(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7002459号
(P7002459)

(45)発行日　令和4年1月20日(2022.1.20)

(24)登録日　令和4年1月4日(2022.1.4)

(51)国際特許分類

Ｇ０６Ｆ　16/90　(2019.01)	Ｇ０６Ｆ	16/90	１００
Ｇ０６Ｆ　16/28　(2019.01)	Ｇ０６Ｆ	16/28	
	Ｇ０６Ｆ	16/90	

ＦＩ

請求項の数　10　（全80頁）

(21)出願番号　特願2018-543097(P2018-543097)
(86)(22)出願日　平成29年8月22日(2017.8.22)
(65)公表番号　特表2019-531517(P2019-531517A)
(43)公表日　令和1年10月31日(2019.10.31)
(86)国際出願番号　PCT/US2017/048051
(87)国際公開番号　WO2018/039257
(87)国際公開日　平成30年3月1日(2018.3.1)
審査請求日　令和2年5月11日(2020.5.11)
(31)優先権主張番号　62/378,143
(32)優先日　平成28年8月22日(2016.8.22)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　62/378,146
(32)優先日　平成28年8月22日(2016.8.22)
最終頁に続く

(73)特許権者　502303739
オラクル・インターナショナル・コーポレイション
アメリカ合衆国カリフォルニア州９４０６５レッドウッド・シティー，オラクル・パークウェイ５００
(74)代理人　110001195
特許業務法人深見特許事務所
(72)発明者　ストジャノビク，アレクサンダー・サシャ
アメリカ合衆国、９５０３０ カリフォルニア州、ロス・ガトス、ウエスト・セントラル・アベニュ、１４
(72)発明者　ナマルバー，ハッサン・ヘイダリ
アメリカ合衆国、９４０６５ カリフォ
最終頁に続く

(54)【発明の名称】　統計プロファイリングおよびリファレンススキーママッチングによるオントロジー帰納のためのシステムおよび方法

(57)【特許請求の範囲】
【請求項１】
コンピュータにより実行されて、データ統合またはその他のコンピューティング環境で使用される方法であって、前記コンピュータが実行する処理は、
１つ以上のスキーマを規定する入力を受信するステップと、
前記１つ以上のスキーマにアクセスすることにより、１つ以上のスキーマのリファレンスによって与えられるエンティティに対応付けられた１つ以上のエンティティ定義を取得するステップと、
前記１つ以上のスキーマから前記１つ以上のエンティティのサンプルデータを生成するステップと、
前記サンプルデータをプロファイリングすることにより、前記サンプルデータに対応付けられた１つ以上のメトリクスを判定するステップと、
前記エンティティ定義に基づいて１つ以上のルールを生成するステップと、
データ入力の処理において使用するために、前記生成した１つ以上のルールに基づいて関数型システムを生成するステップとを含む、方法。
【請求項２】
前記１つ以上のルールは、プロファイリングされたデータ属性または合成値メトリクスについて規定されたデータルールと、エンティティと属性ベクトルとの対応関係を規定する関係ルールと、データルールおよび関係ルールの組み合わせから導出できる複合ルールとを含む、請求項１に記載の方法。

【請求項３】
前記１つ以上のスキーマはリファレンスＨＵＢに与えられる、請求項１または２に記載の方法。
【請求項４】
前記関数型システムはナレッジソースに永続化される、請求項１～３のいずれかに記載の方法。
【請求項５】
前記ナレッジソースはシステムＨＵＢである、請求項４に記載の方法。
【請求項６】
前記１つ以上のスキーマは、型によるタグ付け、比較、分類、または、登録されたＨＵＢから提供されたメタデータスキーマまたはオントロジーの評価に使用される、リファレンスオントロジーとして機能する、請求項１～５のいずれかに記載の方法。
【請求項７】
前記方法は、クラウドまたはクラウドベースのコンピューティング環境で実行される、請求項１～６のいずれかに記載の方法。
【請求項８】
データ統合またはその他のコンピューティング環境で使用されるスキーマ定義のオントロジー解析のためのシステムであって、
前記システムは１つ以上のプロセッサを備え、
前記１つ以上のプロセッサは、
１つ以上のスキーマを規定する入力を受信し、
前記１つ以上のスキーマにアクセスすることにより、１つ以上のスキーマのリファレンスによって与えられるエンティティに対応付けられた１つ以上のエンティティ定義を取得し、
前記１つ以上のスキーマから前記１つ以上のエンティティのサンプルデータを生成し、
前記サンプルデータをプロファイリングすることにより、前記サンプルデータに対応付けられた１つ以上のメトリクスを判定し、
前記エンティティ定義に基づいて１つ以上のルールを生成し、
データ入力の処理において使用するために、前記生成した１つ以上のルールに基づいて関数型システムを生成する、ように動作可能である、システム。
【請求項９】
コンピュータシステムに、請求項１～７のいずれかに記載の方法を実行させるためのコンピュータプログラム。
【請求項１０】
請求項１～７のいずれかに記載の方法を実行するための手段を備える装置。
【発明の詳細な説明】
【技術分野】
【０００１】

【０００２】
優先権の主張
本願は、２０１６年８月２２日に出願され「SYSTEM AND METHOD FOR AUTOMATED MAPPING OF DATA TYPES BETWEEN CLOUD AND DATABASE SERVICES」と題された米国仮特許出願第６２／３７８，１４３号、２０１６年８月２２日に出願され「SYSTEM AND METHOD FOR DYNAMIC, INCREMENTAL RECOMMENDATIONS WITHIN REAL-TIME VISUAL SIMULATION」と題された米国仮特許出願第６２／３７８，１４６号、２０１６年８月２２日に出願され「SYSTEM AND METHOD FOR INFERENCING

OF DATA TRANSFORMATIONS THROUGH PATTERN DECOMPOSITION」と題された米国仮特許出願第６２／３７８，１４７号、２０１６年８月２２日に出願され「SYSTEM AND METHOD FOR ONTOLOGY INDUCTION THROUGH STATISTICAL PROFILING AND REFERENCE SCHEMA MATCHING」と題された米国仮特許出願第６２／３７８，１５０号、２０１６年８月２２日に出願され「SYSTEM AND METHOD FOR META DATA-DRIVEN EXTERNAL INTERFACE GENERATION OF APPLICATION PROGRAMMING INTERFACES」と題された米国仮特許出願第６２／３７８，１５１号、および、２０１６年８月２２日に出願され「SYSTEM AND METHOD FOR DYNAMIC LINEAGE TRACKING AND RECONSTRUCTION OF COMPLEX BUSINESS ENTITIES WITH HIGH-LEVEL POLICIES」と題された米国仮特許出願第６２／３７８，１５２号に基づく優先権を主張し、上記出願各々を本明細書に引用により援用する。

【０００３】

発明の分野

本発明の実施形態は、概して各種ソースから得たデータを統合する方法に関し、具体的には統計プロファイリングおよびリファレンススキーママッチングによるオントロジー帰納に関する。

【背景技術】

【０００４】

背景

現代のコンピューティング環境の多くは、大量のデータをさまざまなタイプのソフトウェアアプリケーション間で共有する機能を必要とする。しかしながら、分散アプリケーションは、たとえばサポートされるそれぞれのデータ型またはそれぞれの実行環境の相違のために、その構成が大幅に異なっている場合がある。アプリケーションの構成は、たとえば、そのアプリケーションプログラミングインターフェイス、ランタイム環境、デプロイ手法、ライフサイクル管理、またはセキュリティ管理に応じて異なり得る。

【０００５】

このような分散アプリケーションの開発に使用されることを目的とするソフトウェアデザインツールは、リソース集中型の傾向があり、アプリケーションおよびデータ統合を管理するためにドメインモデル専門家のサービスを必要とすることが多い。その結果、さまざまな種類の実行環境の中でさまざまな種類のデータを統合するのに使用される複雑でスケーラブルな分散アプリケーションを構築するという課題に取り組むアプリケーション開発者は、一般的に、これらのアプリケーションの設計、構築、および構成のために多大な労力を費やさねばならない。

【発明の概要】

【０００６】

概要

各種実施形態に従い、本明細書では、データの流れ（データフロー(dataflow)、ＤＦ)の管理および複合データフローソフトウェアアプリケーション（データフローアプリケーション、パイプライン）の構築に使用される機械学習（machine learning：ＭＬ、データフロー機械学習、ＤＦＭＬ）を活用する、データ統合またはその他のコンピューティング環境において使用されるシステム（データ人工知能（Artificial Intelligence）システム、データＡＩシステム）について説明する。ある実施形態に従うと、本システムは、スキーマ定義のオントロジー解析を実行することにより、このスキーマに対応付けられたデータおよびデータセットまたはエンティティそれぞれの型を判別し、データセットまたはエンティティとそれらの属性との関係に基づいて規定されたオントロジーを含むリファレンススキーマからモデルを生成またはアップデートすることができる。１つ以上のスキーマを含むリファレンスＨＵＢを用いてデータフローを解析し、さらに、分類する、または、たとえば入力データの変換、エンリッチ化、フィルタリング、もしくはクロスエンティティデータ融合等の、レコメンデーションを行うことができる。

【図面の簡単な説明】

【０００７】
【図１】ある実施形態に係る、データフロー人工知能を提供するシステムを示す図である。
【図２】ある実施形態に係る、システムに使用するイベントコーディネータを含むイベント駆動型アーキテクチャを示す図である。
【図３】ある実施形態に係る、データフロー内のステップを示す図である。
【図４】ある実施形態に係る、複数のソースを含むデータフローの一例を示す図である。
【図５】ある実施形態に係る、データフローをパイプラインとともに使用する例を示す図である。
【図６】ある実施形態に係る、インジェスト／パブリッシュエンジンおよびインジェスト／パブリッシュサービスをパイプラインとともに使用する例を示す図である。
【図７】ある実施形態に係る、ＨＵＢからのインジェストおよびトレーニングのプロセスを示す図である。
【図８】ある実施形態に係る、モデルを構築するプロセスを示す図である。
【図９】ある実施形態に係る、新たに追加されたＨＵＢからのデータセットまたはエンティティを分類するプロセスを示す図である。
【図１０】ある実施形態に係る、新たに追加されたＨＵＢからのデータセットまたはエンティティを分類するプロセスをさらに示す図である。
【図１１】ある実施形態に係る、新たに追加されたＨＵＢからのデータセットまたはエンティティを分類するプロセスをさらに示す図である。
【図１２】ある実施形態に係る、関数型分類に使用するオブジェクト図を示す図である。
【図１３】ある実施形態に係る、ディメンション関数型分類の一例を示す図である。
【図１４】ある実施形態に係る、キューブ関数型分類の一例を示す図である。
【図１５】ある実施形態に係る、ビジネスエンティティの関数型を評価するための関数型分類の使用の一例を示す図である。
【図１６】ある実施形態に係る、関数変換に使用するオブジェクト図を示す図である。
【図１７】ある実施形態に係る、レコメンデーションエンジンの動作を示す図である。
【図１８】ある実施形態に係る、データレイクの使用を示す図である。
【図１９】ある実施形態に係る、データ駆動型戦略を使用してデータレイクを管理することを示す図である。
【図２０】ある実施形態に係る、プロセス駆動型戦略を使用してデータレイクを管理することを示す図である。
【図２１】ある実施形態に係る、パイプラインコンパイラの使用を示す図である。
【図２２】ある実施形態に係る、パイプライングラフの一例を示す図である。
【図２３】ある実施形態に係る、データパイプラインの一例を示す図である。
【図２４】ある実施形態に係る、データパイプラインの別の例を示す図である。
【図２５】ある実施形態に係る、オーケストレーションパイプラインの一例を示す図である。
【図２６】ある実施形態に係る、オーケストレーションパイプラインの一例をさらに示す図である。
【図２７】ある実施形態に係る、メッセージングシステムを含むコーディネーションファブリックの使用を示す図である。
【図２８】ある実施形態に係る、メッセージングシステムを含むコーディネーションファブリックの使用をさらに示す図である。
【図２９】ある実施形態に係る、システムに使用するオンプレミスエージェントを示す図である。
【図３０】ある実施形態に係る、データフロープロセスを示す図である。
【図３１】ある実施形態に係る、データ型の自動マッピングを示す図である。
【図３２】ある実施形態に係る、マッピングの生成のための自動マップサービスを示す図である。
【図３３】ある実施形態に係る、ソーススキーマとターゲットスキーマとの間のマッピン

グの一例を示す図である。
【図３４】ある実施形態に係る、ソーススキーマとターゲットスキーマとの間のマッピングの別の例を示す図である。
【図３５】ある実施形態に係る、データ型の自動マッピングを提供するプロセスを示す図である。
【図３６】ある実施形態に係る、アクセスされたデータに対して有効にされた１つ以上のセマンティックアクションを表示するシステムを示す図である。
【図３７】ある実施形態に係る、アクセスされたデータに対して有効にされた１つ以上のセマンティックアクションを表示するグラフィカルユーザインターフェイスを示す図である。
【図３８】ある実施形態に係る、アクセスされたデータに対して有効にされた１つ以上のセマンティックアクションを表示するグラフィカルユーザインターフェイスをさらに示す図である。
【図３９】ある実施形態に係る、アクセスされたデータに対して有効にされた１つ以上のセマンティックアクションを表示するプロセスを示す図である。
【図４０】ある実施形態に係る、１つ以上のアプリケーション各々ごとに生成された１つ以上の関数式（functional expression）についてデータフロー内の変換のパターンを特定する手段を示す図である。
【図４１】ある実施形態に係る、１つ以上の関数式についてデータフロー内の変換のパターンを特定する例を示す図である。
【図４２】ある実施形態に係る、１つ以上のアプリケーション各々ごとに生成された１つ以上の関数式についてデータフロー内の変換のパターンを特定する際に使用するオブジェクト図を示す図である。
【図４３】ある実施形態に係る、１つ以上のアプリケーション各々ごとに生成された１つ以上の関数式についてデータフロー内の変換のパターンを特定するプロセスを示す図である。
【図４４】ある実施形態に係る、関数型ルールを生成するシステムを示す図である。
【図４５】ある実施形態に係る、関数型ルールを生成するシステムをさらに示す図である。
【図４６】ある実施形態に係る、関数型ルールの生成に使用するオブジェクト図を示す図である。
【図４７】ある実施形態に係る、生成された１つ以上のルールに基づいて関数型システムを生成するプロセスを示す図である。
【図４８】ある実施形態に係る、他言語関数インターフェイスを介して提供された情報に基づくデータフローに関するレコメンデーションの提供において使用するパターンを特定するためのシステムを示す図である。
【図４９】ある実施形態に係る、他言語関数インターフェイスを介して提供された情報に基づくデータフローに関するレコメンデーションの提供において使用するパターンを特定することを示す図である。
【図５０】ある実施形態に係る、他言語関数インターフェイスを介して提供された情報に基づくデータフローに関するレコメンデーションの提供において使用するパターンを特定することをさらに示す図である。
【図５１】ある実施形態に係る、他言語関数インターフェイスを介して提供された情報に基づくデータフローに関するレコメンデーションの提供において使用するパターンを特定するプロセスを示す図である。
【図５２】ある実施形態に係る、１つ以上の層全体にわたる系統トラッキングのための、サンプリングされたデータまたはアクセスされたデータの管理を示す図である。
【図５３】ある実施形態に係る、１つ以上の層全体にわたる系統トラッキングのための、サンプリングされたデータまたはアクセスされたデータの管理をさらに示す図である。
【図５４】ある実施形態に係る、１つ以上の層全体にわたる系統トラッキングのための、サンプリングされたデータまたはアクセスされたデータの管理をさらに示す図である。

【図５５】ある実施形態に係る、１つ以上の層全体にわたる系統トラッキングのための、サンプリングされたデータまたはアクセスされたデータの管理をさらに示す図である。
【図５６】ある実施形態に係る、１つ以上の層全体にわたる系統トラッキングのための、サンプリングされたデータまたはアクセスされたデータの管理をさらに示す図である。
【図５７】ある実施形態に係る、１つ以上の層全体にわたる系統トラッキングのための、サンプリングされたデータまたはアクセスされたデータの管理をさらに示す図である。
【図５８】ある実施形態に係る、１つ以上の層全体にわたる系統トラッキングのための、サンプリングされたデータまたはアクセスされたデータの管理をさらに示す図である。
【図５９】ある実施形態に係る、１つ以上の層全体にわたる系統トラッキングのための、サンプリングされたデータまたはアクセスされたデータの管理のプロセスを示す図である。
【発明を実施するための形態】
【０００８】
詳細な説明
これまでの説明は、他の実施形態およびその特徴とともに、明細書および請求項を含む以下の記載ならびに添付の図面を参照することによって明らかになるであろう。以下の記載では、説明を目的とする具体的な詳細事項が、本発明のさまざまな実施形態の十分な理解を得るために記載されている。しかしながら、さまざまな実施形態はこれらの具体的な詳細事項なしで実施できることは明らかであろう。明細書および請求項を含む以下の記載ならびに添付の図面は、限定することを意図したものではない。
【０００９】
はじめに
各種実施形態に従い、本明細書では、データの流れ（データフロー、ＤＦ)の管理および複合データフローソフトウェアアプリケーション（データフローアプリケーション、パイプライン）の構築に使用される機械学習（ＭＬ、データフロー機械学習、ＤＦＭＬ）を活用する、データ統合またはその他のコンピューティング環境において使用されるシステム（データ人工知能システム、データＡＩシステム）について説明する。
【００１０】
ある実施形態に従うと、このシステムは、本明細書においていくつかの実施形態ではＨＵＢと呼ぶ１つ以上のデータソースまたはデータターゲット間における、複合データ構造、データセットまたはエンティティの自動マッピングに対するサポートを提供することができる。自動マッピングは、メタデータ、スキーマ、およびデータセットの統計プロファイリングによって駆動することができ、自動マッピングを使用することにより、入力ＨＵＢに対応付けられたソースデータセットまたはエンティティを、ターゲットデータセットまたはエンティティに、またはその逆にマッピングして、１つ以上の出力ＨＵＢで使用されるフォーマットまたは組織（プロジェクション）で準備された出力データを生成することができる。
【００１１】
ある実施形態に従うと、このシステムは、本明細書においていくつかの実施形態ではパイプラインエディタまたはLambda Studio IDEと呼ぶ、上記システムで使用される視覚環境を提供するグラフィカルユーザインターフェイスと、ソフトウェア開発コンポーネントとを含み得る。これは、入力ＨＵＢからアクセスされたデータに対するセマンティックアクションの実行のために、当該データに対応付けられた意味またはセマンティクスの理解に基づいてリアルタイムレコメンデーションを提供することを含む。
【００１２】
ある実施形態に従うと、このシステムは、ソフトウェアアプリケーションのデータフローの関数分解から特定されたパターンに基づいて、入力データに対するアクションおよび変換をレコメンドするためのサービスを提供することができ、これは、後のアプリケーションにおいて上記データフローに可能な変換を判定することを含む。データフローは、データの変換、述語、およびデータに適用されるビジネスルールを記述するモデルと、データフロー内で使用される属性とに分解することができる。

【００１３】
ある実施形態に従うと、このシステムは、スキーマ定義のオントロジー解析を実行することにより、このスキーマに対応付けられたデータおよびデータセットまたはエンティティそれぞれのタイプを判別し、データセットまたはエンティティとそれらの属性との関係に基づいて規定されたオントロジーを含むリファレンススキーマからモデルを生成またはアップデートすることができる。１つ以上のスキーマを含むリファレンスＨＵＢを用いてデータフローを解析し、さらに、分類する、または、たとえば、入力データの変換、エンリッチ化、フィルタリング、もしくはクロスエンティティデータ融合等の、レコメンデーションを行うことができる。
【００１４】
ある実施形態に従うと、このシステムは、本明細書においていくつかの実施形態では他言語関数インターフェイスと呼ぶプログラマティックインターフェイスを提供する。このインターフェイスにより、ユーザまたは第三者は、サービス、関数型およびビジネス型、セマンティックアクション、ならびに関数型およびビジネス型に基づくパターンまたは予め定められた複合データフローを、宣言的に規定することにより、システムの機能を拡張することができる。
【００１５】
ある実施形態に従うと、このシステムはデータガバナンス機能を提供することができる。これはたとえば、特定のスナップショットに時間的に関連するデータのスライスごとの、履歴情報（provenance）（特定のデータはどこから来たデータか）、系統（lineage）（このデータはどのようにして取得／処理されたか）、セキュリティ（誰がこのデータの責任者だったか）、分類（このデータは何に関連するデータか）、影響力（impact）（このデータがビジネスにどれほどの影響があるか）、保持時間（retention）（このデータはどれだけの時間存続すべきか）、および有効性（このデータは解析／処理のために除外される／含まれるべきか否か）である。よって、これらはライフサイクルの決定およびデータフローのレコメンデーションにおいて使用することができる。
【００１６】
ある実施形態に従うと、このシステムは、サービスとして、たとえばクラウドベースのコンピューティング環境内で提供されるクラウドサービスとして実現することができ、設計、シミュレーション、デプロイ、開発、オペレーション、およびソフトウェアアプリケーションに使用するデータの解析のための単一のコントロールポイントの役割を果たすことができる。これは、１つ以上のデータソース（たとえばある実施形態では入力ＨＵＢ）からのデータ入力を有効にすること、このデータ用のアプリケーションをユーザが指定できるようにするグラフィカルユーザインターフェイスを提供すること、および、目的とするデータの宛先、用途、またはターゲット（たとえばある実施形態では出力ＨＵＢ）に応じてデータをスケーリングすることを含む。
【００１７】
ある実施形態に従うと、本明細書において、特定のＨＵＢとの関連で使用されるときの「入力」および「出力」という用語は、特定のユースケースまたは例におけるデータの明白な流れを反映するラベルとしてのみ提供されるのであって、特定のＨＵＢの種類または機能を制限することを意図している訳ではない。
【００１８】
たとえば、ある実施形態に従うと、データのソースとして機能する入力ＨＵＢは、同時にまたは別の時に、出力ＨＵＢとしてまたはこのデータもしくは別のデータを受信するターゲットとしても機能することができ、その逆も可能である。
【００１９】
加えて、例示のために本明細書に記載の例のうちのいくつかは入力ＨＵＢおよび出力ＨＵＢの使用を示しているが、ある実施形態に従うと、実際の実装例においてデータ統合またはその他のコンピューティング環境はこのようなＨＵＢを複数含み得る。これらのうちの少なくともいくつかは、入力ＨＵＢおよび／または出力ＨＵＢ双方として機能する。

【００２０】
ある実施形態に従うと、このシステムにより、大規模な、たとえばクラウドベースのコンピューティング環境において、ソフトウェアアプリケーションを急速に開発することができる。このような環境において、データモデルは急速に発展するであろう。また、たとえばサーチ、レコメンデーション、または提案等の特徴は有用なビジネス条件である。このような環境において、人工知能（ＡＩ）とセマンティックサーチとを組合わせることにより、ユーザに対して、彼らの既存のシステムを用いてより多くを達成する権限を与える。たとえば、属性レベルマッピング等の統合インタラクションを、メタデータ、データ、およびシステムとのユーザのやり取りの理解に基づいてレコメンドすることができる。
【００２１】
ある実施形態に従うと、このシステムを用いることにより、複雑なケースを提案することもできる。それはたとえば、情報解析に使用することができ、かつ、それらのデータ内の今まで知られていなかった事実をユーザが発見できるようにする、興味深いディメンションエッジ（dimensional edge）である。
【００２２】
いくつかの実施形態において、このシステムはグラフィカルユーザインターフェイスを提供する。このグラフィカルユーザインターフェイスは、マニュアルタスク（たとえばレコメンデーションまたは提案）の自動化を可能にし、機械学習および確率的な知識の統合を活用することにより、ユーザにとって有用なコンテキストを提供するとともに、発見およびセマンティック駆動型のソリューションを可能にすることができる。それはたとえば、データウェアハウスの作成、サービスのスケーリング、データの作成およびエンリッチ化、ならびにソフトウェアアプリケーションの設計およびモニタリングである。
【００２３】
各種実施形態に従うと、本システムは、以下の特徴のうちの一部またはすべてを含むまたは利用することができる。
【００２４】
設計時システム：ある実施形態に係る、たとえば機械学習機能を提供するデータＡＩサブシステムの使用を含む、ソフトウェアアプリケーション（たとえばデータフローアプリケーション、パイプライン、またはLambdaアプリケーション）の設計、作成、モニタリング、および管理を可能にする計算環境。
【００２５】
実行時システム：ある実施形態に係る、ソフトウェアアプリケーション（たとえばデータフローアプリケーション、パイプライン、またはLambdaアプリケーション）の実行を可能にし、入力を設計時システムから受けてレコメンデーションを設計時システムに提供する計算環境。
【００２６】
パイプライン：ある実施形態に係る、複数の段またはセマンティックアクションを有する処理パイプラインを規定する宣言手段。複数の段またはセマンティックアクションは各々、出力データとして作成するための、入力データのたとえばフィルタリング、結合、エンリッチ化、変換、またはフュージョンのうちの１つ以上等の機能に対応する。たとえばＤＦＭＬにおけるデータフローを表すデータフローソフトウェアアプリケーションまたはデータフローアプリケーション。ある実施形態に従うと、システムは、バッチ（履歴）データ処理にもリアルタイム（ストリーミング）データ処理にも同一のコードベースを（たとえばSpark実行時プラットフォームとともに）使用できる宣言型パイプライン設計をサポートし、また、リアルタイムデータ解析のためにリアルタイムデータストリームに対して作業することができるパイプラインまたはアプリケーションの構築もサポートする。パイプラインの設計変更に伴うデータの再処理は、デプロイされたパイプラインのアップグレードのローリングを通して処理することができる。ある実施形態に従うと、パイプラインは、異なるバッチレイヤおよびリアルタイムレイヤ内のリアルタイムデータとバッチデータの処理に対応することができるLambdaアプリケーションとして提供することができる。

【００２７】
ＨＵＢ：ある実施形態に係る、データセットまたはエンティティを含むデータソースまたはターゲット（クラウドまたはオンプレミス）。イントロスペクトすることができるデータソースであって、このデータソースからデータを消費することができる、または、データをこのデータソースに対してパブリッシュすることができる。データソースはデータセットまたはエンティティを含み、これは属性、セマンティクス、またはその他のデータセットまたはエンティティとの関係を有する。ＨＵＢの例は、ストリーミングデータ、遠隔測定、バッチベース、構造化もしくは非構造化、またはその他のタイプのデータソースを含む。データは、ソースデータセットまたはエンティティに対応付けられたＨＵＢから受信することができ、同じまたは別のＨＵＢのターゲットデータセットまたはエンティティにマッピングすることができる。
【００２８】
システムＨＵＢ：ある実施形態に従うと、システムＨＵＢは、他のＨＵＢに対応付けることができる他のメタデータおよびプロファイル情報ならびにデータセットまたはエンティティを上記他のメタデータに格納するナレッジソースとして機能することができ、また、処理対象のデータのソースまたは受信者としてのレギュラーＨＵＢのように機能することもできる。たとえばＤＦＭＬにおける、メタデータおよびシステムの状態が管理される中央リポジトリである。
【００２９】
データセット（エンティティ）：ある実施形態に係る、属性（たとえばカラム）を含むデータ構造。これは、１つ以上のＨＵＢによって所有されるまたは１つ以上のＨＵＢに対応付けることができる、たとえばデータベーステーブル、ビュー、ファイル、またはＡＰＩである。ある実施形態に従うと、１つ以上のビジネスエンティティ、たとえば顧客記録であり、セマンティックビジネス型として機能することができ、データコンポーネントとして格納される、たとえばＨＵＢ内のテーブルである。データセットまたはエンティティは、属性たとえばテーブル内のカラム、および、データ型たとえばストリングまたは整数とともに、その他のデータセットまたはエンティティとの関係を有することができる。ある実施形態に従うと、当該システムは、たとえばエンリッチ化、準備、変換、モデルトレーニング、またはスコアリング動作中、すべてのタイプのデータ（たとえば構造化、半構造化、または非構造化データ）の、スキーマに依存しない処理をサポートする。
【００３０】
データＡＩサブシステム：ある実施形態に係る、たとえばデータＡＩシステム等のシステムのコンポーネントであり、機械学習およびセマンティック関連機能を担う。これは、サーチ、プロファイリング、レコメンデーションエンジンの提供、または自動マッピングのサポートのうちの１つ以上を含む。データＡＩサブシステムは、イベントコーディネータを介して、設計時システムたとえばLambda Studio等のソフトウェア開発コンポーネントの動作をサポートすることができ、データフローアプリケーション（たとえばパイプライン、Lambdaアプリケーション）によるデータの連続処理に基づいてレコメンデーションを提供することにより、たとえば、既存のたとえばパイプラインの修正をレコメンドして処理中のデータを活用することができる。データＡＩサブシステムは、入力データの量を解析し連続的にドメインナレッジモデルをアップデートすることができる。データフローアプリケーション（たとえばパイプライン）の処理中、たとえばパイプラインの各段は、データＡＩサブシステムが提供するレコメンドされた代案または選択肢に基づいて、アップデートされたドメインモデルおよびユーザからの入力を処理することにより、たとえば、レコメンドされたセマンティックアクションを受容または拒否することができる。
【００３１】
イベントコーディネータ：ある実施形態に係る、設計時システムと実行時システムとの間で動作するイベント駆動型アーキテクチャ（event-driven architecture：ＥＤＡ）コンポーネントであり、データフローアプリケーション（たとえばパイプライン、Lambdaアプリケーション）の設計、作成、モニタリング、および管理に関連するイベントをコーデ

ィネートする。たとえば、イベントコーディネータは、ＨＵＢからデータ（たとえば既知のデータ型に従う新たなデータ）のパブリッシュされた通知を受信し、このＨＵＢからのデータを正規化し、正規化したデータを、サブスクライバのセットに対し、たとえばパイプラインまたはその他の下流のコンシューマによる使用のために、提供する。また、イベントコーディネータは、システム内の状態トランザクションの通知を、一時スライスの作成およびスキーマ進化を含む、系統トラッキングまたはロギングに使用するために、受信することもできる。

【００３２】

プロファイリング：ある実施形態に係る、ＨＵＢからデータのサンプルを抽出し、このＨＵＢから提供されたデータ、ならびにこのＨＵＢ内のデータセットまたはエンティティおよび属性をプロファイリングするとともに、ＨＵＢのサンプリングに関連するメトリクスを決定し、ＨＵＢに関連するメタデータをアップデートすることによりデータのプロファイルをこのＨＵＢに反映させる動作。

【００３３】

ソフトウェア開発コンポーネント（Lambda Studio）：ある実施形態に係る、グラフィカルユーザインターフェイスを提供することにより、セマンティックアクションのパイプラインとしてのLambdaアプリケーションまたはパイプラインのライフサイクルをユーザが作成、モニタリング、および管理できるようにする、設計時システムツール。たとえばパイプラインやLambdaアプリケーションをユーザが設計できるようにするグラフィカルユーザインターフェイス（ＵＩ、ＧＵＩ）またはスタジオ。

【００３４】

セマンティックアクション：ある実施形態に係る、データ変換関数、たとえば関係代数演算。データフローアプリケーション（たとえばパイプライン、Lambdaアプリケーション）が、ＨＵＢ内のデータセットまたはエンティティに対し、別のエンティティへのプロジェクションのために実行できるアクション。セマンティックアクションは、データセット入力を受信しデータセット出力を生成できる異なるモデルまたはＨＵＢに使用可能な高次関数として機能する。セマンティックアクションはマッピングを含み得る。セマンティックアクションは、たとえばパイプラインまたはLambdaアプリケーションの一部としてのデータの処理に応じて、たとえばデータＡＩサブシステムによって連続的にアップデート可能なマッピングを含み得る。

【００３５】

マッピング：ある実施形態に係る、たとえばデータＡＩサブシステムによって提供され、設計時システムを通して、たとえばソフトウェア開発コンポーネントであるLambda Studioを介してアクセス可能にされた、第１の（たとえばソース）データセットまたはエンティティと別の（たとえばターゲット）データセットまたはエンティティとの間のセマンティックアクションのリコメンドされるマッピングである。たとえば、データＡＩサブシステムはサービスとして自動マッピングを提供することができる。自動マッピングは、ＨＵＢに対応付けられたメタデータまたはデータ入力の機械学習解析に基づいて、メタデータ、スキーマ、およびデータセットの統計プロファイリングにより駆動することができる。

【００３６】

パターン：ある実施形態に係る、データフローアプリケーション（たとえばパイプライン、Lambdaアプリケーション）が実行できるセマンティックアクションのパターン。テンプレートを使用することにより、他のアプリケーションが再利用できるパターンの定義を提供することができる。通常ビジネスセマンティクスおよびプロセスに関連付けられるデータおよび関連する変換の論理フロー。

【００３７】

ポリシー：ある実施形態に係る、データフローアプリケーション、たとえばパイプライン、Lambdaアプリケーションが如何にしてスケジュールされるか、どのユーザまたはコンポーネントがどのＨＵＢおよびセマンティックアクションにアクセスできるか、ならびにデータは如何にしてエージングされるべきか、またはその他の検討事項を制御する一組の

ポリシー。如何にしてたとえばパイプラインがたとえばスケジュール、実行、またはアクセスされるべきかを規定するコンフィギュレーション設定。

【００３８】

アプリケーション設計サービス：ある実施形態に従うと、データフローたとえばパイプライン、Lambdaアプリケーション向けのサービスたとえば検証、コンパイル、パッケージング、その他のたとえばＤＦＭＬサービス（たとえばＵＩ、システムファサード）へのデプロイ等を提供する。ソフトウェア開発コンポーネントたとえばLambda Studio（たとえばその入力および出力）におけるパイプラインたとえばLambdaアプリケーションを検証し、パイプラインを永続化（persist）し、パイプライン、Lambdaアプリケーションを実行のためにシステムに（たとえばSparkクラスタに）デプロイすることを制御し、その後、アプリケーションのライフサイクルまたは状態の管理に使用できる、設計時システムコンポーネント。

【００３９】

エッジレイヤ：ある実施形態に係る、データを収集したとえば格納および転送レイヤとしてスケーラブル入出力レイヤにデータを転送するレイヤ。インターネットにアクセス可能なたとえばゲートウェイを介してデータを受信することができ、たとえばデータＡＩシステムへの安全なアクセスをサポートするセキュリティおよびその他の特徴を含む１つ以上のノードを含む実行時システムコンポーネント。

【００４０】

計算レイヤ：ある実施形態に係る、アプリケーション実行およびデータ処理レイヤ（例：Spark）。分散処理コンポーネント、たとえばSparkクラウドサービス、計算ノードのクラスタ、仮想マシンの集合体、またはその他のコンポーネントもしくはノードとして機能し、例としてパイプライン、Lambdaアプリケーションの実行に使用される、実行時システムコンポーネント。マルチテナント環境において、計算レイヤ内のノードは、テナントに対して、これらのテナントによるパイプラインまたはLambdaアプリケーションの実行において使用されるために、割り当てることができる。

【００４１】

スケーラブル入出力（Ｉ／Ｏ）レイヤ：ある実施形態に従うと、トピックおよびパーティションとして構成されたスケーラブルデータパーシステンスおよびアクセスレイヤを提供する（例：Kafka）。データを、システム内で移動できるようにし、システムのさまざまなコンポーネント間で共有できるようにする、キューまたはその他論理ストレージを提供する実行時システムコンポーネント、たとえばKafka環境。マルチテナント環境において、スケーラブルＩ／Ｏレイヤは複数のテナント間で共有できる。

【００４２】

データレイク：ある実施形態に係る、システムＨＵＢまたはその他のコンポーネントからの情報のパーシステンスのリポジトリ。通常は、例としてパイプライン、Lambdaアプリケーションによって正規化または処理され、その他のパイプライン、Lambdaアプリケーションまたはパブリッシュレイヤによって消費される、例としてＤＦＭＬにおけるデータのリポジトリ。

【００４３】

レジストリ：ある実施形態に係る、例としてパイプライン、Lambdaアプリケーションをそれらの関数コンポーネントに分解するのに使用される、例として関数型およびビジネス型の格納のための、１つ以上の情報リポジトリ。

【００４４】

データフロー機械学習（ＤＦＭＬ）：ある実施形態に係る、機械学習（ＭＬ）を活用することにより複合データフローアプリケーションの構築を支援する、データ統合、データフロー管理システム。

【００４５】

メタデータ：ある実施形態に係る、データセットまたはエンティティおよび属性ならびにそれらの関係の、根底にある定義、記述。例としてＤＦＭＬにおけるアーティファクトに

関する記述データでもあり得る。
【００４６】
データ：ある実施形態に係る、データセットまたはエンティティによって表されるアプリケーションデータ。これらはバッチでもストリームでもよい。たとえば、顧客、オーダー、または製品。
【００４７】
システムファサード：ある実施形態に係る、例としてＤＦＭＬイベント駆動型アーキテクチャの機能にアクセスするための統一ＡＰＩレイヤ。
【００４８】
データＡＩサブシステム：ある実施形態に従うと、限定されないがたとえばサーチ、自動マップ、レコメンデーション、またはプロファイリングを含む人工知能（ＡＩ）サービスを提供する。
【００４９】
ストリーミングエンティティ：ある実施形態に係る、データの速度の強調をサポートし得る、データおよび準リアルタイム処理および出力条件の連続入力。
【００５０】
バッチエンティティ：ある実施形態に係る、ボリュームの強調によって特徴付けることができる、スケジュールされたまたはリクエストに応じて実施されるデータのインジェスチョン。
【００５１】
データスライス：ある実施形態に係る、通常は時間によってマークされるデータのパーティション。
【００５２】
ルール：ある実施形態に従うと、例としてＤＦＭＬにおけるアーティファクトを左右するディレクティブを表し、たとえば、データルール、関係ルール、メタデータルール、および複合またはハイブリッドルールである。
【００５３】
レコメンデーション（データＡＩ）：ある実施形態に係る、例としてパイプライン、Lambdaアプリケーションの設計を支援するための１つ以上のセマンティックアクションまたはきめ細かいディレクティブによって通常表される、推奨される一連のアクション。
【００５４】
サーチ（データＡＩ）：ある実施形態に係る、関連するアーティファクトを返すための、コンテキストおよびユーザの意図によって特徴付けられる、例としてＤＦＭＬにおけるセマンティックサーチ。
【００５５】
自動マップ（データＡＩ）：ある実施形態に係る、データフローで試用されることになる候補ソースまたはターゲットデータセットもしくはエンティティをショートリストに入れるある種のリコメンデーション。
【００５６】
データプロファイリング（データＡＩ）：ある実施形態に係る、データセットまたはエンティティに属する属性におけるデータを特徴付けるいくつかのメトリクスの、たとえば最小値、最大値、四分位数間領域、または散在度の、集合体。
【００５７】
アクションパラメータ：ある実施形態に係る、セマンティックアクションの実行対象であるデータセットに対するリファレンス。たとえば、例としてパイプライン、Lambdaアプリケーションにおける等価結合（equi-join）に対するパラメータ。
【００５８】
他言語関数インターフェイス：ある実施形態に係る、例としてＤＦＭＬ　Lambdaアプリケーションフレームワークの一部としてサービス（およびセマンティックアクション）を登録し呼び出すメカニズム。例としてＤＦＭＬにおける機能または変換語彙を拡張するの

に使用できる。
【００５９】
サービス：ある実施形態に係る、データ統合段（たとえば準備、発見、変換、または視覚化）によって特徴付けることができるセマンティックアクションの集合体の、例としてＤＦＭＬにおける未対応のアーティファクト。
【００６０】
サービスレジストリ：ある実施形態に係る、サービス、それらのセマンティックアクションおよびその他のインスタンス情報のリポジトリ。
【００６１】
データライフサイクル：ある実施形態に係る、インジェスチョンで始まりパブリッシュで終わる、例としてＤＦＭＬ内のデータの使用における段。
【００６２】
メタデータハーベスティング：ある実施形態に係る、通常はＨＵＢの登録後に、プロファイリングのためにメタデータおよびサンプルデータを収集すること。
【００６３】
パイプライン正規化：ある実施形態に係る、例としてパイプライン、Lambdaアプリケーションによる消費を容易にする特定のフォーマットでデータを標準化すること。
【００６４】
モニタリング：ある実施形態に係る、例としてパイプライン、Lambdaアプリケーションの実行を、特定、測定、および評価すること。
【００６５】
インジェスト：ある実施形態に係る、例としてＤＦＭＬにエッジレイヤを通してデータを取り込むこと。
【００６６】
パブリッシュ：ある実施形態に係る、例としてＤＦＭＬからデータをターゲットエンドポイントに書き込むこと。
【００６７】
データＡＩシステム
図１は、ある実施形態に係る、データフロー人工知能を提供するためのシステムを示す図である。
【００６８】
図１に示すように、ある実施形態に従うと、システム、たとえばデータＡＩシステム１５０は、たとえばビジネスデータ、消費者データ、および企業データ等のデータを処理および変換するための１つ以上のサービスを提供することができ、これは、たとえばデータベース、クラウドデータウェアハウス、ストレージシステム、またはストレージサービス等の多様な計算アセットとともに使用される機械学習処理の使用を含む。
【００６９】
ある実施形態に従うと、計算アセットは、クラウドベースであっても企業ベースであってもオンプレミスもしくはエージェントベースであってもよい。このシステムのさまざまな要素は１つ以上のネットワーク１３０によって接続することができる。
【００７０】
ある実施形態に従うと、このシステムは、１つ以上の入力ＨＵＢ１１０（たとえばデータのソース、データソース）と、出力ＨＵＢ１８０（たとえばデータのターゲット、データターゲット）とを含み得る。
【００７１】
ある実施形態に従うと、各入力ＨＵＢ、たとえばＨＵＢ１１１は、複数の（ソース）データセットまたはエンティティ１９２を含み得る。
【００７２】
ある実施形態に従うと、入力ＨＵＢの例は、データベース管理システム（ＤＢ、ＤＢＭＳ）１１２（たとえばオンライントランザクション処理システム（on-line transaction pr

ocessing system：ＯＬＴＰ）、ビジネスインテリジェンスシステム、またはオンライン解析処理システム（on-line analytical processing system：ＯＬＡＰ））を含み得る。このような例において、たとえばデータ管理システム等のソースが提供するデータは、構造化データであっても半構造化データであってもよい。

【００７３】

ある実施形態に従うと、入力ＨＵＢの他の例は、非構造化データを有するオブジェクトバケットまたはクリックストリームソースであってもよいクラウドストア／オブジェクトストア１１４（たとえばＡＷＳ　Ｓ３または別のオブジェクトストア）、データクラウド１１６（たとえば第三者クラウド）、ストリーミングデータソース１１８（たとえばＡＷＳキネティックスまたは別のストリーミングデータソース）、またはその他入力ソース１１９を含み得る。

【００７４】

ある実施形態に従うと、入力ＨＵＢはデータソースを含み得る。データソースは、たとえばOracle Big Data Prep (ＢＤＰ)サービスからデータを受ける。

【００７５】

ある実施形態に従うと、このシステムは１つ以上の出力ＨＵＢ１８０（たとえば出力宛先）を含み得る。各出力ＨＵＢ、たとえばＨＵＢ１８１は、複数の（ターゲット）データセットまたはエンティティ１９４を含み得る。

【００７６】

ある実施形態に従うと、出力ＨＵＢの例は、パブリッククラウド１８２、データクラウド１８４（たとえばＡＷＳおよびAzure）、オンプレミスクラウド１８６、またはその他の出力ターゲット１８７を含み得る。このシステムが提供するデータ出力は、出力ＨＵＢにおいてアクセス可能なデータフローアプリケーション（たとえばパイプライン、Lambdaアプリケーション）のために生成することができる。

【００７７】

ある実施形態に従うと、パブリッククラウドの例として、たとえばOracle Public Cloudを挙げることができる。これは、たとえば、Big Data Prepクラウドサービス、Exadataクラウドサービス、Big Data Discoveryクラウドサービス、およびBusiness Intelligenceクラウドサービスを含み得る。

【００７８】

ある実施形態に従うと、このシステムは、サービスとして（たとえばサービスとしてのソフトウェアとして）ユーザに配信される、ストリーミングおよびオンデマンド（バッチ）データ処理のための統一されたプラットフォームとして実現することができ、複数の入力ＨＵＢのためのスケーラブルなマルチテナントデータ処理を提供する。データは、機械学習技術と、サービスの一部としてグラフィカルユーザインターフェイスによって提供される視覚的洞察およびモニタリングとを用いて、リアルタイムで解析される。データセットは、出力ＨＵＢへの出力のために複数の入力ＨＵＢからフュージングすることができる。たとえば、このシステムが提供するデータ処理サービスを通して、データをデータウェアハウス用に生成し１つ以上の出力ＨＵＢにポピュレートすることができる。

【００７９】

ある実施形態に従うと、このシステムは、データの変換、エンリッチ化、ルーティング、分類、およびブレンドのために宣言型およびプログラミングトポロジを提供し、設計時システム１６０と実行時システム１７０とを含み得る。ユーザは、データ処理の実行用に設計された、たとえばデータフローアプリケーション（たとえばパイプライン、Lambdaアプリケーション）１９０等のアプリケーションを作成することができる。

【００８０】

ある実施形態に従うと、設計時システムは、ユーザが、データフロー処理用に、データフローアプリケーションを設計し、データフローを規定し、データを規定できるようにすることができる。たとえば、設計時システムは、データフローアプリケーションの作成のためにグラフィカルユーザインターフェイスを提供するソフトウェア開発コンポーネント１

６２（本明細書ではある実施形態においてLambda Studioと呼ぶ）を提供することができる。

【００８１】

たとえば、ある実施形態に従うと、ユーザは、ソフトウェア開発コンポーネントを用いて、アプリケーション用のデータフローを作成するために入力ＨＵＢおよび出力ＨＵＢを指定することができる。グラフィカルユーザインターフェイスは、データ統合用のサービスのためのインターフェイスを示すことができる。これにより、ユーザは、アプリケーション用のデータフローを作成、操作、および管理することができる。これはデータフローパイプラインを動的にモニタリングおよび管理する機能を含み、それはたとえばデータ系統を観察することおよび法解析を実行すること等である。

【００８２】

ある実施形態に従うと、設計時システムはまた、データフローアプリケーションを実行時システムにデプロイするためのアプリケーション設計サービス１６４を含み得る。

【００８３】

ある実施形態に従うと、設計時システムはまた、データフローを処理するためにメタデータを格納するための１つ以上のシステムＨＵＢ１６６（たとえばメタデータリポジトリ）を含み得る。１つ以上のシステムＨＵＢは、データのサンプル、たとえば関数型およびビジネスデータ型を含むデータ型等のデータのサンプルを格納することができる。システムＨＵＢ内の情報を用いることにより、本明細書に開示する技術のうちの１つ以上を実行することができる。データレイク１６７コンポーネントは、システムＨＵＢからの情報のパーシステンスのためにリポジトリとして動作することができる。

【００８４】

ある実施形態に従うと、設計時システムはまた、データ人工知能処理のための動作を実行するデータ人工知能（ＡＩ）サブシステム１６８を含み得る。上記動作は、ＭＬ技術、たとえばサーチおよび取出を使用することを含み得る。データＡＩサブシステムは、システムＨＵＢ用のメタデータを生成するためにデータをサンプリングすることができる。

【００８５】

ある実施形態に従うと、データＡＩサブシステムは、入力ＨＵＢごとに、スキーマオブジェクト解析、メタデータ解析、サンプルデータ、相関解析、および分類解析を実行することができる。データＡＩサブシステムは、入力されたデータに対して連続的に実行することにより、リッチデータをデータフローアプリケーションに提供することができ、かつ、たとえばパイプライン、Lambdaアプリケーションに、レコメンデーション、洞察、およびタイプ誘導を提供することができる。

【００８６】

ある実施形態に従うと、設計時システムにより、ユーザは、ユースケースの機能的要求を規定するポリシー、アーティファクト、およびフローを作成することができる。

【００８７】

たとえば、ある実施形態に従うと、設計時システムは、グラフィカルユーザインターフェイスを提供することにより、ＨＵＢを作成してデータをインジェストしインジェストポリシーを規定することができる。インジェストポリシーは、時間ベースであってもよく、または、関連するデータフローからの要求に応じたものであってもよい。入力ＨＵＢが選択されると、データをこの入力ＨＵＢからサンプリングすることにより、ソースをプロファイリングすることができる。これはたとえば、メタデータクエリを実行すること、サンプルを取得すること、およびユーザ定義入力を取得すること等である。プロファイルはシステムＨＵＢに格納することができる。グラフィカルユーザインターフェイスは、データフローパイプラインを規定するために複数のソースを結合できるようにする。これは、スクリプトを作成することによって、または、ガイドされたエディタを用いることによって行うことができる。ガイドされたエディタによりデータを各ステップで視覚化することができる。グラフィカルユーザインターフェイスは、データクラウドを如何にしてたとえば修正、エンリッチ化、または結合し得るかを示唆するレコメンデーションサービスへのアク

セスを提供することができる。
【００８８】
ある実施形態に従うと、設計時間中、アプリケーション設計サービスは、得られたコンテンツを解析するのに適した構造を示唆することができる。アプリケーション設計サービスは、ナレッジサービス（関数型分類）を用いることにより、処置および関連するディメンション階層を示唆することができる。これが完了すると、設計時システムは、先のパイプラインからのブレンドされたデータを取得しディメンションターゲット構造をポピュレートするのに必要なデータフローをレコメンドすることができる。依存性解析に基づいて、オーケストレーションフローを導き出して生成することにより、ターゲットスキーマをロード／リフレッシュすることもできる。フォワードエンジニアリングユースケースの場合、設計時システムは、ＨＵＢを生成することにより、ターゲット構造をホストしターゲットスキーマを作成することもできる。
【００８９】
ある実施形態に従うと、実行時システムは、データ処理のためのサービスの実行時間中に処理を実行することができる。
【００９０】
ある実施形態に従うと、実行時または動作モードにおいて、ユーザが作成したポリシーおよびフローの規定が適用および／または実行される。たとえば、このような処理は、インジェスト、変換、モデル、およびパブリッシュサービスを呼び出してパイプライン内でデータを処理することを含み得る。
【００９１】
ある実施形態に従うと、実行時システムは、エッジレイヤ１７２と、スケーラブル入出力（Ｉ／Ｏ）レイヤ１７４と、分散処理システムまたは計算レイヤ１７６とを含み得る。実行時において（たとえば１つ以上の入力ＨＵＢ１１０からデータがインジェストされたときに）、イベントのためにエッジレイヤがデータを受けることができる。このイベントによってデータは生成される。
【００９２】
ある実施形態に従うと、イベントコーディネータ１６５は、設計時システムと実行時システムとの間で動作して、データフローアプリケーション（たとえばパイプライン、Lambdaアプリケーション）の設計、作成、モニタリング、および管理に関連するイベントを調整する。
【００９３】
ある実施形態に従うと、エッジレイヤは、データをスケーラブル入出力レイヤに送る。スケーラブル入出力レイヤはこのデータを分散処理システムまたは計算レイヤにルーティングする。
【００９４】
ある実施形態に従うと、分散処理システムまたは計算レイヤは、パイプラインプロセスを（テナント毎に）実現することにより、データを出力のために処理することができる。分散処理システムは、たとえばApache SparkおよびAlluxioを用いて実現することができ、データをデータレイクにサンプリングしその後このデータを出力ＨＵＢに出力することができる。分散処理システムは、データを起動して処理するためにスケーラブル入出力レイヤと通信することができる。
【００９５】
ある実施形態に従うと、上記コンポーネントのうちのいくつかまたはすべてを含むデータＡＩシステムは、たとえば１つ以上のプロセッサ（ＣＰＵ）とメモリとパーシステント記憶装置（１９８）とを含む１つ以上のコンピュータに提供することができる、またはこのコンピュータによって実行されることができる。
【００９６】
イベント駆動型アーキテクチャ
先に述べたように、ある実施形態に従うと、このシステムは、イベント駆動型アーキテク

チャ（ＥＤＡ）コンポーネントまたはイベントコーディネータを含み得る。これは、設計時システムと実行時システムとの間で動作することにより、データフローアプリケーション（たとえばパイプライン、Lambdaアプリケーション）の設計、作成、モニタリング、および管理に関連するイベントを調整する。

【００９７】

図２は、ある実施形態に係る、システムで使用するイベントコーディネータを含むイベント駆動型アーキテクチャを示す図である。

【００９８】

図２に示すように、ある実施形態に従うと、イベントコーディネータは、イベントキュー２０２（たとえばKafka）と、イベントブートストラッパサービス２０４（たとえばExecutorService）と、イベントコンフィギュレーションパブリッシャ／イベントコンシューマ２０６（たとえばＤＢＣＳ）とを含み得る。

【００９９】

ある実施形態に従うと、システムファサード２０８（たとえばイベントＡＰＩ拡張）で受信したイベントは、１つ以上のイベントブローカー２１０、たとえばKafkaコンシューマによって、システムのさまざまなコンポーネントに伝達することができる。たとえば、例として外部データ２１２（たとえばＳ３、ＯＳＣＳ、またはＯＧＧデータ）等のデータおよび／またはイベント、または、グラフィカルユーザインターフェイス２１４（たとえばブラウザまたはＤＦＭＬ　ＵＩ）からの入力は、イベントコーディネータを介して、その他のコンポーネント、たとえば上述のアプリケーション実行時２１６、データレイク、システムＨＵＢ、データＡＩサブシステム、アプリケーション設計サービス、および／またはインジェスト２２０、パブリッシュ２３０、スケジューリング２４０、またはその他のコンポーネントに、伝達することができる。

【０１００】

ある実施形態に従うと、イベントブローカーを、ストリームイベントのコンシューマとして構成してもよい。イベントブートストラッパは、複数の構成されたイベントブローカーを開始させることにより、登録されたサブスクライバに代わってイベントを処理することができる。所定のイベントの処理のために、各イベントブローカーは、イベントの処理を、登録されたコールバックエンドポイントに委任する。イベントコーディネータは、イベントタイプの登録、イベンティングエンティティの登録、イベントの登録、およびサブスクライバの登録を可能にする。表１は、パブリッシュイベントおよびサブスクライブイベントを含むさまざまなイベントオブジェクトの一例を提供する。

【０１０１】

【表１】

パブリッシュイベント	イベンティングエンティティによるイベントのパブリッシュを登録する２要素（イベントタイプ、イベンティングエンティティ）からなるオブジェクト。
サブスクライブイベント	別のイベンティングエンティティ（パブリッシャ）によってパブリッシュされたイベントへのサブスクリプションを登録する２要素（イベント、イベンティングエンティティ）からなるオブジェクト。

表１

【０１０２】

イベントタイプ

ある実施形態に従うと、イベントタイプは、システムにとって重要なイベントの状態変化を規定する。それはたとえば、ＨＵＢの作成、データフローアプリケーションたとえばパ

イプライン、Lambdaアプリケーションの修正、データセットまたはエンティティのためのデータのインジェスト、またはターゲットＨＵＢに対するデータのパブリッシュ等である。データフォーマットの一例と、さまざまなイベントタイプの例を、以下と表２に示す。

【０１０３】

【表２】

```
{
  "Id":"" ,  // generated at the time of creation
  "Name": "Hub - Creation"
  "Type":"User Event"
}
```

POST /eventTypes	新たな eventType を作成。
PUT /eventType/{eventide}	所定の ID を有する eventType を修正。
GET /eventTypes	システム内のすべての eventTypes を取り出す。
GET /eventTypes/{eventTypeId}	所定の ID を有する eventType の表現を取り出す。
GET /eventTypes/{eventide}/publishers	この eventType のすべてのパブリッシャを取り出す。このタイプのイベントをパブリッシュするために登録された２つ以上のイベンティングエンティティが存在し得る。その場合このタイプのイベントをパブリッシュするすべての異なるイベンティングエンティティが返される。
GET /eventTypes/{eventide}/subscribers	この eventType のすべてのサブスクライバを取り出す。このタイプのイベントへのサブスクライバとして登録された２つ以上のイベンティングエンティティが存在し得る。その場合このタイプのイベントにサブスクライブするすべての異なるイベンティングエンティティが返される。
DELETE /eventTypes/{eventide}	所定の ID を有するイベントを削除。

表２

【０１０４】

イベンティングエンティティ

ある実施形態に従うと、イベンティングエンティティは、イベントのパブリッシャおよび／またはサブスクライバであってもよい。たとえば、イベンティングエンティティは、１つ以上のイベントをパブリッシュするために登録することができる、および／または１つ以上のイベントのコンシューマとなることができる。これは、パブリッシュに対するアクノレッジを通知または送信しサブスクライブされたイベントの処理を委任するのに使用される、エンドポイントまたはコールバックＵＲＬの登録を含む。イベンティングエンティティの例は、メタデータサービス、インジェストサービス、システムＨＵＢアーティファ

クト、およびパイプライン、Lambdaアプリケーションを含み得る。データフォーマットの一例と、さまざまなイベンティングエンティティの例を、以下と表３に示す。

【０１０５】

【表３】

```
{
  "Id":"" , // generated at the time of creation
  "Name": "Metadata Service",
  "endPointURL":"localhost:9010/processEvent",
  "entityType":"DFMLService"
}
```

POST /eventingEntities	新たなイベンティングエンティティを作成。
PUT /eventingEntities/{entityId}	ID によって特定された既存のイベンティングエンティティを修正。
GET /eventingEntities	登録されたすべてのイベンティングエンティティを取り出す。
GET /eventingEntities/{entityId}	所定の ID を有するイベンティングエンティティの表現を取り出す。
GET /eventingEntities/{entityId}/eventsPublished	このイベンティングエンティティによるパブリッシュのために登録されたすべてのイベントを取り出す。
GET /eventingEntities/{entityId}/eventsSubscribed	このイベンティングエンティティによるサブスクライブのために登録されたすべてのイベントを取り出す。
DELETE /eventingEntities/{entityId}	このイベンティングエンティティをシステムから削除。

表３

【０１０６】

イベント

ある実施形態に従うと、イベントは、パブリッシャとして登録されたイベンティングエンティティに対応付けられたイベントタイプのインスタンスであり、サブスクライバ（イベンティングエンティティ）を有し得る。たとえば、メタデータサービスは、パブリッシュのためにＨＵＢ作成イベントを登録することができ、このイベント用のイベントインスタンス（作成したＨＵＢ１つ当たり１つのインスタンス）のうち１つ以上をパブリッシュすることができる。さまざまなイベントの例が表４に示される。

【０１０７】

【表４】

POST /events	イベンティングエンティティによってパブリッシュされる新たなイベントを作成。 { "Id":"" , "acknowledgeURL": "locahost:9010/eventAcknowledge", "onProcessingOf":"/eventType/{eventId}", "eventType":"/eventType/{eventId}", "eventingEntity":"/eventingEntity/{entitId}" }
PUT /events/{eventid}	イベンティングエンティティによるパブリッシュのために登録された既存のイベントを修正。
POST /events/{eventid}/publish	消費のためにイベントをキューに入れるイベントインスタンスをパブリッシュ。 { "event_type": "data", "subtype": "publication", "state": "ready", "context": {“eventContextId”:”{eventId}”,”accessToken”:””} "message": { _actual event data goes here_ } }
GET /events	システムに登録されたすべてのイベントを取り出す。
GET /events/{eventide}	所定の ID を有するイベントを取り出す。

GET /events/{eventide}/publisher	このイベントのパブリッシャを取り出す。
GET /events/{eventide}/subscribers	このイベントのサブスクライバを取り出す。
POST /events/{eventid}/subscribers	イベントのサブスクライバを登録。 { "Id":"" , // generated at the time of creation "processingURL": "locahost:9010/eventProcess", "subscribingEvent":"/events/{eventId}" , "callbackMethod":"REST", "subscriberentity":"/eventingEntity/{entitId}" }
PUT /events/{eventid}/subscribers/{subscriberId}	このイベントのサブスクライバのプロパティを修正。
DELETE /events/{eventid}/subscribers/{subscriberId}	このイベントのサブスクライバを削除。
DELETE /events/{eventide}	イベントを削除。

表４

【０１０８】

例

ある実施形態に従うと、以下の例は、イベントタイプの作成、パブリッシュイベントの登録、サブスクライバの登録、イベントのパブリッシュ、イベントタイプの取得、イベントタイプのためのパブリッシャの取得、およびイベントタイプのためのサブスクライバの取得を示す。

【０１０９】

【数１】

```
POST  http://den00tnk:9021/dfml/service/eventType

{"name":"DATA_INGESTED",
 "type":"SystemEvent"
}
```

【０１１０】

これは、ユニバーサル一意ＩＤ（universally unique ID：ＵＵＩＤ）、たとえば「8e8

7039b-a8b7-4512-862c-fdb05b9b8888」を返す。イベンティングオブジェクトは、システム内のイベントをパブリッシュまたはサブスクライブすることができる。たとえばインジェストサービス、メタデータサービス、およびアプリケーション設計サービス等のサービスエンドポイントは、アクノレッジ、通知、エラー、または処理のためにスタティックなエンドポイントとともに、パブリッシュまたはサブスクライブイベントであってもよい。ＤＦＭＬアーティファクト（たとえばDFMLEntity, DFMLLambdaApp, DFMLHub）も、イベンティングオブジェクトとして登録することができ、これらのタイプのインスタンスは、イベンティングオブジェクトとしての登録なしでイベントをパブリッシュまたはサブスクライブすることができる。

【０１１１】

【数２】

```
POST http://den00tnk:9021/dfml/service/eventEntity

{
  "name": "DFMLEntity",
  "endPointURL":"localhost:9010/<publisherURL>",

"notificationEndPointURL":"http://den00tnk:9021/<publisherURL>/notification
",
  "exceptionEndPointURL":"http://den00tnk:9021/<publisherURL>/exception",
  "errorEndPointURL":"http://den00tnk:9021/<publisherURL>/error",
  "entityType":"DFMLEntity"
}
```

【０１１２】

以下の例は、DFMLLambdaApps（タイプ）をイベンティングオブジェクトとして登録する。

【０１１３】

【数３】

```
{
  "name": "DFMLLambdaApps",
  "endPointURL":"localhost:9010/<publisherURL>",

"notificationEndPointURL":"http://den00tnk:9021/<publisherURL>/notification
",
  "exceptionEndPointURL":"http://den00tnk:9021/<publisherURL>/exception",
  "errorEndPointURL":"http://den00tnk:9021/<publisherURL>/error",
  "entityType":"DFMLLambdaApps"
}
```

【０１１４】

ＨＵＢ、EntityおよびLambdaAppタイプのイベンティングエンティティの場合、〈publisherURL〉をＲＥＳＴエンドポイントＵＲＬにアノテートすることができ、イベント駆動型アーキテクチャは、実際のＵＲＬを、ＤＦＭＬアーティファクトインスタンスＵＲＬを置換することによって導き出す。たとえば、notificationEndpointURLが、http://den00tnk:9021/〈publisherURL〉/notificationとして登録され、メッセージの一部と

して特定されたパブリッシャＵＲＬがhubs/1234/entities/3456である場合、通知のために呼び出されるＵＲＬは、http://den00tnk:9021/ hubs/1234/entities/3456 /notificationであろう。ＰＯＳＴはＵＵＩＤ、たとえば「185cb819-7599-475b-99a7-65e0bd2ab947」を返す。

【０１１５】

パブリッシュイベントの登録

ある実施形態に従うと、パブリッシュイベントは次のように登録できる。

【０１１６】

【数４】

```
POST http://den00tnk:9021/dfml/service/event

{
  "acknowledgeURL":"http://den00tnk:9021/<publisherURL>/acknowledge",
  "onProcessingOf":"/eventType/{eventId}",
  "eventType":"7ea92c868e87039b-a8b7-4512-862c-fdb05b9b8888",
  "publishingEntity":"185cb819-7599-475b-99a7-65e0bd2ab947"
}
```

【０１１７】

上記eventTypeは、イベントタイプDATA_INGESTEDの登録のために返されるＵＵＩＤであり、上記publishingEntityは、イベンティングオブジェクトとして登録されるDFML Entityタイプである。この登録は、ＵＵＩＤ、たとえば「2c7a4b6f-73ba-4247-a07a-806ef659def5」を返す。

【０１１８】

サブスクライバの登録

ある実施形態に従うと、サブスクライバは次のように登録できる。

【０１１９】

【数５】

```
POST http://den00tnk:9021/dfml/service/event/2c7a4b6f-73ba-4247-a07a-
806ef659def5/subscribers
```

【０１２０】

パブリッシュイベント登録から返されたＵＵＩＤは、サブスクライバの登録のための経路セグメントとして使用される。

【０１２１】

【数６】

```
{
  "processingURL": "http://den00tnk:9021/dfml/service/eventType/process3",
  "callbackMethod":"SYNC_POST",
  "subscriberEntity":"7599916b-baab-409c-bfe0-5334f111ef41",
  "publisherURL":"/hubs/1234/entities/3456",
  "publishingObjectType":"DFMLEntity",
  "subscribingObjectType":"DFMLLambdaApps",
  "subscriberURL":"/lambdaApps/123456"
}
```

【０１２２】
上記publisherURLおよびpublishingObjectTypeは、パブリッシャオブジェクトのインスタンスおよびタイプである。ここで、データフロー（たとえばLambda）アプリケーションは、ＵＲＩ /lambdaApps/123456の特定において、エンティティ /hubs/1234/entities/3456からのDATA_INGESTEDイベントへのサブスクライブに注目する。この登録はＵＵＩＤ、たとえば「1d542da1-e18e-4590-82c0-7fe1c55c5bc8」を返す。
【０１２３】
イベントのパブリッシュ
ある実施形態に従うと、イベントは以下のようにパブリッシュすることができる。
【０１２４】
【数７】

```
POST http://den00tnk:9021/dfml/service/event/publish

{
    "event_type":"DATA_AVAILABLE",
    "subtype":"publication",
    "state":"ready",
    "eventTypeId":"2c7a4b6f-73ba-4247-a07a-806ef659def5",
    "publisherURL":"dfml/service/eventType",
    "message": {"id":"1234",
        "descr":"something happened here testing this and this again"
}
}
```

【０１２５】
上記publisherURLは、パブリッシュオブジェクトが、DFMLEntity、DFMLHubまたはDFMLLambdaAppsのうちの１つである場合に使用され、サブスクライバが参加するメッセージをパブリッシュするイベンティングオブジェクトのインスタンスをチェックするために使用される。また、パブリッシャＵＲＬは、サブスクライバがメッセージの処理に成功したときに通知ＵＲＬを導き出すために使用される。このパブリッシュは、パブリッシュされたイベントの一部であったメッセージ本体を返す。
【０１２６】
イベントタイプの取得
ある実施形態に従うと、イベントタイプは以下のように求めることができる。
【０１２７】
【数８】

```
GET  http://den00tnk:9021/dfml/service/eventType

{
  "eventTypes": [
    {
      "Id": "8e87039b-a8b7-4512-862c-fdb05b9b8888",
      "name": "DATA_INGESTED",
      "type": "SystemEvent",
      "createdBy": " ",
      "updatedBy": " ",
      "description": " ",
      "typeQualifier": " ",
      "resourceType": " ",
      "verb": " ",
      "operationType": " ",
      "status": " ",
      "annotation": " "
    },
    {
      "Id": "7ea92c86-8db5-42d6-992a-2578a6d025ce",
      "name": "DATA_AVAILABLE",
      "type": "SystemEvent",
      "createdBy": " ",
      "updatedBy": " ",
      "description": " ",
      "typeQualifier": " ",
      "resourceType": " ",
      "verb": " ",
      "operationType": " ",
      "status": " ",
      "annotation": " "
    }
  ]
}
```

【０１２８】
イベントタイプ用のパブリッシャを取得
【０１２９】
【数９】

```
GET  http://den00tnk:9021/dfml/service/eventType/7ea92c86-8db5-42d6-992a-
2578a6d025ce/publishers

{
  "eventingObjects": [
    {
      "Id": "185cb819-7599-475b-99a7-65e0bd2ab947",
      "name": "DFMLEntity",
      "entityType": "DFMLEntity",
      "endPointURL": "localhost:9010/<publisherURL>",
      "notificationEndpointURL":
"http://den00tnk:9021/<publisherURL>/notification",
      "exceptionEndpointURL":
"http://den00tnk:9021/<publisherURL>/exception",
      "errorEndpointURL": "http://den00tnk:9021/<publisherURL>/error",
      "acknowledgeEndpointURL": " ",
      "description": " ",
      "entityQualifier": " ",
      "status": " ",
      "annotation": " "
    }
  ]
}
```

【０１３０】
イベントタイプ用のサブスクライバを取得
ある実施形態に従うと、イベントタイプ用のサブスクライバは以下のように求めることができる。
【０１３１】
【数１０】

```
GET  http://den00tnk:9021/dfml/service/eventType/7ea92c86-8db5-42d6-992a-
2578a6d025ce/subscribers

{
  "eventingObjects": [
    {
      "Id": "7599916b-baab-409c-bfe0-5334f111ef41",
      "name": "DFMLLambdaApps",
      "entityType": "DFMLLambdaApps",
      "endPointURL": "localhost:9010/<publisherURL>",
      "notificationEndpointURL":
"http://den00tnk:9021/<publisherURL>/notification",
      "exceptionEndpointURL":
"http://den00tnk:9021/<publisherURL>/exception",
      "errorEndpointURL": "http://den00tnk:9021/<publisherURL>/error",
      "acknowledgeEndpointURL": " ",
      "description": " ",
      "entityQualifier": " ",
      "status": " ",
      "annotation": " "
    }
  ]
}
```

【０１３２】
上記説明は、イベントコーディネータ、イベントタイプ、イベンティングエンティティ、およびイベントの特定の実施形態を説明するために、一例として示したものである。他の実施形態に従うと、他の種類のＥＤＡを用いて、設計時システムと実行時システムとの間で機能するシステム内における通信を提供することにより、データフローアプリケーションの設計、作成、モニタリング、および管理に関するイベントを調整することができ、他の種類のイベントタイプ、イベンティングエンティティ、およびイベントをサポートすることができる。
【０１３３】
データフロー機械学習（ＤＦＭＬ）のフロー
先に述べたように、各種実施形態に従うと、本システムは、データのフロー（データフロー、ＤＦ）の管理および複合データフローソフトウェアアプリケーション（たとえばデータフローアプリケーション、パイプライン、Lambdaアプリケーション）の構築に用いられる機械学習（ＭＬ、データフロー機械学習、ＤＦＭＬ）を活用する、データ統合またはその他のコンピューティング環境で使用することができる。
【０１３４】
図３は、ある実施形態に係る、データフロー内のステップを示す図である。
図３に示すように、ある実施形態に従うと、ＤＦＭＬデータフロー２６０の処理は、インジェストステップ２６２を含む複数のステップを含み得る。インジェストステッにおいて、データを、さまざまなソース、たとえばSalesforce（ＳＦＤＣ）、Ｓ３、またはＤＢａａＳから、インジェストすることができる。
【０１３５】
データ準備ステップ２６４において、インジェストされたデータを、たとえば、重複排除

、標準化、またはエンリッチ化することによって準備することができる。
【０１３６】
変換ステップ２６６において、本システムは、１以上のマージ、フィルタ、またはデータセットのルックアップを実行することにより、データを変換することができる。
【０１３７】
モデルステップ２６８において、１つ以上のモデルが、当該モデルに対するマッピングとともに生成される。
【０１３８】
パブリッシュステップ２７０において、本システムは、モデルをパブリッシュし、ポリシーおよびスケジュールを特定し、ターゲットデータ構造をポピュレートすることができる。
【０１３９】
ある実施形態に従うと、本システムは、そのデータ準備ステップ、変換ステップ、およびモデルステップ各々の全体を通して、サーチ／レコメンド機能２７２の使用をサポートする。ユーザは、データ統合フレームワーク内に機能の幅が封じ込められた一組の明確に規定されたサービスを通して、システムとのやり取りを行うことができる。この一組のサービスは、システムの論理ビューを規定する。たとえば、設計モードにおいて、ユーザは、特定のユースケースの機能的要求を規定するフロー、アーティファクト、およびポリシーを作成することができる。
【０１４０】
図４は、ある実施形態に係る、複数のソースを含むデータフローの一例を示す図である。
【０１４１】
図４に示す一例としてのデータフロー２８０に示されるように、ある実施形態に従うと、必要なのは、ここではＳＦＤＣおよびＦＡＣＳ（Fusion Apps Cloud Service）として示されている複数のソース２８２からのコンテンツを、ＯＳＣＳ（Oracle Storage Cloud Service）内のいくつかのファイルとともに取り込んで、この情報を、所望のコンテンツの解析に使用できるようにブレンドし、ターゲットキューブおよびディメンションを導き出し、ブレンドしたコンテンツをターゲット構造にマッピングし、このコンテンツがディメンションモデルとともにオラクル・ビジネス・インテリジェンス・クラウドサービス（Oracle Business Intelligence Cloud Service：ＢＩＣＳ）環境で使用できるようにすることであり、これは、インジェスト、変換２６６Ａ／２６６Ｂ、モデル、オーケストレート２９２、およびデプロイ２９４ステップを含む。
【０１４２】
示されている例は、本明細書に記載の技術を説明するために提供され、本明細書に記載の機能は、これら特定のデータソースとともに使用することに限定されない。
【０１４３】
ある実施形態に従うと、インジェストステップにおいて、ＳＦＤＣコンテンツにアクセスしこれをインジェストするために、ＨＵＢがデータレイクにおいて作成されてこのコンテンツを受ける。これは、たとえば、関連するアクセスモード（ＪＤＢＣ、ＲＥＳＴ、ＳＯＡＰ）のＳＦＤＣアダプタを選択し、ＨＵＢを作成し、名称を与え、時間ベースの、または関連するデータフローの要求に応じたインジェストポリシーを規定することによって、実施することができる。
【０１４４】
ある実施形態に従うと、その他２つのソースについて同様のプロセスを実行することができる。違いは、ＯＳＣＳソースの場合、スキーマは、開始時点でわかっていない場合があり、代わりに何らかの手段（たとえばメタデータクエリ、サンプリング、またはユーザ規定）によって取得できる点である。
【０１４５】
ある実施形態に従うと、データのソースを、任意でプロファイリングすることにより、さらに調査することができる。これは、統合フローにおいて後にレコメンデーションを導き出すのに役立ち得る。

【０１４６】
ある実施形態に従うと、次のステップは、如何にして別々のソースを中心アイテムの周りで結合するかを規定することである。これは、典型的には解析の基礎（ファクト）であり、データフローパイプラインを規定することによって実現できる。これは、直接、パイプラインドメイン固有言語（ＤＳＬ）スクリプトを作成することによって、または、ガイドされたエディタを用いて、行うことができる。ガイドされたエディタの場合、ユーザは、各ステップでデータに対する効果を知ることができ、データを如何にしてたとえば修正、エンリッチ化、結合できるかを示唆するレコメンデーションサービスを使用することができる。
【０１４７】
この時点で、ユーザは、システムが、得られたコンテンツを解析するのに適した構造を示唆することを、要求できる。たとえば、ある実施形態に従うと、システムは、ナレッジサービス（関数型分類）を用いることにより、対策および関連するディメンション階層を示唆することができる。これが完了すると、システムは、先のパイプラインからのブレンドされたデータを取得しディメンションターゲット構造をポピュレートするのに必要なデータフローをレコメンドすることができる。これはまた、依存性解析に基づいて、オーケストレーションフローを導き出して生成することにより、ターゲットスキーマをロード／リフレッシュする。
【０１４８】
ある実施形態に従い、本システムは次に、ターゲット構造をホストするＨＵＢを生成し、これを、アダプタを介して、ターゲットスキーマを作成するのに必要なデータ定義言語（data definition language：ＤＤＬ）を生成するＤＢＣＳに対応付け、たとえば、新たに作成されたスキーマにアクセスするのに必要なモデルを生成するのにＢＩＣＳが使用できるＸＤＭＬまたは何らかの形態をデプロイすることができる。これは、オーケストレーションフローを実行し排出（exhaust）サービスをトリガすることによってポピュレートすることができる。
【０１４９】
図５は、ある実施形態に係る、パイプラインを有するデータフローの使用の一例を示す図である。
【０１５０】
図５に示すように、ある実施形態に従うと、本システムにより、ユーザは、この例ではパイプラインステップＳ１～Ｓ５を含むデータフローを表すパイプライン３０２を規定することにより、アプリケーションとして構築／実行３０４されるときのデータの処理を記述することができる。
【０１５１】
たとえば、ある実施形態に従うと、ユーザは、インジェスト、変換、モデル、およびパブリッシュサービス、またはその他のサービスたとえばポリシー３０６、実行３１０、またはパーシステンスサービス３１２を呼び出して、パイプライン内のデータを処理することができる。また、ユーザは、ソリューション（すなわちコントロールフロー）を規定することにより、関連するパイプラインを統合することができる統一フローを特定することもできる。典型的に、ソリューションは、完全なユースケースたとえば売上キューブおよび関連するディメンションのローディングをモデル化する。
【０１５２】
データＡＩシステムコンポーネント
ある実施形態に従うと、アダプタは、さまざまなエンドポイントへの接続およびさまざまなエンドポイントからのデータのインジェストを可能にし、アプリケーションまたはソースタイプに固有である。
【０１５３】
ある実施形態に従うと、本システムは、予め定められたアダプタのセットを含み得る。これらのアダプタのうちのいくつかは、他のＳＯＡアダプタを活用することができ、追加の

アダプタをフレームワークに登録できるようにする。所定の接続タイプに対して２つ以上のアダプタがあってもよい。その場合、インジェストエンジンは、ＨＵＢの接続タイプ構成に基づいて最も適したアダプタを選択する。

【０１５４】

図６は、ある実施形態に係る、インジェスト／パブリッシュエンジンおよびインジェスト／パブリッシュサービスの使用の一例を示す図である。

【０１５５】

図６に示すように、ある実施形態に従うと、パイプライン３３４は、インジェスト／パブリッシュエンジン３３０に、インジェスト／パブリッシュサービス３３２を介してアクセスすることができる。パイプライン３３４は、この例において、データ（たとえば売上データ）を入力ＨＵＢ（たとえばＳＦＤＣ　ＨＵＢ１）からインジェストし３３６、インジェストしたデータを変換し３３８、このデータを出力ＨＵＢ（たとえばOracle HUB）に対してパブリッシュする３４０ように、設計されている。

【０１５６】

ある実施形態に従うと、インジェスト／パブリッシュエンジンは、複数の接続タイプ３３１をサポートする。これらのタイプの１つである接続タイプ３４２は、ＨＵＢへのアクセスを提供する１つ以上のアダプタ３４４に対応付けられている。

【０１５７】

たとえば、図６の例に示されるように、ある実施形態に従うと、ＳＦＤＣ接続タイプ３５２は、ＳＦＤＣ　ＨＵＢ３５８および３５９へのアクセスを提供するＳＦＤＣ－Ａｄｐ１アダプタ３５４およびＳＦＤＣ－Ａｄｐ２アダプタ３５６に対応付けることができ、ＥｘＤａａＳ接続タイプ３６２は、ＥｘＤａａｓ　ＨＵＢ３６６へのアクセスを提供するＥｘＤａａＳ－Ａｄｐアダプタ３６４に対応付けることができ、Oracle接続タイプ３７２は、Ｏｒａｃｌｅ　ＨＵＢ３７６へのアクセスを提供するＯｒａｃｌｅ　Ａｄｐアダプタ３７４に対応付けることができる。

【０１５８】

レコメンデーションエンジン

ある実施形態に従うと、本システムは、データに対して実行できる可能ないくつかのアクションの中から最も関連性が高いものを予測／示唆する専門フィルタリングシステムとして動作するレコメンデーションエンジンまたはナレッジサービスを含み得る。

【０１５９】

ある実施形態に従うと、レコメンデーションを連結することにより、ユーザが所定の最終ゴールを達成するためにこれらのレコメンデーションを辿り易くすることができる。たとえば、レコメンデーションエンジンは、データセットをデータキューブに変換してターゲットＢＩシステムに対してパブリッシュするときの一組のステップを通してユーザを案内することができる。

【０１６０】

ある実施形態に従うと、レコメンデーションエンジンは、（Ａ）ビジネス型分類、（Ｂ）関数型分類、および（Ｃ）ナレッジベースという３つの側面を利用する。データセットまたはエンティティに対するオントロジー管理およびクエリ／サーチ機能は、たとえば、クエリＡＰＩ、ＭＲＳおよび監査リポジトリとの、ＹＡＧＯ３に由来する共有オントロジーによって、提供することができる。ビジネスエンティティ分類は、たとえば、ビジネス型を識別するＭＬパイプラインベースの分類によって提供することができる。関数型分類は、たとえば、演繹的なルールベースの関数型分類によって提供することができる。アクションレコメンデーションは、たとえば、帰納的なルールベースのデータ準備、変換、モデル、依存性、および関連するレコメンデーションによって提供することができる。

【０１６１】

分類サービス

ある実施形態に従うと、本システムは、ビジネス型分類と関数型分類とにカテゴライズできる分類サービスを提供する。各々について以下でさらに説明する。

【０１６２】
ビジネス型分類
ある実施形態に従うと、エンティティのビジネス型はその表現型（phenotype）である。エンティティ内の個々の属性の観測可能な特性は、エンティティのビジネス型の特定において、定義と同様に重要である。分類アルゴリズムは、データセットまたはエンティティの概略的定義を使用するが、データを用いて構築したモデルを利用してデータセットまたはエンティティのビジネス型を分類することもできる。
【０１６３】
たとえば、ある実施形態に従うと、ＨＵＢからインジェストされたデータセットは、システムが知っている既存のビジネス型（メインＨＵＢからシードされたもの）のうちの１つとして分類することができる、または、既存のビジネス型に分類できない場合は新たな型として追加することができる。
【０１６４】
ある実施形態に従うと、ビジネス型分類は、レコメンデーションを行う際に、帰納的推論（パイプライン内の類似するビジネス型に対して規定された変換からの）、または、分類ルートエンティティから導出された単純な命題に基づいて、利用される。
【０１６５】
概要を述べる。ある実施形態に従うと、分類プロセスは以下の一組のステップによって説明される。これらのステップは、メイン（トレーニング）ｈｕｂからインジェストしシードするステップ、モデルを構築しカラム統計を計算しこれらを分類に使用するために登録するステップ、プロファイルの作成／カラム統計の計算を含む、新たに追加されたｈｕｂからのデータセットまたはエンティティを分類するステップ、データセットまたはエンティティを分類して、構造およびカラム統計に基づき、使用するエンティティモデルのショートリストを提供するステップ、ならびに、マルチクラス分類を含むデータセットまたはエンティティを分類し、モデルを用いて計算／予測するステップである。
【０１６６】
図７は、ある実施形態に従う、ＨＵＢからのインジェストおよびトレーニングのプロセスを示す図である。
【０１６７】
図７に示すように、ある実施形態に従うと、ＨＵＢ３８２（たとえばこの例ではＲｅｌａｔｅｄＩＱソース）からのデータは、レコメンデーションエンジン３８０が、データセット３９０（たとえばレジリエント分散データセット（Resilient Distributed Dataset：ＲＤＤ）として読み出すことができる。このデータセットは、この例では、アカウントデータセット３９１、イベントデータセット３９２、コンタクトデータセット３９３、リストデータセット３９４、およびユーザデータセット３９５を含む。
【０１６８】
ある実施形態に従うと、複数の型分類ツール４００を、ＭＬパイプライン４０２とともに使用することができる。たとえば、ＧｒａｐｈＸ４０４、Ｗｏｌｆｒａｍ／Ｙａｇｏ４０６、および／またはＭＬｌｉｂ統計４０８は、ＨＵＢが最初に登録されたときにナレッジグラフ４４０にエンティティメタデータ（トレーニングまたはシードデータ）をシードする際に使用することができる。
【０１６９】
ある実施形態に従うと、データセットまたはエンティティメタデータおよびデータは、ソースＨＵＢからインジェストされデータレイクに格納される。モデル生成４１０中、エンティティメタデータ（属性および他のエンティティとの関係）は、たとえばＦＰ－ｇｒｏｗｔｈロジスティック回帰４１２を通して、モデル４２０およびナレッジグラフの生成に使用される。ナレッジグラフは、すべてのデータセットまたはエンティティ、この例ではイベント４２２、アカウント４２４、コンタクト４２６、およびユーザ４２８を表す。シードの一部として、回帰モデルをデータセットまたはエンティティデータを用いて構築し、属性統計（最小値、最大値、平均値、または確率密度）を計算する。

【０１７０】
図８は、ある実施形態に係る、モデル構築プロセスを示す図である。
図８に示すように、ある実施形態に従うと、たとえばＳｐａｒｋ環境４３０内で実行する場合、Ｓｐａｒｋ　ＭＬｌｉｂ統計を用いて、ナレッジグラフに属性プロパティとして追加されたカラム統計を計算することができる。計算したカラム統計を、他のデータセットまたはエンティティメタデータとともに用いて、その回帰モデルが分類の新たなエンティティのテストにおいて使用されるエンティティを、ショートリストに入れることができる。
【０１７１】
図９は、ある実施形態に係る、新たに追加されたＨＵＢからのデータセットまたはエンティティを分類するプロセスを示す図である。
【０１７２】
図９に示すように、ある実施形態に従うと、新たなＨＵＢ、この例ではＯｒａｃｌｅ　ＨＵＢ４４２が追加されると、このＨＵＢから提供されたデータセットまたはエンティティ、たとえば当事者情報４４４および顧客情報４４６が、モデルにより、先に作成されたトレーニングまたはシードデータに基づいて、当事者４４８として分類される。
【０１７３】
たとえば、ある実施形態に従うと、カラム統計を、新たなデータセットまたはエンティティのデータから計算し、エンティティのサブグラフを表す一組の述語（predicate）を、この情報を、インジェストの一部として利用できる他のメタデータとともに用いて、作成する。
【０１７４】
ある実施形態に従うと、カラム統計の計算は、最尤推定（maximum likelihood estimation：ＭＬＥ）法において有用であり、一方、サブグラフはデータセットの回帰モデルにおいて有用である。新たなエンティティのために生成された一組のグラフ述語を用いて、新たなエンティティのテストおよび分類のために、候補エンティティモデルをショートリストに入れる。
【０１７５】
図１０は、ある実施形態に係る、新たに追加されたＨＵＢからのデータセットまたはエンティティを分類するプロセスをさらに示す図である。
【０１７６】
図１０に示すように、ある実施形態に従うと、分類対象である、新たなデータセットまたはエンティティのサブグラフを表す述語を、既にナレッジグラフの一部であるデータセットまたはエンティティを表す同様のサブグラフと比較する４５０。一致の確率に基づく一致するエンティティのランキングを、新たなエンティティの分類のためのテストで使用するエンティティモデルをショートリストに入れる際に使用する。
【０１７７】
図１１は、ある実施形態に係る、新たに追加されたＨＵＢからのデータセットまたはエンティティを分類するプロセスをさらに示す図である。
【０１７８】
図１１に示すように、ある実施形態に従うと、ショートリストに入れた一致するデータセットまたはエンティティの回帰モデルを、新たなデータセットまたはエンティティからのデータのテストに使用する。ＭＬパイプラインを拡張して追加の分類方法／モデルを包含することにより、プロセスの精度を改善することができる。分類サービスは、許容可能な閾値、たとえばこの例では０．８よりも高い確率以内に一致がある場合、新たなエントリを分類する４５２。そうでなければ、このデータセットまたはエンティティを新たなビジネス型としてナレッジグラフに追加することができる。また、ユーザは、結果を受容または拒否することによって分類を検証することができる。
【０１７９】
関数型分類
ある実施形態に従うと、エンティティの関数型はその遺伝子型（genotype）である。関

数型は、それを通して変換アクションが規定されるインターフェイスとして説明することもできる。たとえば、結合変換またはフィルタは、この場合はリレーショナルエンティティ等の関数型で規定される。要約すると、すべての変換は、パラメータとしての関数型で規定される。

【０１８０】

図１２は、ある実施形態に係る、関数型分類に使用するオブジェクト図を示す図である。

【０１８１】

オブジェクト図４６０による図１２に示すように、ある実施形態に従うと、本システムは、データセットまたはエンティティを評価してその関数型を特定するときの基準となる一組のルールを通して、一般的なケース（この例ではディメンション、レベル、またはキューブ）を説明することができる。

【０１８２】

たとえば、ある実施形態に従うと、多次元キューブは、その測定属性および次元によって説明することができる。これらは各々それ自体それらのタイプおよびその他の特性によって規定できる。ルールエンジンは、ビジネス型エンティティを評価しその関数型を評価に基づいてアノテートする。

【０１８３】

図１３は、ある実施形態に係る、ディメンション関数型分類の一例を示す図である。

図１３に示す一例としての関数型分類４７０の階層に示すように、ある実施形態に従うと、レベルは、たとえばそのディメンションおよびレベル属性によって規定することができる。

【０１８４】

図１４は、ある実施形態に係る、キューブ関数型分類の一例を示す図である。

図１４に示す一例としての関数型分類４８０の階層に示すように、ある実施形態に従うと、キューブは、たとえばその測定属性および次元によって規定することができる。

【０１８５】

図１５は、ある実施形態に係る、ビジネスエンティティの関数型を評価するための関数型分類の使用の一例を示す図である。

【０１８６】

図１５に示すように、この例４９０において、ある実施形態に従うと、売上データセットは、ルールエンジンによって、キューブ関数型として評価されねばならない。同様に、製品、顧客、および時間は、ディメンションおよびレベル（たとえば年齢グループ、性別）として評価されねばならない。

【０１８７】

ある実施形態に従うと、この具体例におけるエンティティの関数型およびデータセットまたはエンティティ要素を特定するルールが以下に示される。これは、同じ関数型の評価のために指定できるいくつかのルールを含む。たとえば、タイプ「Date」のカラムは、親レベルのエンティティに対するリファレンスがあるか否かに関わらず、ディメンションとして考慮することができる。同様に、郵便番号、性別、および年齢は、これらをディメンションとして特定するのにデータルールしか必要としない場合がある。

【０１８８】

【数１１】

顧客
Id, Name → (Dimension isComposedOf DimensionAttrib)
AgeGroup, Gender → (Dimension isComposedOf IdAttrib, IdAttrib references Level)

時間
Day → (Dimension /Level isComposedOf DimensionAttrib /LevelAttrib)
Month → (Dimension isComposedOf IdAttrib, IdAttrib references Level)

売上
Qty, Price, Amount → (Cube isComposedOf CubeAttr and Data rule on this columns, for example numeric min/max values, probability density)
Custid → (DimAttr references Dimension, Cube isComposedOf CubeAttr)
Date → (references Dimension, Cube isComposedOf CubeAttr)

【０１８９】
図１６は、ある実施形態に係る、関数変換に使用するオブジェクト図を示す図である。
図１５に示すように、この例５００において、ある実施形態に従うと、変換関数は関数型で規定することができる。ビジネスエンティティ（ビジネス型）は関数型としてアノテートされ、これは、デフォルトで、複合ビジネス型は関数型「エンティティ」であることを含む。
【０１９０】
図１７は、ある実施形態に係る、レコメンデーションエンジンの動作を示す図である。
図１７に示すように、ある実施形態に従うと、レコメンデーションエンジンは、ビジネス型で規定された一組のアクションであるレコメンデーションを生成する。各アクションは、データセットに対する変換を適用することを要求する指令である。
【０１９１】
ある実施形態に従うと、レコメンデーションコンテキスト５３０は、レコメンデーションのソースを抽出し、レコメンデーションを生成した一組の命題を特定するメタデータを含む。このコンテキストにより、レコメンデーションエンジンは、ユーザのレスポンスに基づいてレコメンデーションを学習しそれに優先順位を付けることができる。
【０１９２】
ある実施形態に従うと、ターゲットエンティティ演繹／マッピング部５１２は、ターゲットの定義（およびデータセットまたはエンティティおよび属性ビジネス型をアノテートする分類サービス）を用いて、現在のデータセットをターゲットにマッピングすることを容易にする変換レコメンデーションを行う。これは、ユーザが、わかっているターゲットオブジェクト（たとえば売上キューブ）で開始し、パイプラインを構築してキューブをインスタンス化するときに、よくあることである。
【０１９３】
ある実施形態に従うと、テンプレート（パイプライン／ソリューション）５１４は、再使用可能な一組のパイプラインステップおよび変換を規定して所望の最終結果を得る。たとえば、テンプレートがステップを含むことにより、エンリッチ化し、変換し、データマートに対してパブリッシュしてもよい。この場合の一組のレコメンデーションは、テンプレート設計を反映する。
【０１９４】
ある実施形態に従うと、分類サービス５１６は、ＨＵＢからデータレイクにインジェストしたデータセットまたはエンティティのビジネス型を特定する。エンティティに対するレ

コメンデーションは、同様のエンティティ（ビジネス型）に適用された変換に基づいて、または、ターゲットエンティティ演繹／マッピング部に関連して行うことができる。

【０１９５】

ある実施形態に従うと、関数型サービス５１８は、規定されたルールに基づいてデータセットまたはエンティティが持つことができる関数型をアノテートする。たとえば、所定のデータセットからキューブを生成するためまたはそれをディメンションテーブルに結合するためには、キューブの関数型を規定するルールにデータセットが適合するか否かを評価することが重要である。

【０１９６】

ある実施形態に従うと、パイプラインコンポーネント５２０からのパターン推論により、レコメンデーションエンジンは、類似するコンテキストの既存のパイプライン定義における所定のビジネス型に基づいて実行された変換をサマライズし、類似する変換を現在のコンテキストのレコメンデーションとして提案することができる。

【０１９７】

ある実施形態に従うと、レコメンデーションコンテキストを用いて、レコメンデーション５３２を処理することができ、これは、アクション５３４、変換関数５３５、アクションパラメータ５３６、関数パラメータ５３７、およびビジネス型５３８を含む。

【０１９８】

データレイク／データ管理戦略

先に述べたように、ある実施形態に従うと、データレイクは、システムＨＵＢまたはその他のコンポーネントからの情報のパーシステンスのリポジトリを提供する。

【０１９９】

図１８は、ある実施形態に係るデータレイクの使用を示す図である。

図１８に示すように、ある実施形態に従うと、データレイクは、１つ以上のデータアクセスＡＰＩ５４０、キャッシュ５４２、およびパーシステンスストア５４４に対応付けることができる。これらは共に動作することにより、正規化されているインジェストされたデータを、複数のパイプライン５５２、５５４、５５６で使用するために受ける。

【０２００】

ある実施形態に従うと、多種多様なデータ管理戦略を用いてデータ（パフォーマンス、スケーラビリティ）を管理するとともにデータレイクにおけるそのライフサイクルを管理することができる。これは、大まかにデータ駆動型またはプロセス駆動型に分類できる。

【０２０１】

図１９は、ある実施形態に係る、データ駆動型戦略を使用してデータレイクを管理することを示す図である。

【０２０２】

図１９に示すように、ある実施形態に従うと、データ駆動型アプローチでは、管理単位がＨＵＢまたはデータサーバ規定に基づいて導出される。たとえば、このアプローチにおいて、Ｏｒａｃｌｅ１　ＨＵＢからのデータは、このＨＵＢに対応付けられた第１のデータセンター５６０に格納することができ、ＳＦＨＵＢ１からのデータは、このＨＵＢに対応付けられた第２のデータセンター５６２に格納することができる。

【０２０３】

図２０は、ある実施形態に係る、プロセス駆動型戦略を使用してデータレイクを管理することを示す図である。

【０２０４】

図２０に示すように、ある実施形態に従うと、プロセス駆動型アプローチでは、管理単位がデータにアクセスする関連のパイプラインに基づいて導出される。たとえば、このアプローチにおいて、売上パイプラインに対応付けられたデータは、このパイプラインに対応付けられた第１のデータセンター５６４に格納することができ、他のパイプライン（たとえばパイプライン１、２、３）からのデータは、これら他のパイプラインに対応付けられた第２のデータセンター５６６に対応付けることができる。

【０２０５】
パイプライン
ある実施形態に従うと、パイプラインは、インジェストされたデータに対して実行すべき変換または処理を規定する。処理済みのデータは、データレイクに格納されてもよく、または、たとえばＤＢＣＳのような別のエンドポイントに対してパブリッシュされてもよい。
【０２０６】
図２１は、ある実施形態に係るパイプラインコンパイラの使用を示す図である。
図２１に示すように、ある実施形態に従うと、パイプラインコンパイラ５８２は、設計環境５７０と実行環境５８０との間で動作する。これは、１つ以上のパイプラインメタデータ５７２およびＤＳＬ、たとえばＪａｖａ（登録商標） ＤＳＬ５７４、ＪＳＯＮ ＤＳＬ５７６、Ｓｃａｌａ ＤＳＬ５７８を受け取ることと、実行環境で使用される出力を、たとえばＳｐａｒｋアプリケーション５８４および／またはＳＱＬステートメント５８６として与えることとを含む。
【０２０７】
図２２は、ある実施形態に係る、パイプライングラフの一例を示す図である。
図２２に示すように、ある実施形態に従うと、パイプライン５８８はパイプラインステップのリスト含む。異なる種類のパイプラインステップは、このパイプライン内で実行できる異なる種類の動作を表している。各パイプラインステップは、一般的にパイプラインステップパラメータによって記述される、複数の入力データセットと複数の出力データセットとを含み得る。パイプライン内における動作の処理順序は、前のパイプラインステップからの出力パイプラインステップパラメータを、次のパイプラインステップに結合することによって規定される。このようにして、パイプラインステップと、パイプラインステップパラメータ間の関係とが、有向非巡回グラフ（directed acyclic graph：ＤＡＧ）を形成する。
【０２０８】
ある実施形態に従うと、パイプラインは、パイプラインの入力および出力パイプラインステップパラメータを表す１つ以上の固有パイプラインステップ（シグネチャパイプライン）をパイプラインが含む場合、別のパイプラインで再使用することができる。囲んでいるパイプラインは、（パイプライン使用（pipeline usage））パイプラインステップで再使用されるパイプラインである。
【０２０９】
図２３は、ある実施形態に係る、データパイプラインの一例を示す図である。
図２３に示す一例としてのデータパイプライン６００に示されるように、ある実施形態に従うと、データパイプラインはデータ変換を実行する。パイプライン内におけるデータの流れは、パイプラインステップパラメータの結合として表される。さまざまな種類のパイプラインステップが、異なる変換動作のためにサポートされる。これはたとえば、エンティティ（データをデータレイクから取り出す、または処理済みのデータをデータレイク／他のＨＢＵに対してパブリッシュする）、および結合（複数のソースの融合）を含む。
【０２１０】
図２４は、ある実施形態に係る、データパイプラインの別の例を示す図である。
図２４に示す一例としてのデータパイプライン６１０に示されるように、ある実施形態に従うと、データパイプラインＰ１は、別のデータパイプラインＰ２で再使用することができる。
【０２１１】
図２５は、ある実施形態に係る、オーケストレーションパイプラインの一例を示す図である。
【０２１２】
図２５に示す一例としてのオーケストレーションパイプライン６２０に示されるように、ある実施形態に従うと、オーケストレーションパイプラインを使用した場合、パイプラインステップを用いて、オーケストレーションフロー全体において実行する必要があるタス

クまたはジョブを表すことができる。オーケストレーションパイプライン内のパイプラインステップはすべて、１つの入力パイプラインステップパラメータと１つの出力パイプラインステップパラメータとを有するものとする。タスク間の実行依存性は、パイプラインステップパラメータ間の結合として表すことができる。

【０２１３】

ある実施形態に従うと、タスクの並列実行は、パイプラインステップが、条件なしで、前にある同じパイプラインステップに依存している場合（すなわちフォーク（fork））、スケジュールすることができる。あるパイプラインステップが、前にある複数のパスに依存している場合、このパイプラインステップは、それ自体の実行に先立って複数のパスすべてが完了するのを待つ（すなわち結合（join））。しかしながら、これは、タスクが並列に実行されることを常に意味する訳ではない。オーケストレーションエンジンは、利用できるリソースに応じて、タスクを直列で実行するか並列で実行するか否かを判定することができる。

【０２１４】

図２５に示す例において、ある実施形態に従うと、パイプラインステップ１が最初に実行される。パイプラインステップ２およびパイプラインステップ３が並列で実行される場合、パイプラインステップ４は、パイプラインステップ２およびパイプラインステップ３がいずれも完了してから実行される。オーケストレーションエンジンはまた、このオーケストレーションパイプラインを、パイプラインステップ間の依存性を満たしている限り、直列に（パイプラインステップ１、パイプラインステップ２、パイプラインステップ３、パイプラインステップ４）または（パイプラインステップ１、パイプラインステップ３、パイプラインステップ２、パイプラインステップ４）として実行することができる。

【０２１５】

図２６は、ある実施形態に係る、オーケストレーションパイプラインの一例をさらに示す図である。

【０２１６】

図２６に示す一例としてのパイプライン６２５に示されるように、ある実施形態に従うと、各パイプラインステップは、自身のセマンティクスに応じて、たとえば成功またはエラーステータス等のステータス６３０をリターンすることができる。２つのパイプラインステップ間の依存性は、パイプラインステップのリターンステータスに基づいて条件付きであってもよい。示されている例では、パイプラインステップ１が最初に実行され、首尾よく終了した場合はパイプラインステップ２が実行され、そうでなければパイプラインステップ３が実行される。パイプラインステップ２またはパイプライン３いずれかが実行された後に、パイプラインステップ４が実行される。

【０２１７】

ある実施形態に従うと、オーケストレーションパイプラインを入れ子にすることにより、あるオーケストレーションパイプラインが別のオーケストレーションパイプラインをパイプライン使用（pipeline usage）を通して参照できるようにしてもよい。オーケストレーションパイプラインは、パイプライン使用（pipeline usage）としてのデータパイプラインを参照することもできる。オーケストレーションパイプラインとデータパイプラインとの相違は、オーケストレーションパイプラインが、シグネチャパイプラインステップを含まないデータパイプラインを参照するのに対し、データパイプラインは、シグナチャパイプラインステップを含む別のデータパイプラインを再使用できる点にある。

【０２１８】

ある実施形態に従うと、パイプラインステップの種類およびコード最適化に応じて、データパイプラインを、Ｓｐａｒｋクラスタ内で実行する単一のＳｐａｒｋアプリケーションとして、または、ＤＢＣＳ内で実行される複数のＳＱＬステートメントとして、または、ＳＱＬおよびＳｐａｒｋコードの混合として、生成することができる。オーケストレーションパイプラインの場合、下にある実行エンジン内、またはたとえばＯｏｚｉｅ等のワークフロースケジュールコンポーネント内で実行するために生成することができる。

【０２１９】
コーディネーションファブリック
ある実施形態に従うと、コーディネーションファブリックまたはファブリックコントローラは、フレームワークコンポーネント（サービスプロバイダ）をデプロイし管理するのに必要なツールと、（ユーザ設計）アプリケーションとを提供し、アプリケーションの実行とリソース要求／割当を管理し、統合フレームワーク（メッセージングバス）を提供することにより、各種コンポーネント間のやり取りを容易にする。
【０２２０】
図２７は、ある実施形態に係る、メッセージングシステムを含むコーディネーションファブリックの使用を示す図である。
【０２２１】
図２７に示すように、ある実施形態に従うと、メッセージングシステム（たとえばＫａｆｋａ）６５０は、リソースマネージャ６６０（たとえばＹａｒｎ／Ｍｅｓｏｓ）と、スケジューラ６６２（たとえばＣｈｒｏｎｏｓ）と、アプリケーションスケジューラ６６４（たとえばＳｐａｒｋ)と、ここではノード６５２、６５４、６５６、６５８として示されている複数のノードとの間のインタラクションを調整する。
【０２２２】
ある実施形態に従うと、リソースマネージャを用いてデータ計算タスク／アプリケーションのライフサイクルを管理する。これは、スケジューリング、モニタリング、アプリケーション実行、リソース調停および割当、ロードバランシングを含み、メッセージ駆動型コンポーネント統合フレームワーク内における（メッセージの作成者および消費者である）コンポーネントの構成およびデプロイを管理すること、ダウンタイムなしでコンポーネント（サービス）をアップグレードすること、および、サービスの混乱を最小にまたはなくした状態でインフラストラクチャをアップグレードすることを含む。
【０２２３】
図２８は、ある実施形態に係る、メッセージングシステムを含むコーディネーションファブリックの使用をさらに示す図である。
【０２２４】
図２８に示すように、ある実施形態に従うと、コーディネーションファブリック内のコンポーネント間の依存性が、簡単なデータ駆動型パイプライン実行ユースケースによって示されている。（ｃ）は消費者（consumer）、（ｐ）は生産者（producer）を示す。
【０２２５】
図２８に示す実施形態に従うと、スケジューラ（ｐ）が、ＨＵＢへのデータのインジェストを開始することにより、プロセスを開始する（１）。インジェストエンジン（ｃ）は、要求を処理し（２）、データをＨＵＢからデータレイクにインジェストする。インジェストプロセスの終了後、インジェストエンジン（ｐ）は、終了ステータスを伝達する（３）ことにより、パイプライン処理を開始する。スケジューラがデータ駆動型の実行をサポートする場合、実行するパイプラインプロセスを自動的に開始する（３ａ）ことができる。パイプラインエンジン（ｃ）は、データの実行を待っているパイプラインを計算する（４）。パイプラインエンジン（ｐ）は、実行のためにスケジューリングするパイプラインアプリケーションのリストを伝える（５）。スケジューラは、パイプラインのための実行スケジュール要求を取得し（６）、パイプラインの実行を開始する（６ａ）。アプリケーションスケジューラ（たとえばＳｐａｒｋ）は、リソース割当についてリソースマネージャとの調停を行い（７）パイプラインを実行する。アプリケーションスケジューラは、割り当てられたノード内の実行部に、パイプラインを実行のために送る（８）。
【０２２６】
オンプレミスエージェント
ある実施形態に従うと、オンプレミスエージェントは、ローカルデータへのアクセスを容易にし、かつ、限られたやり方で、分散型パイプライン実行を容易にする。オンプレミスエージェントは、たとえばクラウドＤＩサービスと通信するようにプロビジョニングおよ

び構成され、データアクセスおよび遠隔パイプライン実行要求を処理する。
【０２２７】
図２９は、ある実施形態に係る、システムで使用するオンプレミスエージェントを示す図である。
【０２２８】
図２９に示すように、ある実施形態に従うと、クラウドエージェントアダプタ６８２は、オンプレミスエージェント６８０をプロビジョニングし（１）、通信のためにエージェントアダプタエンドポイントを構成する。
【０２２９】
インジェストサービスは、メッセージングシステムを通して、ＨＵＢ１に対するローカルデータアクセス要求を開始する（２）。クラウドエージェントアダプタは、インジェストサービスを通して開始された要求へのアクセスを提供することにより、オンプレミスエージェントとメッセージングシステムとの間の仲介として動作し（３）、オンプレミスエージェントからのデータをデータレイクに書き込み、メッセージングシステムを通してタスクの終了を通知する。
【０２３０】
プレミスエージェントは、データアクセス要求の処理のためにクラウドエージェントアダプタをポーリングする（４）またはデータをクラウドにアップロードする。クラウドエージェントアダプタは、データをデータレイクに書き込み（５）、メッセージングシステムを通してパイプラインに通知する。
【０２３１】
ＤＦＭＬフロープロセス
図３０は、ある実施形態に係るデータフロープロセスを示す図である。
【０２３２】
図３０に示すように、ある実施形態に従うと、インジェストステップ６９２において、データは、さまざまなソースたとえばＳＦＤＣ、Ｓ３、またはＤＢａａＳからインジェストされる。
【０２３３】
データ準備ステップ６９３において、インジェストされたデータは、たとえば重複排除、標準化、またはエンリッチ化によって準備される。
【０２３４】
変換ステップ６９４において、システムは、マージ、フィルタ、またはデータセットのルックアップを実行することにより、データを変換する。
【０２３５】
モデルステップ６９５において、１つ以上のモデルが、当該モデルに対するマッピングとともに生成される。
【０２３６】
パブリッシュステップ６９６において、システムは、モデルをパブリッシュし、ポリシーおよびスケジュールを指定し、ターゲットデータ構造をポピュレートすることができる。
【０２３７】
メタデータおよびデータ駆動型自動マッピング
ある実施形態に従うと、このシステムは、１つ以上のデータソースまたはターゲット（本明細書においていくつかの実施形態ではＨＵＢと呼ぶ）間における、複合データ構造、データセット、またはエンティティの自動マッピングのサポートを提供することができる。自動マッピングは、メタデータ、スキーマ、およびデータセットの統計プロファイリングによって駆動することができる。自動マッピングを使用することにより、入力ＨＵＢに対応付けられたソースデータセットまたはエンティティを、ターゲットデータセットまたはエンティティに、またはその逆にマッピングすることにより、１つ以上の出力ＨＵＢで使用されるフォーマットまたは組織（プロジェクション）で準備された出力データを生成することができる。

【０２３８】
たとえば、ある実施形態に従うと、ユーザは、データフロー、パイプライン、またはLambdaアプリケーションを実装（たとえば構築）するために、入力ＨＵＢ内のソースまたは入力データセットまたはエンティティから、出力ＨＵＢ内のターゲットまたは出力データセットまたはエンティティにマッピングされるデータを選択することを所望する場合がある。
【０２３９】
ある実施形態に従うと、ＨＵＢおよびデータセットまたはエンティティの、非常に大きな組について、入力ＨＵＢから出力ＨＵＢへのデータのマップを手作業で作成することは、極めて長い時間を要する非効率的なタスクとなり得るので、自動マッピングにより、データのマッピングのためのレコメンデーションをユーザに提供することによって、データフローアプリケーションたとえばパイプライン、Lambdaアプリケーションの単純化にユーザが集中できるようにする。
【０２４０】
ある実施形態に従うと、データＡＩサブシステムは、グラフィカルユーザインターフェイス（たとえばLambda Studio統合開発環境（Lambda Studio Integrated Development Environment）（ＩＤＥ））を介して、自動マップサービスを求める自動マップ要求を受けることができる。
【０２４１】
ある実施形態に従うと、この要求は、自動マップサービスの実行対象であるアプリケーションに対して指定されたファイルを、入力ＨＵＢを特定する情報、データセットまたはエンティティ、および１つ以上の属性とともに、含み得る。アプリケーションファイルは、アプリケーションのためのデータに関する情報を含み得る。データＡＩサブシステムは、アプリケーションファイルを処理することにより、属性名およびデータ型を含む、エンティティ名とエンティティの他の形状特性とを抽出することができる。自動マップサービスは、これを、マッピングのための可能な候補セットを発見するためのサーチにおいて使用することができる。
【０２４２】
ある実施形態に従うと、本システムは、たとえばデータウェアハウス等のＨＵＢへの変換のために、データにアクセスすることができる。アクセスされたデータは、半構造化および構造化データを含むさまざまな種類のデータを含み得る。データＡＩサブシステムは、アクセスされたデータに対し、データの１つ以上の形状、特徴、または構造を判別することを含む、メタデータ解析を実行することができる。たとえば、メタデータ解析により、データ型（たとえばビジネス型および関数型）、およびデータのカラム形状を判別することができる。
【０２４３】
ある実施形態に従うと、データのメタデータ解析に基づいて、データの１つ以上のサンプルを特定することができ、サンプリングされたデータに機械学習プロセスを適用することにより、アクセスされたデータ内のデータのカテゴリを判別し、モデルをアップデートすることができる。データのカテゴリは、たとえば、データ内のファクトテーブル等のデータの関連部分を示し得る。
【０２４４】
ある実施形態に従うと、機械学習は、たとえばロジスティック回帰モデルまたは機械学習のために実装できる他の種類の機械学習モデルを用いて、実装することができる。ある実施形態に従うと、データＡＩサブシステムは、データ内の１つ以上のデータアイテムの関係を、データのカテゴリに基づいて解析することができる。この関係は、データのカテゴリについて、データ内の１つ以上のフィールド示す。
【０２４５】
ある実施形態に従うと、データＡＩサブシステムは、特徴抽出のためのプロセスを実行することができ、これは、アクセスされたデータの属性についてランダムにサンプリングさ

れたデータの統計プロファイル、データ型、および１つ以上のメタデータを判別することを含む。

【０２４６】

たとえば、ある実施形態に従うと、データＡＩサブシステムは、アクセスされたデータのプロファイルを、そのデータのカテゴリに基づいて生成することができる。このプロファイルは、データを出力ＨＵＢに変換するために生成することができ、たとえばグラフィカルユーザインターフェイスに表示できる。

【０２４７】

ある実施形態に従うと、このようなプロファイルを作成した結果、このモデルは、入力データセットまたはエンティティに対する候補データセットまたはエンティティの類似度に関するある程度の信頼度で、レコメンデーションをサポートすることができる。レコメンデーションは、フィルタリングしソートした後に、グラフィカルユーザインターフェイスを介してユーザに提供することができる。

【０２４８】

ある実施形態に従うと、自動マップサービスは、ユーザがデータフローアプリケーションたとえばパイプライン、Lambdaアプリケーションを構築しているステージに基づいて、レコメンデーションを動的に示唆することができる。

【０２４９】

エンティティレベルにおけるレコメンデーションの一例は、ある実施形態に従うと、属性のレコメンデーション、たとえば別の属性または別のエンティティに自動的にマッピングされるエンティティのカラムを、含み得る。サービスは、ユーザの過去の活動に基づいて、継続的にレコメンデーションを提供しユーザを案内することができる。

【０２５０】

ある実施形態に従うと、レコメンデーションは、たとえば入力ＨＵＢに対応付けられたソースデータセットまたはエンティティから、出力ＨＵＢに対応付けられたターゲットデータセットまたはエンティティに、自動マップサービスから提供されたアプリケーションプログラミングインターフェイス（ＡＰＩ）（たとえばＲＥＳＴ　ＡＰＩ）を用いて、マッピングすることができる。レコメンデーションは、たとえば属性、データ型、および表現等のデータのプロジェクションを示すことができ、上記表現は、データ型に関する属性のマッピングであってもよい。

【０２５１】

ある実施形態に従うと、本システムは、レコメンデーションに基づいてアクセスされたデータの変換のための出力ＨＵＢを選択するために、グラフィカルユーザインターフェイスを提供することができる。たとえば、グラフィカルユーザインターフェイスにより、ユーザは、データを出力ＨＵＢに変換するためのレコメンデーションを選択することができる。

【０２５２】

自動マッピング

ある実施形態に従うと、自動マッピング機能は数学的に定義することができ、エンティティ集合Ｅは次のように定義される。

【０２５３】

【数１２】

$$\mathrm{E} = \{e_1, e_2, \ldots, e_n \mid \forall\, e_i \in S\}$$

$$Shape{:}\, S = \{MetaData \times DataType \times StatisticalProfile\}$$

【０２５４】

式中、形状（shape）集合Ｓは、メタデータ（metadata）、データ型（data type）および統計プロファイリングディメンション（statistical profiling dimension）を含む。

目的は、ｅｉとｅｊとの間の類似度の確率が最大になるｊを見出すことである。
【０２５５】
【数１３】

$$e_j^* = \arg\max_j p\{Sim(e_i, e_j) \mid entity = e_i\}$$

【０２５６】
データセットまたはエンティティレベルにおいて、問題はバイナリ問題である。すなわち、データセットまたはエンティティが類似しているか類似していないかである。ｆｓ、ｆｔ、ｈ（ｆｓ，ｆｔ）がソース、ターゲットの特徴、およびソースとターゲットとの間のインタラクティブ特徴のセットを表すとする。したがって、目的は類似度の確率を推定することである。
【０２５７】
【数１４】

$$p = g(f_s, f_t, h(f_s, f_t); \boldsymbol{\beta})$$

$$g(.):\ [0,1]^Q \rightarrow [0,1]$$

【０２５８】
対数尤度関数は次のように定義される
【０２５９】
【数１５】

$$\ell(\boldsymbol{\beta}) = \sum_{q=1}^{Q} c_q \log p_q(\boldsymbol{\beta}) + (1 - c_q) \log(1 - p_q(\boldsymbol{\beta}))$$

【０２６０】
したがって、ロジスティック回帰モデルにおいて、未知の係数は次のように推定できる。
【０２６１】
【数１６】

$$g(x; \boldsymbol{\beta}) = \frac{1}{1 + e^{-\boldsymbol{\beta}^T x}}$$

$$\boldsymbol{\beta}^* = \arg\max_{\beta} \ell(\boldsymbol{\beta})$$

【０２６２】
ある実施形態に従うと、自動マップサービスは、たとえば、システムファサードサービスからＨＴＴＰ　ＰＯＳＴ要求を受けたことにより、トリガすることができる。システムファサードＡＰＩは、データフローアプリケーションたとえばパイプライン、LambdaアプリケーションＪＳＯＮファイルを、ＵＩから自動マップＲＥＳＴ　ＡＰＩに送り、パーサモジュールは、アプリケーションＪＳＯＮファイルを処理し、属性名およびデータ型を含むデータセットまたはエンティティのエンティティ名および形状を抽出する。

【０２６３】
ある実施形態に従うと、自動マップサービスは、サーチを利用することにより、マッピングのための有力候補セットを素早く発見する。候補セットは関連性が高いセットである必要があるので、特別なインデックスおよびクエリを用いてこれを実現することができる。この特別なインデックスは、エンティティのすべての属性が格納されすべてがＮ－ｇｒａｍのコンビネーションでトークン化されている特別なサーチフィールドを取り入れる。クエリ時に、サーチクエリビルダモジュールは、たとえばレーベンシュタイン距離に基づいてファジーサーチ特徴を活用することにより、所定のエンティティのエンティティ名と属性名との両方を用いて特別なクエリを構成し、サーチブースト機能を活用することにより、結果を、ストリング類似度という点における関連性によってソートする。
【０２６４】
ある実施形態に従うと、レコメンデーションエンジンは、複数の関連結果、たとえば多くの場合は上位Ｎの結果の選択を、ユーザに対して示す。
【０２６５】
ある実施形態に従うと、高い精度を得るために、機械学習モデルは、ソースとターゲットを比較し、抽出された特徴に基づいてエンティティの類似度のスコアを求める。特徴抽出は、各属性についてランダムにサンプリングされたデータの統計プロファイル、データ型、およびメタデータを含む。
【０２６６】
ある実施形態に従うと、本明細書の記載は概してロジスティック回帰モデルを用いてOracle Business Intelligence（ＯＢＩ）系統マッピングデータから得た自動マッピングの例を学習することを説明しているが、その他のスーパーバイズされる機械学習モデルを代わりに使用することができる。
【０２６７】
ある実施形態に従うと、ロジスティック回帰モデルの出力は、統計的な意味において、候補データセットまたはエンティティの、入力データセットまたはエンティティに対する類似の程度の総信頼度を表す。的確なマッピングを発見するために、その他１つ以上のモデルを用いて、類似する特徴を使用してソース属性とターゲット属性との類似度を計算することができる。
【０２６８】
最後に、ある実施形態に従うと、レコメンデーションがフィルタリングおよびソートされてシステムファサードに送り返されユーザインターフェイスに送られる。自動マップサービスは、データフローアプリケーションたとえばパイプラインまたはLambdaアプリケーション設計中に、ユーザがどのステージにいるかに基づいて、レコメンデーションを動的に提示する。サービスは、ユーザの過去の活動に基づいて、継続的にレコメンデーションを提供しユーザをガイドすることができる。自動マッピングは、フォワード・エンジニアリング方向またはリバースエンジニアリング方向いずれかで実施することができる。
【０２６９】
図３１は、ある実施形態に係る、データ型の自動マッピングを示す図である。
図３１に示すように、ある実施形態に従うと、システムファサード７０１および自動マップＡＰＩ７０２により、データフローアプリケーションたとえばパイプラインまたはLambdaアプリケーションを、ソフトウェア開発コンポーネントたとえばLambda Studioから受けることができる。パーサ７０４は、アプリケーションのＪＳＯＮファイルを処理し、属性名およびデータ型を含むエンティティ名および形状を抽出する。
【０２７０】
ある実施形態に従うと、サーチインデックス７０８を用いてプライマリサーチ７１０をサポートすることにより、マッピングのためのデータセットまたはエンティティの有力候補セットを発見する。サーチクエリビルダモジュール７０６は、所定のエンティティのエンティティ名および属性名の両方を用いてクエリを構成することにより、選択データセットまたはエンティティ７１２を求める。

【０２７１】
ある実施形態に従うと、機械学習（ＭＬ）モデルを用いてソースとターゲットの対を比較し、抽出された特徴に基づいて、データセットまたはエンティティの類似度のスコアを求める。特徴抽出７１４は、各属性についてランダムにサンプリングされたデータの統計プロファイル、データ型、およびメタデータを含む。
【０２７２】
ある実施形態に従うと、ロジスティック回帰モデル７１６は、出力として、入力エンティティに対する候補エンティティの類似の程度の総信頼度を提供する。より正確なマッピングを発見するために、カラムマッピングモデル７１８を用いてソース属性とターゲット属性との類似度をさらに評価する。
【０２７３】
ある実施形態に従うと、次に、自動マッピング７２０として、レコメンデーションを、ソフトウェア開発コンポーネントたとえばLambda Studioに返すために、ソートする。自動マップサービスは、データフローアプリケーションたとえばパイプラインまたはLambdaアプリケーション設計中に、ユーザがどのステージにいるかに基づいて、レコメンデーションを動的に提示する。サービスは、ユーザの過去の活動に基づいて、継続的にレコメンデーションを提供しユーザをガイドすることができる。
【０２７４】
図３２は、ある実施形態に係る、マッピングの生成のための自動マップサービスを示す図である。
【０２７５】
図３２に示すように、ある実施形態に従うと、自動マップサービスを、マッピングの生成のために提供することができる。これは、ＵＩクエリ７２８を受け、クエリ理解エンジン７２９に送り、その後クエリ分解７３０コンポーネントに送ることを含む。
【０２７６】
ある実施形態に従うと、プライマリサーチ７１０をデータＨＵＢ７２２を用いて実行することにより、後のメタデータおよび統計プロファイル処理７３２で使用する候補データセットまたはエンティティ７３１を求める。
【０２７７】
ある実施形態に従うと、結果は、Get Statsプロファイル７３４コンポーネント、データＡＩシステム７２４、および特徴抽出７３５に送られる。結果は、合成７３６、モデル７２３に従う最終信頼度マージおよびランキング７３９に使用され、レコメンデーションおよび関連信頼度７４０に与えられる。
【０２７８】
自動マップの例
図３３は、ある実施形態に係る、ソーススキーマとターゲットスキーマとの間のマッピングの一例を示す図である。
【０２７９】
図３３に示すように、この例７４１は、ある実施形態に従い、たとえば（ａ）上位語（hypernym）、（ｂ）同義語（synonym）、（ｃ）相等性（equality）、（ｄ）Soundex、および（ｅ）ファジーマッチングに基づく、単純な自動マッピングの例を示す。
【０２８０】
図３４は、ある実施形態に係る、ソーススキーマとターゲットスキーマとの間のマッピングの別の例を示す図である。
【０２８１】
図３４に示すように、専らメタデータに基づくアプローチは、この情報が無関係の場合、失敗する。ある実施形態に従い、図３４は、ソース属性名およびターゲット属性名に全く情報がない場合の例７４２を示す。メタデータ特徴がないとき、本システムは、特徴の統計プロファイリングを含むモデルを用いて類似するエンティティを発見することができる。
【０２８２】

自動マッププロセス
図３５は、ある実施形態に係る、自動化されたデータ型のマッピングを提供するプロセスを示す図である。
【０２８３】
図３５に示すように、ステップ７４４で、ある実施形態に従うと、アクセスされたデータを処理することにより、アクセスされたデータのメタデータ解析を実行する。
【０２８４】
ステップ７４５において、アクセスされたデータの１つ以上のサンプルを特定する。
ステップ７４６において、機械学習プロセスを適用することにより、アクセスされたデータ内のデータのカテゴリを判別する。
【０２８５】
ステップ７４８において、アクセスされたデータの自動マッピングに使用するために、判別したデータのカテゴリに基づいて、アクセスされたデータのプロファイルを生成する。
【０２８６】
動的レコメンデーションおよびシミュレーション
ある実施形態に従うと、このシステムは、（本明細書においていくつかの実施形態ではLambda Studioと呼ぶ）ソフトウェア開発コンポーネントと、上記システムで使用される視覚環境を提供する（本明細書においていくつかの実施形態ではパイプラインエディタまたはLambda Studio IDEと呼ぶ）グラフィカルユーザインターフェイスとを含み得る。これは、入力ＨＵＢからアクセスされたデータに対するセマンティックアクションの実行のために、当該データに対応付けられた意味またはセマンティクスの理解に基づいてリアルタイムレコメンデーションを提供することを含む。
【０２８７】
たとえば、ある実施形態に従うと、グラフィカルユーザインターフェイスは、入力ＨＵＢからアクセスされたデータに対する動作（セマンティックアクションとも呼ぶ）を実行するためにリアルタイムレコメンデーションを提供することができる。これは、部分データおよびこのデータの形状またはその他の特性を含む。セマンティックアクションは、データに対し、このデータに対応付けられた意味またはセマンティクスに基づいて実行することができる。データの意味を用いて、このデータに対して実行できるセマンティックアクションを選択することができる。
【０２８８】
ある実施形態に従うと、セマンティックアクションは、１つ以上のデータセットに対するオペレータを表し得るものであり、システム内で宣言的に規定されたベースセマンティックアクションまたは関数を参照することができる。処理済みの１つ以上のデータセットは、セマンティックアクションを実行することによって生成できる。セマンティックアクションは、特定の関数型またはビジネス型に対応付けられたパラメータによって規定することができる。これらは、処理対象の特定のアップストリームデータセットを表す。グラフィカルユーザインターフェイスをメタデータ駆動型とし、グラフィカルユーザインターフェイスが動的に生成されてレコメンデーションをデータ内で特定されたメタデータに基づいて動的に提供するようにしてもよい。
【０２８９】
図３６は、ある実施形態に係る、アクセスされたデータに対して有効にされた１つ以上のセマンティックアクションを表示するシステムを示す図である。
【０２９０】
図３６に示すように、ある実施形態に従うと、ユーザ入力エリア７５２を有するグラフィカルユーザインターフェイス７５０を用いて、アクセスされたデータに対して有効にされたセマンティックアクションに対するクエリが、システムのナレッジソースに送られる。このクエリは、アクセスされたデータの分類を示す。
【０２９１】
ある実施形態に従うと、クエリに対するレスポンスをナレッジソースから受ける。このレ

スポンスは、アクセスされたデータに対して有効にされ、データの分類に基づいて特定された、１つ以上のセマンティックアクションを示す。

【０２９２】

ある実施形態に従うと、アクセスされたデータに対して有効にされたセマンティックアクションのうち選択されたセマンティックアクションが、選択のためおよびアクセスされたデータとともに使用するために表示される。これは、アクセスされたデータの処理中に、アクセスされたデータに対して有効にされたセマンティックアクション７５６のうち選択されたセマンティックアクションまたはレコメンデーション７５８のリストを、自動的に提供またはアップデートすることを含む。

【０２９３】

ある実施形態に従うと、レコメンデーションは、静的データに基づいて予め計算するのではなく、動的に提供することができる。たとえば、システムは、たとえばユーザプロファイルまたはユーザの経験レベル等の情報を考慮して、リアルタイムでアクセスされるデータに基づき、リアルタイムでレコメンデーションを提供することができる。リアルタイムデータに対してシステムが提供するレコメンデーションは、データフローアプリケーションたとえばパイプライン、Lambdaアプリケーションを生成するために、注目すべき、関連する、正確なものであってもよい。レコメンデーションは、特定のメタデータに対応付けられたデータに関して、ユーザの行動に基づいて提供することができる。システムは、情報に対するセマンティックアクションをレコメンドすることができる。

【０２９４】

たとえば、ある実施形態に従うと、システムは、データをインジェストし、変換し、統合し、任意のシステムに対してパブリッシュすることができる。システムは、エンティティを用いてその数値基準のうちのいくつかを興味深い解析手法で解析することをレコメンドすることができ、このデータをさまざまなディメンションでピボットし、どれが興味深いディメンションであるかを示し、ディメンション階層に関してデータを要約し、より多くのインサイトでデータをエンリッチ化する。

【０２９５】

ある実施形態に従うと、レコメンデーションは、たとえばデータのメタデータ解析のような技術を用いるデータの解析に基づいて、提供することができる。

【０２９６】

ある実施形態に従うと、メタデータ解析は、たとえばデータの形状、特徴、および構造等のデータの分類を判別することを含み得る。メタデータ解析は、データ型（たとえばビジネス型および関数型）を判別することができる。メタデータ解析は、データのカラム形状を示すこともできる。ある実施形態に従うと、データを、メタデータ構造（たとえば形状および特徴）と比較することにより、データ型およびデータに対応付けられた属性を判別することができる。メタデータ構造は、システムのシステムＨＵＢ（たとえばナレッジソース）で規定することができる。

【０２９７】

ある実施形態に従うと、システムは、メタデータ解析を用いて、システムＨＵＢにクエリすることにより、メタデータに基づいてセマンティックアクションを特定することができる。レコメンデーションは、入力ＨＵＢからアクセスされたデータのメタデータの解析に基づいて決定されたセマンティックアクションであってもよい。具体的には、セマンティックアクションは、メタデータにマッピングすることができる。たとえば、セマンティックアクションは、これらのアクションが許可されたおよび／または適用可能なメタデータにマッピングすることができる。セマンティックアクションは、ユーザによって規定されてもよく、および／またはデータの構造に基づいて規定されてもよい。

【０２９８】

ある実施形態に従うと、セマンティックアクションは、メタデータに対応付けられた条件に基づいて規定することができる。システムＨＵＢは、セマンティックアクションが修正、削除、または増補されるように、修正することができる。

【０２９９】
ある実施形態に従うと、セマンティックアクションの例は、キューブの構築、データのフィルタリング、データのグループ分け、データの集約、または、データに対して実行できるその他のアクションを含み得る。メタデータに基づいてセマンティックアクションを規定することにより、データに対して許可されるセマンティックアクションを決定するのにマッピングまたはスキームは不要になる。セマンティックアクションは、新たな異なるメタデータ構造が発見されたときに規定してもよい。よって、システムは、入力として受けたデータについて解析されたメタデータを用いて、セマンティックアクションの特定に基づき、レコメンデーションを動的に求めることができる。
【０３００】
ある実施形態に従うと、セマンティックアクションを第三者が規定し、この第三者がデータを、たとえばメタデータに対応付けられた１つ以上のセマンティックアクションを規定するデータを、供給できるようにすることができる。システムは、システムＨＵＢに動的にクエリすることにより、メタデータに対して利用できるセマンティックアクションを決定することができる。よって、システムＨＵＢを修正し、その時点でこのような修正に基づいて許可されるセマンティックアクションをシステムが決定するようにしてもよい。システムは、動作（たとえばフィルタリング、検出、および登録）を実行することにより、第三者から取得したデータを処理することができる。このとき、データは、セマンティックアクションを規定し、上記処理によって特定されたセマンティックアクションに基づいてセマンティックアクションが利用できるようにすることができる。
【０３０１】
図３７および図３８は、ある実施形態に係る、アクセスされたデータに対して有効にされた１つ以上のセマンティックアクションを表示するグラフィカルユーザインターフェイスを示す図である。
【０３０２】
図３７に示すように、ある実施形態に従うと、ソフトウェア開発コンポーネント（たとえばLambda Studio）は、グラフィカルユーザインターフェイス（たとえばパイプラインエディタまたはLambda Studio IDE）７５０を提供することができる。これは、出力ＨＵＢへのプロジェクションのために入力データを処理するまたは入力データの処理をシミュレートする際に使用するレコメンデーションセマンティックアクションを表示することができる。
【０３０３】
たとえば、ある実施形態に従うと、図３７のインターフェイスによって、ユーザは、データフローアプリケーションたとえばパイプライン、Lambdaアプリケーションに対応付けられたオプション７５２を表示することができる。これは、たとえば入力ＨＵＢ規定７５４を含む。
【０３０４】
ある実施形態に従うと、データフローアプリケーションたとえばパイプライン、Lambdaアプリケーションの作成中、または、データフローアプリケーションたとえばパイプライン、Lambdaアプリケーションの、入力データに対するシミュレーション中に、１つ以上のセマンティックアクション７５６またはその他のレコメンデーション７５８を、ユーザによる検討用にグラフィカルユーザインターフェイスに表示することができる。
【０３０５】
ある実施形態に従うと、シミュレーションモードにおいて、ソフトウェア開発コンポーネント（たとえばLambda Studio）は、サンドボックス（sandbox）環境を提供する。この環境によって、ユーザは、出力に対するさまざまなセマンティックアクションの実行の結果を即時見ることができる。これは、アクセスされたデータの処理中に、アクセスされたデータに適したセマンティックアクションのリストを自動的にアップデートすることを含む。
【０３０６】

たとえば、図３８に示すように、ある実施形態に従うと、何らかの情報を探しているユーザの開始点から、システムは、情報に対する動作７６０をレコメンドすることができる。たとえば、エンティティを用いてその数値基準のうちのいくつかを興味深い解析手法で解析することを推奨し、このデータをさまざまなディメンションでピボットし、どれが興味深いディメンションであるかを示し、ディメンション階層に関してデータを要約し、より多くのインサイトでデータをエンリッチ化する。

【０３０７】

ある実施形態に従うと、示されている例では、ソースおよびディメンションの双方が、システム内の解析可能なエンティティに対して推奨され、多次元キューブを構築するタスクを、概ねポイントおよびクリックのうちの一方にする。

【０３０８】

典型的には、このような作業には多くの経験とドメイン専用知識が必要である。機械学習を用いてデータ特性と一般的な統合パターンに対するユーザの行動パターンとの両方を、セマンティックサーチと機械学習からのレコメンデーションとの組み合わせとともに解析することにより、ビジネス専用アプリケーションの構築のためにアプリケーションを開発するための先端ツーリング（tooling）を可能にする。

【０３０９】

図３９は、ある実施形態に係る、アクセスされたデータに対して有効にされた１つ以上のセマンティックアクションを表示するプロセスを示す図である。

【０３１０】

図３９に示すように、ステップ７７２で、ある実施形態に従い、アクセスされたデータを処理することにより、アクセスされたデータのメタデータ解析を実行する。メタデータ解析は、アクセスされたデータの分類を判別することを含む。

【０３１１】

ステップ７７４で、システムのナレッジソースに、アクセスされたデータに対して有効にされたセマンティックアクションのクエリを送信する。このクエリはアクセスされたデータの分類を示す。

【０３１２】

ステップ７７５で、ナレッジソースから、上記クエリに対するレスポンスを受信する。このレスポンスは、アクセスされたデータに対して有効にされ上記データの分類に基づいて特定された１つ以上のセマンティックアクションを示す。

【０３１３】

ステップ７７６で、アクセスされたデータに対して有効にされたセマンティックアクションの中から選択されたセマンティックアクションを、選択のためかつアクセスされたデータに対して使用するために、グラフィカルユーザインターフェイスに表示する。これは、アクセスされたデータの処理中に、アクセスされたデータに対して有効にされたセマンティックアクションの中から選択されたセマンティックアクションのリストを自動的に提供またはアップデートすることを含む。

【０３１４】

データフローの関数分解

ある実施形態に従うと、このシステムは、ソフトウェアアプリケーションのデータフローの関数分解から特定されたパターンに基づいて、入力データに対するアクションおよび変換をレコメンドするためのサービスを提供することができ、これは、後のアプリケーションにおいて上記データフローに可能な変換を判定することを含む。データフローは、データの変換、述語、およびデータに適用されるビジネスルールを記述するモデルと、データフロー内で使用される属性とに分解することができる。

【０３１５】

図４０は、ある実施形態に係る、パイプライン、Lambdaアプリケーションを評価してその構成要素を求めることにより、パターン検出および帰納的学習を容易にすることをサポートすることを示す。

【０３１６】
図４０に示すように、ある実施形態に従うと、関数分解ロジック８００すなわちソフトウェアコンポーネントは、コンピュータシステムまたはその他の処理デバイスによって実行可能なソフトウェアまたはプログラムコードとして提供することができ、（たとえばパイプラインエディタまたはLambda Studio IDE内の）ディスプレイ８０５に対し、関数分解８０２およびレコメンデーション８０４を提供するために使用することができる。たとえば、このシステムは、データフローアプリケーションたとえばパイプライン、Lambdaアプリケーションのデータフローの関数分解から特定されたパターン／テンプレートに基づいて、データに対するアクションおよび変換をレコメンドするためのサービスを提供することができる、すなわち、データフローの関数分解を通して、後のアプリケーションにおいてデータフローに可能な変換を判定するためのパターンを観察することができる。
【０３１７】
ある実施形態に従うと、このサービスは、データフローを、データの変換、述語、およびデータに適用されるビジネスルールを記述するモデルと、データフローで使用される属性とに分解または分類することができるフレームワークによって実現できる。
【０３１８】
従来、アプリケーションのデータフローは、データに対する一連の変換を表すことができ、データに適用される変換のタイプは、コンテキスト性が高い。大抵のデータ統合フレームワークにおいて、通常、プロセスの系統は、データフローが如何にして永続化され、解析され、生成されるかに関しては、限定的であるまたは存在しない。ある実施形態に従うと、このシステムによって、セマンティックリッチエンティティタイプに基づいてフローまたはグラフからコンテキストが関連するパターンを導出することができ、さらにデータフロー文法およびモデルを学習し、これを用いて、所定の類似するコンテキストが与えられたときに複合データフローグラフを生成することができる。
【０３１９】
ある実施形態に従うと、システムは、データフローの設計仕様に基づいて、パターンおよびテンプレートを規定する１つ以上のデータ構造を生成することができる。データフローを分解して、関数式を規定するデータ構造にすることにより、パターンおよびテンプレートを決定することができる。データフローを用いて、データ変換のレコメンデーションのためのパターンを決定するための関数式を予測し生成することができる。レコメンデーションは、分解されたデータフローおよび固有パターンの帰納的学習によって導出されたモデルに基づいており、粒度を細かくすることができる（たとえば、特定の属性に対するスカラー変換をレコメンド、または、１つ以上の属性をフィルタリングまたは結合のために述語において使用）。
【０３２０】
ある実施形態に従うと、データフローアプリケーションたとえばパイプライン、Lambdaアプリケーションにより、ユーザは、データに対するセマンティックアクションに基づいて複雑なデータ変換を生成することができる。このシステムは、データ変換を、パイプライン、Lambdaアプリケーションのためのデータのフローを規定する１つ以上のデータ構造として、格納することができる。
【０３２１】
ある実施形態に従うと、データフローアプリケーション、たとえばパイプライン、Lambdaアプリケーションのデータフローの分解を利用することにより、データのパターン解析を判断し、関数式を生成することができる。分解は、セマンティックアクションと、変換および述語、またはビジネスルールに対して実行することができる。先のアプリケーションのセマンティックアクション各々は、分解を通じて特定することができる。帰納のプロセスを用いて、ビジネスロジックを、そのコンテキスト要素（ビジネス型および関数型）を含むデータフローから、抽出することができる。
【０３２２】
ある実施形態に従い、プロセスに対してモデルを生成することができ、帰納に基づいて、

コンテキストがリッチな規範データフロー設計レコメンデーションを生成することができる。これらのレコメンデーションは、モデルから推論されたパターンに基づいていてもよく、各レコメンデーションは、アプリケーションのためのデータに対して実行できるセマンティックアクションに対応し得る。

【０３２３】

ある実施形態に従い、システムは、関数分解に基づいてデータ変換のパターンを推論するプロセスを実行することができる。システムは、１つ以上のデータフローアプリケーションたとえばパイプライン、Lambdaアプリケーションのデータフローにアクセスすることができる。データフローを処理することにより、１つ以上の関数式を求めることができる。関数式は、データフロー内で特定されたアクション、述語、またはビジネスルールに基づいて生成することができる。アクション、述語、またはビジネスルールを用いることにより、データフローに対する変換のパターンを特定（たとえば推論）することができる。変換のパターンの推論は、受動プロセスであってもよい。

【０３２４】

ある実施形態に従うと、変換のパターンは、異なるアプリケーションのデータフローの受動解析に基づいて、クラウドソーシング法で決定することができる。このパターンは、機械学習（たとえば深層強化学習）を用いて決定することができる。

【０３２５】

ある実施形態に従うと、変換のパターンは、データフローアプリケーションたとえばパイプライン、Lambdaアプリケーションに対して生成された関数式について特定することができる。１つ以上のデータフローを分解することにより、データ変換のパターンを推論することができる。

【０３２６】

ある実施形態に従うと、システムは、このパターンを用いて、新たなデータフローアプリケーションたとえばパイプライン、Lambdaアプリケーションのデータフローに対する１つ以上のデータ変換をレコメンドすることができる。貨幣交換のためのデータに対する処理のデータフローの例において、このシステムは、データに対する変換のパターンを特定することができる。また、このシステムは、アプリケーションの新たなデータフローに対する１つ以上の変換をレコメンドすることができ、このデータフローは、同様の貨幣交換のためのデータを含む。変換は、新たなデータフローを変換に従って修正することにより同様の貨幣交換を生み出すように、パターンに従って同様のやり方で実行することができる。

【０３２７】

図４１は、ある実施形態に係る、１つ以上のアプリケーション各々ごとに生成された１つ以上の関数式についてデータフロー内の変換のパターンを特定する手段を示す図である。

【０３２８】

先に述べたように、ある実施形態に従うと、パイプラインたとえばLambdaアプリケーションにより、ユーザは、関係演算法におけるオペレータに対応するセマンティックアクションに基づいて、複雑なデータ変換を規定することができる。データ変換は通常、有向非巡回グラフもしくはクエリとして、または、ＤＦＭＬの場合は入れ子関数として、永続化される。データフローアプリケーションたとえばパイプライン、Lambdaアプリケーションを分解し入れ子関数としてシリアライズすることにより、データフローのパターン解析を可能にし、同様のコンテキストにおいてデータセットに対する複雑な変換を抽出する関数式を生成するのに使用することができるデータフローモデルを帰納する。

【０３２９】

ある実施形態に従うと、入れ子関数分解を、セマンティックアクション（行またはデータセットオペレータ）のレベルだけでなく、スカラー変換および述語構造でも実行する。これにより、複合データフローの深い系統機能を可能にする。帰納モデルに基づくレコメンデーションは、高粒度にすることができる（たとえば、特定の属性に対するスカラー変換をレコメンド、またはフィルタリングまたは結合のために述語における１つ以上の属性を

使用）。
【０３３０】
ある実施形態に従うと、関数分解に含まれる要素は概ね以下の通りである。
アプリケーションは、トップレベルデータフロー変換を表す。
【０３３１】
アクションは、１つ以上のデータセット（固有のデータフレーム）に対するオペレータを表す。
【０３３２】
アクションは、システムにおいて宣言的に規定されたベースセマンティックアクションまたは関数を参照する。アクションは、１つ以上のアクションパラメータを有することができ、各アクションパラメータは、特定の役割（イン、アウト、イン／アウト）およびタイプを有することができ、１つ以上の処理済みのデータセットを返すことができ、数レベルの深さに埋め込むまたは入れ子にすることができる。
【０３３３】
アクションパラメータは、アクションによって所有され、特定の関数型またはビジネス型を有し、処理対象の特定のアップストリームデータセットを表す。結合パラメータは、変換に使用されるＨＵＢ内のデータセットまたはエンティティを表す。値パラメータは、現在の変換のコンテキストにおいて処理される中間または一時データ構造を表す。
【０３３４】
スコープリゾルバにより、全体のデータフローにおいて使用されるデータセットまたはデータセット内の要素に対し、プロセス系統を導出することができる。
【０３３５】
図４２は、ある実施形態に係る、１つ以上のアプリケーション各々ごとに生成された１つ以上の関数式についてデータフロー内の変換のパターンを特定する際に使用するオブジェクト図を示す図である。
【０３３６】
図４２に示すように、ある実施形態に従うと、関数分解ロジックを用いて、データフローアプリケーションの、たとえばパイプライン、Lambdaアプリケーションの、データフローを、データの変換、述語、およびデータに適用されるビジネスルールを記述するモデルと、データフロー内で使用される属性とに分解または分類することができる。これは、たとえば、パターンまたはテンプレート８１２（テンプレートがLambdaアプリケーションに対応付けられている場合はパイプライン）、サービス８１４、関数８１６、関数パラメータ８１８、および関数型８２０に分解することができる。
【０３３７】
ある実施形態に従うと、これらの関数コンポーネント各々は、データフローアプリケーションたとえばパイプライン、Lambdaアプリケーションを反映する、たとえばタスク８２２またはアクション８２４に、さらに分解することができる。
【０３３８】
ある実施形態に従うと、スコープリゾルバ８２６を用いて、特定の属性または埋め込まれたオブジェクトに対するリファレンスを、そのスコープを通してリゾルブすることができる。たとえば、図４２に示すように、スコープリゾルバは、属性または埋め込まれたオブジェクトに対するリファレンスを、その隣接スコープを通してリゾルブする。たとえば、フィルタおよび別のテーブルの出力を用いる結合関数は、そのスコープリゾルバへのリファレンスを有し、InScopeOf動作と組合わせて使用することにより、リーフノードそのルートノードにリゾルブすることができる。
【０３３９】
図４３は、ある実施形態に係る、１つ以上のアプリケーション各々ごとに生成された１つ以上の関数式についてデータフロー内の変換のパターンを特定するプロセスを示す図である。
【０３４０】

図４３に示すように、ある実施形態に従うと、ステップ８４２において、１つ以上のソフトウェアアプリケーション各々について、データフローにアクセスする。
【０３４１】
ステップ８４４において、１つ以上のソフトウェアアプリケーションのデータフローを処理することにより、このデータフローを表す１つ以上の関数式を生成する。この１つ以上の関数式は、データフロー内で特定されたセマンティックアクションおよびビジネスルールに基づいて生成される。
【０３４２】
ステップ８４５において、１つ以上のソフトウェアアプリケーション各々について生成した１つ以上の関数式に対し、データフロー内の変換のパターンを特定する。セマンティックアクションおよびビジネスルールを用いて、データフロー内の変換のパターンを特定する。
【０３４３】
ステップ８４７において、データフロー内で特定された変換のパターンを用いて、１つ以上のデータ変換のレコメンデーションを、別のソフトウェアアプリケーションのデータフローに対して提供する。
【０３４４】
オントロジー学習
ある実施形態に従うと、このシステムは、スキーマ定義のオントロジー解析を実行することにより、このスキーマに対応付けられたデータおよびデータセットまたはエンティティのタイプを判別し、エンティティとそれらの属性との関係に基づいて規定されたオントロジーを含むリファレンススキーマからモデルを生成またはアップデートすることができる。１つ以上のスキーマを含むリファレンスＨＵＢを用いてデータフローを解析し、さらに、分類する、または、たとえば入力データの変換、エンリッチ化、フィルタリング、もしくはクロスエンティティデータ融合等の、レコメンデーションを行うことができる。
【０３４５】
ある実施形態に従うと、本システムは、スキーマ定義のオントロジー解析を実行することにより、リファレンススキーマ内のデータおよびエンティティのタイプのオントロジーを決定することができる。言い換えると、このシステムは、エンティティとその属性との間の関係に基づいて規定されたオントロジーを含むスキーマからモデルを生成することができる。リファレンススキーマは、システムによって与えられるものであっても、デフォルトリファレンススキーマであっても、代わりに、ユーザから供給されるもしくは第三者リファレンススキーマであってもよい。
【０３４６】
データ統合フレームワークのうちのいくつかは、エンジニアメタデータを周知のシステムソースタイプからリバースする場合があるが、パターン定義およびエンティティ分類に使用出来る関数型システムを構築するためのメタデータの解析は提供しない。また、メタデータハーベスティングは、範囲が限られており、抽出されたデータセットまたはエンティティのためのデータプロファイリングまで拡張されるものではない。ユーザが、（同様のトポロジー空間における）エンティティ分類に加えて複雑なプロセス（ビジネスロジック）および統合パターンにおいて使用する関数型システムオントロジー学習のためにリファレンススキーマを特定できるようにする機能は、現在利用できない。
【０３４７】
ある実施形態に従うと、１つ以上のスキーマをリファレンスＨＵＢに格納することができる。これ自体は、システムＨＵＢの中にまたはその一部として設けることができる。リファレンススキーマと同様、リファレンスＨＵＢも、ユーザから提供されるもしくは第三者リファレンスＨＵＢであってもよく、または、マルチテナント環境において、特定のテナントに対応付けられたとえばデータフローＡＰＩを通してアクセスされてもよい。
【０３４８】
ある実施形態に従うと、リファレンスＨＵＢを使用してデータフローを解析し、さらに分

類またはレコメンデーションを行うことができる。それはたとえば、変換、エンリッチ化、フィルタリング、またはクロスエンティティ・データフュージョンである。

【０３４９】

たとえば、ある実施形態に従うと、このシステムは、リファレンスＨＵＢをオントロジー解析のためのスキーマとして規定する入力を受けることができる。リファレンスＨＵＢは、インポートされてエンティティ定義（属性定義、データ型、およびデータセットまたはエンティティ、制約、またはビジネスルール間の関係）を取得してもよい。リファレンスＨＵＢ内のサンプルデータ（たとえば、例としてカラムデータ等の属性ベクトル）を、すべてのデータセットまたはエンティティおよびプロファイリングされたデータについて抽出することにより、データのいくつかのメトリクスを導出してもよい。

【０３５０】

ある実施形態に従うと、型システムは、リファレンススキーマの用語集に基づいてインスタンス化することができる。システムは、オントロジー解析を実行することにより、データの型を記述するオントロジー（たとえば一組のルール）を導出することができる。オントロジー解析はデータルールを決定することができる。データルールは、プロファイリングされたデータ（たとえば属性または合成値）メトリクスについて規定されたものであり、ビジネス型要素（たとえばＵＯＭ、ＲＯＩＬ、または通貨の種類）を、そのデータプロファイルとともに記述する。オントロジー解析は、関係ルールおよび複合ルールを決定することができる。関係ルールは、データセットまたはエンティティと属性ベクトル（リファレンススキーマからインポートされた制約またはリファレンス）との対応関係を規定し、複合ルールは、データルールおよび関係ルールの組み合わせから導出できる。そうすると、型システムは、メタデータハーベスティングおよびデータサンプリングを通して得られたルールに基づいて規定することができる。

【０３５１】

ある実施形態に従うと、オントロジー解析を用いてインスタンス化した型システムに基づいて、システムＨＵＢからのパターンおよびテンプレートを利用することができる。そうすると、システムは型システムを用いてデータフロー処理を実行することができる。

【０３５２】

たとえば、ある実施形態に従うと、データセットまたはエンティティの分類および型アノテーションは、登録されたＨＵＢの型システムによって特定することができる。型システムを用いて、リファレンススキーマから導出した関数型およびビジネス型についてルールを規定することができる。型システムを用いて、たとえばブレンド、エンリッチ化、および変換レコメンデーション等のアクションを、型システムに基づいてデータフロー内で特定したエンティティに対して実行することができる。

【０３５３】

図４４は、ある実施形態に係る、関数型ルールを生成するためのシステムを示す図である。

【０３５４】

図４４に示すように、ある実施形態に従うと、コンピュータシステムまたはその他の処理デバイスによって実行可能なソフトウェアまたはプログラムコードとして提供されるルール帰納ロジック８５０またはソフトウェアコンポーネントによって、ルール８５１を関数型システム８５２に対応付けることができる。

【０３５５】

図４４は、ある実施形態に係る、関数型ルールを生成するためのシステムを示す図である。

【０３５６】

図４５に示すように、ある実施形態に従うと、ＨＵＢ１はリファレンスオントロジーとして機能することができる。これは、他の（たとえば新たに登録された）ＨＵＢたとえばＨＵＢ２およびＨＵＢ３から提供されたメタデータスキーマまたはオントロジーの、型タグ付け、比較、分類、そうでなければ評価に使用することができ、データＡＩシステムが使用する適切なルールを作成する。

【０３５７】

図４６は、ある実施形態に係る、関数型ルールの生成に使用するオブジェクト図を示す図である。

【０３５８】

ある実施形態に従うと、たとえば図４６に示すように、ルール帰納ロジックにより、ルールを、一組の関数型８５３（たとえばＨＵＢ、データセットまたはエンティティ、および属性）を有する関数型システムに対応付け、データフローアプリケーションたとえばパイプライン、Lambdaアプリケーションの作成に使用するためにレジストリに格納することができる。これは、各関数型８５４を関数型ルール８５６およびルール８５８に対応付けることができることを含む。各ルールはルールパラメータ８６０に対応付けることができる。

【０３５９】

ある実施形態に従うと、最初にリファレンススキーマが処理され、このスキーマに適した一組のルールを含むオントロジーを作成することができる。

【０３６０】

ある実施形態に従うと、次に新たなＨＵＢまたは新たなスキーマが評価され、そのデータセットまたはエンティティを、既存のオントロジーおよび作成したルールと比較し、新たなＨＵＢ／スキーマおよびそのエンティティの解析、ならびに、システムのさらなる学習に、使用することができる。

【０３６１】

データ統合フレームワークにおけるメタデータハーベスティングはリバースエンジニアリングのエンティティ定義（属性およびそのデータ型、場合によっては関係）に限定される場合があるが、ある実施形態に従うと、本明細書に記載のシステムが提供するアプローチは以下の点が異なる。すなわち、スキーマ定義をリファレンスオントロジーとして使用できるようにし、そこからビジネス型および関数型を導出することができ、また、リファレンススキーマにおけるデータセットまたはエンティティのためのデータプロファイリングメトリクスを導出することができる。そうすると、このリファレンスＨＵＢを用いてその他のＨＵＢ（データソース）内のビジネスエンティティを解析することにより、さらに分類またはレコメンデーション（たとえばブレンドまたはエンリッチ化）を行うことができる。

【０３６２】

ある実施形態に従うと、システムは、リファレンススキーマを用いるオントロジー学習に対して以下の一組のステップを使用する。

【０３６３】

ユーザは、新たに登録されたＨＵＢをリファレンススキーマとして使用するためのオプションを指定する。

【０３６４】

エンティティ定義（たとえば属性定義、データ型、エンティティ間の関係、制約またはビジネスルールがインポートされる）。

【０３６５】

すべてのデータセットまたはエンティティについてサンプルデータが抽出され、データがプロファイリングされてデータに関するいくつかのメトリクスが導出される。

【０３６６】

型システムがリファレンススキーマの用語集に基づいてインスタンス化される（関数型およびビジネス型）。

【０３６７】

ビジネス型を記述する一組のルールが導出される。

データルールは、プロファイリングされたデータメトリクスに関して規定され、ビジネス型要素の性質を記述する（たとえば、ＵＯＭ、ＲＯＩまたは通貨の種類は、そのデータプロファイルとともにビジネス型要素として規定できる）。

【０３６８】

要素（リファレンススキーマからインポートされた制約またはリファレンス）におけるアソシエーションを規定する関係ルールが生成される。
【０３６９】
データおよび関係ルールの組み合わせを通して導出できる複合ルールが生成される。
型システム（関数およびビジネス）は、メタデータハーベスティングおよびデータサンプリングを通して導出されたルールに基づいて規定される。
【０３７０】
そうすると、パターンまたはテンプレートは、リファレンススキーマに基づいてインスタンス化された型を用いて複雑なビジネスロジックを規定することができる。
【０３７１】
そうすると、システムに登録されたＨＵＢをリファレンススキーマのコンテキストで解析することができる。
【０３７２】
新たに登録されたＨＵＢにおいて、データセットまたはエンティティの分類および型アノテーションを、リファレンススキーマから導出した関数型およびビジネス型についてのルールに基づいて、実施することができる。
【０３７３】
型アノテーションに基づいて、データセットまたはエンティティに対し、ブレンド、エンリッチ化、変換レコメンデーションを実行することができる。
【０３７４】
図４７は、ある実施形態に係る、生成された１つ以上のルールに基づいて関数型システムを生成するプロセスを示す図である。
【０３７５】
図４７に示すように、ある実施形態に従い、ステップ８６２において、リファレンスＨＵＢを規定する入力を受信する。
【０３７６】
ステップ８６３において、リファレンスＨＵＢにアクセスすることにより、リファレンスＨＵＢによって与えられる、データセットまたはエンティティに対応付けられた１つ以上のエンティティ定義を取得する。
【０３７７】
ステップ８６４において、リファレンスＨＵＢから１つ以上のデータセットまたはエンティティのサンプルデータを生成する。
【０３７８】
ステップ８６５において、このサンプルデータをプロファイリングすることにより、このサンプルデータに対応付けられた１つ以上のメトリクスを判定する。
【０３７９】
ステップ８６６において、エンティティ定義に基づいて１つ以上のルールを生成する。
ステップ８６７において、生成した１つ以上のルールに基づいて関数型システムを生成する。
【０３８０】
ステップ８６８において、データ入力の処理において使用するために、関数型システムおよびサンプルデータのプロファイルを永続化する。
【０３８１】
他言語関数インターフェイス
ある実施形態に従うと、このシステムは、(本明細書においていくつかの実施形態では他言語関数インターフェイスと呼ぶ）プログラマティックインターフェイスを提供する。このインターフェイスにより、ユーザまたは第三者は、サービス、関数型およびビジネス型、セマンティックアクション、ならびに関数型およびビジネス型に基づくパターンまたは予め定められた複合データフローを、宣言的に規定することにより、システムの機能を拡張することができる。

【０３８２】
先に述べたように、現在のデータ統合システムが提供し得るインターフェイスは限られており、タイプに対するサポートがなく、また、オブジェクト組成およびパターン定義について明確に規定されたインターフェイスがない。このような短所のために、フレームワークを拡張するサービス全体にわたってセマンティックアクションを呼び出すためのクロスサービスレコメンデーションまたは統一アプリケーション設計プラットフォームのような複雑な機能は、現在のところ提供されていない。
【０３８３】
ある実施形態に従うと、多言語関数インターフェイスにより、ユーザは、定義またはその他の情報を（たとえば顧客から他の第三者に）宣言型で提供することによってシステムの機能を拡張することができる。
【０３８４】
ある実施形態に従うと、システムはメタデータ駆動型であり、多言語関数インターフェイスを通して受けた定義を処理することによってメタデータを求め、当該メタデータの分類たとえばデータ型（たとえば関数型およびビジネス型）を判別し、これらのデータ型（関数およびビジネス両方）を既存のメタデータと比較して、型の一致があるか否かを判断することができる。
【０３８５】
ある実施形態に従うと、多言語関数インターフェイスを通して受けたメタデータを、システムＨＵＢに格納することにより、システムがデータフローを処理するためにメタデータにアクセスできるようにしてもよい。たとえば、メタデータにアクセスすることにより、入力として受けたデータセットのタイプに基づいてセマンティックアクションを決定することができる。システムは、このインターフェイスを通して提供されたデータのタイプについて許可されるセマンティックアクションを決定することができる。
【０３８６】
ある実施形態に従うと、共通の宣言型インターフェイスを提供することにより、システムは、ユーザが、サービスネイティブのタイプおよびアクションを、プラットフォームネイティブのタイプおよびアクションにマッピングすることを可能にすることができる。これにより、タイプおよびパターンの発見を通して、統一されたアプリケーション設計体験が可能になる。これはまた、プラットフォームを拡張するさまざまなサービスのコンポーネント、および、各セマンティックアクションのためのネイティブコードの生成を必要とする、純粋に宣言型のデータフロー定義および設計を容易にする。
【０３８７】
ある実施形態に従うと、多言語関数インターフェイスを通して受けたメタデータを自動で処理し、そこに記述されているオブジェクトまたはアーティファクト（たとえばデータ型またはセマンティックアクション）を、システムが処理するデータフローの動作において使用することができる。１つ以上の第三者システムから受けたメタデータ情報を用いて、サービスを規定する、１つ以上の関数型およびビジネス型を表示する、１つ以上のセマンティックアクションを表示する、または、１つ以上のパターン／テンプレートを表示してもよい。
【０３８８】
たとえば、ある実施形態に従うと、データの関数型およびビジネス型といった、アクセスされたデータの分類を、判別することができる。この分類は、情報とともに受けたデータに関する情報に基づいて特定することができる。１つ以上の第三者システムからデータを受けることによって、システムの機能を拡張することにより、第三者から受けた情報（たとえばサービス、セマンティックアクション、またはパターン）に基づいてデータフローのデータ統合を実行することができる。
【０３８９】
ある実施形態に従うと、システムＨＵＢ内のメタデータをアップデートすることにより、データに関して特定される情報を含むようにすることができる。たとえば、サービスおよ

びパターン／テンプレートを、アップデートすることにより、多言語関数インターフェイスを通して受けたメタデータにおいて特定された情報（たとえばセマンティックアクション）に基づいて、実行することができる。このようにして、システムを、データフローの処理を妨害することなく、多言語関数インターフェイスを通して強化することができる。

【０３９０】

ある実施形態に従うと、後続のデータフローは、システムＨＵＢ内のメタデータを、そのアップデート後に用いて処理することができる。メタデータ解析は、データフローアプリケーション、たとえばパイプライン、Lambdaアプリケーションのデータフローに対して実行することができる。次に、システムＨＵＢを用いて、他言語関数インターフェイスを介して提供された定義を考慮して変換のレコメンデーションを決定することができる。変換は、サービスを実行するためにセマンティックアクションを規定するのに使用されるパターン／テンプレートに基づいて決定することができ、セマンティックアクションも同様に他言語関数インターフェイスを介して提供された定義を考慮することができる。

【０３９１】

図４８は、ある実施形態に係る、多言語関数インターフェイスを介して提供された情報に基づくデータフローに関するレコメンデーションの提供において使用するパターンを特定するためのシステムを示す図である。

【０３９２】

図４８に示すように、ある実施形態に従うと、他言語関数インターフェイス９００を介して受けた定義を用いて、システムＨＵＢ内の、サービスレジストリ９０２、関数およびビジネス型レジストリ９０４、またはパターン／テンプレート９０６をアップデートすることができる。

【０３９３】

ある実施形態に従うと、アップデートした情報を、ルールエンジン９０８を含むデータＡＩサブシステムが使用することによって、たとえば、システムＨＵＢ内のタイプがアノテートされたＨＵＢ、データセットもしくはエンティティ、または属性９１０を判別し、これらのデータセットまたはエンティティを、ソフトウェア開発コンポーネント（たとえばLambda Studio）を介したデータフローアプリケーションたとえばパイプライン、Lambdaアプリケーションに関するレコメンデーションの提供において使用するために、レコメンデーションエンジン９１２に提供することができる。

【０３９４】

図４９は、ある実施形態に係る、他言語関数インターフェイスを介して提供された情報に基づくデータフローに関するレコメンデーションの提供において使用するパターンを特定することをさらに示す図である。

【０３９５】

図４９に示すように、ある実施形態に従うと、第三者メタデータ９２０を他言語関数インターフェイスで受けることができる。

【０３９６】

図５０は、ある実施形態に係る、他言語関数インターフェイスを介して提供された情報に基づくデータフローに関するレコメンデーションの提供において使用するパターンを特定することをさらに示す図である。

【０３９７】

図５０に示すように、ある実施形態に従うと、多言語関数インターフェイスで受けた第三者メタデータを用いてシステムの機能を拡張することができる。

【０３９８】

ある実施形態に従うと、このシステムにより、明確に規定されたインターフェイスを通してフレームワークを拡張することが可能になる。このインターフェイスにより、サービス、サービスのネイティブなタイプ、サービスによって実現されるセマンティックアクションを、特にサービスの一部として利用できる予め規定されたアルゴリズムを抽出するタイプ付きパラメータ、パターンまたはテンプレートとともに、登録することができる。

【０３９９】
ある実施形態に従うと、共通の宣言型プログラミングパラダイムを提供することにより、プラガブルサービスアーキテクチャは、サービスネイティブのタイプおよびアクションを、プラットフォームネイティブのタイプおよびアクションにマッピングすることを可能にする。これにより、タイプおよびパターンの発見を通して、統一されたアプリケーションの設計体験が可能になる。これはまた、プラットフォームを拡張するさまざまなサービスのコンポーネント、および、各セマンティックアクションのためのネイティブコードの生成を必要とする、純粋に宣言型のデータフロー定義および設計を容易にする。
【０４００】
ある実施形態に従うと、プラガブルサービスアーキテクチャはまた、プラグインのための、コンパイル、生成、デプロイ、および実行時実行フレームワーク（統一アプリケーション設計サービス）を規定する。レコメンデーションエンジンは、プラグインされたすべてのサービスのセマンティックアクションおよびパターンの機械学習および推論を行うことができ、分散型複合データフロー設計および開発に関するクロスサービスセマンティックアクションレコメンデーションを行うことができる。
【０４０１】
図５１は、ある実施形態に係る、他言語関数インターフェイスを介して提供された情報に基づくデータフローに関するレコメンデーションの提供において使用するパターンを特定するプロセスを示す図である。
【０４０２】
図５１に示すように、ある実施形態に従うと、ステップ９３２において、データの処理において使用するメタデータの１つ以上の定義を他言語関数インターフェイスを介して受信する。
【０４０３】
ステップ９３４において、他言語関数インターフェイスを介して受信したメタデータを処理することにより、受信したメタデータによって規定される、分類、セマンティックアクション、パターンを規定するテンプレート、またはサービスのうちの１つ以上を含む、受信したメタデータに関する情報を特定する。
【０４０４】
ステップ９３６において、他言語関数インターフェイスを介して受信したメタデータをシステムＨＵＢに格納する。システムＨＵＢは、アップデートされることにより、受信したメタデータに関する情報を含み、かつ、システムのサポートされるタイプ、セマンティックアクション、テンプレート、およびサービスを含むシステムの機能を拡張する。
【０４０５】
９３８において、他言語関数インターフェイスを介してシステムＨＵＢにおいてアップデートされた情報に基づいてデータフローに関するレコメンデーションを提供するためのパターンを特定する。
【０４０６】
ポリシーベースのライフサイクル管理
ある実施形態に従うと、このシステムはデータガバナンス機能を提供することができる。これはたとえば、特定のスナップショットに時間的に関連するデータのスライスごとの、履歴情報（特定のデータはどこから来たデータか）、系統（このデータはどのようにして取得／処理されたか）、セキュリティ（誰がこのデータの責任者だったか）、分類（このデータは何に関連するデータか）、影響力（このデータがビジネスにどれほどの影響があるか）、保持時間（このデータはどれだけの時間存続すべきか）、および有効性（このデータは解析／処理のために除外される／含まれるべきか否か）である。これらは、ライフサイクルの決定およびデータフローのレコメンデーションにおいて使用することができる。
【０４０７】
データライフサイクルの管理についての現在のアプローチは、一時パーティションに全体にわたるデータ特性の変化に基づくガバナンス関連機能またはデータの進化（データプロ

ファイルまたはドリフトの変化）のトラッキングを含まない。システムが観察したまたは導出したデータ特性（分類、変化の頻度、変化のタイプ、またはプロセスにおける用途）は、ライフサイクルの決定またはデータについてのレコメンデーション（保持時間、セキュリティ、有効性、取得間隔）には使用されない。

【０４０８】

ある実施形態に従うと、システムは、系統トラッキングに基づいてデータフローのライフサイクルを表示できるグラフィカルユーザインターフェイスを提供することができる。ライフサイクルは、どこでデータが処理されたか、および、そのデータの処理中にエラーが生じたか否かを示すことができ、データのタイムライン図（たとえば、データセットの数、データセットのボリューム、およびデータセットの使用）として示すことができる。上記インターフェイスは、データのある時点のスナップショットを提供することができ、かつ、データの処理中にその視覚的インジケータを提供することができる。よって、このインターフェイスにより、データの完全な監査、または、ライフサイクルに基づくデータのシステムスナップショット（たとえばパフォーマンスメトリクス、またはリソース使用法）が可能である。

【０４０９】

ある実施形態に従うと、システムは、（インジェストされたデータから周期的にサンプリングされた）サンプルデータおよびユーザ規定アプリケーションによる処理のために取得したデータに基づいて、データのライフサイクルを求めることができる。データライフサイクル管理のいくつかの側面は、インジェストされたデータすなわちストリーミングデータおよびバッチデータ（リファレンスおよびインクリメンタル）のカテゴリ全体にわたって同様である。インクリメンタルデータの場合、システムは、スケジュールされたログ収集イベント駆動型方法を用いてデータの一時スライスを取得し以下の機能をカバーするアプリケーションインスタンス全体にわたるスライスの割当を管理することができる。

【０４１０】

ある実施形態に従うと、システムは、データ損失の場合、システムＨＵＢ内で管理されるメタデータから層全体にわたる系統を用いてデータを再構成することができる。

【０４１１】

たとえば、ある実施形態に従うと、インクリメンタルデータ属性カラム、または、ユーザ構成設定を、特定することにより、インクリメンタルデータを取得することができ、データインジェスト全体にわたり高および低ウォーターマークを維持する。クエリまたはＡＰＩおよび対応するパラメータ（タイムスタンプまたはＩＤカラム）を、インジェストされたデータに対応付けることができる。

【０４１２】

ある実施形態に従うと、システムは、層全体にわたって系統情報を管理することができる。それはたとえば、エッジレイヤ内のクエリまたはログメタデータ、スケーラブルＩ／Ｏレイヤ内のインジェストごとのトピック／パーティションオフセット、データレイク内のスライス（ファイルパーティション）、このデータを用いた後続のダウンストリーム処理済データセットのプロセス系統（データおよびそれに対応付けられたパラメータを生成するアプリケーションの特定の実行インスタンス）のリファレンス、ターゲットエンドポイントに対してパブリッシュされることが「マークされた」データセットおよびデータレイク内のそれに対応するデータスライスに関するトピック／パーティション、ならびに、ジョブ実行インスタンスのパブリッシュおよび処理されてターゲットエンドポイントに対してパブリッシュされるパーティション内のオフセット、である。

【０４１３】

ある実施形態に従うと、レイヤ（たとえばエッジ、スケーラブルＩ／Ｏ、データレイク、またはパブリッシュ）の障害の場合、データはアップストリームレイヤから再構成するかまたはソースから取得することができる。

【０４１４】

ある実施形態に従うと、システムは、その他のライフサイクル管理機能を果たすことがで

きる。
【０４１５】
たとえば、ある実施形態に従うと、セキュリティは、データスライスのこれらのレイヤ各々で強化および監査される。データスライスは、処理またはアクセスの対象から除外する、または（既に除外されていた場合は）その対象にすることができる。これにより、スプリアスなまたは破損したデータスライスは処理されないようにすることができる。保持ポリシーは、スライディングウィンドウを通してデータのスライスに対して強化することができる。データのスライスについて影響力を解析する（たとえば、所定のウィンドウのスライスをタグ付けする能力を、四半期ごとの報告のために構築されたデータマートのコンテキストにおいて影響力があると解析）。
【０４１６】
ある実施形態に従うと、データを、システム内で規定された関数型またはビジネス型をタグ付けすることによって分類する（たとえば、データセットを、ビジネス型（たとえばオーダー、顧客、製品、または時間）とともに関数型でタグ付けする（キューブまたはディメンションまたは階層データとして））。
【０４１７】
ある実施形態に従うと、システムは、１つ以上のＨＵＢからデータにアクセスすることを含む方法を実行することができる。このデータはサンプリングすることができ、システムは、データの一時スライスを決定し、スライスを管理する。これは、システムのシステムＨＵＢにアクセスし、サンプリングされたデータに関するメタデータを取得することを含む。サンプリングされたデータは、システム内の１つ以上の層全体にわたる系統トラッキングのために管理することができる。
【０４１８】
ある実施形態に従うと、サンプルデータに関するインクリメンタルデータおよびパラメータは、インジェストされたデータに対して管理することができる。データは、サンプルデータに対応付けられたデータの型のタグ付けによって分類できる。
【０４１９】
図５２は、ある実施形態に係る、１つ以上の層全体にわたる系統トラッキングのための、サンプリングされたデータまたはアクセスされたデータの管理を示す図である。
【０４２０】
たとえば、図５２に示すように、ある実施形態に従うと、このシステムを用いて、ＨＵＢ９５２、この例ではＯｒａｃｌｅデータベース、および、ＨＵＢ９５４、この例ではＳ３またはその他の環境から、データを受けることがきる。エッジレイヤにおいて入力ＨＵＢから受けたデータは、データフローアプリケーションたとえばパイプライン、Lambdaアプリケーションによる使用のために、１つ以上のトピックスとして、スケーラブルＩ／Ｏレイヤに与えることができる（トピックス各々は分散パーティションとして提供することができる）。
【０４２１】
ある実施形態に従うと、典型的にはパーティションへのオフセットによって表される、インジェストされたデータは、コンピュートレイヤによって正規化９６４することができ、システムの層にまたがる１つ以上の一時スライスとしてデータレイクに書き込むことができる。
【０４２２】
ある実施形態に従うと、このデータはその後、データフローアプリケーションたとえばパイプライン、Lambdaアプリケーション９６６、９６８によって使用され、最終的には１つ以上の追加のトピックス９６０、９６２に対してパブリッシュ９７０され、その後、この例ではＤＢＣＳ環境等の１つ以上の出力ＨＵＢでターゲットエンドポイント（たとえばテーブル）に対してパブリッシュされることができる。
【０４２３】
図５２に示すように、ある実施形態に従うと、最初、データ再構成および系統トラッキン

グ情報は、たとえば、履歴情報（provenance）（Hub 1、S3）、系統（lineage）（Source Entity in Hub 1）、セキュリティ（security）（Connection Credential used）、またはデータのインジェストに関するその他の情報を、含み得る。

【０４２４】

図５３は、ある実施形態に係る、１つ以上の層全体にわたる系統トラッキングのための、サンプリングされたデータまたはアクセスされたデータの管理をさらに示す図である。図５３に示すように、その後、データ再構成および系統トラッキング情報をアップデートすることにより、たとえば、アップデートされた履歴情報（→Ｔ１）、系統（→Ｔ１（インジェストプロセス（Ingest Process））、またはその他の情報等の情報を含むようにすることができる。

【０４２５】

図５４は、ある実施形態に係る、１つ以上の層全体にわたる系統トラッキングのための、サンプリングされたデータまたはアクセスされたデータの管理をさらに示す図である。図５４に示すように、その後、データ再構成および系統トラッキング情報をさらにアップデートすることにより、たとえば、アップデートされた履歴情報（→Ｅ１）、系統（→Ｅ１（正規化（Normalize））、またはその他の情報等の情報を含むようにすることができる。

【０４２６】

ある実施形態に従うと、１つ以上のデータフローアプリケーションたとえばパイプライン、Lambdaアプリケーションによって使用される、一時スライス９７２は、システムの層にまたがるように作成することができる。障害、たとえば、データレイクへの書込において障害が発生した場合、システムは、未処理の１つ以上のデータスライスを判別し、そのデータスライスの処理を、全体としてまたはインクリメンタルに、完了することができる。

【０４２７】

図５５は、ある実施形態に係る、１つ以上の層全体にわたる系統トラッキングのための、サンプリングされたデータまたはアクセスされたデータの管理をさらに示す図である。図５５に示すように、その後、データ再構成および系統トラッキング情報をさらにアップデートし追加の一時スライスを作成することにより、たとえば、アップデートされた履歴情報（→Ｅ１１（App1））、セキュリティ（Role Executing App 1）、またはその他の情報等の情報を含むようにすることができる。

【０４２８】

図５６は、ある実施形態に係る、１つ以上の層全体にわたる系統トラッキングのための、サンプリングされたデータまたはアクセスされたデータの管理をさらに示す図である。図５６に示すように、その後、データ再構成および系統トラッキング情報をさらにアップデートし追加の一時スライスを作成することにより、たとえば、アップデートされた系統（→Ｅ１２（App2））、セキュリティ（Role Executing App 2）、またはその他の情報等の情報を含むようにすることができる。

【０４２９】

図５７は、ある実施形態に係る、１つ以上の層全体にわたる系統トラッキングのための、サンプリングされたデータまたはアクセスされたデータの管理をさらに示す図である。図５７に示すように、その後、データ再構成および系統トラッキング情報をさらにアップデートし追加の一時スライスを作成することにより、たとえば、アップデートされた系統（→Ｔ２（Publish））、セキュリティ（Role Executing Publish to I/O Layer）、またはその他の情報等の情報を含むようにすることができる。

【０４３０】

図５８は、ある実施形態に係る、１つ以上の層全体にわたる系統トラッキングのための、サンプリングされたデータまたはアクセスされたデータの管理をさらに示す図である。図５８に示すように、その後、データ再構成および系統トラッキング情報をさらにアップデートすることにより、ターゲットエンドポイント９７６へのデータの出力を反映することができる。

【０４３１】
データライフサイクル管理
ある実施形態に従うと、上記系統トラッキングに基づくデータライフサイクル管理は、いくつかの機能エリアに向けられている。これらのエリアのうちのいくつかは、ユーザによって構成されることができ（アクセスコントロール、保持時間、有効性）、いくつかは導出され（履歴情報、系統）、それ以外は機械学習アルゴリズムを使用する（分類、影響力）。たとえば、データ管理は、サンプルデータ（インジェストされたデータから定期的にサンプリング）にも、ユーザ規定アプリケーションによる処理のために取得されたデータにも、適用される。データライフサイクル管理のいくつかの側面は、インジェストされたデータすなわちストリーミングデータおよびバッチデータ（リファレンスおよびインクリメンタル）のカテゴリ全体にわたって同様である。インクリメンタルデータの場合、ＤＦＭＬは、スケジュールされたログ収集イベント駆動型方法を用いてデータの一時スライスを取得し以下の機能をカバーするアプリケーションインスタンス全体にわたるスライスの割当を管理する。
【０４３２】
データ損失の場合、システムＨＵＢにおいて管理されるメタデータから層全体にわたる系統を用いてデータを再構成。
【０４３３】
インクリメンタルデータ属性カラムまたはユーザ構成設定を特定することにより、インクリメンタルデータを取得し、インジェスト全体にわたり高および低ウォーターマークを維持。
【０４３４】
インジェストごとに、クエリまたはＡＰＩおよび対応するパラメータ（タイムスタンプまたはＩＤカラム）を対応付ける。
【０４３５】
層全体にわたる系統情報の管理。エッジレイヤ内のクエリまたはログメタデータ。スケーラブルＩ／Ｏレイヤ内のインジェストごとのトピック／パーティションオフセット。データレイク内のスライス（ファイルパーティション）。このデータを用いた後続のすべてのダウンストリーム処理済データセットのプロセス系統（データおよびそれに対応付けられたパラメータを生成するアプリケーションの特定の実行インスタンス）のリファレンス。ターゲットエンドポイントに対してパブリッシュされることが「マークされた」データセットのトピック／パーティションオフセットおよびデータレイク内の対応するデータスライス。ジョブ実行インスタンスのパブリッシュおよび処理されてターゲットエンドポイントに対してパブリッシュされるパーティション内のオフセット。
【０４３６】
レイヤの障害の場合、データはアップストリームレイヤから再構成するかまたはソースから取得することができる。セキュリティは、データスライスのこれらのレイヤ各々で強化および監査される。データスライスは、処理またはアクセスの対象から除外する、または（すでに除外されていた場合は）その対象にすることができる。これにより、スプリアスなまたは破損したデータスライスは処理されないようにすることができる。保持時間ポリシーは、スライディングウィンドウを通してデータのスライスに対して強化することができる。データのスライスについて影響力を解析する（たとえば、所定のウィンドウについてスライスをタグ付けする能力を、四半期ごとの報告のために構築されたデータマートのコンテキストにおいて影響力があると解析）。
【０４３７】
システム内で規定された関数型またはビジネス型のタグ付けによってデータを分類（たとえば、データセットを、ビジネス型（たとえばオーダー、顧客、製品、または時間）とともに、関数型（キューブまたは次元または階層データ）でタグ付け）。
【０４３８】
図５９は、ある実施形態に係る、１つ以上の層全体にわたる系統トラッキングのための、

サンプリングされたデータまたはアクセスされたデータの管理のプロセスを示す図である。
【０４３９】
図５９に示すように、ある実施形態に従うと、ステップ９８２において、１つ以上のＨＵＢからデータにアクセスする。
【０４４０】
ステップ９８３において、アクセスしたデータをサンプリングする。
ステップ９８４において、サンプリングされたデータおよびアクセスされたデータについて、一時スライスを特定する。
【０４４１】
ステップ９８５において、システムＨＵＢにアクセスすることにより、一時スライスによって表されるサンプリングされたデータまたはアクセスされたデータに関するメタデータを取得する。
【０４４２】
ステップ９８６において、一時スライスによって表されるサンプリングされたデータまたはアクセスされたデータについて分類情報を判断する。
【０４４３】
ステップ９８７において、一時スライスによって表されるサンプリングされたデータまたはアクセスされたデータを、システム内の１つ以上の層全体にわたる系統トラッキングのために管理する。
【０４４４】
本発明の実施形態は、汎用または専用デジタルコンピュータ、コンピューティングデバイス、マシン、またはマイクロプロセッサを用いて実現可能である。これは、本開示の教示に従ってプログラムされた、１つ以上のプロセッサ、メモリおよび／またはコンピュータ読取可能な記憶媒体を含む。熟練プログラマーは本開示の教示に基づいて適切なソフトウェアコーディングを簡単に作成でき、それはソフトウェア技術の当業者には明らかであろう。
【０４４５】
いくつかの実施形態において、本発明は、コンピュータをプログラムすることにより本発明のプロセスのうちのいずれかを実行するために使用できる命令が格納された非一時的なコンピュータ読取可能媒体（複数の媒体）であるコンピュータプログラムプロダクトを含む。記憶媒体の例は、フロッピー（登録商標）ディスク、光ディスク、ＤＶＤ、ＣＤ－ＲＯＭ、マイクロドライブ、および光磁気ディスク、ＲＯＭ、ＲＡＭ、ＥＰＲＯＭ、ＥＥＰＲＯＭ、ＤＲＡＭ、ＶＲＡＭ、フラッシュメモリデバイス、磁気または光カード、名のシステム（分子メモリＩＣを含む）、または、命令および／またはデータの非一時的な格納に適したその他のタイプの記憶媒体または装置を含み得るが、これらに限定される訳ではない。
【０４４６】
これまでの本発明の記載は例示および説明を目的として提供されている。すべてを網羅するまたは本発明を開示されている形態そのものに限定することは意図されていない。数多くの変更および変形が当業者には明らかであろう。
【０４４７】
たとえば、上記実施形態のうちのいくつかは、たとえばWolfram、Yago、Chronos、およびSpark等の製品を使用することによりさまざまな計算を実行すること、および、たとえばＢＤＰ、ＳＦＤＣおよびＳ３等のデータソースを使用することによりデータのソースまたはターゲットとして機能することを示しているが、本明細書に記載の実施形態は、同様のタイプの機能を提供するその他のタイプの製品およびデータソースとともに使用することもできる。
【０４４８】
加えて、上記実施形態のうちのいくつかは、さまざまな実施形態のコンポーネント、レイヤ、オブジェクト、ロジックまたはその他の特徴を示しているが、このような特徴は、コ

ンピュータシステムまたはその他処理装置によって実行可能なソフトウェアまたはプログラムコードとして提供することができる。

【０４４９】

実施形態は、本発明の原理とその実際の応用を最適に説明することにより、意図する特定の用途に適したさまざまな実施形態およびさまざまな変更を伴うこの発明を当業者が理解できるようにすることを目的として、選択され説明されている。変更および変形は、開示されている特徴の関連する任意の組合わせを含む。本発明の範囲は以下の請求項およびその均等物によって規定されることが意図されている。

【図面】

【図１】

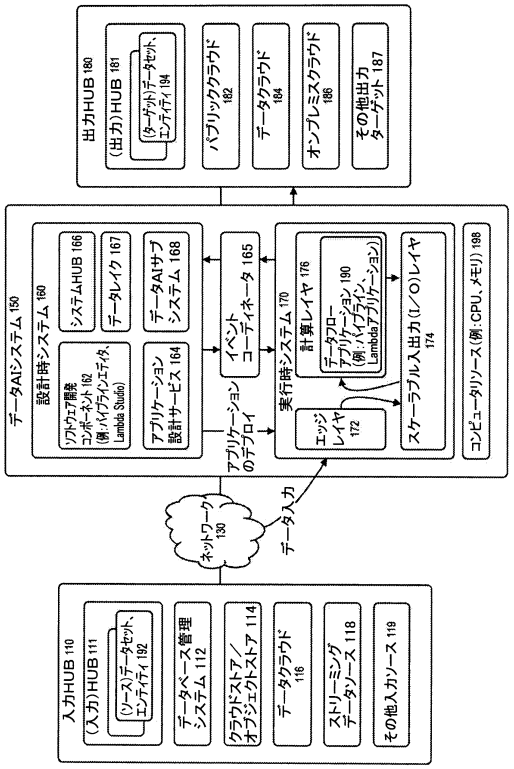

FIGURE 1

【図２】

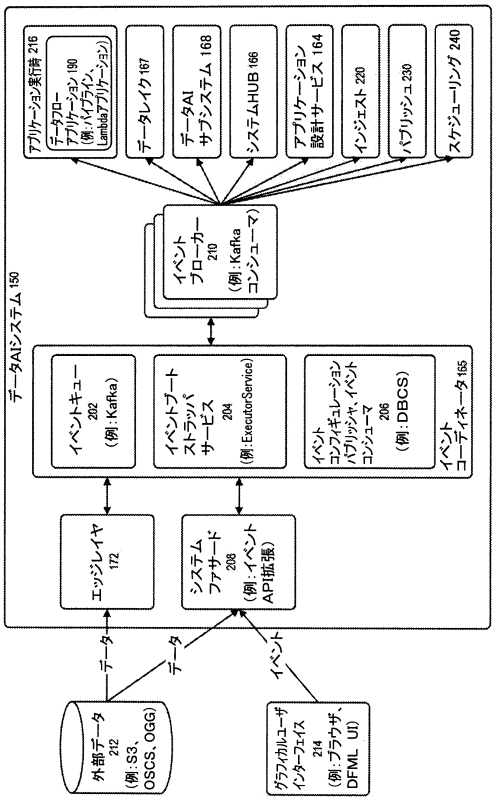

FIGURE 2

【図３】

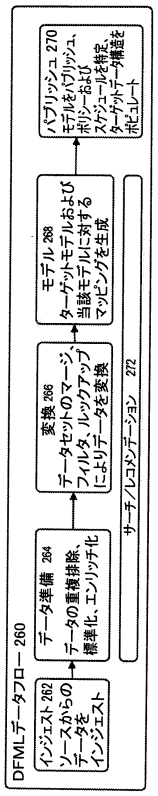

FIGURE 3

【図４】

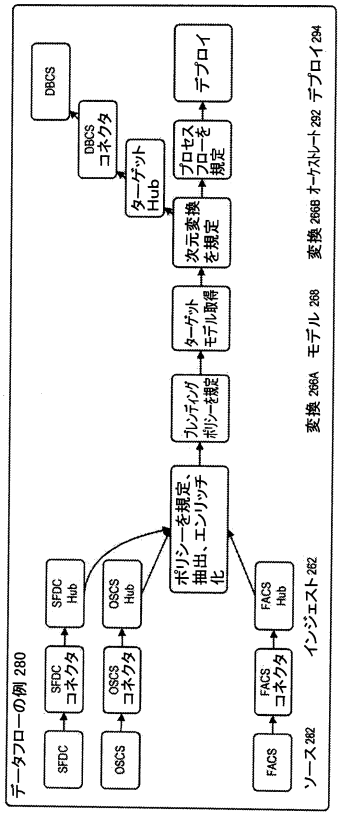

FIGURE 4

【図５】

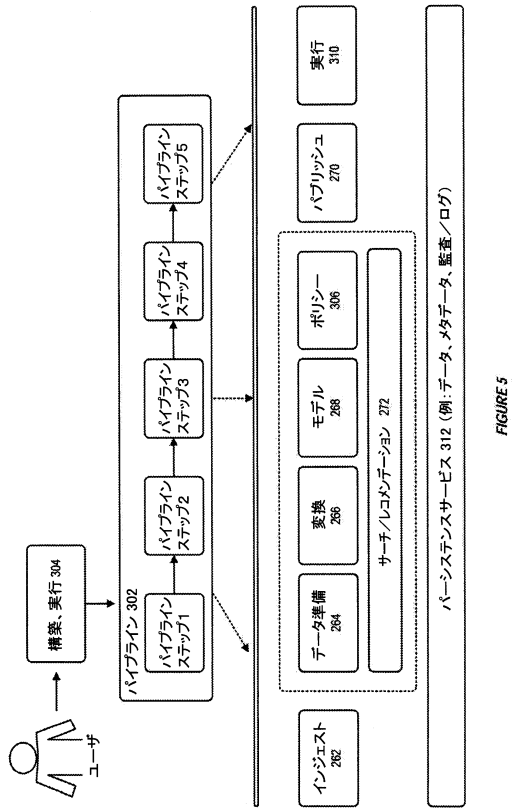

FIGURE 5

【図６】

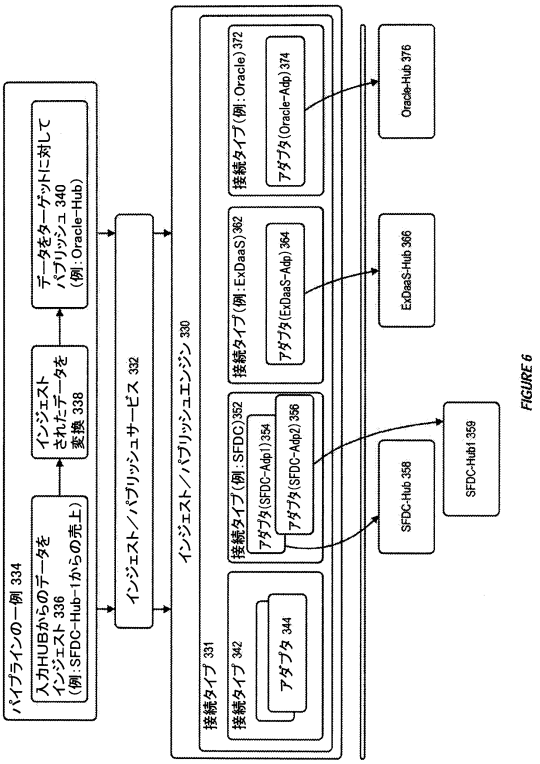

FIGURE 6

【図７】

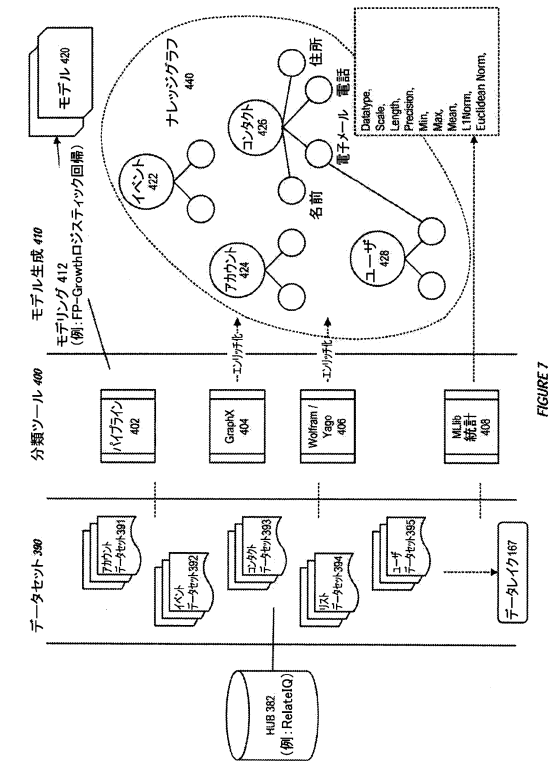

FIGURE 7

【図８】

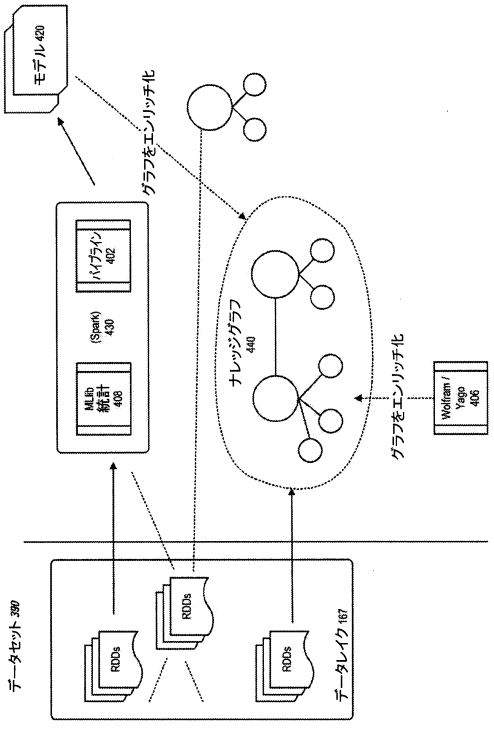

FIGURE 8

【図９】

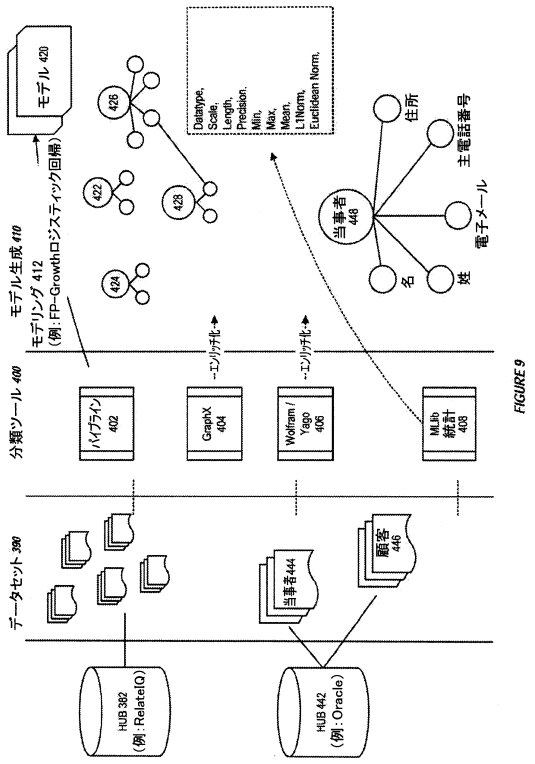

FIGURE 9

【図１０】

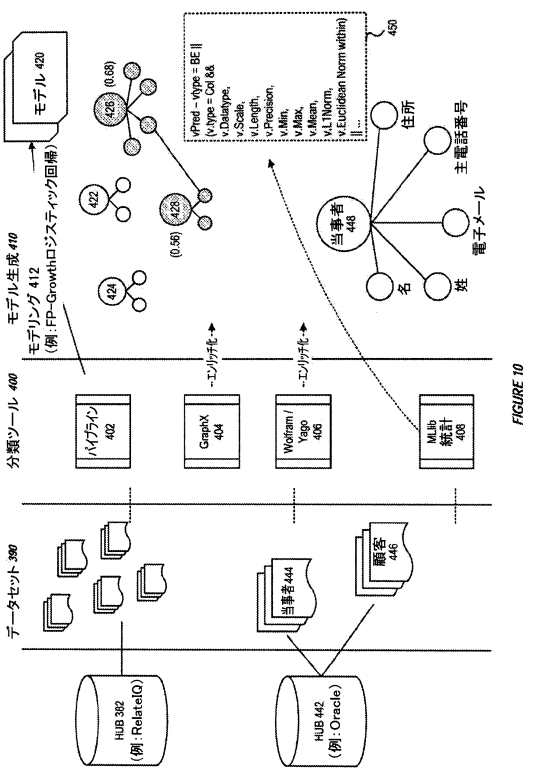

FIGURE 10

【図１１】

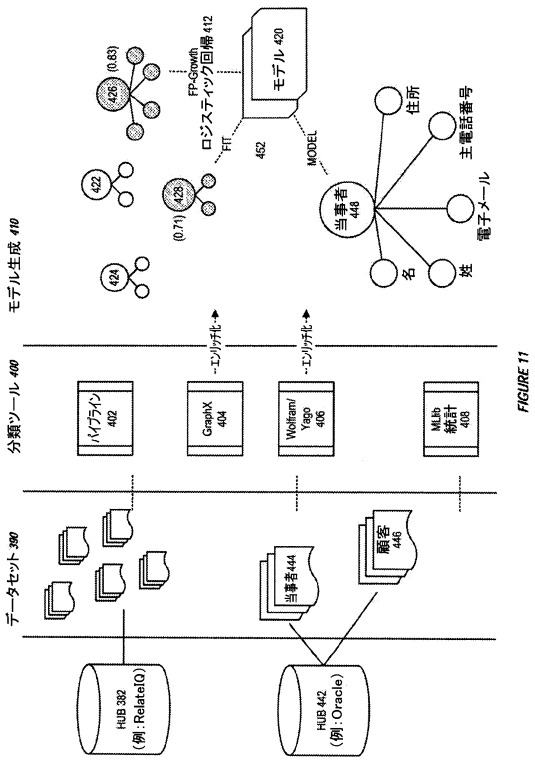

FIGURE 11

【図１２】

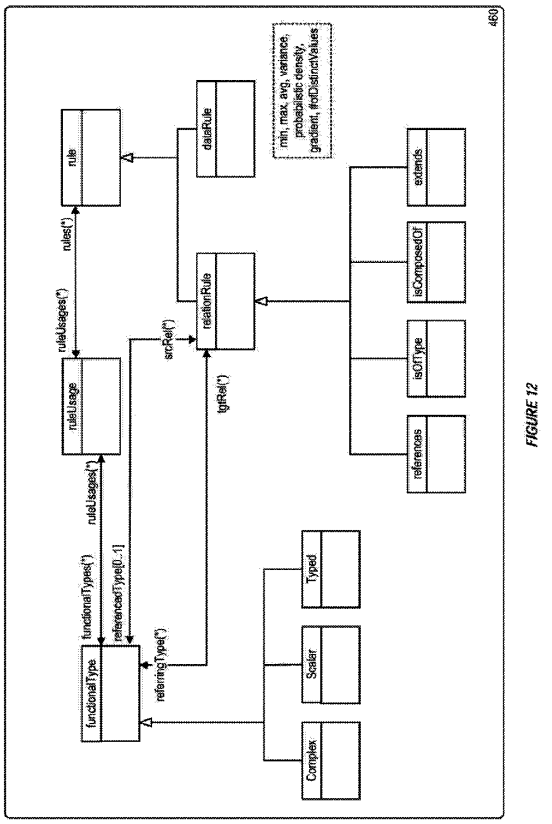

FIGURE 12

【図１３】

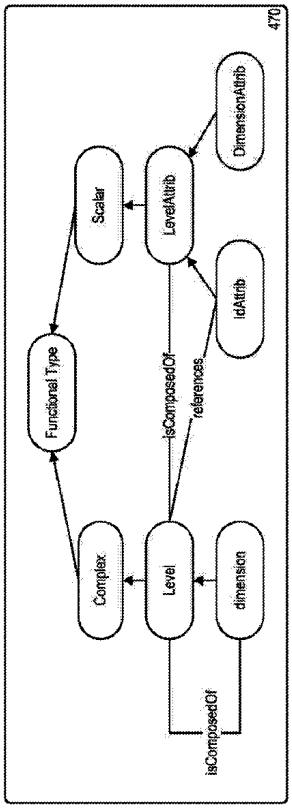

FIGURE 13

【図１４】

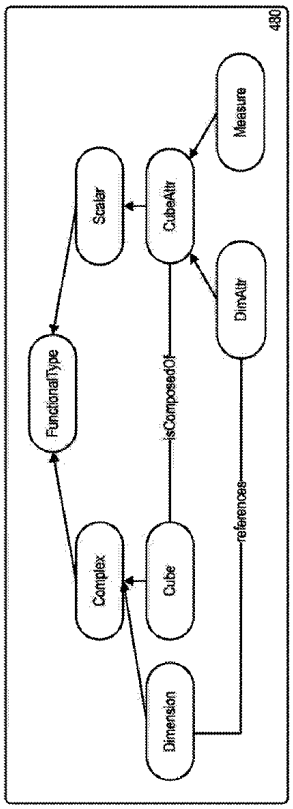

FIGURE 14

【図１５】

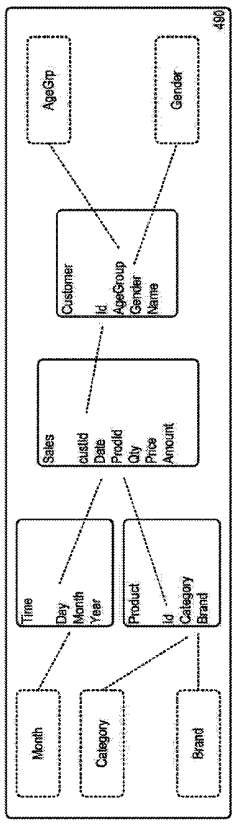

FIGURE 15

【図１６】

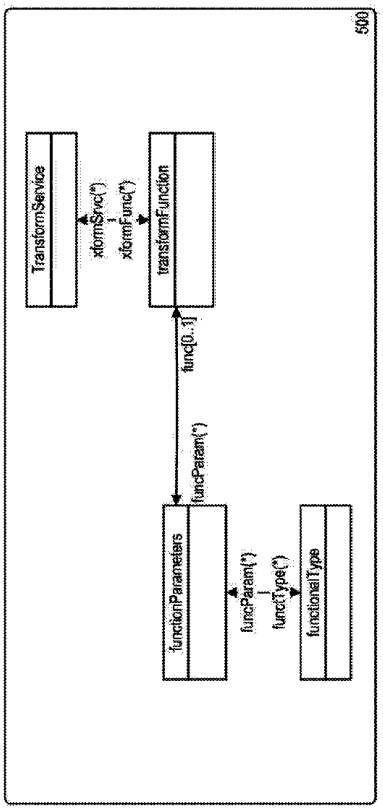

FIGURE 16

【図１７】

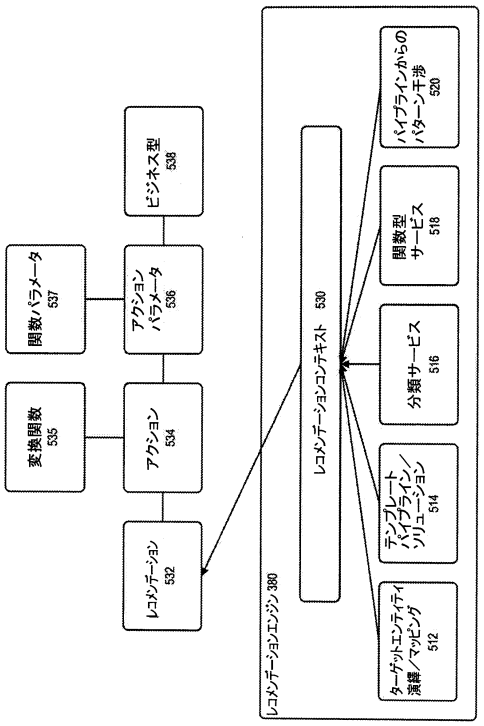

FIGURE 17

【図１８】

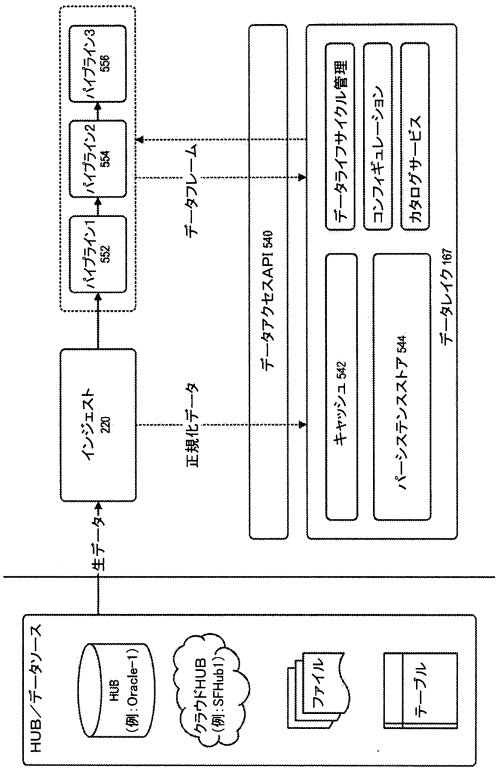

FIGURE 18

【図１９】

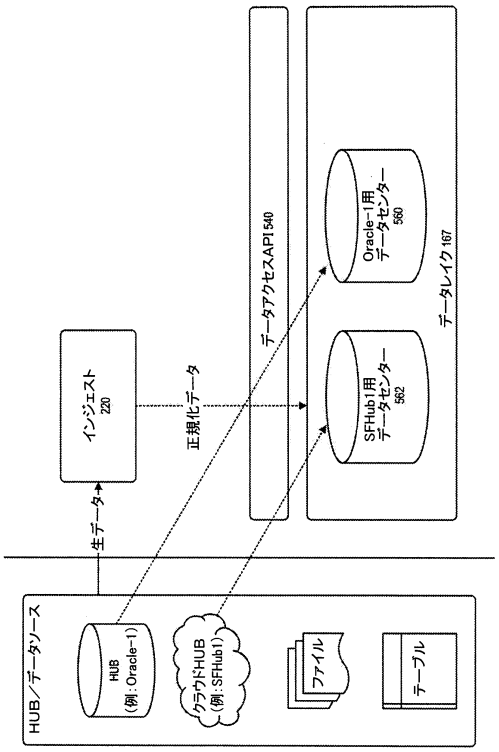

FIGURE 19

【図２０】

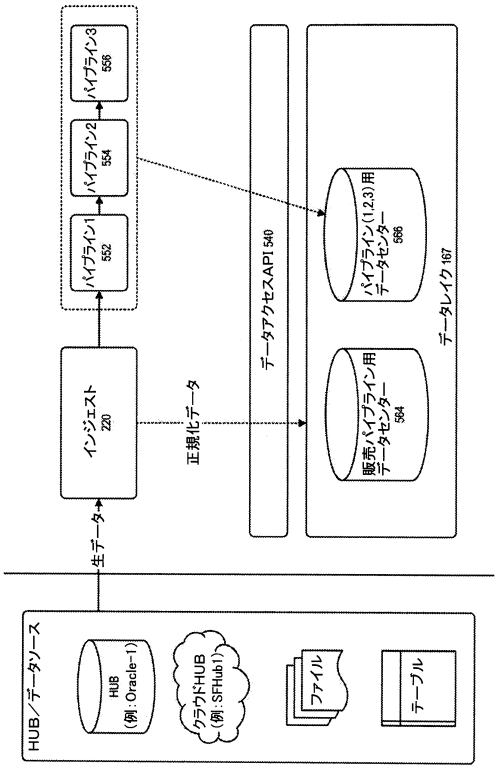

FIGURE 20

【図２１】

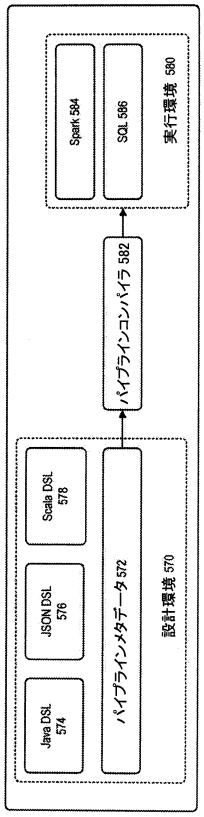

FIGURE 21

【図２２】

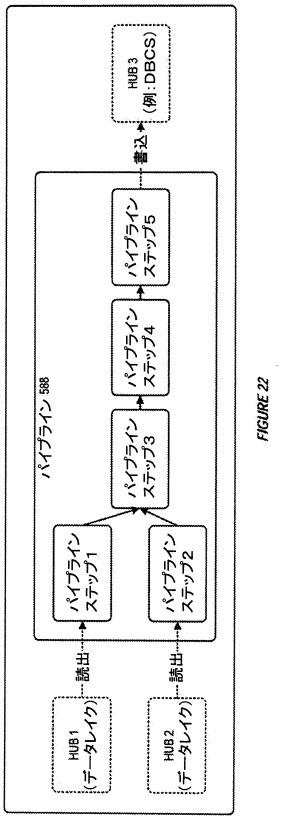

FIGURE 22

【図２３】

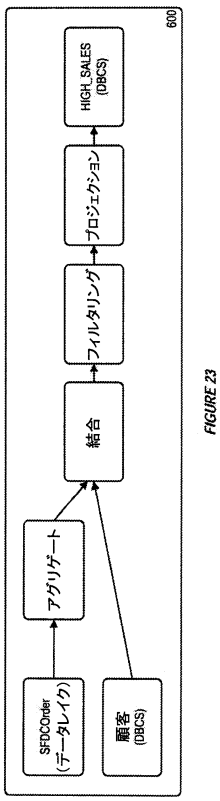

FIGURE 23

【図２４】

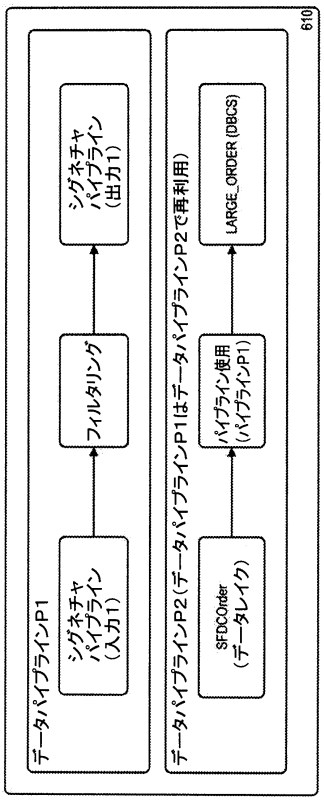

FIGURE 24

【図２５】

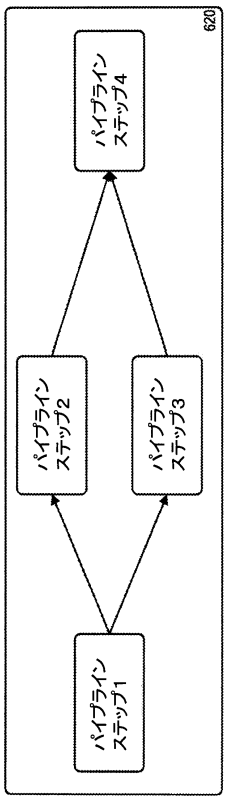

FIGURE 25

【図２６】

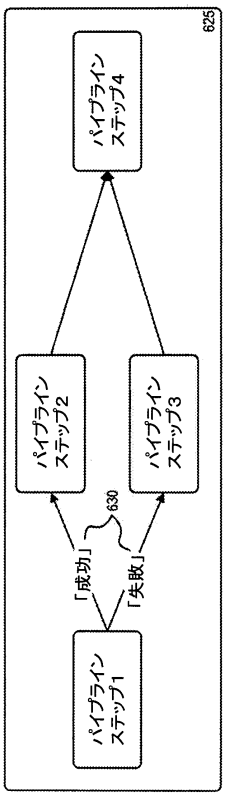

FIGURE 26

【図２７】

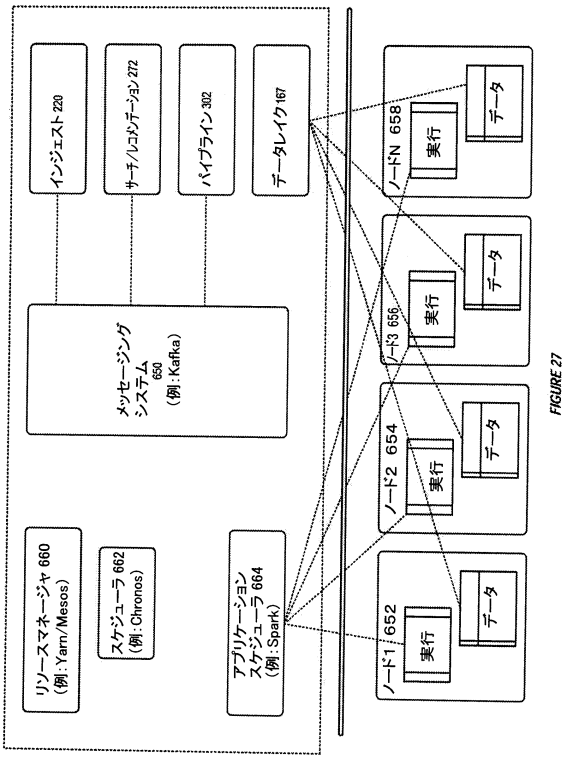

FIGURE 27

【図２８】

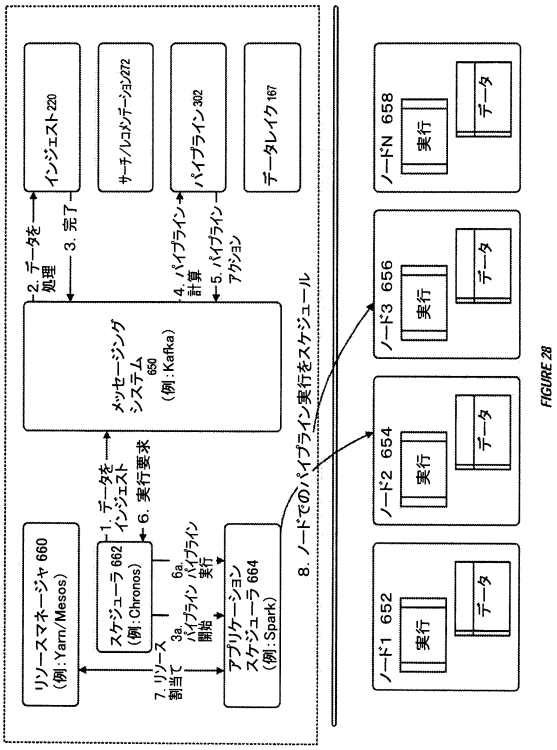

FIGURE 28

【図２９】

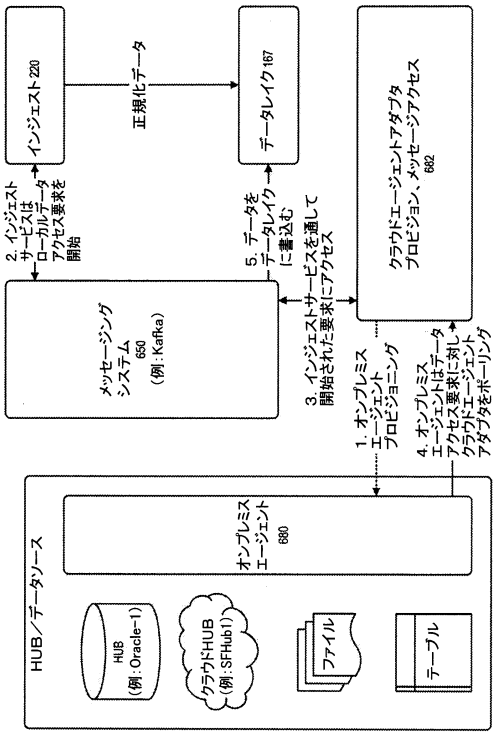

FIGURE 29

【図３０】

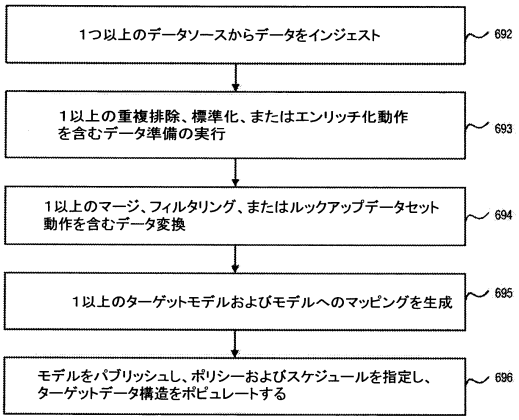

FIGURE 30

【図３１】

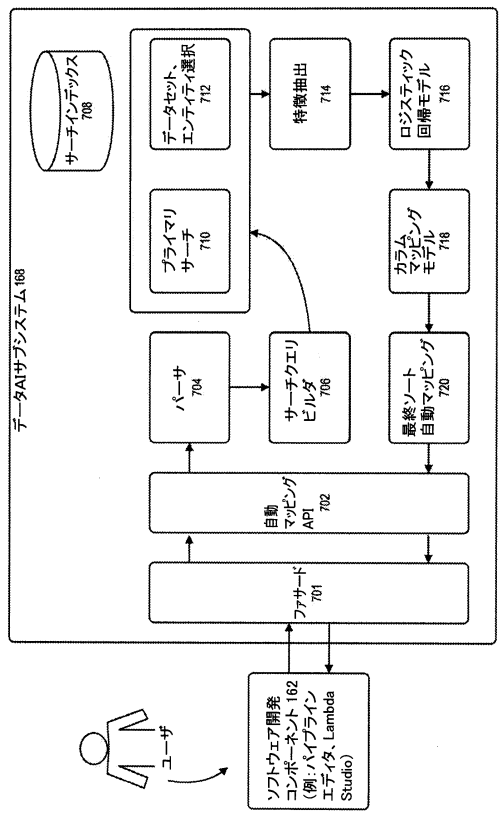

FIGURE 31

【図３２】

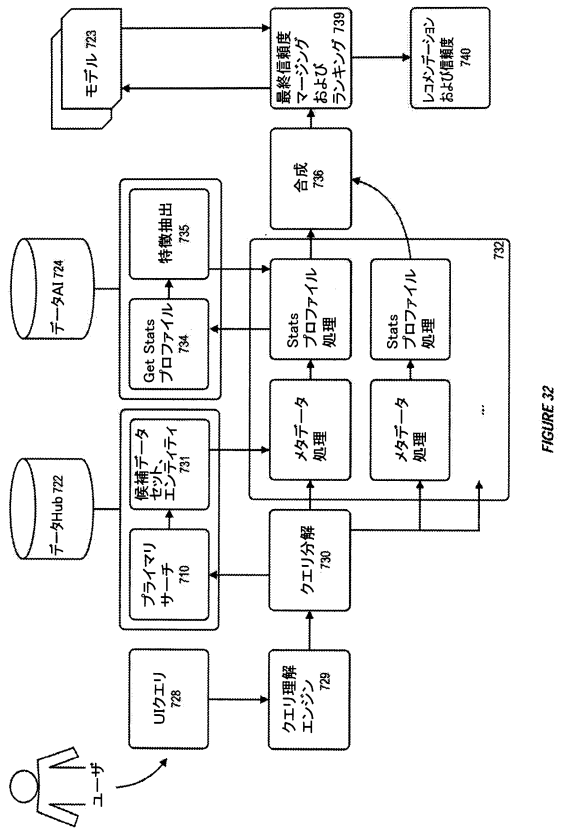

FIGURE 32

【図３３】

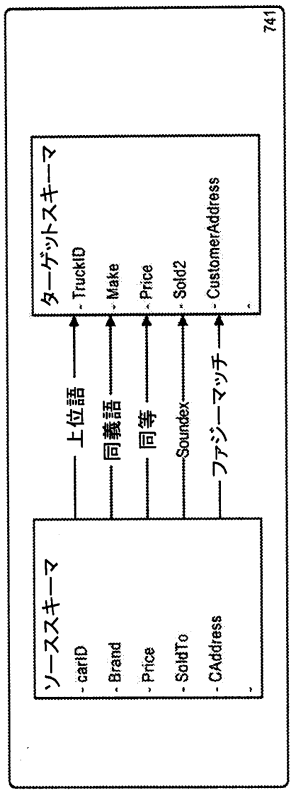

FIGURE 33

【図３４】

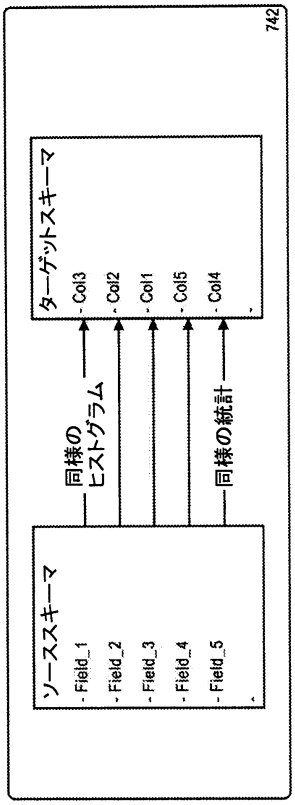

FIGURE 34

【図３５】

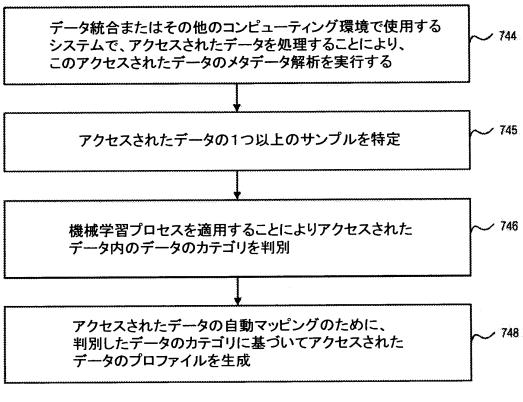

FIGURE 35

【図３６】

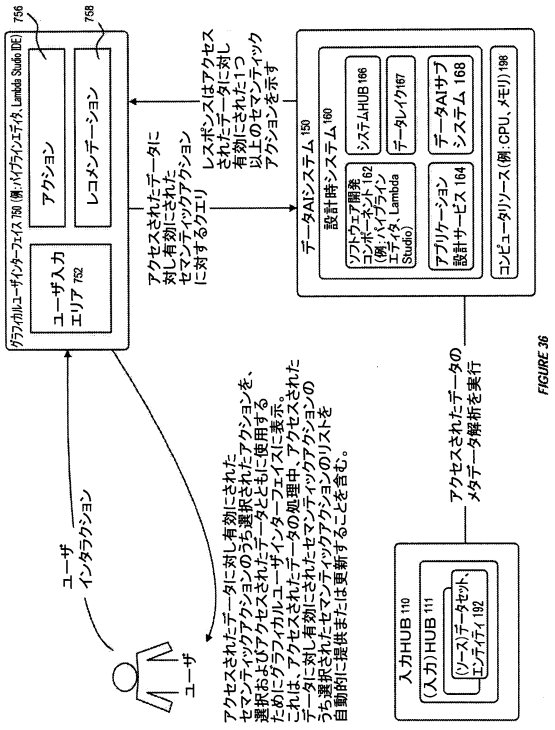

FIGURE 36

【図３７】

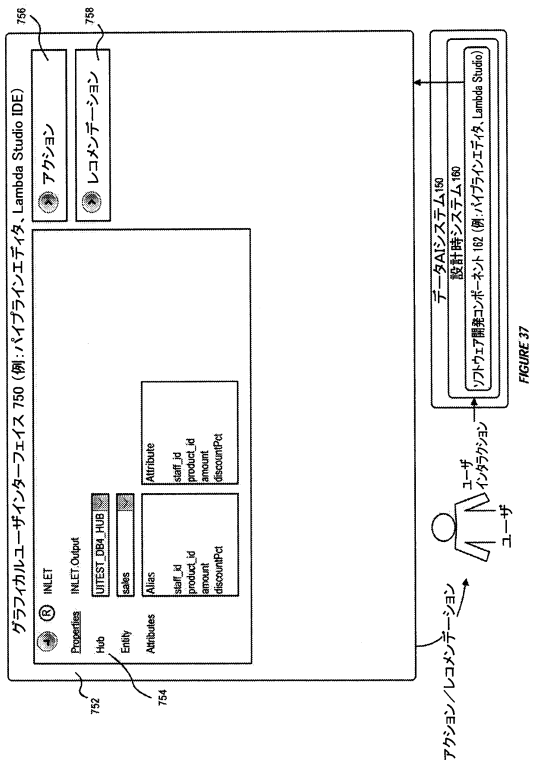

FIGURE 37

【図３８】

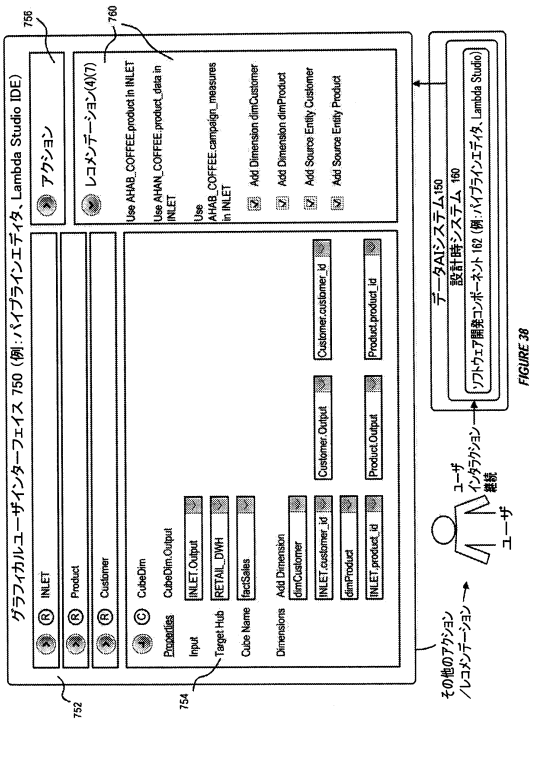

FIGURE 38

【図３９】

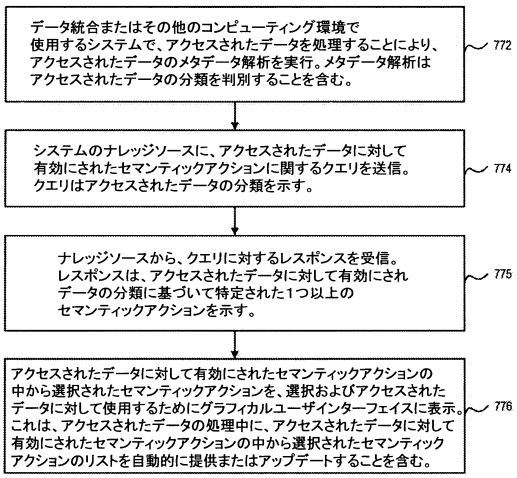

FIGURE 39

【図４０】

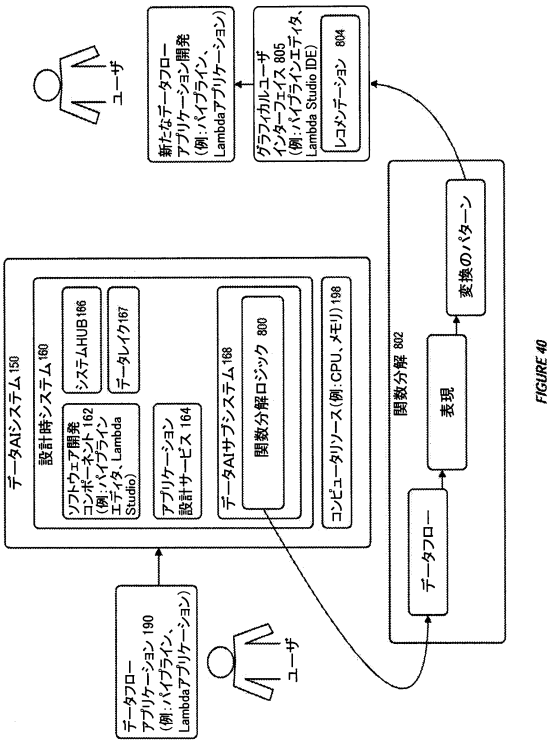

FIGURE 40

【図４１】

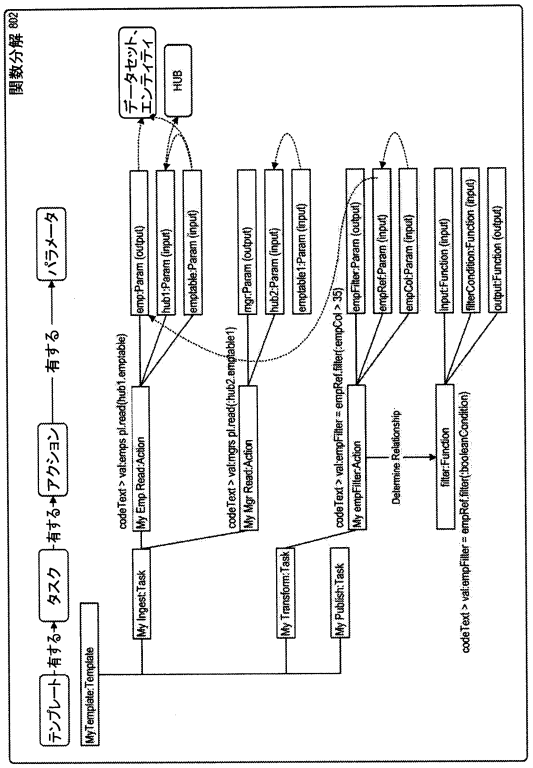

FIGURE 41

【図４２】

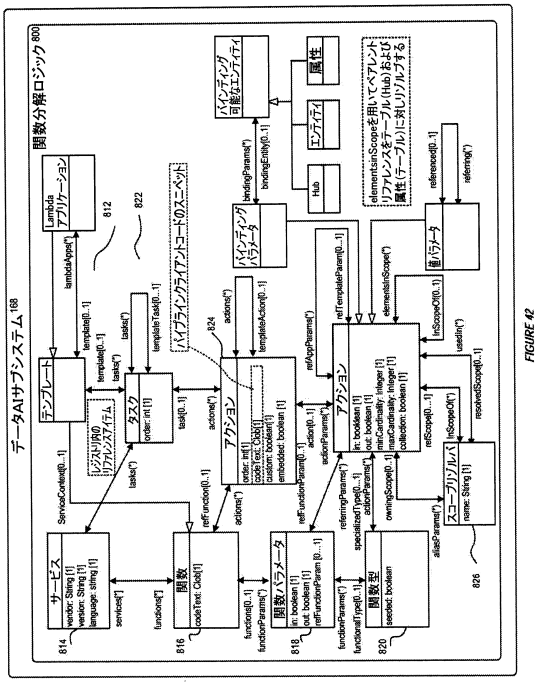

FIGURE 42

【図４３】

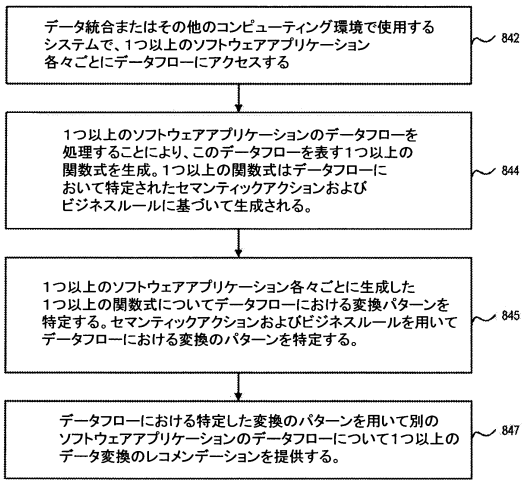

FIGURE 43

【図４４】

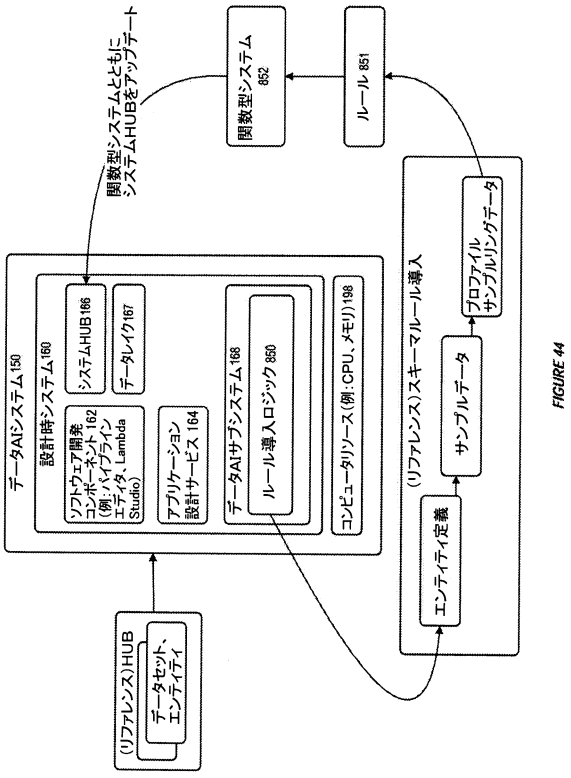

FIGURE 44

【図４５】

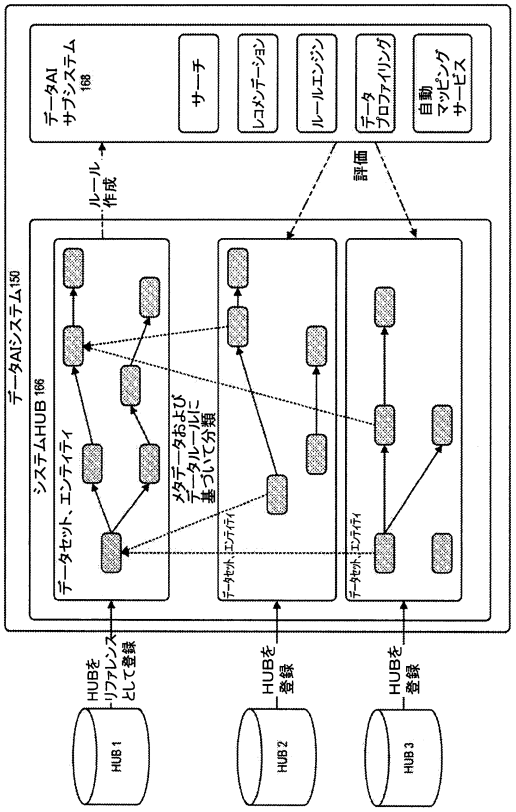

FIGURE 45

【図４６】

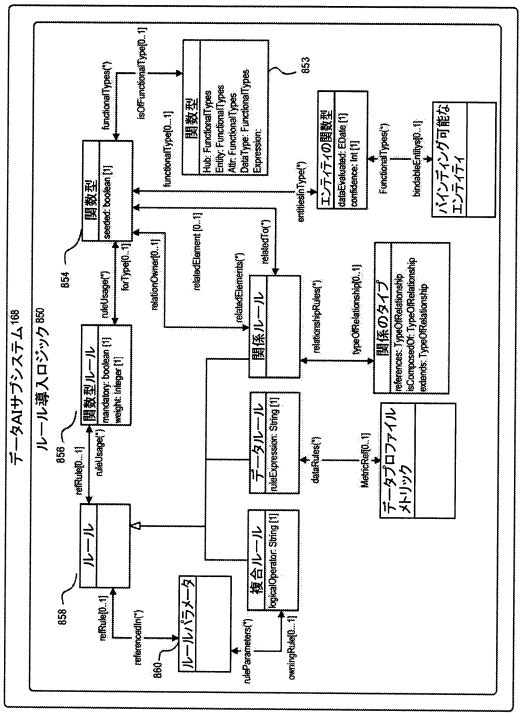

FIGURE 46

【図４７】

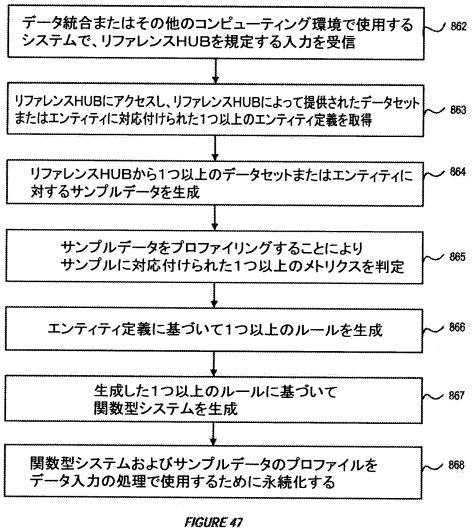

FIGURE 47

【図４８】

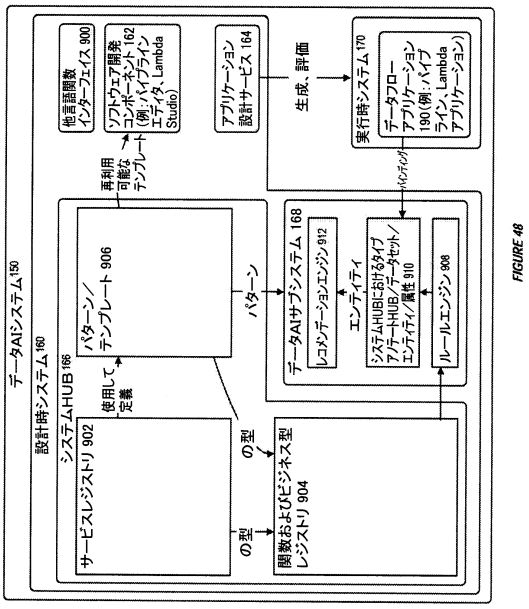

FIGURE 48

【図４９】

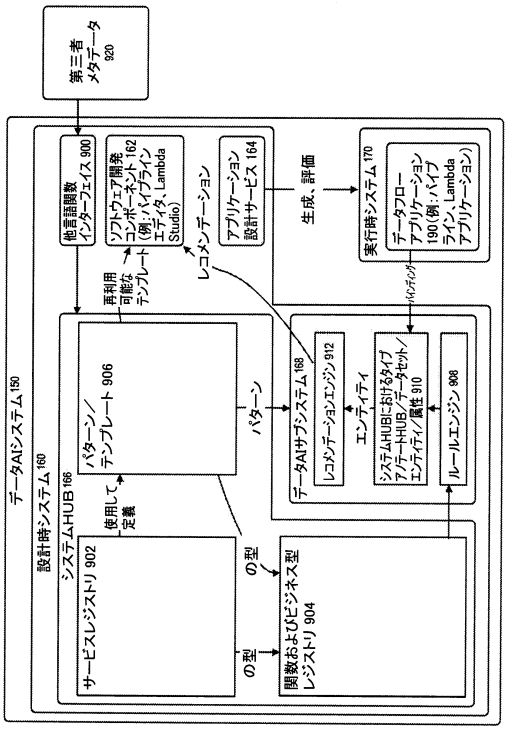

FIGURE 49

【図５０】

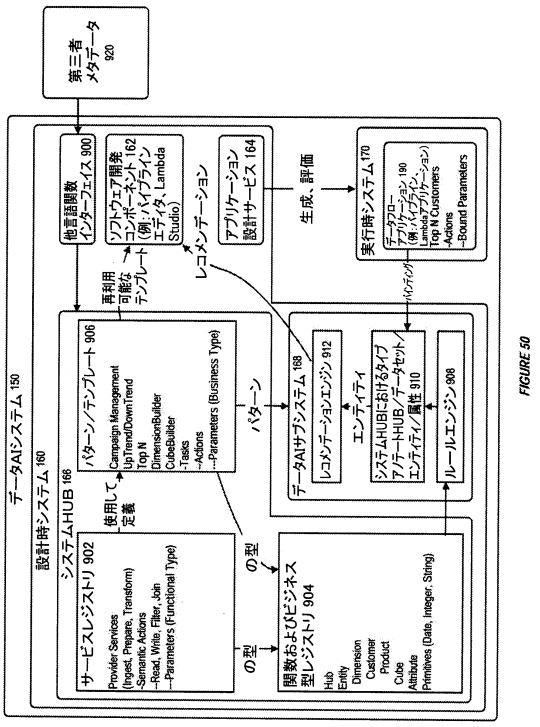

FIGURE 50

【図５１】

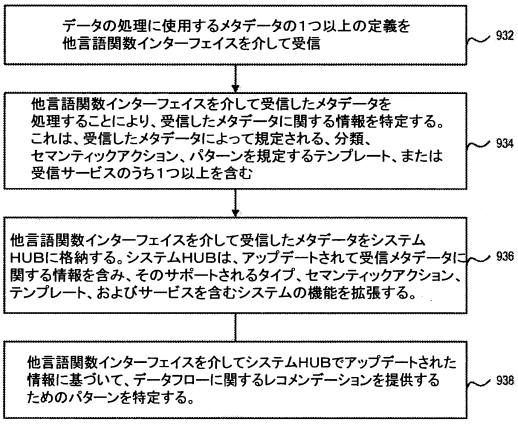

FIGURE 51

【図５２】

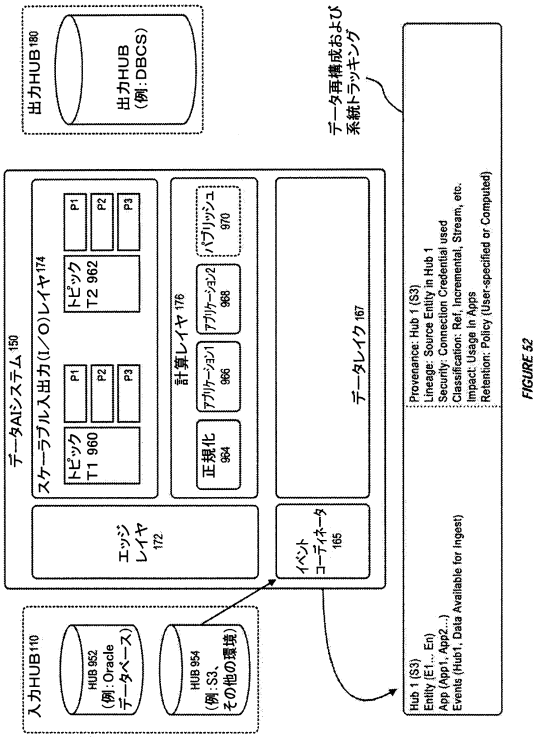

FIGURE 52

【図５３】

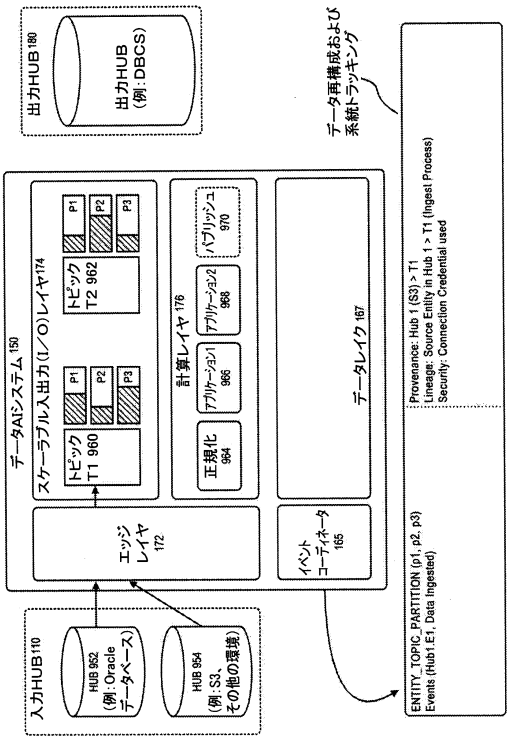

FIGURE 53

【図５４】

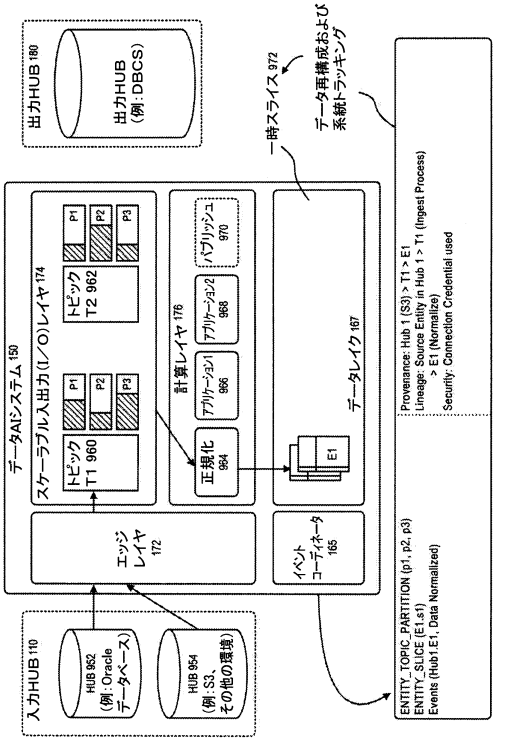

FIGURE 54

【図５５】

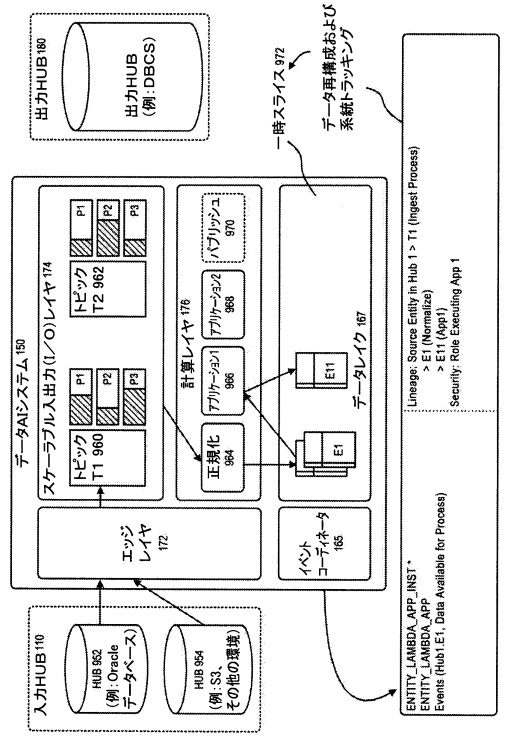

FIGURE 55

【図５６】

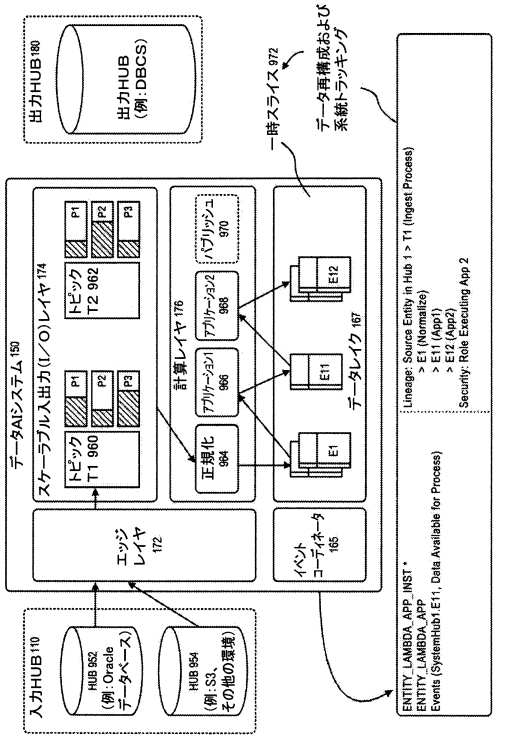

FIGURE 56

【図５７】

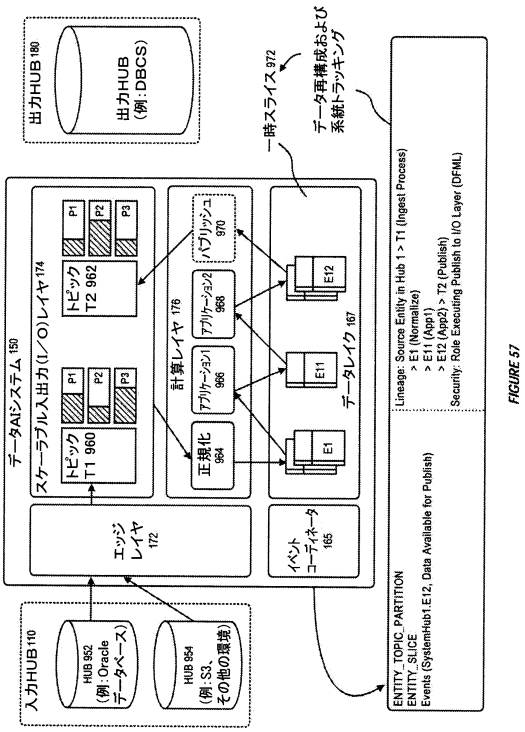

FIGURE 57

【図５８】

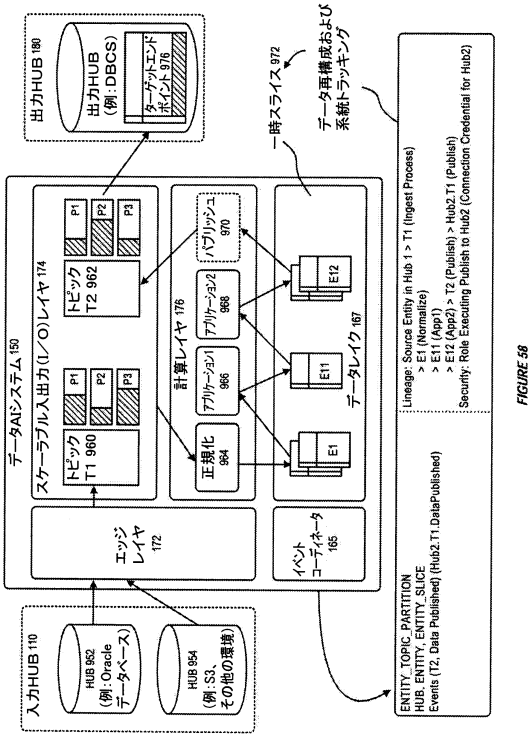

FIGURE 58

【図５９】

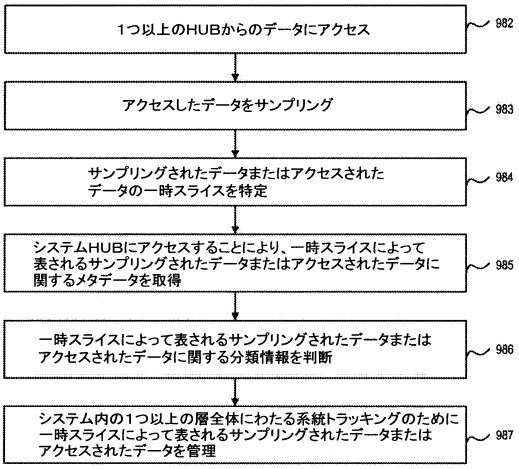

FIGURE 59

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)
(31)優先権主張番号　62/378,147
(32)優先日　平成28年8月22日(2016.8.22)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　62/378,150
(32)優先日　平成28年8月22日(2016.8.22)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　62/378,151
(32)優先日　平成28年8月22日(2016.8.22)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　62/378,152
(32)優先日　平成28年8月22日(2016.8.22)
(33)優先権主張国・地域又は機関
米国(US)
ルニア州、レッドウッド・ショアーズ、オラクル・パークウェイ、５００、エム／エス・５・オゥ・ピィ・７
(72)発明者　アラン，デイビッド
アメリカ合衆国、９４０６５ カリフォルニア州、レッドウッド・ショアーズ、オラクル・パークウェイ、５００、エム／エス・５・オゥ・ピィ・７
(72)発明者　シーサラマン，ガネーシュ
アメリカ合衆国、９４０６５ カリフォルニア州、レッドウッド・ショアーズ、オラクル・パークウェイ、５００、エム／エス・５・オゥ・ピィ・７
審査官　鹿野 博嗣
(56)参考文献　特開２００５－０６３３３２（ＪＰ，Ａ）
特開２０１１－１０８０８５（ＪＰ，Ａ）
(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１６／９０
Ｇ０６Ｆ　１６／２８