データ処理システムの要素を示す図である。

【図 2 B】データ処理システムの要素を示す図である。

【図 2 C】データ処理システムの要素を示す図である。

【図 3】クエリを実行するためのユーザインターフェースを示す図である。

【図 4 A】データフローグラフを示す図である。

【図 4 B】データフローグラフの部分グラフを示す図である。

【図 4 C】コンポーネントの実行可能コードを示す図である。

【図 5 A】外部リソースのカタログの内容を示す図である。

【図 5 B】外部リソースのカタログの内容を示す図である。

【図 5 C】外部リソースのカタログの内容を示す図である。

【図 6 A】クエリを実行することに関連付けられたプロセスのためのフローチャートである。

【図 6 B】クエリを実行することに関連付けられたプロセスのためのフローチャートである。

【図 6 C】クエリを実行することに関連付けられたプロセスのためのフローチャートである。

【図 6 D】クエリを実行することに関連付けられたプロセスのためのフローチャートである。

【発明を実施するための形態】

【0024】

データ処理システムはクエリ（SQLクエリなど）をデータフローグラフなどのコンピュータプログラムに変換することができる。コンピュータプログラムは、実行されると、クエリによって指定された操作と等価の操作（例えば、データ処理操作）を実施するコンポーネントを含む。

【0025】

図 1 は、クエリ 102 を処理し、クエリ 102 によって指定された結果 104 を生成することができるデータ処理システム 100 を示す。結果 104 は、元のクエリ 102 によって参照された外部リソース 108 から受信されたデータ 106 に基づく。データ処理システム 100 は、データ処理システム 100 がクエリ 102 に基づいて生成し、外部リソース 108 へ送信する命令 110 に応答してデータ 106 を受信する。

【0026】

クエリ 102 は、クエリによって指定されたデータを取得するために用いられる。1つの種類のクエリ 102 は構造化クエリ言語（SQL）クエリである。本説明はSQLクエリを一例として用いることになるが、ここで説明される諸技法は多次元式（MDX）クエリなどの他の種類のクエリとともに用いることもできるであろう。

【0027】

SQLクエリ（SQL文とも呼ばれる）は、構造化クエリ言語（SQL）によって定義されたコマンドおよび構文を用いる。概して、クエリは、クエリ内で示された1つまたは複数のデータセット内のデータのサブセットを指定する文である。指定されたサブセットは、クエリを処理するシステムによって、クエリを発行したシステムへ返されることが可能である。クエリによって指定され、クエリに応答して返されるデータは概して、クエリによって示されたデータセット内に記憶されている全データの一部分である。SQLクエリの一例は、「SELECT last\_name FROM current\_customers」であり得るであろう。このSQLクエリは操作、SELECTを含む。操作、SELECTは、クエリを実行するシステムに、SELECT操作の引き数に従ってデータを取得するように命令する。SQLの構文内において、引き数は、データベーステーブルなどのデータのセットの、「current\_customers」、およびデータベーステーブルの列である、「last\_name」である。システムがクエリを解釈し、クエリの操作を実行すると、システムはクエリに応答してlast\_name列のデータ（例えば、last\_name列内に含まれているデータの各部分

10

20

30

40

50

）を返すことになる。SQLは、2008年4月7日に発行された、Alex KrieglおよびBoris Trukhnovによる「SQL Bible, 2nd Edition」、ISBN 978-0470229064に詳細に説明されている。この文献は、本明細書において参照により組み込まれる。

#### 【0028】

クエリに応答してデータを返すリソースの典型的な例はリレーショナルデータベースである。リレーショナルデータベースは、1つまたは複数のデータベーステーブル、ならびにSQLクエリを解釈すること、データをテーブルから読み出すこと、データをテーブルに書き込むこと、およびその他の種類のデータ処理機能を実施することなどのデータ処理操作を管理するシステムの集合である。データベーステーブルは、a)レコードを各々表す行、およびb)行内に記憶されたデータのカテゴリを各々表す列の形で配列されたデータの集合である。例えば、「current\_customers」と呼ばれるデータベーステーブルは、事業の現在の顧客を各々表す行を有してもよく、顧客の氏名、顧客の住所、顧客によって購入された一番最近の製品等などのデータのカテゴリを表す列を有してもよい。

10

#### 【0029】

リレーショナルデータベーステーブルは、データを、属性に各々対応するデータの要素で構成された、タプルの形式で記憶する一種のデータベーステーブルである。タプルはリレーショナルデータベース内の行の形式を採ることができ、属性はリレーショナルデータベース内の列の形式を採ることができる。それゆえ、各タプルは、データベーステーブルの属性のうちの1つに各々対応するデータの要素（時として、属性値と呼ばれる）を含む。さらに、複数のタプルのグループ化は時として関係と呼ばれる。

20

#### 【0030】

リレーショナルデータベース管理システム(RDBMS)は、リレーショナルデータベース内に記憶されたデータを作成および変更することを対象とする命令を処理するシステムである。RDBMSは、クエリを解釈し、クエリによって指定されたデータを返すための機能性を含む。クエリを解釈することと、クエリによって指定されたデータを返すこととの組み合わせは、時として、クエリを実行することと称される。例えば、いくつかのRDBMSの実装形態は、a)SQLクエリを構文解析し、b)構造化クエリ言語によって定義された操作を識別し、c)コマンドの引き数を識別し、d)引き数に従って操作を実施する（例えば、実行する）エンジンを含む。

30

#### 【0031】

上述されたように、SQLクエリ「SELECT last\_name FROM current\_customers」は操作、SELECTを含み、この操作はRDBMSに、SELECT操作の引き数に従ってデータを取得するように命令する。引き数は、RDBMSによって管理されているデータベーステーブルである、「current\_customers」、およびデータベーステーブルの列である、「last\_name」である。RDBMSがクエリを解釈し、クエリの操作を実行すると、RDBMSはクエリに回答してlast\_name列のデータ（例えば、last\_name列内に含まれているデータの各部分）を返すことになる。RDBMSのいくつかの実装形態では、クエリプランナと呼ばれるモジュールが、実施されるべき操作を識別することになる。このように、SQLは、リレーショナルデータベースに適用可能な、例えば、RDBMSによって維持されているデータに適用可能なクエリ言語である。

40

#### 【0032】

外部リソース108はデータ処理システム100の外部にある。（データ処理システム100のいくつかの構成要素は図2A～図2Cに詳細に示される）。例えば、外部リソース108は、ネットワーク（例えば、インターネット）を用いて通信するファシリティとすることができる。データ処理システム100は、データをネットワークへ送信すること、およびデータをネットワークから受信することによって、外部リソース108と通信する。外部リソース108は、記憶媒体、例えば、有形の非一時的なコンピュータ可読媒体上に記憶されたデータベーステーブル、データファイル、またはその他のデータ構造などのデータのセットを含んでもよい。

50

## 【 0 0 3 3 】

データ処理システム 1 0 0 は、たとえ、クエリ 1 0 2 によって指定された外部リソース 1 0 8 が R D B M S を含まない場合でも、クエリ 1 0 2 によって指定された結果 1 0 4 を生成する能力を有する。言い換えれば、外部リソース 1 0 8 は R D B M S であるか、またはその構成要素の間に R D B M S を含む必要がない。外部リソース 1 0 8 のいくつかの例は R D B M S を含むことになり、その一方で、外部リソース 1 0 8 の他の例は R D B M S を含まないことになる。

## 【 0 0 3 4 】

使用時に、データ処理システム 1 0 0 はクエリ 1 0 2 に基づいてコンピュータプログラム 1 0 8 ( 時として、データ処理プログラムと呼ばれる ) を生成する。例えば、コンピュータプログラム 1 0 8 は、クエリを入力として受け取り、データフローグラフを出力として作成するエンジン ( 例えば、データ処理システム 1 0 0 のサブシステムを形成するエンジン ) を用いて生成されることが可能である。ここではデータフローグラフが一例として用いられているが、コンピュータプログラムは、プログラムコードによって表された命令を実施するために実行されることが可能であるプログラムコードを含む任意の種類のプログラムとすることができる。

## 【 0 0 3 5 】

このように、図 1 に示されるコンピュータプログラム 1 0 8 などのコンピュータプログラムは、実行されると、例えば、R D B M S などのデータベース管理システムによる、対応するクエリ 1 0 2 の実行と同じ出力を作成することができる。それゆえ、クエリ 1 0 2 は、S Q L などのクエリ言語を用いて書くことができる。しかし、対応するデータ処理操作を実施するシステム、例えば、データ処理システム 1 0 0 のサブシステムは、クエリ 1 0 2 を実行したシステム ( 例えば、R D B M S ) によって実施されるであろう操作と等価である操作を実施するために、コンピュータプログラム 1 0 8 を実行することができる。( 概して、R D B M S は、コンピュータプログラム 1 0 8 を実行する能力を有する機能性を有しない ) 。 2 つの操作は等価であるという時、2 つの操作は、同じ入力データを提供されると、実質的に同じ出力データを作成することを意味する。一例として、2 つの操作は、同じ入力データを提供されると、全く同じ出力データを作成してもよい。一例として、同じ入力データを提供された 2 つの操作は、データのフォーマットが異なるのみである出力データを作成してもよく、例えば、1 つの操作はコンマ区切りの出力データを作成してもよく、等価な操作は、そうでない場合にはコンマ区切りの出力データと同一であるタブ区切りの出力データを作成してもよい。

## 【 0 0 3 6 】

データ処理システム 1 0 0 は、たとえ、クエリ 1 0 2 が R D B M S 以外のデータソースへの参照 1 1 2 を含む場合でも、クエリ 1 0 2 によって指定された結果 1 0 4 を生成することができる。例えば、クエリ 1 0 2 は、R D B M S でない外部リソース 1 0 8 を参照してもよい。外部リソース 1 0 8 は R D B M S でないため、外部リソース 1 0 8 は、クエリ、例えば、S Q L クエリを解釈するための機能性に関連付けられていない。いくつかの実装形態では、データ処理システム 1 0 0 が外部リソース 1 0 8 のデータのレコードフォーマットへのアクセスを有する場合には、データ処理システム 1 0 0 はクエリを、外部リソース 1 0 8 から受信されたデータに適用することができる。レコードフォーマット ( 時として、スキーマと呼ばれる ) はデータの本体の編成の記述である。

## 【 0 0 3 7 】

外部リソース 1 0 8 は、データ処理システム 1 0 0 の外部にあるデータの任意のソースとすることができる。データ処理システム 1 0 0 の外部にあることによって、リソースはデータ処理システム 1 0 0 の構成要素のうちの 1 つでないことを意味する。( データ処理システム 1 0 0 の構成要素のいくつかの例は図 2 A ~ 図 2 C に詳細に示される ) 。例えば、外部リソース 1 0 8 は、ネットワーク 1 1 4 ( 例えば、ここでは「雲」によって表されている、インターネット ) を用いて通信するファシリティとすることができる。データ処理システム 1 0 0 は、データをネットワーク 1 1 4 へ送信すること、およびデータをネットワー

10

20

30

40

50

ク 1 1 4 から受信することによって、外部リソース 1 0 8 と通信する。いくつかの例では、外部リソース 1 0 8 は、TCP/IP または UDP/IP などのインターネットベースのプロトコルを用いて通信するウェブサイトまたは別のファシリティとすることができる。

#### 【 0 0 3 8 】

いくつかの実装形態では、外部リソース 1 0 8 は、ネットワーク 1 1 4 には不可視である R D B M S を含んでもよい。これによって、外部リソース 1 0 8 は、外部リソース 1 0 8 のデータを記憶するリレーショナルデータベース管理システムを含んでもよいが、リレーショナルデータベース管理システムは、限られた場合（例えば、システムアドミニストレータがクエリを提出することを可能にするシステムアドミニストレータインターフェースを経由するなどの場合）を除いて、ネットワーク 1 1 4 を経由して到達する S Q L クエリなどのクエリを受け付けないことを意味する。例えば、外部リソース 1 0 8 が、インターネット上で利用可能なウェブサイトである場合には、このとき、外部リソース 1 0 8 は、データを記憶する「後置型」リレーショナルデータベースを有してもよい。本例では、リレーショナルデータベースは、ウェブブラウザインターフェース、モバイルアプリケーション、または外部リソース 1 0 8 の多くの、またはほとんどのユーザによって利用されているその他のアクセス技法からの S Q L クエリを受け付けない。その代わりに、外部リソース 1 0 8 のデータは、ハイパーテキスト転送プロトコル（HTTP）要求などの、S Q L クエリを含まない技法、またはアプリケーションプログラミングインターフェース（API、後述される）を経由して提出された命令、あるいは別の技法を経由して主にアクセスされる。

#### 【 0 0 3 9 】

外部リソース 1 0 8 はクエリ 1 0 2 を解釈しないため、データ処理システム 1 0 0 は、外部リソース 1 0 8 によって解釈されることが可能である、外部リソース 1 0 8 へ伝送すべき命令 1 1 0 を決定する。命令 1 1 0 は元のクエリ 1 0 2 の形式以外の形式のものである。例えば、元のクエリ 1 0 2 が S Q L クエリである場合には、命令 1 1 0 は S Q L クエリでない（例えば、命令 1 1 0 は SQL コマンドまたは引き数を含まない）。データ処理システム 1 0 0 は、クエリ 1 0 2 に基づいて、および外部リソース 1 0 8 を記述する他の情報に基づいて、いかなる命令 1 1 0 を外部リソース 1 0 8 へ送信するべきかを決定することができる。いくつかの実装形態では、データ処理システム 1 0 0 は、クエリ 1 0 2 を解釈し、クエリ 1 0 2 によって指定された結果 1 0 4 を生成するために用いられるモジュールを有する。いくつかの実装形態では、データ処理システム 1 0 0 は、クエリ 1 0 2 を実施する操作に対応する操作を含むコンピュータプログラムを生成し、実行されると、命令 1 1 0 を外部リソース 1 0 8 へ送信する。

#### 【 0 0 4 0 】

図 2 A ~ 図 2 C は、外部リソース 1 0 8 を参照するクエリ 1 0 2 を実行するために用いることができるデータ処理システム 1 0 0 の要素を示す。図 2 A を参照すると、クエリ 1 0 2 がデータ処理システム 1 0 0 によって受信されると、クエリ 1 0 2 はコンピュータプログラム生成エンジン 1 2 0 へ提供される。コンピュータプログラム生成エンジン 1 2 0 は、実行されると、クエリ 1 0 2 に対応する操作を実施するコンピュータプログラム 1 3 2 を生成する。例えば、クエリ 1 0 2 は、S Q L クエリ、例えば、構造化クエリ言語によって定義された 1 つまたは複数のコマンド、ならびに操作に関連付けられた引き数を含むクエリであってもよい。本例では、コンピュータプログラム 1 3 2 は、S Q L クエリを実施する操作と等価である実行可能な機能性を含む。コンピュータプログラム 1 3 2 が（例えば、実行エンジン 1 4 0 によって）実行されると、コンピュータプログラム 1 3 2 は、S Q L クエリ内で定義された同じ引き数に基づいて実行する。

#### 【 0 0 4 1 】

データ処理システム 1 0 0 は、操作を実施するために R D B M S の機能性、例えば、R D B M S のクエリ解釈機能性に頼らない技法を用いて、クエリ 1 0 2 を実施する操作（時として、クエリ 1 0 2 を実行することと称されるプロセス）を実施することができる。その代わりに、クエリは、コンピュータプログラム 1 3 2 を実行することによって実施される

10

20

30

40

50

ことが可能である。コンピュータプログラム 132 が生成され、構成されると、リレーショナルデータベースのクエリ解釈機能性は、クエリ 102 に基づいて出力を生成するために用いられない。さらに、データ処理システム 100 は、たとえ、クエリ 102 内で識別されたデータソースが、クエリ 102 の形式のクエリを用いて動作するデータベースでない場合でも、クエリ 102 を実行することができる。例えば、外部リソース 108 は、SQL の形式で指定された命令を受け付けるように構成されていなくてもよい。クエリ 102 が SQL クエリであり、外部リソース 108 を参照する場合には、このとき、データ処理システム 100 はクエリ 102 を受信し、それに応答して、コンピュータプログラム 132 が構成され、実行されると、コンピュータプログラム 132 の出力がクエリ 102 の実行の出力と等価になるように、いかなる操作が実施されるべきであるかを決定することができる。

10

#### 【0042】

このように、SQL クエリを用いて、データをリレーショナルデータベースシステム以外のシステムから取得することができる。SQL は、クエリを指定するために用いられる共通言語であるため、多くのユーザは SQL クエリの書き方を知っており、多くのレガシーシステムは、SQL クエリを自動的に生成するように構成されている。ここで説明されている諸技法は、ユーザおよびレガシーシステムが SQL クエリを書くか、または生成することを可能にし、SQL クエリは、データを、SQL クエリを解釈するための機能性を有しない外部リソースから取得するために実施されることが可能である。さらに、データは、外部リソースから、SQL クエリを実行するリレーショナルデータベースへデータをコピーするのではなく、データ処理システム 100 によって外部リソースから取得されることが可能である。

20

#### 【0043】

いくつかの実装形態では、コンピュータプログラム 132 はデータフローグラフを含む。データフローグラフは、入力データに対して実施されるべき操作を表すコンポーネント、およびデータの流れを表すコンポーネント（時として、ノードと呼ばれる）の間のリンクを含むコンピュータプログラムである。コンポーネントによって表された操作は、入力データを処理することによって、入力データに基づいて出力データを生成する。コンポーネントが他のコンポーネントにリンクされている場合には、コンポーネントは入力データを他のコンポーネントへ提供し、出力データを他のコンポーネントから受信することができる。この場合、2 つのコンポーネントの間の各リンクはコンポーネントの一方から他方のコンポーネントへのデータの流れを表す。グラフのコンポーネントのサブセット（例えば、グラフの 1 つまたは複数のコンポーネント）は時として、グラフの部分グラフと呼ばれる。

30

#### 【0044】

データフローグラフがグラフベースの処理システムによって実行されると、コンポーネントの各々が実行され、例えば、コンピュータプログラム、またはコンピュータプログラムの部分が実行され、コンポーネントによって表された操作を実施する。実行時に、データフローグラフは入力データを受信し、入力データは処理され（例えば、データフローグラフのコンポーネントの操作によって操作を受ける）、出力データを生成する。

40

#### 【0045】

データフローグラフのコンポーネントのいくつかまたは全ては、コンポーネントに関連付けられた操作を実施するための実行可能プログラムコードを呼び出すための情報に各々関連付けられている。いくつかの実装形態では、データフローグラフを表すデータ構造は、実行可能コードを参照するデータを含むことができる。データ構造は、データフローグラフをインスタンス化するために用いることができる。これによって、データ構造は、データフローグラフに関連付けられた操作を実施するコードを実行するために用いることができることを意味する。例えば、コンポーネントは、コンポーネントに関連付けられた操作、例えば、データを処理し、出力することを実施するためのコンピュータ実行可能命令を含むコンピュータ可読記憶装置内に記憶されたコンピュータプログラムへの参照に関連付

50

けられていてもよい。

【 0 0 4 6 】

いくつかの例では、データフローグラフのいくつかまたは全てのコンポーネントは、データをプログラムコードが利用できるようにするための情報に各々関連付けられている。例えば、コンポーネントは、データを、コンポーネントに関連付けられた実行可能プログラムへ送出的ために呼び出すことができる関数コールに関連付けられていてもよく、またはコンポーネントは、コンポーネントに関連付けられた実行可能プログラムへ送出的されたデータを受信することができるネットワークポートに関連付けられていてもよく、またはコンポーネントは、データを、コンポーネントに関連付けられた実行可能プログラムへ送出的するための別の技法に関連付けられていてもよい。このように、各コンポーネントはデータを受信し、処理し、出力することができる。

10

【 0 0 4 7 】

いくつかの例では、データフローグラフはパラメータ化可能である。これによって、データフローグラフは、データフローグラフが実行のために準備される際に、パラメータの値を用いて構成されることが可能であることを意味する。パラメータ値を提供されたデータフローグラフのインスタンスは時として、データフローグラフのパラメータ化インスタンスと呼ばれる。パラメータは、パラメータが属するプログラムの挙動を変更するために変更されることが可能であるデータの種類である。例えば、パラメータの値は、プログラムが入力データを処理し、出力データを作成する仕方を変更するために、プログラムへ提供されることが可能である。データフローグラフの場合には、データフローグラフの各コンポーネントは1つまたは複数のパラメータに関連付けられていてもよい。同様に、パラメータのうちの1つまたは複数の単一のコンポーネントに関連付けられているか、または複数のコンポーネントに関連付けられていてもよい。

20

【 0 0 4 8 】

グラフベースのシステムの一例が、「Managing Parameters for Graph-Based Applications」と題する米国特許出願公開第 2 0 0 7 / 0 0 1 1 6 6 8 号に詳細に記載されている。この特許出願は本明細書において参照により組み込まれる。グラフベースの計算を実行するためのシステムが、「Executing Computations Expressed as Graphs」と題する米国特許第 5 , 9 6 6 , 0 7 2 号に記載されている。この特許は本明細書において参照により組み込まれる。さらに、データフローグラフのコンポーネントはクエリ 1 0 2 の操作に置き換えることができる。この置き換えに係る諸技法が、「Managing Data Queries」と題する米国特許出願公開第 2 0 1 1 / 0 1 7 9 0 1 4 ( A 1 ) 号、および同じく「Managing Data Queries」と題する米国特許出願公開第 2 0 1 2 / 0 2 8 4 2 5 5 ( A 1 ) 号にさらに記載されている。これらの特許出願は本明細書において参照により組み込まれる。いくつかの実装形態では、データフローグラフはクエリ 1 0 2 から作成することができる。

30

【 0 0 4 9 】

図 2 B を参照すると、いくつかの例では、コンピュータプログラム 1 3 2 は、実行されると、外部リソース 1 0 8 と通信する部分 1 3 4 を含む。例えば、部分 1 3 4 は、データを外部リソース 1 0 8 へ伝送し、および / またはデータを外部リソース 1 0 8 から受信するように構成された実行可能機能性（例えば、実行可能プログラムコード）を含んでもよい。1 3 4 が実行されると（例えば、部分 1 3 4 のプログラムコードが実行エンジンによって解釈され、実施されると）、コンピュータプログラム 1 3 4 はデータを外部リソース 1 0 8 へ伝送し、および / またはデータを外部リソース 1 0 8 から受信する。いくつかの実装形態では、部分 1 3 4（およびコンピュータプログラム 1 3 2 の他の部分）は、データ処理システム 1 0 0 に提供されており、外部リソース 1 0 8 の識別に基づいて取得されるプログラムコードを含んでもよい。例えば、アプリケーション開発者（図示されていない）は、部分 1 3 4 のプログラムコードを、外部リソースの技術的要求に適合するように書いていてもよい。いくつかの実装形態では、技術的要求は、後述される、アプリケーションプログラミングインターフェース（API）を含む。いくつかの実装形態では、部分 1 3

40

50



4 はデータフローグラフのコンポーネント、またはデータフローグラフの部分グラフである。

【 0 0 5 0 】

いくつかの実装形態では、コンピュータプログラム生成エンジン 1 2 0 は、コンピュータプログラムの 1 つまたは複数の既存の部分にアクセスし、部分を組み立ててコンピュータプログラム 1 3 2 を形成することによって、コンピュータプログラム 1 3 2 を生成する。例えば、コンピュータプログラム生成エンジン 1 2 0 は、既存のコンポーネント、例えば、実行可能プログラムコードの既存の部分を記憶しているコンポーネントライブラリ 1 2 6 へのアクセスを有してもよい。例えば、既存のコンポーネントは、データフローグラフ内に含めるために適したコンポーネントであってもよく、またはデータフローグラフ内に含めるために適した別のグラフ（例えば、部分グラフ）であってもよい。

10

【 0 0 5 1 】

いくつかの実装形態では、コンポーネントライブラリ 1 2 6 は、特定の種類の外部リソース 1 0 8 に各々対応する既存のコンポーネントを含んでもよい。例えば、コンポーネントライブラリ 1 2 6 は、クエリ 1 0 2 内の参照 1 1 2 によって示された外部リソース 1 0 8 に対応するコンポーネント 1 2 4 を含んでもよい。いくつかの例では、コンポーネント 1 2 4 は、開発者によって、データ処理システム 1 0 0 が、参照 1 1 2 に対応する外部リソース 1 0 8 にアクセスすることを可能にする目的のために開発されていてもよい。コンポーネント 1 2 4 は、外部リソース 1 0 8 の識別に基づいてコンポーネントライブラリ 1 2 6 から選定されることが可能である。

20

【 0 0 5 2 】

コンピュータプログラム生成エンジン 1 2 0 が 1 つまたは複数のコンポーネント 1 2 4 を（例えば、コンポーネントライブラリ 1 2 6 から）受信すると、コンピュータプログラム生成エンジン 1 2 0 はまた、コンポーネント 1 2 4 またはコンポーネント群を構成する。その際に、コンピュータプログラム生成エンジン 1 2 0 は、構成された部分を有するコンピュータプログラム 1 3 2 を生成する。例えば、構成されたコンピュータプログラム 1 3 2 は、外部リソース 1 0 8 と通信するための実行可能機能性を含むコンポーネント 1 2 4 またはコンポーネント群に対応する、構成された部分 1 3 4 を含む。

【 0 0 5 3 】

いくつかの実装形態では、構成された部分 1 3 4 は、コンピュータプログラム生成エンジン 1 2 0 によって受信された特性 1 3 6 に基づいて構成されることが可能である。特性 1 3 6 は、外部リソース 1 0 8 の特徴を記述するデータを含む。このデータは、コンピュータプログラム 1 3 2 によって、どのようにデータを外部リソース 1 0 8 へ送信し、そこから受信するのかを決定するために用いられる。いくつかの例では、外部リソース 1 0 8 はデータを特定のフォーマットで送信および受信してもよい。これらの例では、特性 1 3 6 は、外部リソース 1 0 8 へ送信され、そこから受信されるべきデータのフォーマットの仕様を含むことができる。いくつかの例では、外部リソース 1 0 8 はユーザ名および/またはパスワードなどの認証情報の使用を必要としてもよい。これらの例では、特性 1 3 6 は識別子の仕様を含むことができる。いくつかの例では、外部リソース 1 0 8 は特定のアドレスまたはその他のロケーションにおいてアクセス可能であってもよい。例えば、外部リソース 1 0 8 は、特定の IP（インターネットプロトコル）アドレス、または特定のサーバ名、または別の種類のアドレスにおいてアクセス可能であってもよい。これらの例では、特性 1 3 6 はアドレス情報を含むことができる。

30

40

【 0 0 5 4 】

特性 1 3 6 は、例えば、外部リソース 1 0 8 の動作の変化に応答して、更新されることが可能であるソースから受信されることが可能である。例えば、特性 1 3 6 は、データ処理システム 1 0 0 が通信する能力を有する外部リソース 1 0 8 に対応する特性を表すデータを含むカタログ 1 2 2 内に記憶されていてもよい。カタログ 1 2 2 は、データベース、単層ファイル、または任意の他の種類のデータ記憶機構とすることができる。いくつかの実装形態では、カタログ 1 2 2 は複数のエンティティによって同時に読み出されること、お

50

よび書き込まれることが可能である。例えば、カタログ 1 2 2 は、並行したデータ読み出し操作および書き込み操作を管理するための機能性を有するデータベースまたはその他のデータ記憶技法として実装することができるであろう。並行したデータ読み出し操作および書き込み操作を管理するための機能性の一例は、データの一部分がいつ書き込まれるのかを示すロックまたはセマフォの使用である。用いられ得るであろう 1 つの種類のロッキング機能性は、1 つまたは複数のロックが獲得され、読み出し操作または書き込み操作が実施され、その後、ロックが解除される、2 相ロッキングである。並行したデータ読み出し操作および書き込み操作を管理することによって、単一のカタログ 1 2 2 が生成エンジン 1 2 0 の多くのインスタンスによって用いられることが可能である。さらに、単一のカタログ 1 2 2 が多くの種類の外部リソース 1 0 8 のための特性 1 3 6 を記憶することができる。たとえば、特性 1 3 6 が頻繁にアクセスされるほど（例えば、カタログ 1 3 6 内に記憶された少なくともいくつかの特性が毎秒数回以上読み出されるか、または書き込まれるほど）、カタログ 1 2 2 によってサポートされている外部リソース 1 0 8 の数が十分に大きい場合でも、外部リソース 1 0 8 に関連付けられた特性 1 3 6 はいずれも随時更新されることが可能である。

#### 【 0 0 5 5 】

いくつかの例では、外部リソース 1 0 8 の動作が変化した場合には、外部リソース 1 0 8 と通信するための実行可能コード（例えば、実行可能コンポーネント 1 2 4 の一部を構成する実行可能コード）は変更される必要はなく、特性のみが更新される。例えば、外部リソース 1 0 8 は、それが出力データを他のシステムへ提供する仕方を変更してもよい。特性 1 3 6 は、コンピュータプログラム生成エンジン 1 2 0 が、構成されたコンピュータプログラム 1 3 2 の部分 1 3 4 を、構成された部分 1 3 4 が、データを、外部リソース 1 0 8 から受信されることになるフォーマットで受け付けるように構成されることになるように構成することを可能にする。さらに、構成されたコンピュータプログラム 1 3 2 は、外部リソースへ送信されるべき命令 1 1 0 を生成することができ、外部リソース 1 0 8 によって解釈されると、外部リソース 1 0 8 に、命令 1 1 0 に応答してデータ 1 0 6 を返信させることになる。命令 1 1 0 は、構成された部分 1 3 4 によって、特性 1 3 6 によって指定された命令のフォーマットに基づいてフォーマットされることが可能である。いくつかの実装形態では、命令のフォーマットは、カタログ 1 2 2 内に記憶されたデータから導出される。

#### 【 0 0 5 6 】

いくつかの実装形態では、カタログ 1 2 2 は、データ処理システム 1 0 0 が通信する能力を有する外部リソース 1 0 8 のデータ 1 0 6 のレコードフォーマットについての情報を含む。例えば、コンピュータプログラム 1 3 2 は、カタログ 1 2 2 内に記憶されたレコードフォーマット 1 2 8 を用いて、対応する外部リソース 1 0 8 から受信されたデータ 1 0 6 を解釈することができる。いくつかの例では、カタログ 1 2 2 内に記憶されたレコードフォーマット 1 2 8 は、外部リソース 1 0 8 から受信されたデータ 1 0 6 の構造を指定する。レコードフォーマットは、データがレコードに編成され、それにより、各レコードが複数のフィールドを有する、データの構造を指定してもよい。コンピュータプログラム 1 3 2 はレコードフォーマット 1 2 8 を用いて、外部リソース 1 0 8 から受信されたデータ 1 0 6 内のレコードおよびフィールドを識別することができる。いくつかの例では、コンピュータプログラム 1 3 2 はレコードフォーマット 1 2 8 を用いてデータ 1 0 6 を解釈し、例えば、データのレコードおよびフィールドを異なるフォーマット（例えば、コンピュータプログラム 1 3 2 の出力において用いられるべきフォーマット）に変形することができる。

#### 【 0 0 5 7 】

いくつかの実装形態では、コンピュータプログラム 1 3 2 はカタログ 1 2 2 のレコードフォーマット 1 2 8 を用いて、データを、SQL クエリの結果に期待される形式に対応する形式で出力することができる。いくつかの例では、リレーショナルデータベース管理システムは、例えば、SQL クエリに応答して、データをレコードおよびフィールドの形式で

10

20

30

40

50

返す。それゆえ、コンピュータプログラム 132 が外部リソース 108 からのデータ 106 内のレコードおよびフィールドも識別することができる場合には、コンピュータプログラム 132 は SQL クエリに回答して出力をレコードおよびフィールドの形式で提供することができる。これは、たとえ、外部リソース 108 から受信されたデータ 106 が、リレーショナルデータベースから通例受信されるデータの形式、例えば、データベーステーブル、を採らない場合でも、実施されることが可能である。

#### 【0058】

いくつかの実装形態では、コンピュータプログラム 132 はパラメータを用いて構成することができる。例えば、パラメータは、プログラムの挙動を変更するために変更されることが可能である値であってもよい。具体例として、パラメータは「filename」であってもよく、パラメータの値はファイルシステム内のファイルのロケーションとすることができる。パラメータの値は、プログラムを、異なるファイルにアクセスするように構成するために、異なるファイルのロケーションに変更されることが可能である。同じプログラムの 2 つのインスタンス（例えば、同じ実行可能プログラムコードのインスタンス）が、同じプログラムの 2 つのインスタンスの挙動を変更することになる、異なるパラメータ値を用いて構成されることが可能である。

10

#### 【0059】

図 2C を参照すると、コンピュータプログラム生成エンジン 120 は特性を用いて、構成されたコンピュータプログラム 132 の部分 134 を構成するために用いられるパラメータ値 138 を生成することができる。いくつかの例では、コンピュータプログラム 132 は、特性 136 の各々に対応するパラメータを有してもよい。例えば、コンピュータプログラム 132 は、コンピュータプログラム 132 がデータ 106 をいかなるフォーマットで受信することになるのかを決定するために用いられる「record\_format」と呼ばれるパラメータを有してもよい。コンピュータプログラム生成エンジン 120 は、外部リソース 108 のための特性 136 の一部として受信されたレコードフォーマットに基づいて「record\_format」パラメータのためのパラメータ値 138 を生成することができる。別の例として、コンピュータプログラム 132 は、外部リソース 108 に接続する際にユーザ名を供給するために用いられる「username」と呼ばれるパラメータを有してもよい。コンピュータプログラム生成エンジン 120 は、外部リソース 108 のための特性 136 の一部として受信されたユーザ名データに基づいて「username」パラメータのためのパラメータ値 138 を生成することができる。

20

30

#### 【0060】

いくつかの実装形態では、パラメータ値 138 は、コンピュータプログラム 132 によって可読のフォーマットによるパラメータファイルの形式でコンピュータプログラム 132 へ提供される。例えば、パラメータファイルはデータ操作言語（DML）でフォーマットされてもよい。

#### 【0061】

いくつかの実装形態では、コンピュータプログラム 132 は、外部リソース 108 に関連付けられた実行可能コードに基づいて生成される。例えば、カタログ 122 は、実行可能コードが（例えば、特性 136 などの構成データを用いて）構成されると、その後、実行可能コードを用いて外部リソース 108 と通信することができる、実行可能コードのロケーションを指定してもよい。このように、実行可能コードが、例えば、実行可能コードを変更することを許可されたシステムアドミニストレータまたはその他のエンティティによって、更新された場合には、カタログ 122 は、更新された実行可能コードのロケーションを記憶する。コンピュータプログラム生成エンジン 120 がコンピュータプログラム 132 を生成すると、コンピュータプログラム生成エンジン 120 はカタログにアクセスし、実行可能コードの最新バージョンのロケーションを決定することができる。

40

#### 【0062】

コンピュータプログラム 132 が生成されると、実行エンジン 140 がコンピュータプログラム 132 を受信する。次に、実行エンジン 140 はコンピュータプログラム 132 を

50

実行し、例えば、コンピュータプログラム 132 に関連付けられたプログラムコードによって指定された命令を実施する。実行されると、コンピュータプログラム 132（例えば、コンピュータプログラムの構成された部分 134）は命令 110 を生成し、命令 110 を外部リソース 108 へ伝送する。いくつかの実装形態では、外部リソース 108 は、命令を外部リソース 108 へ送信し、データを外部リソース 108 から受信するために用いられる API 142（アプリケーションプログラミングインターフェース）を公開する。概して、API 142 は、コンピュータプログラム 132 が外部リソース 108 と対話することを可能にする任意のファシリティとすることができる。例えば、API 142 は、外部リソース 108 が受信して実施するように構成されている命令の種類を指定してもよい。API によって通例指定される命令の一例は関数コールである。関数は実行可能プログラムコードの一部分である。関数コールを用いる際に、コンピュータプログラム 132 は、命令 110 の一部として、関数の名前、および関数に渡されるべき（例えば、関数の実行可能プログラムコードによって用いられる）引き数を伝送する。命令 110 は、多くの関数コール、またはその他の種類の命令、あるいはその両方を含んでもよい。

#### 【0063】

いくつかの実装形態では、コンピュータプログラム 132 がデータフローグラフである場合には、実行エンジン 140 は、時として、グラフオペレーティングシステムと呼ばれる、専用オペレーティングシステムを含む。グラフオペレーティングシステムは、データフローグラフの個々のコンポーネントの基礎をなす操作を実行する能力を有するコンピュータプログラムである。例えば、データフローグラフのコンポーネントが、データ処理システムによって実施されるべき操作を表す場合には、グラフオペレーティングシステムは、データ処理システムに、操作を実施するように命令するタスクを課される。

#### 【0064】

命令 110 を外部リソース 108 へ送信した後に、コンピュータプログラム 132 は命令 110 に応答してデータ 106 を外部リソースから受信する。次に、実行エンジン 140 は、受信されたデータ 106 をクエリ 102 の結果 104 にフォーマットする。このように、データ処理システム 100 はクエリ 102 を実行し、クエリによって指定された結果 104 を生成することができる。

#### 【0065】

いくつかの実装形態では、外部リソース 108 は、データ 106 を、データを属性 - 値の対の形式で指定する JSON（JavaScript オブジェクト表記法）、またはデータを、データのためのカテゴリを示すタグによって区切って指定する XML（拡張可能マーク付け言語）などのフォーマットで提供する。いくつかの実装形態では、データ 106 のためのレコードフォーマット 128 は、外部リソース 108 に関連付けられた他のデータとともにカタログ 122 内に記憶されており、JSON または XML データを解釈するために用いることができる。

#### 【0066】

コンピュータプログラム 132 が実行エンジン 140 によって実行されると、コンピュータプログラム 132（例えば、結果 104）は、クエリ 102 を実行するが、構成されたコンピュータプログラム 132 を実行しない（実行エンジン 140 以外の）システムの出力と等価である。このように、コンピュータプログラム 132 は、クエリ 102 に対応するコンピュータプログラムの一例である。いくつかの実装形態では、コンピュータプログラム 132 は結果 104 をデータベーステーブル 144 の形式でフォーマットする。このように、データ処理システム 100 は、クエリ 102、例えば、SQL クエリなどのデータベースクエリ）を実行し、RDBMS などのシステムがクエリ 102 を実行し、クエリによって指定された結果を提供すれば提供されるであろう同じ種類の出力を提供することができる。

#### 【0067】

さらに、SQL クエリが、データベーステーブル内に記憶されたデータを取得するために通例用いられるが、ここで示されるクエリ 102 は、いくつかの可能な形式のうちの任意

10

20

30

40

50

のもので記憶されたデータを取得するために用いることができる。外部リソース 108 から受信されたデータ 106 は、データベーステーブル（例えば、リレーショナルデータベーステーブル）以外の形式、例えば、単層ファイル、によるものであってもよい。単層ファイルは、データ要素間の構造的関係を含まないデータファイルである。単層ファイルのデータは、レコードおよびフィールドの形式で表される能力を有してもよい。いくつかの例では、外部リソース 108 から受信されたデータ 106 は、外部リソース 108 にネイティブのフォーマットによるものであってもよい。これによって、データ 106 は、外部リソース 108 によって、データを記憶および処理するために用いられるフォーマットによるものであってもよいことを意味する。

#### 【0068】

図 3 は、クエリを実行するためのユーザインターフェース 300 を示す。このユーザインターフェース 300 は、図 1 および図 2 に示されるデータ処理システム 100 に関連付けられている。このユーザインターフェース 300 は、ユーザ（図示されていない）が SQL クエリ 302 をユーザインターフェース 300 のテキストボックス 304 内に入力することを可能にする。SQL クエリ 302 は、「SELECT」コマンド 306、およびデータソースのための識別子である引き数 308 を含む。ここでは、引き数 308 は「web.wikipedia」である。引き数「web.wikipedia」は外部リソース（例えば、図 1 に示される外部リソース 108）に関連付けられている。

#### 【0069】

いくつかの例では、データ処理システム 100 は、識別子のリスト、および各識別子に関連付けられた外部リソースのリストを記憶してもよい。多数の技法を、識別子を外部リソースに関連付けるために用いることができるであろう。いくつかの例では、図 2 B ~ 図 2 C を参照すると、カタログ 122 が、外部リソースに各々対応する、識別子のリストを記憶していてもよい。例えば、各識別子は、データ処理システム 100 に提出されたクエリ 102 内に含まれる参照 112 に対応してもよい。例えば、参照 112 は、図 3 においてクエリ 300 内に示される引き数 308 「web.wikipedia」であってもよい。このように、参照 112 をクエリ 102 から抽出し、カタログ 122 内の対応する外部リソースを検索するために用いることができる。さらに、カタログは各識別子を特性 136 のそれぞれのセットに関連付けていてもよい。カタログ 122 は各識別子をコンポーネントライブラリ 126 内の 1 つまたは複数のコンポーネントへの参照に関連付けていてもよい。このように、外部リソースへの参照を用いて、カタログ 122 内において、特性の対応するセット、および対応するコンポーネント（例えば、図 2 B ~ 図 2 C に示されるとおりのコンポーネント 124）への参照を識別することができる。別の例として、コンポーネントライブラリ 126 が識別子を記憶していてもよく、例えば、各コンポーネント 124 のための識別子を記憶していてもよく、それにより、識別子（外部リソースの名前など）を用いて、1 つまたは複数の対応するコンポーネント（例えば、図 2 B ~ 図 2 C に示されるとおりのコンポーネント 124）を識別することができる。

#### 【0070】

図 3 に示されるように、「web.wikipedia」は、ハイパーテキスト転送プロトコル（HTTP）を介してインターネット上で利用可能なウィキペディアと呼ばれるウェブサイトを参照する識別子である。ユーザがクエリ実行ボタン 310 をクリックすると、ユーザインターフェース 300 はデータ処理システム 100（図 1）に SQL クエリ 302 を実行させる。

#### 【0071】

SQL クエリ 302 が実行されると、実行の結果 312 がユーザインターフェース 300 内に示される。本例では、結果 312 はデータベーステーブルの形式で表示される。結果 312 は、データの要素を含む行 314 および列 316 を含む。引き数 308 に関連付けられた外部リソースはリレーショナルデータベースでないが、データ処理システム 100 は、クエリ 302 が、外部リソース（ここでは、ウィキペディア）をデータソースとして用いて実行されることを可能にし、結果 312 が、データベーステーブルとしてフォーマ

10

20

30

40

50

ットされることを可能にする。例えば、結果 3 1 2 は、ウィキペディアによって返されたデータ内のレコードおよびフィールドを識別することによって、データベーステーブルとしてフォーマットすることができる。識別は、ウィキペディアに関連付けられたレコードフォーマット 1 2 8 (図 2 B ~ 図 2 C) を用いることによって実施することができる。ここで、クエリ 3 0 2 は、結果 3 1 2 によって満足されなければならない基準を示す ' where ' コマンド 3 1 8 含んでいた。コマンド 3 1 8 は、結果がテキスト内に用語「SQL」を含まなければならないことを示す「subject = ' SQL '」の引き数 3 2 0 (この種の引き数は時として、述語と呼ばれる) を有する。それゆえ、結果 3 1 2 は全て用語「SQL」を含む。

#### 【 0 0 7 2 】

図 4 A は、クエリ 3 0 2 の実行に回答して生成されたデータフローグラフ 4 0 0 を示す。データフローグラフ 4 0 0 は、図 2 A ~ 図 2 C に示される、構成されたコンピュータプログラム 1 3 2 の一例である。いくつかの実装形態では、データフローグラフ 4 0 0 は、他のコンポーネントを含む別のデータフローグラフの部分グラフであってもよい。クエリ 3 0 2 が実行されると、データ処理システム (図 1 および図 2) はデータフローグラフ 4 0 0 を生成し、構成する。さらに、実行エンジン 1 4 0 (図 2 A ~ 図 2 C) はデータフローグラフ 4 0 0 を実行し、図 3 に示される結果 3 1 2 を生成する。データフローグラフ 4 0 0 は、外部リソース 1 0 8 (図 1) を表すコンポーネント 4 0 2、および外部リソース 1 0 8 への入力を表すコンポーネント 4 0 4 を含む。ここでは、コンポーネント 4 0 2 はウィキペディアを表す。例えば、コンポーネント 4 0 2、4 0 4 はともに、図 2 B ~ 図 2 C に示される、構成された部分 1 3 4 の一例であってもよい。データフローグラフ 4 0 0 が (例えば、グラフオペレーティングシステムによって) 実行されると、コンポーネント 4 0 2 は命令 (例えば、図 1 および図 2 に示される命令 1 1 0) を外部リソース 1 0 8 へ伝送する。命令は、入力コンポーネント 4 0 4 から受信された入力データに基づく。例えば、ウィキペディアは、ハイパーテキスト転送プロトコル (HTTP) を介して提出された平文クエリを実行する探索クエリ関数を公開してもよい。本例では、コンポーネント 4 0 2 は、平文探索クエリをウィキペディアへ伝送するように構成されている。例えば、SQL クエリ 3 0 2 は「subject = ' SQL '」の引き数 3 2 0 を含んでいた。ここでは、入力コンポーネント 4 0 4 は、用語「SQL」を、ウィキペディアと通信するコンポーネント 4 0 2 へ提供するように構成されることが可能である。その結果、コンポーネント 4 0 2 は、(例えば、入力コンポーネント 4 0 4 から受信された) 用語「SQL」を平文探索クエリの一部として伝送するように構成される。平文探索クエリに回答して、コンポーネント 4 0 2 は HTTP を介して、ハイパーテキストマーク付け言語 (HTML) に従ってフォーマットされていてもよい、結果を受信する。コンポーネント 4 0 2 はまた、受信された HTML データを構文解析し、構文解析されたデータをデータベーステーブル、例えば、図 3 に示される結果 3 1 2 などの形式にフォーマットするように構成される。

#### 【 0 0 7 3 】

ウィキペディアと通信するために、コンポーネント 4 0 2、4 0 4 は、ウィキペディアに固有の特性 (例えば、図 2 B ~ 図 2 C に示される特性 1 3 6) を用いて構成される。例えば、特性 1 3 6 は、ウィキペディアが探索結果を提供するフォーマット、例えば、探索結果において用いられる特定の HTML フォーマットを記述する情報を含んでもよい。このように、ウィキペディアがその出力のフォーマットを変更した場合には、(例えば、図 2 B ~ 図 2 C に示されるカタログ 1 2 2 内に記憶された) 特性 1 3 6 は、更新されたフォーマットを反映するように更新されることが可能である (例えば、カタログ 1 2 2 のアドミニストレータユーザによって更新される)。ウィキペディアと通信するコンポーネント 4 0 2 が構成されると、特性 1 3 6 の最も直近のバージョンにアクセスし、利用することができる。

#### 【 0 0 7 4 】

コンポーネント 4 0 2、4 0 4 は、特性に基づくパラメータ値を用いて構成される。一例として、コンポーネント 4 0 4 は、「input\_text」と呼ばれるパラメータを有してもよく、そのため、コンポーネント 4 0 4 は、テキスト文字列「SQL」を「input\_text」パラメ

10

20

30

40

50

ータのための値として用いるように構成されることが可能である。このように、SQLクエリ内で定義された引き数320によって定義された基準が、ウィキペディアに照会する際に用いられる。

【0075】

いくつかの実装形態では、ウィキペディアなどの外部リソースと通信するコンポーネント402は複数の実行可能コンポーネントで構成される。いくつかの例では、コンポーネント402は、1つまたは複数のネットワークリソースとのネットワーク接続を確立するための実行可能命令を含む1つまたは複数のコンポーネントを含んでもよい。例えば、ネットワークリソースは、ウィキペディアから発出したデータを提供することができるサーバであってもよい。

10

【0076】

いくつかの例では、コンポーネント402は、外部リソースから受信されたデータを構文解析するための実行可能命令を含む1つまたは複数のコンポーネントを含んでもよい。例えば、ウィキペディアから受信されたデータは、SQLクエリ302に回答するデータ、ならびにマーク付けデータ（例えば、XMLタグなどのタグ）、受信されたデータのサイズもしくは文字セットを記述するデータなどのメタデータ、またはクエリに回答しないその他のデータなどの、他の種類のデータの両方を含んでもよい。外部リソースから受信されたデータを構文解析するための実行可能命令を含むコンポーネントは、外部リソースから受信されたデータを処理し、クエリに回答するデータを、クエリに回答しないデータから分離することができる。クエリに回答しないデータは破棄されるか、または別の目的のために用いられてもよい。例えば、受信されたデータのサイズを記述するデータは、予想された量のデータが外部リソースから受信されたことを決定するために用いることができる。

20

【0077】

いくつかの例では、コンポーネント402は、外部リソースから受信されたデータをフォーマットするための実行可能命令を含む1つまたは複数のコンポーネントを含んでもよい。例えば、コンポーネント402は、特定の仕方でフォーマットされた出力データを（例えば、データフローグラフ400の他の部分へ）提供してもよい。出力データは、例えば、複数のテキスト行、またはアレイの複数の要素、あるいは別の種類のフォーマットとしてフォーマットされることが可能であろう。いくつかの実装形態では、外部リソースから受信されたデータをフォーマットするための実行可能命令を含むコンポーネントは、構文解析されたデータ（例えば、外部リソースから受信されたデータを構文解析するための実行可能命令を含むコンポーネントによって構文解析されたデータ）を受信し、構文解析されたデータを、データフローグラフ400の出力データのために指定されたフォーマットでフォーマットすることができる。例えば、データフローグラフ400の出力データのために指定されたフォーマットはデータベーステーブルのフォーマットであってもよい。

30

【0078】

複数のコンポーネントで構成された1つの種類のコンポーネントは部分グラフと呼ばれる。図4Bは、ウィキペディアと通信するコンポーネント402を構成するコンポーネント412a~gを含む部分グラフ410の一例を示す。コンポーネント412a~gは、ウィキペディアと通信するために必要な計算操作の一部分を実施する実行可能機能性を各々含む。例えば、1つのコンポーネント412aは、ネットワーク上のサーバ、例えば、ハイパーテキスト転送プロトコル（HTTP）を用いるサーバと通信するための実行可能コードを含む「ウェブサービス呼び出し」コンポーネントである。「ウェブサービス呼び出し」コンポーネント412aは要求をサーバへ伝送し、要求に回答してデータを受信する。いくつかの実装形態では、「ウェブサービス呼び出し」コンポーネント412aはシンプルオブジェクトアクセスプロトコル（SOAP）要求を外部リソースへ伝送する。いくつかの実装形態では、「ウェブサービス呼び出し」コンポーネント412aは外部リソースのアプリケーションプログラミングインターフェース（API）にアクセスする。

40

【0079】

「ウェブサービス呼び出し」コンポーネント412aは、要求内に含めて伝送されるべき

50

データを指定する入力を受信する。

【 0 0 8 0 】

複製コンポーネント 4 1 2 b および再フォーマットコンポーネント 4 1 2 c は、「ウェブサービス呼び出し」コンポーネント 4 1 2 a によって伝送されるべきデータを準備する。例えば、複製コンポーネント 4 1 2 b はデータを、図 4 A に示される入力コンポーネント 4 0 4 から受信してもよい。いくつかの実装形態では、再フォーマットコンポーネント 4 1 2 c は、次に、外部リソースへ伝送するのに不適當であるデータを削除することができる。例えば、データの一部は、外部リソースへ伝送するのに不適當である場合があり、例えば、データの一部は、外部リソースの探索機能性へ伝送されることが可能である探索語であり得るであろう。いくつかの例では、データの一部は、外部リソースへ伝送するのに適当でないデータであり得るであろう。例えば、外部リソースは、元のクエリ（例えば、図 1 に示されるクエリ 1 0 2 ）に応答するようにデータを処理するための機能性を欠く場合がある。一例として、データの一部は正規表現であってもよく、外部リソースは、外部リソースの探索機能性によって受信された正規表現を評価するための機能性を欠く場合がある。

10

【 0 0 8 1 】

第 2 の再フォーマットコンポーネント 4 1 2 d が、外部リソースから受信されたデータのフォーマットを変更し、例えば、「ウェブサービス呼び出し」コンポーネント 4 1 2 a によって出力として提供されたデータを構文解析してフォーマットする。例えば、「ウェブサービス呼び出し」コンポーネント 4 1 2 a から受信されたデータは XML データなどのタグ付きデータであってもよい。再フォーマットコンポーネント 4 1 2 d は、タグによって区切られたデータを抽出し、抽出されたデータを、XML タグを用いないフォーマットで出力することができる。再フォーマットコンポーネント 4 1 2 d はまた、出力データ内に必要ないデータを削除してもよい。例えば、破棄されるデータは、XML タグ、または XML タグによって区切られているが、クエリに応答しないデータであってもよい。いくつかの実装形態では、再フォーマットコンポーネント 4 1 2 d は、データを再フォーマットする前に、レコードフォーマット 1 2 8（図 2 B ~ 図 2 C）を用いて、「ウェブサービス呼び出し」コンポーネント 4 1 2 a から受信されたデータの構造を解釈し、例えば、「ウェブサービス呼び出し」コンポーネント 4 1 2 a から受信されたデータ内のレコードおよびフィールドを識別する。

20

30

【 0 0 8 2 】

複製コンポーネント 4 1 2 b はまた、データを第 3 の再フォーマットコンポーネント 4 1 2 e へ提供することができる。いくつかの実装形態では、この再フォーマットコンポーネント 4 1 2 e は、第 1 の再フォーマットコンポーネント 4 1 2 c による削除であった外部リソースへ伝送するのに不適當であるデータを識別する。ここで、外部リソースへ伝送するのに不適當であるデータは、再フォーマットコンポーネント 4 1 2 e によって、2 つのコンポーネントから受信されたデータを組み合わせる結合コンポーネント 4 1 2 f へ提供されることが可能である。

【 0 0 8 3 】

第 2 の再フォーマットコンポーネント 4 1 2 d もまた、その出力を結合コンポーネント 4 1 2 f へ提供する。それゆえ、結合コンポーネント 4 1 2 f は、第 2 の再フォーマットコンポーネント 4 1 2 d および第 3 の再フォーマットコンポーネント 4 1 2 e からの入力を表す出力を提供する。

40

【 0 0 8 4 】

フィルタコンポーネント 4 1 2 g が結合コンポーネント 4 1 2 f の出力を受信する。フィルタコンポーネント 4 1 2 g は、クエリ（例えば、図 1 に示されるクエリ 1 0 2 ）に응答しないデータを削除する。一例として、「ウェブサービス呼び出し」コンポーネント 4 1 2 a がクエリの 1 つまたは複数の要素（例えば、部分）を提供されなかったがゆえに、「ウェブサービス呼び出し」コンポーネント 4 1 2 a は、元のクエリに응答しないデータを出力する場合がある。クエリの 1 つまたは複数の要素は、「ウェブサービス呼び出し」コ

50



ンポーネント 4 1 2 a によって表された外部リソースへ提供するのに適当でない場合がある、正規表現であってもよい。ここで、フィルタコンポーネント 4 1 2 g は、正規表現（例えば、第 3 の再フォーマットコンポーネント 4 1 2 e によって結合コンポーネント 4 1 2 f へ出力され、第 1 の再フォーマットコンポーネント 4 1 2 c によって、「ウェブサービス呼び出し」コンポーネント 4 1 2 a へ提供された出力から削除されたおりのもの）を受信することができるであろう。フィルタコンポーネント 4 1 2 g は、正規表現に基づいて「ウェブサービス呼び出し」コンポーネント 4 1 2 a から最初に受信されたデータをフィルタリングすることができる。ここでは正規表現の例が用いられているが、フィルタコンポーネント 4 1 2 g は、クエリの他の種類の要素、または他の種類の基準、例えば、クエリ内で指定されていない基準に基づいてデータをフィルタリングすることができるであろう。

10

#### 【 0 0 8 5 】

図 4 B に示される部分グラフ 4 1 0 のコンポーネントの各々は、部分グラフ 4 1 0 が実行されると実施されるそれ自身の実行可能コードを有する。例えば、図 4 C は、図 4 B に示されるウェブサービス呼び出しコンポーネント 4 1 2 を構成する実行可能コード 4 4 0 の一例を示す。実行可能コード 4 4 0 は、外部リソースへの要求をフォーマットするためのコード 4 4 2、外部リソースからの出力を構文解析するためのコード 4 4 4、要求を外部リソースへ伝送するためのコード 4 4 6、および外部リソースからの出力を他のコンポーネントへ提供するためのコード 4 4 8 を含む。

#### 【 0 0 8 6 】

図 2 B ~ 図 2 C に関して上述されたように、カタログ 1 2 2 は、データ処理システム 1 0 0 が通信する能力を有する外部リソース 1 0 8 に対応するデータを含む。カタログ 1 0 8 内に記憶されたデータ、例えば、特性 1 3 6 は、コンピュータプログラム 1 3 2 を構成するために用いられる。

20

#### 【 0 0 8 7 】

図 5 A に示されるように、いくつかの実装形態では、ユーザインターフェース 5 0 0 を、カタログ 1 2 2 内に記憶されたデータを見て編集するために用いることができる。ユーザインターフェース 5 0 0 は、ユーザ（例えば、カタログ 1 2 2 のアドミニストレータ）がカタログ内のエントリ 5 0 2 のためのデータを見て編集することを可能にする。カタログのエントリ 5 0 2 は、特定の外部リソース、例えば、ウィキペディアに関連するデータを記憶する。ここではユーザインターフェース 5 0 0 およびエントリ 5 0 2 の一例が示されているが、他の様式のカatalogは他の様式のエントリを有してもよく、他の様式のエントリは異なる種類のデータを含んでもよい。

30

#### 【 0 0 8 8 】

このエントリ 5 0 2 はエントリについての一般情報 5 0 4 a などのデータを含む。エントリ 5 0 2 はまた、後述されるように、外部リソースから受信されたデータのフォーマットを指定する、エントリ 5 0 2 に関連付けられた外部リソースのレコードフォーマット 5 0 4 b を含むことができる。エントリ 5 0 2 はまた、後述されるように、このエントリ 5 0 2 に関連付けられた外部リソースに照会するために用いることができるデータのための名前を指定するエントリのためのインデックス 5 0 4 c を含むことができる。例えば、エントリ 5 0 2 はまた、エントリのためのキー 5 0 4 d を含むことができる。キーは、データをレコードの形式で記憶する、リレーショナルデータベースなどの、データソースのために用いられる。その属性のためのレコードの記憶された値が一意的であるキー属性。エントリ 5 0 2 はまた、エントリのためのパラメータ 5 0 4 e を含むことができる。いくつかの実装形態では、図 2 C を参照すると、パラメータ、例えば、パラメータ 1 3 8 が、コンピュータプログラム 1 3 2 を構成するために用いられる。いくつかの実装形態では、パラメータ 5 0 4 e は、このエントリ 5 0 2 に関連付けられた外部リソースの特徴を記述するデータを含む、図 2 B ~ 図 2 C に示される特性 1 3 6 のためのデータ値を含む。例えば、パラメータ 5 0 4 e は、外部リソース 1 0 8 への許可されたアクセスを得るために用いられる認証情報（例えば、ユーザ名およびパスワード）などの構成データを含んでもよい。

40

50

エントリはまた、いかなるエンティティ（例えば、カタログ 1 2 2 のアドミニストレータなどのユーザインターフェース 5 0 0 のいかなるユーザ）がエントリ 5 0 2 にアクセスし、および／またはそれを変更することができるのかを示す、エントリ 5 0 2 のための権限 5 0 4 f を含むことができる。エントリはまた、このエントリ 5 0 2 に関連付けられた外部リソース 1 0 8 についての統計データ、例えば、外部リソース 1 0 8 において利用可能なデータ量などの統計データを指定する、エントリ 5 0 2 のための統計 5 0 4 g を含むことができる。

#### 【 0 0 8 9 】

図 5 A に示されるように、エントリの一般情報 5 0 4 a は、記憶された実行可能コードのパスへの参照 5 0 6 を含む。いくつかの実装形態では、図 1 を参照すると、記憶された実行可能コードは、外部リソース 1 0 8 と通信するコンピュータプログラム 1 3 2 の部分 1 3 4 である。参照 5 0 6 は、生成エンジン 1 2 0 によって、記憶された実行可能コードにアクセスし、記憶された実行可能コードを用いてコンピュータプログラム 1 3 2 を生成するために用いられることが可能である。

#### 【 0 0 9 0 】

図 5 B に示されるように、エントリ 5 0 2 はまた、エントリ 5 0 2 に関連付けられた外部リソースのレコードフォーマット 5 0 4 b を含むことができる。レコードフォーマット 5 0 4 b は、図 2 B ~ 図 2 C に示されるレコードフォーマット 1 2 8 の一例であってもよい。レコードフォーマット 5 0 4 b は、外部リソースから受信されるデータのフォーマットを指定する。検討のために、レコードフォーマット 5 0 4 b は埋め込みレコードフォーマット 5 0 8 およびフィールド 5 1 0 を含むことができる。フィールド 5 1 0 は、外部リソースへ伝送され、そこから受信されるデータのための名前を指定する。例えば、ここでは、フィールド 5 1 0 は、ウィキペディアへ伝送されるデータのための名前である「サブジェクト」（例えば、百科事典へのクエリ内で用いられるサブジェクト）、およびウィキペディアから受信されるデータのための名前である「行」（例えば、サブジェクトに対応する百科事典エントリの行）である。埋め込みレコードフォーマット 5 0 8 は、外部リソースへ提供され、そこから受信されるデータについてのフォーマット情報を指定する。例えば、この埋め込みレコードフォーマット 5 0 8 は、「サブジェクト」および「行」フィールドは、特定の種類のユニコード文字符号化である、UTF - 8 を用いてフォーマットされることを示す。

#### 【 0 0 9 1 】

図 5 C に示されるように、エントリ 5 0 2 はまた、このエントリ 5 0 2 に関連付けられた外部リソースに照会するために用いることができるデータのための名前を指定するエントリのためのインデックス 5 0 4 c を含むことができる。例えば、このエントリ 5 0 2 は、「サブジェクト」と呼ばれるインデックス 5 1 4 を有する。それゆえ、外部リソースは、「サブジェクト」のためのデータ値を受信し、データ値に応答してデータを返すように構成されている。例えば、ウィキペディアは、各エントリに関連付けられた「サブジェクト」フィールド内にデータ値が現れた百科事典エントリを返すように構成されていてもよい。

#### 【 0 0 9 2 】

図 6 A は、クエリを実行するための手続き 6 0 0 を表すフローチャートを示す。手続き 6 0 0 は、例えば、図 1 に示されるデータ処理システム 1 0 0 のコンポーネントによって実施されることが可能である。

#### 【 0 0 9 3 】

手続きは SQL クエリを受信する 6 0 2。例えば、SQL クエリは、図 1 に示されるクエリ 1 0 2 の一例とすることができる。SQL クエリは、データ処理システムの外部にあるリソースに関連付けられた識別子を含む。さらに、このリソースはリレーショナルデータベース管理システムでない。リソースは、図 1 に示される外部リソース 1 0 8 とすることができる。識別子の一例は、図 3 に示される引き数 3 0 8 である。

#### 【 0 0 9 4 】

手続きは SQL クエリに基づいてコンピュータプログラムを生成する 6 0 4。コンピュー

10

20

30

40

50

タプログラムは、図 2 A ~ 図 2 C に示されるコンピュータプログラム 1 3 2 とすることができる。いくつかの例では、コンピュータプログラムは、SQL クエリの操作を表すコンポーネントを含む。いくつかの例では、コンピュータプログラムはデータフローグラフであり、外部リソースと通信するコンピュータプログラムの部分はデータフローグラフのコンポーネントまたは部分グラフである。データフローグラフ 4 0 0 の一例が図 4 A に示されている。いくつかの実装形態では、図 6 B に示される手続き 6 1 0 を、例えば、コンピュータプログラムの生成の間にコンピュータプログラムの 1 つまたは複数の部分を構成するために、コンピュータプログラムの生成の間に実施することができる。いくつかの実装形態では、図 6 D に示される手続き 6 3 0 を、コンピュータプログラムをインスタンス化するために用いることができるデータ構造、例えば、データフローグラフを生成するために実施することができる。

10

**【 0 0 9 5 】**

手続きは、例えば、コンピュータプログラムを構成するために用いられるパラメータ値に基づいて、コンピュータプログラムを実行する 6 0 6。例えば、コンピュータプログラムは、図 2 A ~ 図 2 C に示される実行エンジン 1 4 0 によって実行されてもよい。いくつかの実装形態では、コンピュータプログラムの実行は、図 5 C に示される手続き 6 2 0 に対応する。

**【 0 0 9 6 】**

図 6 B は、コンピュータプログラムを生成する際にコンピュータプログラムの一部分を構成するために用いられる手続き 6 1 0 を表すフローチャートを示す。いくつかの実装形態では、手続き 6 1 0 は、図 6 A に示されるとおりのコンピュータプログラムを生成する 6 0 4 ためのステップを表す。手続き 6 1 0 は、例えば、コンピュータプログラム生成エンジン 1 2 0 などの、図 1 に示されるデータ処理システム 1 0 0 のコンポーネントによって実施されることが可能である。いくつかの実装形態では、コンピュータプログラムは、複数の部分を組み立てることによって生成される。各部分は、コンピュータプログラムが組み立てられている間に構成されてもよい。この手続き 6 1 0 は、組み立てられているコンピュータプログラムの一部分を構成するために用いることができる。

20

**【 0 0 9 7 】**

手続きは、外部リソースに関連付けられた構成データを識別する 6 1 4。例えば、外部リソースは、図 1 および図 2 に示される外部リソース 1 0 8 とすることができる。構成データは、コンピュータプログラムによって、外部リソースにアクセスし、データを外部リソースから受信するために用いられる情報を指定する。いくつかの実装形態では、構成データは、リソースから受信されるべきデータのフォーマット、例えば、図 2 B ~ 図 2 C に示されるレコードフォーマット 1 2 8 を指定する。いくつかの実装形態では、構成データは、ネットワーク上のリソースにアクセスするために用いられるデータを指定する。例えば、構成データは、外部リソースのデータへの許可されたアクセスを入手するために用いられる、ユーザ名およびパスワードなどの、認証情報を含むことができ、ならびに / あるいは構成データは、ホスト名またはユニフォームリソースロケータ (URL) などのネットワークアドレスを含むことができ、ならびに / あるいは構成データは他の種類のデータを含むことができる。例えば、いくつかの例では、構成データは、図 2 B ~ 図 2 C に示される特性 1 3 6 の一例とすることができる。構成データは、リソースによって用いられるデータフォーマットに対する変更に基づいて更新されることができるカタログ内で指定される。

30

40

**【 0 0 9 8 】**

手続きは構成データに基づいてパラメータ値を生成する 6 1 6。例えば、パラメータ値は、図 2 C に示されるパラメータ値 1 3 8 であってもよい。いくつかの実装形態では、パラメータジェネレータがデータを受信し、パラメータファイルを、コンピュータプログラムによって可読なフォーマットで生成する。次に、パラメータファイルはコンピュータプログラムへ提供される。例えば、パラメータジェネレータは、図 1 に示されるコンピュータプログラム生成エンジン 1 2 0 の一部分とすることができる。パラメータジェネレータによって受信されるデータは、図 2 B ~ 図 2 C に示される特性 1 3 6 のデータとすることができる。

50

できる。例えば、パラメータジェネレータは特性 1 3 6 を構文解析してパラメータ値 1 3 8 を決定してもよい。

【 0 0 9 9 】

手続きはパラメータ値をコンピュータプログラム的一部分へ提供する 5 1 6。部分は、リソースと通信する能力を有する。例えば、コンピュータプログラムの部分は、図 2 B ~ 図 2 C に示される構成された部分 1 3 4 であってもよい。コンピュータプログラムの部分は、リソースがアクセスされる仕方を定義する実行可能命令を含む。さらに、実行可能命令は、コンピュータプログラムの部分へ提供されたパラメータに基づいて動作する。

【 0 1 0 0 】

手続き 6 1 0 は、生成されているコンピュータプログラムの任意の部分のために実施することができる。コンピュータプログラムの全ての部分が構成されると、コンピュータプログラムは実行準備が完了する。

10

【 0 1 0 1 】

図 6 C は、クエリに対応するコンピュータプログラムを実行するための手続き 6 2 0 を表すフローチャートを示す。いくつかの実装形態では、手続きは、図 6 A に示されるとおりのコンピュータプログラムを実行する 6 0 6 ためのステップを表す。手続き 6 2 0 は、例えば、コンピュータプログラム実行エンジン 1 4 0 などの、図 1 に示されるデータ処理システム 1 0 0 のコンポーネントによって実施されることが可能である。コンピュータプログラムは、図 2 A ~ 図 2 C に示されるコンピュータプログラム 1 3 2 とすることができる。例えば、手続き 6 2 0 は、コンピュータプログラム実行エンジン 1 4 0 がコンピュータプログラム 1 3 2 を受信し、コンピュータプログラム 1 3 2 の実行可能コードに対応する操作を実施すると、開始されることが可能である。

20

【 0 1 0 2 】

コンピュータプログラムの実行は 1 つまたは複数の命令を外部リソースへ伝送させる 6 2 2。例えば、外部リソースは、図 1 および図 2 に示される外部リソース 1 0 8 とすることができる。命令は、コンピュータプログラムに対応する SQL クエリの操作以外の操作を定義する。例えば、命令は、図 1 に示される命令 1 1 0 であってもよい。いくつかの例では、命令は、SQL クエリに回答して結果を返さない外部リソースのファシリティへ伝送される。例えば、命令は、SQL クエリを入力として受信し、SQL クエリを評価し、SQL クエリによって指定されたデータに回答する外部リソースのファシリティへ伝送されない。このように、いくつかの例では、命令は SQL コマンドまたは引き数などの SQL 命令を含まない。いくつかの実装形態では、リソースは、リソースによって公開されたアプリケーションプログラミングインターフェース (API) を用いてアクセスされる。API 1 4 2 の一例が図 2 C に示されている。

30

【 0 1 0 3 】

コンピュータプログラムの実行は命令に回答してデータをリソースから受信させる 6 2 4。例えば、データは、図 1 に示される結果 1 0 4 とすることができる。結果は、データベーステーブル、例えば、図 1 に示されるデータベーステーブル 1 4 4 の形式でフォーマットされてもよい。命令に回答してリソースから受信されたデータは、SQL クエリによって指定されたデータを含む。このように、受信されたデータは、仮にリソースが SQL クエリを提供され、それに回答して、外部リソースが、SQL クエリによって指定されたデータを返すとした場合に、外部リソースから受信されたであろうデータと同じになる。

40

【 0 1 0 4 】

図 6 D は、クエリに対応するデータフローグラフをインスタンス化するコンピュータ内のデータ構造を生成するための手続き 6 3 0 を表すフローチャートを示す。例えば、データフローグラフは、図 2 A ~ 図 2 C に示されるコンピュータプログラム 1 3 2 とすることができる。クエリに対応するデータフローグラフ 4 0 0 の一例が図 4 A に示されている。

【 0 1 0 5 】

クエリに基づくクエリプランナへの要求が生成される 6 3 2。クエリプランナは、クエリを実行するために実施することができるステップを表すデータを生成するシステムである

50

。クエリプランナはしばしば、データを、R D B M S によって実施されることが可能であるステップの形式で生成する。

【 0 1 0 6 】

クエリプランナによって要求に基づいて生成されたクエリプランが受信される 6 3 4。クエリプランは、リレーショナルデータベースを管理するシステム、例えば、R D B M S によって実施されるべき 1 つまたは複数のステップの記述を表すデータである。

【 0 1 0 7 】

データソースが、例えば、クエリに基づいて、識別される 6 3 6。データソースは、外部リソース、例えば、図 1 および図 2 に示される外部リソース 1 0 8 とすることができる。識別は、クエリ内で表されている参照、例えば、図 1 に示される参照 1 1 2 に基づいて行うことができるであろう。いくつかの例では、データソースはクエリ内の引き数に基づいて識別される。

10

【 0 1 0 8 】

リレーショナルデータベースを管理するシステム以外の実行システムが識別される 6 3 8。実行システムは、データフローグラフを実行するシステムとすることができる。例えば、実行システムは、図 2 A ~ 図 2 C に示される実行エンジン 1 4 0 とすることができる。対照的に、リレーショナルデータベースを管理するシステムの一例は R D B M S である。

【 0 1 0 9 】

データフローグラフをインスタンス化するデータ構造が生成される 6 4 0。図 2 A ~ 図 2 C に関して上述されたように、データフローグラフは、入力データに対して実施されるべき操作を表すコンポーネント、およびデータの流れを表すコンポーネント（時として、ノードと呼ばれる）の間のリンクを含むコンピュータプログラムである。

20

【 0 1 1 0 】

データ構造は、実行されるべき少なくとも 1 つの操作を表すノードを含む。ノードは、実行システムによって、操作を実施するための実行可能プログラムコードを呼び出すために使用可能な情報に関連付けられている。ノードはまた、実行システムによって、データをプログラムコードが利用できるようにするために使用可能な情報にも関連付けられている。データフローグラフの少なくとも 1 つのリンクは、ノードの操作の出力データがデータフローグラフの別のノードの操作への入力データとして提供されることを表す。

【 0 1 1 1 】

操作は、クエリプランによって記述されたステップに基づいて選定される。いくつかの実装形態では、操作は、データソースにおいて利用可能なデータにアクセスすることを含む。例えば、図 4 A に示されるグラフ 4 0 0 のコンポーネント 4 0 2 は、データフローグラフを実行するシステムの外部のデータソースにおいて利用可能なデータにアクセスするノードの一例である。

30

【 0 1 1 2 】

生成された 6 4 0 データフローグラフが実行されると、データフローグラフに基づくプログラムコードが、識別された実行システム上で実行されることが可能である。さらに、データがデータソースから受信されることが可能である。例えば、データは、図 6 C に示されるように受信されることが可能である 6 2 4。

40

【 0 1 1 3 】

手続き 6 0 0、6 1 0、6 2 0、6 3 0 のいくつかの実装形態では、A P I が用いられる。A P I が用いられるときには、手続き 6 0 0 は、S Q L クエリを受信したことに応答して A P I の関数を実行させることを含む。A P I の関数は外部リソースにおいて実行可能である。さらに、（例えば、図 5 C に示される手続き 6 2 0 において）リソースへ伝送された命令は A P I の関数を実行させる。このように、S Q L クエリを、リレーショナルデータベース管理システムでなく、S Q L クエリを解釈しないリソースに照会するために用いることができる。

【 0 1 1 4 】

図 1 に示される生成エンジン 1 2 0 および / または実行エンジン 1 4 0 は、例えば、1 つ

50

または複数の汎用コンピュータ上において、UNIXオペレーティングシステムのバージョンなどの、好適なオペレーティングシステムの制御の下でホストされてもよい。例えば、これは、複数の中央処理装置（CPU）またはプロセッサコア、局所的なもの（例えば、対称型マルチプロセッシング（SMP）コンピュータなどのマルチプロセッサシステム）、または局所的に分散したもの（例えば、クラスタとして結合された複数のプロセッサ、または超並列処理（MPP）システム、あるいは遠隔的なもの、または遠隔的に分散したもの（例えば、ローカルエリアネットワーク（LAN）および/またはワイドエリアネットワーク（WAN）を介して結合された複数のプロセッサ）のいずれか、あるいはそれらの任意の組み合わせを用いたコンピュータシステムの構成を含む、複数ノード並列コンピューティング環境を含むことができる。

10

#### 【0115】

図3に示されるユーザインターフェース300は開発環境の一部であってもよい。開発環境は、いくつかの実装形態では、アプリケーションをデータフローグラフとして開発するためのシステムである。このシステムに従って作成されたデータフローグラフは、グラフコンポーネントによって表された個々のプロセス内へ情報を入れ、そこから出すための方法、プロセス間で情報を移動させるための方法、およびプロセスのための実行順序を定義するための方法を提供する。このシステムは、任意の利用可能な方法からプロセス間通信方法を選定するアルゴリズムを含む（例えば、グラフのリンクに係る通信経路は、TCP/IPまたはUNIXドメインソケットを用いるか、あるいはプロセス間でデータを渡すための共有メモリを用いることができる）。

20

#### 【0116】

上述された諸技法は、好適なソフトウェアを実行するコンピューティングシステムを用いて実装することができる。例えば、ソフトウェアは、少なくとも1つのプロセッサ、（揮発性および/または不揮発性メモリおよび/または記憶要素を含む）少なくとも1つのデータ記憶システム、（少なくとも1つの入力デバイスもしくはポートを用いて入力を受信するため、および少なくとも1つの出力デバイスもしくはポートを用いて出力を提供するための）少なくとも1つのユーザインターフェースを各々含む、（分散型、クライアント/サーバ型、またはグリッド型などの様々なアーキテクチャのものであってもよい）1つまたは複数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ティングシステム上で実行する1つまたは複数のコンピュータプログラム内の手続きを含んでもよい。ソフトウェアは、例えば、データフローグラフの設計、構成、および実行に関連するサービスを提供する、より大きなプログラムの1つまたは複数のモジュールを含んでもよい。プログラムのモジュール（例えば、データフローグラフの要素）は、データリポジトリ内に記憶されたデータモデルに適合するデータ構造またはその他の編成データとして実装されることが可能である。

30

#### 【0117】

ソフトウェアは、CD-ROM、または（例えば、汎用もしくは専用コンピューティングシステムもしくはデバイスによって可読の）その他のコンピュータ可読媒体などの、有形の非一時的な媒体上に提供されるか、あるいはネットワークの通信媒体を通じて、ソフトウェアが実行されるコンピューティングシステムの有形の非一時的な媒体へ（例えば、伝搬信号内に符号化されて）配信されてもよい。処理の一部または全ては、専用コンピュータ上で、あるいはコプロセッサまたはフィールドプログラマブルゲートアレイ（FPGA）または専用の特定用途向け集積回路（ASIC）などの、専用ハードウェアを用いて実施されてもよい。処理は、ソフトウェアによって指定された計算の異なる部分が異なるコンピューティング要素によって実施される分散式で実施されてもよい。このようなコンピュータプログラムは各々、好ましくは、記憶デバイス媒体がコンピュータによって、本明細書において説明されている処理を実施するために読み出されると、コンピュータを構成し、動作させるために、汎用または専用プログラム可能コンピュータによってアクセス可能な記憶デバイスのコンピュータ可読記憶媒体（例えば、ソリッドステートメモリもしくは媒体、または磁気的もしくは光学的媒体）上に記憶されるか、またはそれにダウンロードされ

40

50

る。本発明のシステムはまた、コンピュータプログラムを用いて構成された、有形の非一時的な媒体として実装されると考えられてもよく、このように構成された媒体は、コンピュータを、本明細書において説明されている処理ステップのうちの1つまたは複数を実施するための特定の既定の仕方で動作させる。

【0118】

本発明の多数の実施形態が説明された。しかし、上述の説明は、本発明の範囲を限定することではなく、本発明を例示することが意図されていることを理解されたい。本発明の範囲は添付の請求項の範囲によって定義される。

【0119】

したがって、他の実施形態もまた、添付の請求項の範囲内に含まれる。例えば、様々な変更が、本発明の範囲から逸脱することなくなされ得る。追加的に、上述されたステップのいくつかは順序に依存しなくてもよく、それゆえ、説明されているものと異なる順序で実施されることが可能である。

10

20

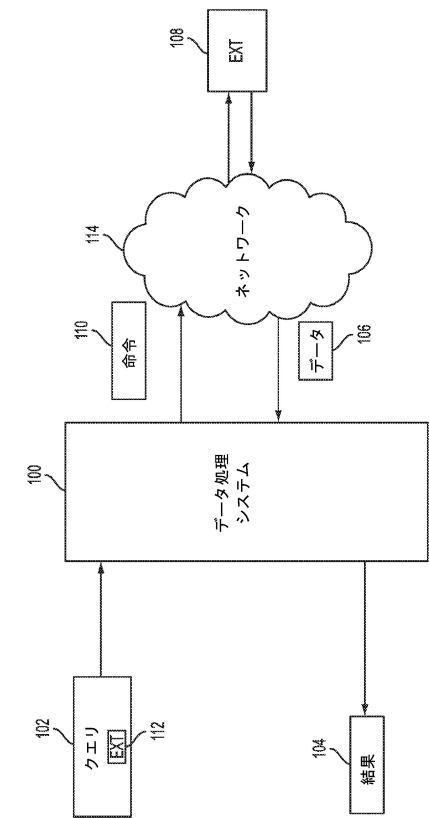
30

40

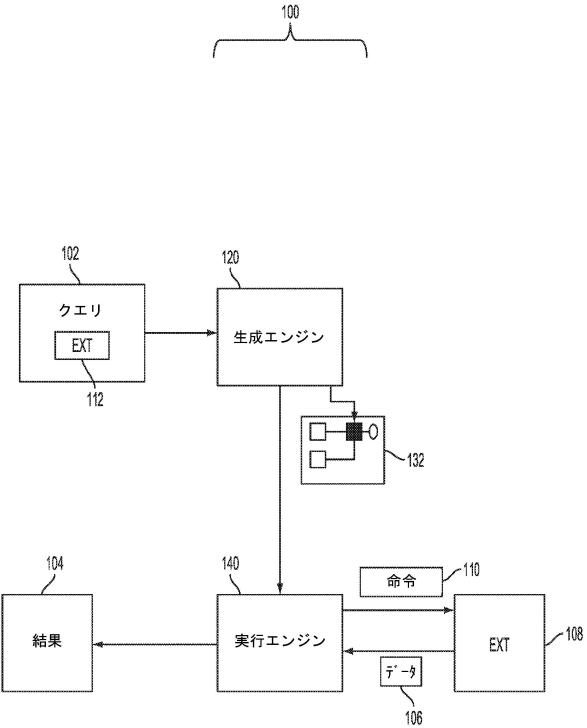
50

【図面】

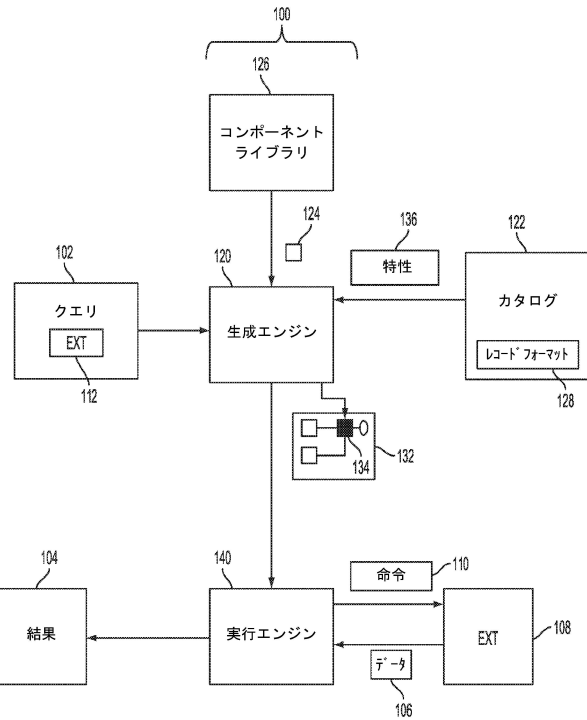
【図 1】



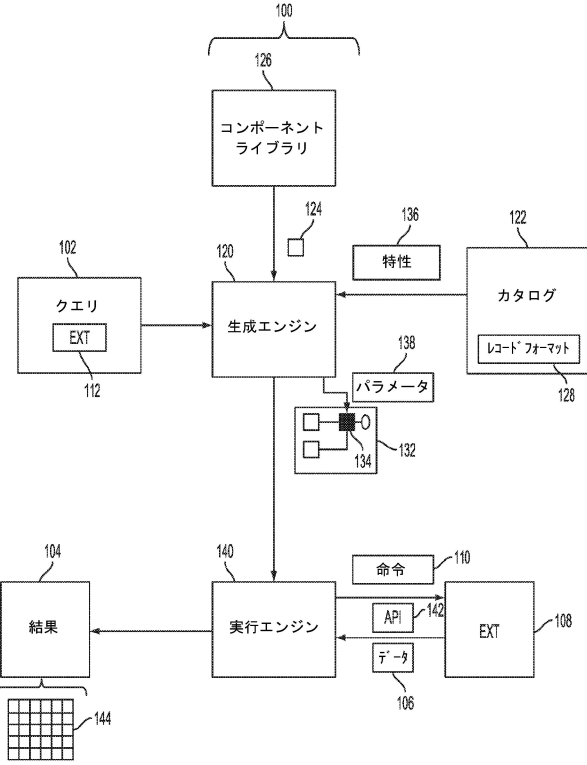
【図 2 A】



【図 2 B】



【図 2 C】



10

20

30

40

50

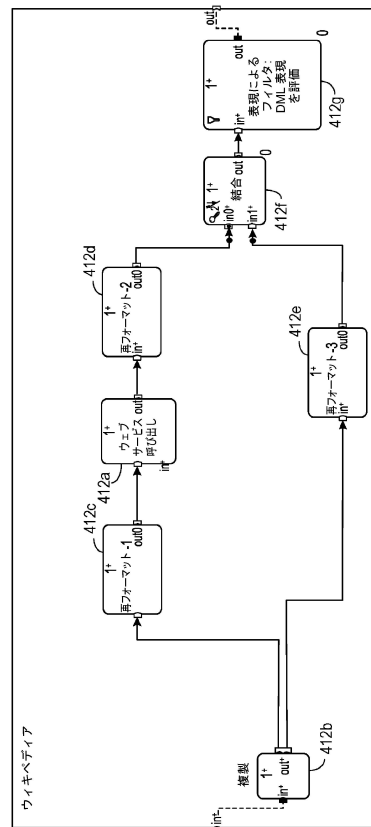


【 図 3 - 2 】



FIG. 3 (Cont.)

【 図 4 B 】



20

40

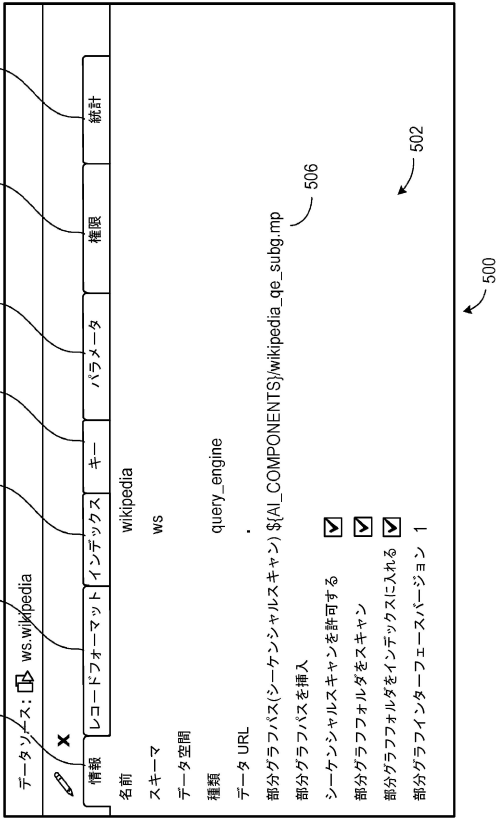
50

【図 4 C】

```
1 type request_type=record
2   ascii_string("\0" request_url;
3   record
4     utf8_string("\0" name;
5     utf8_string("\0" value;
6     end [big endian integer (4)] request_url params = NULL;
7     void (big endian integer (4)) request_body = NULL;
8     end; /*Metadata for records containing request information*/
9
10 type response_type=record
11   // The response type field definitions are shown below.
12   // The definitions are shown here for informational purposes only.
13   // You cannot change the definition of response_type.
14   big_endian_integer (4) response_code;
15   ascii_string (big_endian_integer (4)) response_headers = NULL;
16   void (big_endian_integer (4)) response_body = NULL;
17   end; /*Metadata for records containing response information*/
18
19 type response_vector_type=response_type [int]; /*Response vector*/
20
21 request :: prepare_request (in) =
22   begin
23     request.request_url :: "http://en.wikipedia.org/wiki/" + in.subject;
24   end;
25
26 out :: create_output (in, response) =
27   begin
28     out.* :: response.*;
29     out.subject :: in.subject;
30   end;
```

FIG. 4C

【図 5 A】



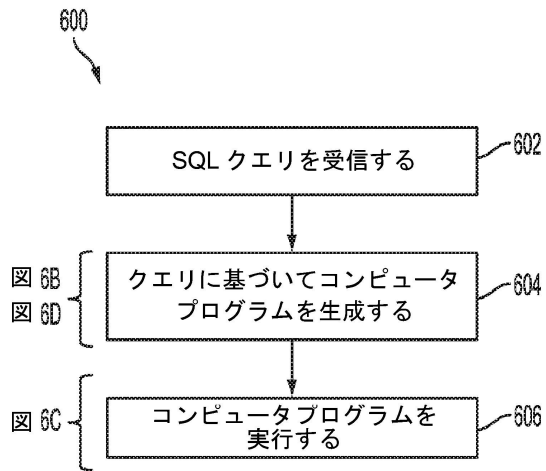
【図 5 B】

データソース: ws.wikipedia			
情報	レコードフォーマット	インデックス	統計
名前	ws.wikipedia		
スキーマ	ws		
データ空間			
種類	query_engine		
データ URL			
部分グラフパス(シーケンシャルスキャン) \${AL_COMPONENTS}/wikipedia_qe_subg.mp			
部分グラフパスを挿入			
シーケンシャルスキャンを許可する			
部分グラフフォルダをスキャン			
部分グラフフォルダをインデックスに入れる			
部分グラフインテリジェントフェーズバリエーション 1			

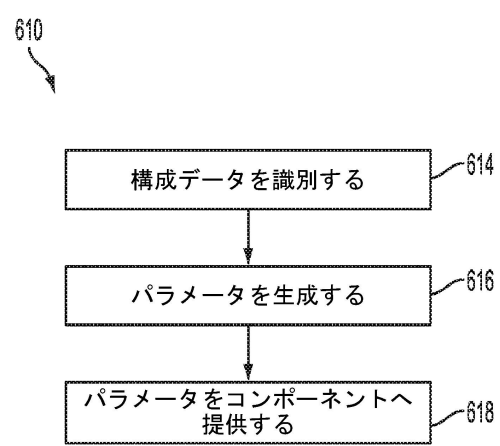
【図 5 C】

データソース: ws.wikipedia			
情報	レコードフォーマット	インデックス	統計
名前	ws.wikipedia		
スキーマ	ws		
データ空間			
種類	query_engine		
データ URL			
部分グラフパス(シーケンシャルスキャン) \${AL_COMPONENTS}/wikipedia_qe_subg.mp			
部分グラフパスを挿入			
シーケンシャルスキャンを許可する			
部分グラフフォルダをスキャン			
部分グラフフォルダをインデックスに入れる			
部分グラフインテリジェントフェーズバリエーション 1			

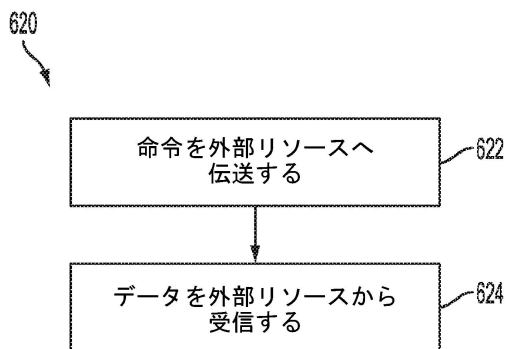
【図 6 A】



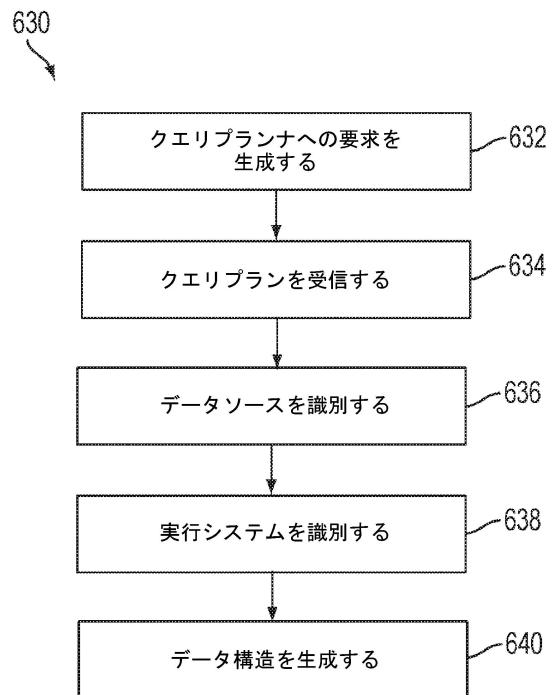
【図 6 B】



【図 6 C】



【図 6 D】



10

20

30

40

50

---

フロントページの続き

(33)優先権主張国・地域又は機関

米国(US)

前置審査

0 2 0 6 7 , シャロン , ブルック ロード 9 4

(72)発明者 アリン , グレン ジョン

アメリカ合衆国 , マサチューセッツ州 0 2 4 7 6 , アーリントン , マウント ヴァーノン ストリート 1 3 2

審査官 原 秀人

(56)参考文献 米国特許出願公開第 2 0 0 9 / 0 0 5 5 3 7 0 ( U S , A 1 )

特開平 0 7 - 0 6 5 0 3 2 ( J P , A )

特表 2 0 1 3 - 5 1 7 5 7 4 ( J P , A )

特表 2 0 1 4 - 5 1 9 0 8 0 ( J P , A )

特開平 0 2 - 0 6 7 6 8 2 ( J P , A )

特開平 1 0 - 0 9 1 4 9 4 ( J P , A )

特開平 1 0 - 0 4 9 4 0 9 ( J P , A )

特開平 0 9 - 3 1 9 7 5 7 ( J P , A )

欧州特許出願公開第 0 0 8 2 2 5 0 5 ( E P , A 2 )

国際公開第 2 0 0 7 / 0 6 1 4 3 0 ( W O , A 1 )

(58)調査した分野 (Int.Cl. , D B 名)

G 0 6 F 1 6 / 0 0 - 1 6 / 9 5 8