

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6412924号
(P6412924)

(45) 発行日 平成30年10月24日 (2018.10.24)

(24) 登録日 平成30年10月5日 (2018.10.5)

(51) Int. Cl.	F I
G06Q 50/10 (2012.01)	G06Q 50/10
G06F 12/00 (2006.01)	G06F 12/00 513A

請求項の数 13 (全 39 頁)

(21) 出願番号	特願2016-513953 (P2016-513953)	(73) 特許権者	502303739
(86) (22) 出願日	平成26年3月26日 (2014.3.26)		オラクル・インターナショナル・コーポレイション
(65) 公表番号	特表2016-525734 (P2016-525734A)		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、オラクル・パークウェイ500
(43) 公表日	平成28年8月25日 (2016.8.25)		
(86) 国際出願番号	PCT/US2014/031921	(74) 代理人	110001195
(87) 国際公開番号	W02014/186058		特許業務法人深見特許事務所
(87) 国際公開日	平成26年11月20日 (2014.11.20)	(72) 発明者	リー, ジョン
審査請求日	平成29年3月10日 (2017.3.10)		アメリカ合衆国、94402 カリフォルニア州、サン・マテオ、ウェスト・カピストラーノ・ウェイ、720
(31) 優先権主張番号	61/824,544		
(32) 優先日	平成25年5月17日 (2013.5.17)		
(33) 優先権主張国	米国 (US)		
(31) 優先権主張番号	14/044,417		
(32) 優先日	平成25年10月2日 (2013.10.2)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 ETLマップ設計のためのプロジェクトおよびセクタコンポーネントタイプの使用

(57) 【特許請求の範囲】

【請求項 1】

データマッピングの生成を促進するための方法であって、

論理設計の1つ以上のコンポーネントを特定する情報を1つ以上のコンピュータシステムにて受け取ることを含み、前記1つ以上のコンポーネントのうちの少なくとも1つは第1のタイプであり、前記方法はさらに、

前記1つ以上のコンピュータシステムに関連付けられる1つ以上のプロセッサにより、前記論理設計における上流のコンポーネントに基づいて、前記第1のタイプである前記1つ以上のコンポーネントのうちの少なくとも1つの前記論理設計における下流のコンポーネントに可視であるオブジェクトのデータ属性のセットを判定することと、

前記第1のタイプである前記オブジェクトのデータ属性のセットを集合させることと、

前記1つ以上のコンピュータシステムに関連付けられる前記1つ以上のプロセッサにより、前記第1のタイプである前記1つ以上のコンポーネントのうちの少なくとも1つの前記論理設計における前記下流のコンポーネントに可視であるオブジェクトのデータ属性のセットを含む集合を示す情報を生成することを含む、方法。

【請求項 2】

前記1つ以上のコンピュータシステムに関連付けられる前記1つ以上のプロセッサにより、前記第1のタイプである前記1つ以上のコンポーネントのうちの少なくとも1つの前記論理設計における前記下流のコンポーネントに可視であるオブジェクトのデータ属性の前記セットを判定することは、前記上流のコンポーネントから可視であるオブジェクトの

10

20

1つ以上の属性を導き出すことと、前記下流のコンポーネントに前記1つ以上の属性をエクスポートすることを含む、請求項1に記載の方法。

【請求項3】

前記論理設計の前記1つ以上のコンポーネントを特定する前記情報を受け取ることは、前記論理設計を通して流れる情報の形を変更するオペレーションを示す情報を受け取ることを含む、請求項1または2に記載の方法。

【請求項4】

前記論理設計の前記1つ以上のコンポーネントを特定する前記情報を受け取ることは、前記論理設計を通して流れる情報のフローを制御するが前記論理設計を通して流れる前記情報の形を変更しないオペレーションを示す情報を受け取ることを含む、請求項1～3のいずれか1項に記載の方法。

10

【請求項5】

前記論理設計の前記1つ以上のコンポーネントを特定する前記情報を受け取ることは、ソースデータストアに格納されるデータの1つ以上の属性を有するソースコンポーネントを示す情報を受け取ることを含む、請求項1～4のいずれか1項に記載の方法。

【請求項6】

前記論理設計の前記1つ以上のコンポーネントを特定する前記情報を受け取ることは、ターゲットデータストアに格納されるデータの1つ以上の属性を有するターゲットコンポーネントを示す情報を受け取ることを含む、請求項1～5のいずれか1項に記載の方法。

【請求項7】

20

前記第1のタイプである前記1つ以上のコンポーネントのうちの少なくとも1つの前記論理設計における前記下流のコンポーネントに可視であるオブジェクトのデータ属性のセットを含む集合を示す前記情報を生成することは、前記下流のコンポーネントに属性のリストをエクスポートすることを含む、請求項1～6のいずれか1項に記載の方法。

【請求項8】

前記論理設計へのコンポーネントまたは属性の導入または除去による前記論理設計における変更を前記1つ以上のコンピュータシステムにて受け取ることと、

前記1つ以上のコンピュータシステムに関連付けられる前記1つ以上のプロセッサにより、前記論理設計における前記変更が前記第1のタイプである前記1つ以上のコンポーネントのうちの少なくとも1つに影響するかどうかを判定することと、

30

前記論理設計における前記変更が前記第1のタイプである前記1つ以上のコンポーネントの少なくとも1つに影響するという判定に基づいて、前記1つ以上のコンピュータシステムに関連付けられる前記1つ以上のプロセッサにより、前記下流のコンポーネントに可視であるオブジェクトのデータ属性の更新されたセットを判定することとをさらに含む、請求項1～7のいずれか1項に記載の方法。

【請求項9】

前記論理設計へのコンポーネントまたは属性の導入による前記論理設計における変更を前記1つ以上のコンピュータシステムにて受け取ることと、

前記1つ以上のコンピュータシステムに関連付けられる前記1つ以上のプロセッサにより、前記論理設計における前記変更が前記第1のタイプである前記1つ以上のコンポーネントのうちの少なくとも1つに影響するかどうかを判定することと、

40

前記論理設計における前記変更が前記第1のタイプである前記1つ以上のコンポーネントのうちの少なくとも1つに影響するという判定に基づいて、前記下流のコンポーネントに可視であるオブジェクトのデータ属性の前記セットを保存することとをさらに含む、請求項1～8のいずれか1項に記載の方法。

【請求項10】

コンポーネントまたは属性をリネームする前記論理設計における変更を、前記1つ以上のコンピュータシステムにて受け取ることと、

前記1つ以上のコンピュータシステムに関連付けられる前記1つ以上のプロセッサにより、前記論理設計における前記変更が前記第1のタイプである前記1つ以上のコンポーネ

50

ントのうちの少なくとも1つに影響するかどうかを判定することと、

前記論理設計における前記変更が前記第1のタイプである前記1つ以上のコンポーネントのうちの少なくとも1つに影響するという判定に基づいて、前記下流のコンポーネントに可視であるオブジェクトのデータ属性の前記セットをリネームすることとをさらに含む、請求項1～9のいずれか1項に記載の方法。

【請求項11】

データマッピングの生成を促進するためのコンピュータ実行可能コードを含むコンピュータ読取可能プログラムであって、

論理設計の1つ以上のコンポーネントを特定する情報を受け取るためのコードを含み、前記1つ以上のコンポーネントのうちの少なくとも1つは第1のタイプであり、

前記論理設計における上流のコンポーネントに基づいて、前記第1のタイプである前記1つ以上のコンポーネントのうちの少なくとも1つの前記論理設計における下流のコンポーネントに可視であるオブジェクトのデータ属性のセットを判定するためのコードと、

前記第1のタイプである前記オブジェクトのデータ属性のセットを集合させるためのコードと、

前記第1のタイプである前記1つ以上のコンポーネントのうちの少なくとも1つの前記論理設計における前記下流のコンポーネントに可視であるオブジェクトのデータ属性のセットを含む集合を示す情報を生成するためのコードとを含む、コンピュータ読取可能プログラム。

【請求項12】

コンピュータによって実行されると、請求項1～10のいずれか1項に記載の方法を当該コンピュータに実施させるコンピュータ読取可能プログラム。

【請求項13】

データマッピングの生成を促進するためのシステムであって、

プロセッサと、

前記プロセッサと通信し、かつ、命令のセットを格納するように構成されるメモリとを含み、前記命令のセットは、前記プロセッサによって実行されると、

論理設計の1つ以上のコンポーネントを特定する情報を受け取るように前記プロセッサを構成し、前記1つ以上のコンポーネントのうちの少なくとも1つは第1のタイプであり、前記命令のセットはさらに、

前記論理設計における上流のコンポーネントに基づいて、前記第1のタイプである前記1つ以上のコンポーネントのうちの少なくとも1つの前記論理設計における下流のコンポーネントに可視であるオブジェクトのデータ属性のセットを判定し、

前記第1のタイプである前記オブジェクトのデータ属性のセットを集合させ、

前記第1のタイプである前記1つ以上のコンポーネントのうちの少なくとも1つの前記論理設計における前記下流のコンポーネントに可視であるオブジェクトのデータ属性のセットを含む集合を示す情報を生成するように前記プロセッサを構成する、システム。

【発明の詳細な説明】

【背景技術】

【0001】

発明の背景

ペースが一層速くなっている今日のビジネス環境において、組織はより特殊化されたソフトウェアアプリケーションを使用する必要がある。さらに、組織は、異種混合のハードウェアプラットフォームおよびシステム上でこれらのアプリケーションの共存を保証する必要があるとともに、アプリケーションとシステムとの間でデータを共有する能力を保証する必要がある。

【0002】

データ統合は、あるシステムから別のシステムへのデータ移行を行なうために具体的に構成する特別に設計されたソフトウェアを有するプロプライエタリサーバを必要とするリソース集中的なプロシージャである。これらのエリアにおいて、パフォーマンスが典型的

10

20

30

40

50

に貧弱または非能率的となっている。

【 0 0 0 3 】

したがって、データ統合シナリオの開発に関係する問題を解決することが望まれており、そのいくつかが本願明細書において論じられ得る。さらに、データ統合シナリオの開発に関係する障害を低減することが望まれており、そのいくつかが本願明細書において論じられ得る。

【 発明の概要 】

【 課題を解決するための手段 】

【 0 0 0 4 】

発明の概要

この開示の以下の部分は、少なくとも主題の基本的な理解を提供する目的のために、この開示において発見される 1 つ以上のイノベーション、実施形態および / または例の簡素化された概要を提供する。この概要は、如何なる特定の実施形態または例の広範囲な概要も提供することを試みてはいない。さらに、この概要は、実施形態または例の主な / 決定的な要素を識別するようには意図されておらず、または、この開示の主題の範囲を定めるようには意図されていない。したがって、この概要の 1 つの目的は、この開示において発見されるいくつかのイノベーション、実施形態および / または例を簡素化された形で、後述されるさらなる詳細な説明の前置きとして提供することであり得る。

【 0 0 0 5 】

さまざまな実施形態において、ユーザはデータ統合システムによって、プラットフォームおよび技術に依存しない論理設計を作成し得る。ユーザは、ユーザがデータをソースとターゲットとの間でどのように流したいかをハイレベルで規定する論理設計を作成し得る。ユーザのインフラストラクチャを考慮して、ツールが論理設計を分析し、物理的設計を作成し得る。論理設計は、設計における各ソースおよびターゲットならびにジョインまたはフィルタのようなオペレーションおよびアクセスポイントに対応する複数のコンポーネントを含み得る。物理的設計に転送された際の各コンポーネントは、データに対してオペレーションを行なうようコードを生成する。存在する技術（たとえば SQL サーバ、オラクル、Hadoop など）およびに使用される言語（SQL、pig など）に依存して、各コンポーネントによって生成されるコードは異なり得る。

【 0 0 0 6 】

1 つの局面において、データ統合システムのユーザは、始めから終わりまで、論理設計における各コンポーネントにて、すべてのデータ属性を特定する必要はない。データ統合システムは、論理設計を通して流れる情報を完全に宣言する必要性を回避する、プロジェクトおよびセレクトタイプのような複数のコンポーネントタイプを提供する。データ統合システムは、所定のコンポーネントタイプによって表わされるオペレーションにて何の属性が必要とされるか決定し得る。これは、設計およびメンテナンスの両方を簡素化する。さまざまな局面において、既存の RDBMS リソースと、パフォーマンスの向上を達成するために別個のプロプライエタリ ETL サーバの必要性を回避する性能とをレバレッジするデータ変換および移行が提供される。

【 0 0 0 7 】

一実施形態において、データマッピングの生成を促進するための方法は、論理設計の 1 つ以上のコンポーネントを特定する情報を受け取ることを含み、1 つ以上のコンポーネントのうちの少なくとも 1 つは第 1 のタイプである。第 1 のタイプである 1 つ以上のコンポーネントのうちの少なくとも 1 つの論理設計における下流のコンポーネントに可視であるデータ属性のセットが、論理設計における上流のコンポーネントに基づいて、判定される。次いで、第 1 のタイプである 1 つ以上のコンポーネントのうちの少なくとも 1 つの論理設計における下流のコンポーネントに可視である属性のセットを示す情報が生成される。1 つの局面において、第 1 のタイプである 1 つ以上のコンポーネントのうちの少なくとも 1 つの論理設計における下流のコンポーネントに可視であるデータ属性のセットを判定することは、上流のコンポーネントから可視である 1 つ以上の属性を導き出すことと、下流

10

20

30

40

50

のコンポーネントに1つ以上の属性をエクスポートする(expose)こととを含み得る。別の局面において、論理設計の1つ以上のコンポーネントを特定する情報を受け取ることは、論理設計を通して流れる情報の形を変更するオペレーションを示す情報を受け取ることを含み得る。

【0008】

いくつかの実施形態において、論理設計の1つ以上のコンポーネントを特定する情報を受け取ることは、論理設計を通して流れる情報のフローを制御するが論理設計を通して流れる情報の形を変更しないオペレーションを示す情報を受け取ることを含み得る。さらに別の実施形態において、論理設計の1つ以上のコンポーネントを特定する情報を受け取ることは、ソースデータストアに格納されるデータの1つ以上の属性を有するソースコンポーネントを示す情報を受け取ることを含み得る。論理設計の1つ以上のコンポーネントを特定する情報を受け取ることは、ターゲットデータストアに格納されるデータの1つ以上の属性を有するターゲットコンポーネントを示す情報を受け取ることを含み得る。

10

【0009】

第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つの論理設計における下流のコンポーネントに可視である属性のセットを示す情報を生成することは、下流のコンポーネントに属性のリストをエクスポートすることを含み得る。一実施形態において、論理設計へのコンポーネントまたは属性の導入または除去による論理設計における変更が受け取られ得る。論理設計における変更が第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つに影響するかどうかの判定がなされる。論理設計における変更が、第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つに影響するという判定に基づいて、下流のコンポーネントに可視であるデータ属性の更新されたセットが判定され得る。

20

【0010】

さらに別の実施形態において、論理設計へのコンポーネントまたは属性の導入による論理設計における変更が受け取られる。論理設計における変更が第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つに影響するかどうかを判定がなされ得る。論理設計における変更が、第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つに影響するという判定に基づいて、下流のコンポーネントに可視であるデータ属性のセットが保存され得る。別の局面において、論理設計における変更が、第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つに影響するという判定に基づいて、下流のコンポーネントに可視であるデータ属性のセットが自動的にリネームされ得る。

30

【0011】

一実施形態において、一時的でないコンピュータ読取可能媒体は、データマッピングの生成を促進するためのコンピュータ実行可能コードを格納する。一時的でないコンピュータ読取可能媒体は、論理設計の1つ以上のコンポーネントを特定する情報を受け取るためのコードを含み、1つ以上のコンポーネントのうちの少なくとも1つは第1のタイプであり、一時的でないコンピュータ読取可能媒体はさらに、論理設計における上流のコンポーネントに基づいて、第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つの論理設計における下流のコンポーネントに可視であるデータ属性のセットを判定するためのコードと、第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つの論理設計における下流のコンポーネントに可視である属性のセットを示す情報を生成するためのコードとを含み得る。

40

【0012】

別の実施形態において、データマッピングの生成を促進するためのシステムは、プロセッサと、プロセッサと通信するとともに命令のセットを格納するように構成されるメモリとを含み得、命令のセットは、プロセッサによって実行されると、論理設計の1つ以上のコンポーネントを特定する情報を受け取るようにプロセッサを構成し、1つ以上のコンポーネントのうちの少なくとも1つは第1のタイプであり、命令のセットはさらに、論理設計における上流のコンポーネントに基づいて、第1のタイプである1つ以上のコンポーネ

50

ントのうちの少なくとも1つの論理設計における下流のコンポーネントに可視であるデータ属性のセットを判定し、第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つの論理設計における下流のコンポーネントに可視である属性のセットを示す情報を生成するようにプロセッサを構成する。

【0013】

一実施形態において、データマッピングの生成を促進するためのシステムは、論理設計の1つ以上のコンポーネントを特定する情報を受け取るように構成される受取部を含み、1つ以上のコンポーネントのうちの少なくとも1つは第1のタイプであり、システムは、論理設計における上流のコンポーネントに基づいて、第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つの論理設計における下流のコンポーネントに可視であるデータ属性のセットを判定するように構成される判定部と、第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つの論理設計における下流のコンポーネントに可視である属性のセットを示す情報を生成するように構成される生成部とを含む。第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つの論理設計における下流のコンポーネントに可視であるデータ属性のセットを判定することは、上流のコンポーネントから可視である1つ以上の属性を導き出すとともに、下流のコンポーネントに1つ以上の属性をエクスポートすることを含み得る。

10

【0014】

論理設計の1つ以上のコンポーネントを特定する情報は、論理設計を通して流れる情報の形を変更するオペレーションを示す情報を含み得る。論理設計の1つ以上のコンポーネントを特定する情報は、論理設計を通して流れる情報のフローを制御するが論理設計を通して流れる情報の形を変更しないオペレーションを示す情報を含み得る。論理設計の1つ以上のコンポーネントを特定する情報は、ソースデータストアに格納されるデータの1つ以上の属性を有するソースコンポーネントを示す情報を含み得る。論理設計の1つ以上のコンポーネントを特定する情報は、ターゲットデータストアに格納されるデータの1つ以上の属性を有するターゲットコンポーネントを示す情報を含み得る。第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つの論理設計における下流のコンポーネントに可視である属性のセットを示す情報を生成することは、下流のコンポーネントに属性のリストをエクスポートすることを含み得る。

20

【0015】

一実施形態において、受取部はさらに、論理設計へのコンポーネントまたは属性の導入または除去による論理設計における変更を受け取るように構成されており、判定部はさらに、論理設計における変更が、第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つに影響するかどうかを判定するように構成されており、判定部はさらに、論理設計における変更が、第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つに影響するという判定に基づいて、下流のコンポーネントに可視であるデータ属性の更新されたセットを判定するように構成される。

30

【0016】

1つの局面において、システムはさらに保存部を含み得、受取部はさらに、論理設計へのコンポーネントまたは属性の導入による論理設計における変更を受け取るように構成されており、判定部はさらに、論理設計における変更が、第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つに影響するかどうかを判定するように構成されており、保存部は、論理設計における変更が、第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つに影響するという判定に基づいて、下流のコンポーネントに可視であるデータ属性のセットを保存するように構成される。

40

【0017】

別の局面において、システムはさらにリネーム部を含み得、受取部はさらに、コンポーネントまたは属性をリネームする論理設計における変更を受け取るように構成されており、判定部はさらに、論理設計における変更が、第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つに影響するかどうかを判定するように構成されており、リネ

50

ーム部は、論理設計における変更が、第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つに影響するという判定に基づいて、下流のコンポーネントに可視であるデータ属性のセットをリネームするように構成される。

【0018】

この開示の主題の性質および均等物（ならびに提供される如何なる固有または明白な利点および改良）のさらなる理解は、この開示の残りの部分、任意の添付の図面および請求の範囲への参照により、上記のセクションに加えて実現されるべきである。

【0019】

この開示において発見されるイノベーション、実施形態および/または例を適切に記載および説明するために、1つ以上の添付の図面に対して参照がなされ得る。1つ以上の添付の図面を説明するよう用いられる付加的な詳細または例は、特許請求される発明のいずれか、ここで記載される実施形態および/または例のいずれか、またはこの開示において示されるいずれかのイノベーションの現在理解されている最良の形態の範囲への限定と見なされるべきではない。

【図面の簡単な説明】

【0020】

【図1】本発明の実施形態を組み込み得るシステムの簡略図である。

【図2】本発明の実施形態に従ったデータ統合システムのブロック図である。

【図3】本発明の実施形態に従った、データ統合システムを実現するために使用され得るハードウェア/ソフトウェアスタックの簡略ブロック図である。

【図4】データ統合シナリオが本発明のさまざまな実施形態において作成され得るさまざまな異種混合のデータソースを有する環境のブロック図である。

【図5A】データ統合システムによって実行され得る従来のデータ統合処理における簡略化されたデータフローを示す図である。

【図5B】データ統合システムによって実行され得る従来のデータ統合処理における簡略化されたデータフローを示す図である。

【図6A】本発明の実施形態に従った、データ統合システムによって実行され得る次世代データ統合処理における簡略化されたデータフローを示す図である。

【図6B】本発明の実施形態に従った、データ統合システムによって実行され得る次世代データ統合処理における簡略化されたデータフローを示す図である。

【図7】本発明に従った一実施形態におけるODIスタジオとデータ統合システムのリポジトリとの間の相互作用の簡略ブロック図である。

【図8】本発明の実施形態に従った、データ統合シナリオを作成するための方法のフローチャートを示す図である。

【図9】本発明の実施形態に従った、データ統合シナリオを作成するためのユーザインターフェイスのスクリーンショットの図である。

【図10】本発明の実施形態に従った、マッピングを作成するための方法のフローチャートを示す図である。

【図11】本発明の実施形態に従った、データ統合シナリオにおいてマッピング情報を提供するためのユーザインターフェイスのスクリーンショットの図である。

【図12】本発明の実施形態に従った、データ統合シナリオにおいてフロー情報を提供するためのユーザインターフェイスのスクリーンショットの図である。

【図13】本発明の実施形態に従った、パッケージを作成するための方法のフローチャートを示す図である。

【図14】本発明の実施形態に従った、データ統合シナリオにおいてパッケージシーケンス情報を提供するためのユーザインターフェイスのスクリーンショットの図である。

【図15】本発明の実施形態に従った、データ統合シナリオを展開するための方法のフローチャートを示す図である。

【図16A】本発明に従った一実施形態におけるマッピングの簡略ブロック図である。

【図16B】本発明に従った一実施形態におけるマッピングの簡略ブロック図である。

【図 1 7】本発明の実施形態に従った、コンポーネントタイプに基づいてコンポーネント属性を導き出すための方法のフローチャートを示す図である。

【図 1 8】本発明に従った一実施形態においてフィルタコンポーネントを有する図 1 6 A および図 1 6 B のマッピングの簡略ブロック図である。

【図 1 9】本発明に従った一実施形態におけるマッピングの簡略ブロック図である。

【図 2 0】本発明の実施形態に従った、コンポーネントを生成するための方法のフローチャートを示す図である。

【図 2 1】本発明の実施形態を実施するために使用され得るコンピュータシステムの簡略ブロック図である。

【図 2 2】実施形態に従ったデータマッピングの生成を促進するためのシステムの簡略ブロック図である。

10

【発明を実施するための形態】

【0021】

発明の詳細な説明

イントロダクション

さまざまな実施形態において、ユーザはデータ統合システムによって、プラットフォームおよび技術に依存しない論理設計を作成し得る。ユーザは、ユーザがデータをソースとターゲットとの間でどのように流したいかをハイレベルで規定する論理設計を作成し得る。ユーザのインフラストラクチャを考慮して、ツールが論理設計を分析し、物理的設計を作成し得る。論理設計は、設計における各ソースおよびターゲットならびにジョインまたはフィルタのようなオペレーションおよびアクセスポイントに対応する複数のコンポーネントを含み得る。物理的設計に転送された際の各コンポーネントは、データに対してオペレーションを行なうようコードを生成する。存在する技術（たとえばSQLサーバ、オラクル、Hadoopなど）およびに使用される言語（SQL、pigなど）に依存して、各コンポーネントによって生成されるコードは異なり得る。

20

【0022】

1つの局面において、データ統合システムのユーザは、始めから終わりまで、論理設計における各コンポーネントにて、すべてのデータ属性を特定する必要はない。データ統合システムは、論理設計を通して流れる情報を完全に宣言する必要性を回避する、プロジェクトおよびセレクトタイプのような複数のコンポーネントタイプを提供する。データ統合システムは、所定のコンポーネントタイプによって表わされるオペレーションにて何の属性が必要とされるか決定し得る。これは、設計およびメンテナンスの両方を簡素化する。

30

【0023】

図 1 は、この開示において発見されるイノベーション、実施形態および/または例のいずれかの実施形態を組み込み得るか、または、実施形態に組み込まれ得るシステム 100 の簡略図である。図 1 は、本発明を組み込む実施形態を単に例示しており、請求の範囲において記載される本発明の範囲を限定しない。当業者は、他の変形例、修正例および代替例を認識するであろう。

【0024】

一実施形態において、システム 100 は 1 つ以上のユーザコンピュータ 110（たとえばコンピュータ 110 A、110 B および 110 C）を含む。ユーザコンピュータ 110 は、（任意の適切な種類のマイクロソフト社の Windows（登録商標）および/またはアップル社の Macintosh（登録商標）オペレーティングシステムを実行するパーソナルコンピュータおよび/もしくはラップトップコンピュータを単に例示として含む）汎用のパーソナルコンピュータ、ならびに/または、さまざまな商業的に利用可能な UNIX（登録商標）もしくはUNIXライクなオペレーティングシステムのいずれかを実行するワークステーションコンピュータであり得る。これらのユーザコンピュータ 110 はさらに、本発明の方法を行なうように構成される 1 つ以上のアプリケーションと、1 つ以上のオフィスアプリケーション、データベースクライアントおよび/またはサーバアプリケーションならびにウェブブラウザアプリケーションとを含むさまざまなアプリケーシ

40

50

ョンのうちのいずれかを有し得る。

【 0 0 2 5 】

代替的には、ユーザコンピュータ 1 1 0 は、ネットワーク（たとえば以下に記載される通信ネットワーク 1 2 0）を介して通信することができ、ならびに／または、ウェブページもしくは他のタイプの電子文書を表示およびナビゲートすることができるシンククライアントコンピュータ、インターネットが有効化された携帯電話、および／または携帯情報端末といった任意の他の電子デバイスであり得る。例示的なシステム 1 0 0 は 3 つのユーザコンピュータを有するよう示されているが、任意の数のユーザコンピュータまたはデバイスがサポートされ得る。

【 0 0 2 6 】

本発明のある実施形態は、通信ネットワーク 1 2 0 を含む得るネットワーク化された環境において動作する。通信ネットワーク 1 2 0 は、TCP/IP、SNA、IPX、および AppleTalk などを含むがこれらに限定されないさまざまな商業的に利用可能なネットワークプロトコルのいずれかをを用いるデータ通信をサポートし得る、当業者が精通している任意のタイプのネットワークであり得る。単に例示として、通信ネットワーク 1 2 0 は、イーサネット（登録商標）ネットワークおよび／もしくはトークンリングネットワークなどを含むがこれらに限定されないローカルエリアネットワーク（「LAN」）、ワイドエリアネットワーク、仮想プライベートネットワーク（VPN：virtual private network）を含むがこれに限定されない仮想ネットワーク、インターネット、イントラネット、エクストラネット、公衆交換電話ネットワーク（「PSTN：public switched telephone network」）、赤外線ネットワーク、IEEE 802.11 スイートのプロトコル、当該技術において公知である Bluetooth（登録商標）プロトコル、および／もしくは任意の他の無線プロトコルのいずれかの下で動作するネットワークを含むがこれに限定されないワイヤレスネットワーク、ならびに／または、これらおよび／もしくは他のネットワークの任意の組合せであり得る。

【 0 0 2 7 】

本発明の実施形態は、1 つ以上のサーバコンピュータ 1 3 0（たとえばコンピュータ 1 3 0 A および 1 3 0 B）を含む得る。サーバコンピュータ 1 3 0 の各々は、上で論じられたオペレーティングシステムのいずれかを含むがこれらに限定されないオペレーティングシステムと、任意の商業的に利用可能なサーバオペレーティングシステムとを有するよう構成され得る。サーバコンピュータ 1 3 0 の各々はさらに、1 つ以上のクライアント（たとえばユーザコンピュータ 1 1 0）および／または他のサーバ（たとえばサーバコンピュータ 1 3 0）にサービスを提供するように構成され得る 1 つ以上のアプリケーションを実行し得る。

【 0 0 2 8 】

単に例示として、サーバコンピュータ 1 3 0 の 1 つは、単に例示としてウェブページまたは他の電子文書についてユーザコンピュータ 1 1 0 からの要求を処理するよう使用され得るウェブサーバであり得る。ウェブサーバはさらに、HTTPサーバ、FTPサーバ、CGIサーバ、データベースサーバ、および Java（登録商標）サーバなどを含むさまざまなサーバアプリケーションを実行し得る。本発明のいくつかの実施形態において、ウェブサーバは、本発明の方法を実行するよう、ユーザコンピュータ 1 1 0 の 1 つ以上の上でウェブブラウザ内で動作され得るウェブページを取り扱うように構成され得る。

【 0 0 2 9 】

いくつかの実施形態において、サーバコンピュータ 1 3 0 は、ユーザコンピュータ 1 1 0 および／または他のサーバコンピュータ 1 3 0 の 1 つ以上の上で実行される、クライアントによってアクセス可能な 1 つ以上のアプリケーションを含む得る 1 つ以上のファイルおよび／またはアプリケーションサーバを含む得る。単に例示として、サーバコンピュータ 1 3 0 の 1 つ以上は、ユーザコンピュータ 1 1 0 および／または他のサーバコンピュータ 1 3 0 に応答して、（いくつかの場合において本発明の方法を行なうように構成され得る）ウェブアプリケーションを含むがこれらに限定されないプログラムまたはスクリプト

10

20

30

40

50

を実行可能な１つ以上の汎用コンピュータであり得る。

【００３０】

単に例示として、ウェブアプリケーションは、Java、CもしくはC++のような任意のプログラミング言語、および／または、Perl、Python、もしくはTCLのような任意のスクリプト言語、および、任意のプログラミング／スクリプト言語の組合せで記述された１つ以上のスクリプトまたはプログラムとして実現され得る。アプリケーションサーバはさらに、オラクル、マイクロソフト、およびIBMなどから商業的に利用可能なものを含むがこれらに限定されないデータベースサーバを含み得、当該データベースサーバは、ユーザコンピュータ１１０の１つおよび／またはサーバコンピュータ１３０の別の１つの上で実行されるデータベースクライアントからの要求を処理し得る。

10

【００３１】

いくつかの実施形態において、アプリケーションサーバは、本発明の実施形態に従って情報を表示するために動的にウェブページを作成し得る。アプリケーションサーバによって提供されるデータは、（たとえばHTML、XML、JavaScript、AJAXなどを含む）ウェブページとしてフォーマットされ得るか、および／または、（たとえば上述したような）ウェブサーバを介してユーザコンピュータ１１０の１つへ転送され得る。同様に、ウェブサーバは、ユーザコンピュータ１１０の１つからウェブページ要求および／もしくは入力データを受け取り得、ならびに／または、アプリケーションサーバへウェブページ要求および／もしくは入力データを転送する。

【００３２】

20

さらに別の実施形態に従うと、サーバコンピュータ１３０の１つ以上は、ファイルサーバとして機能し得、ならびに／または、ユーザコンピュータ１１０の１つおよび／もしくはサーバコンピュータ１３０の別の１つ上で実行されるアプリケーションによって組み込まれる、本発明の方法を実現するのに必要なファイルの１つ以上を含み得る。代替的には、当業者が理解するように、ファイルサーバは、すべての必要なファイルを含み得、ユーザコンピュータ１１０および／またはサーバコンピュータ１３０の１つ以上によってそのようなアプリケーションが遠隔から呼び出されることを可能にする。なお、本願明細書におけるさまざまなサーバ（たとえばアプリケーションサーバ、データベースサーバ、ウェブサーバ、ファイルサーバなど）について記載される機能は、インプリメンテーションに特有のニーズおよびパラメータに依存して、単一のサーバおよび／または複数の特殊化されるサーバによって実行され得る。

30

【００３３】

ある実施形態において、システム１００は、１つ以上のデータベース１４０（たとえばデータベース１４０Ａおよび１４０Ｂ）を含み得る。データベース１４０の位置は任意である。すなわち、単に例示として、データベース１４０Ａは、サーバコンピュータ１３０Ａ（および／またはユーザコンピュータ１１０の１つ以上）に対してローカルな（および／または存在している）記憶媒体上に存在し得る。代替的には、データベース１４０Ｂは、（たとえば通信ネットワーク１２０を介して）ユーザコンピュータ１１０およびサーバコンピュータ１３０の１つ以上と通信し得る限り、ユーザコンピュータ１１０およびサーバコンピュータ１３０のいずれかまたはすべてからリモートであり得る。実施形態の特定の集合では、データベース１４０は、当業者が精通しているストレージエリアネットワーク（SAN）に存在し得る。（同様に、ユーザコンピュータ１１０およびサーバコンピュータ１３０に起因する機能を実行するための任意の必要なファイルが適切のように、それぞれのコンピュータ上にローカルにおよび／またはリモートに格納され得る。）実施形態の１つの集合において、データベース１４０の１つ以上は、SQLフォーマットのコマンドに回答してデータを格納、更新および抽出するよう適合されるリレーショナルデータベースであり得る。たとえば、データベース１４０は、上述したように、データベースサーバによって制御および／または維持され得る。

40

【００３４】

データ統合の概略

50

図 2 は、本発明のある実施形態に従ったデータ統合システム 200 の簡略ブロック図である。図 2 は、この開示において示される 1 つ以上の発明のさまざまな実施形態または実現例を組み込み得るデータ統合システム 200 の簡略図である。図 2 は、本願明細書において開示される発明の実施形態または実現例を単に例示しているだけであり、請求の範囲に記載されるような任意の発明の範囲を限定するべきでない。当業者は、この開示および本願明細書において示される教示を通じて、図において示される実施形態または実現例に対する他の変形例、修正例および / または代替例を認識し得る。

【0035】

この実施形態において、データ統合システム 200 は、情報ソース 202、情報統合部 204、および情報宛先部 206 を含む。一般に、情報ソース 202 から情報統合部 204 に情報が流れ、これにより、情報は、情報宛先部 206 によって、消費され、利用可能になり、または別の態様で使用され得る。データフローは一方向または双方向であり得る。いくつかの実施形態において、1 つ以上のデータフローがデータ統合システム 200 に存在し得る。

【0036】

情報ソース 202 は、データを提供するように構成される 1 つ以上のハードウェアおよび / またはソフトウェア要素を示す。情報ソース 202 は、データへの直接または間接のアクセスを提供し得る。この実施形態において、情報ソース 202 は、1 つ以上のアプリケーション 208 と 1 つ以上のリポジトリ 210 とを含む。

【0037】

アプリケーション 208 は、デスクトップアプリケーション、ホストされたアプリケーション、ウェブベースアプリケーション、またはクラウドベースアプリケーションのような従来のアプリケーションを示す。アプリケーション 208 は、1 つ以上の所定の目的のために、データを受け取り、処理し、かつ維持するように構成され得る。アプリケーション 208 のいくつかの例は、カスタマーリレーションシップマネジメント (CRM: customer relationship management) アプリケーション、金融サービスアプリケーション、管理およびリスクコンプライアンスアプリケーション、人的資本マネジメント (HCM: human capital management) 調達アプリケーション、サプライチェーンマネジメントアプリケーション、または、プロジェクトもしくはポートフォリオマネジメントアプリケーションなどを含む。アプリケーション 208 は、当該技術において公知のようなさまざまな人間が読むことが可能でありマシンが読み取り可能なフォーマットでアプリケーションデータを操作およびエクスポートするために構成される機能を含み得る。アプリケーション 208 はさらに、リポジトリ 210 におけるデータにアクセスするとともに、リポジトリ 210 にデータを格納し得る。

【0038】

リポジトリ 210 は、データへのアクセスを提供するように構成されるハードウェアおよび / またはソフトウェア要素を示す。リポジトリ 210 は、データの論理的および / または物理的なパーティショニングを提供し得る。リポジトリ 210 は、リポーティングおよびデータ分析をさらに提供し得る。リポジトリ 210 のいくつかの例は、データベース、データウェアハウス、またはクラウドストレージなどを含む。リポジトリは、1 つ以上のアプリケーション 208 からのデータを統合することにより作成される中央レポジトリを含み得る。リポジトリ 210 に格納されたデータは、オペレーショナルシステムからアップロードされ得る。データは、ソースにおいて利用可能になる前に付加的なオペレーションを介して渡され得る。

【0039】

情報統合部 204 は、データ統合サービスを提供するように構成される 1 つ以上のハードウェアおよび / またはソフトウェア要素を示す。直接または間接のデータ統合サービスが情報統合部 204 において提供され得る。この実施形態において、情報統合部 204 は、データ移行部 212、データウェアハウジング部 214、マスターデータマネジメント部 216、データ同期部 218、連結部 220、およびリアルタイムメッセージ部 22

10

20

30

40

50

2を含む。情報統合部204は、データ統合機能を提供する1つ以上のモジュール、サービス、または、ここで示された要素以外の付加的な要素を含み得ることが理解されるであろう。

【0040】

データ移行部212は、データ移行を提供するように構成される1つ以上のハードウェアおよび/またはソフトウェア要素を示す。一般に、データ移行部212は、ストレージタイプ、フォーマット、またはシステム同士の間でデータを転送するための1つ以上のプロセスを提供する。データ移行部212は通常、移行を達成するようマニュアルまたはプログラムオプションを提供する。データ移行プロセスにおいて、1つのシステム上のデータまたは1つのシステムによって提供されるデータは、データ抽出およびデータロードのための設計を提供する別のシステムにマッピングされる。データ移行は、1つ以上のフェーズを含み得、当該1つ以上のフェーズとしてはたとえば、第1のシステムのデータ形式を第2のシステムのフォーマットおよび要件に関連させる1つ以上の設計が作成される設計フェーズと、データが第1のシステムから読み出されるデータ抽出フェーズと、データクリーニングフェーズと、データが第2のシステムに書き込まれるデータローディングフェーズとがある。いくつかの実施形態において、データ移行は、データが上記のフェーズのうちのいずれかにおいて正確に処理されるかどうか判定するよう、データ検証フェーズを含み得る。

【0041】

データウェアハウジング部214は、リポーティングおよびデータ分析のために使用されるデータベースを提供するように構成される1つ以上のハードウェアおよび/またはソフトウェア要素を示す。データウェアハウスは典型的に、1つ以上の異種のソースからのデータを統合することにより作成されるデータの中央レポジトリとして閲覧される。データウェアハウジング部214は、現在のデータの格納と履歴データの格納とを含み得る。データウェアハウジング部214は、典型的な抽出、変換、ロード（ETL：extract, transform, load）ベースのデータウェアハウスを含み得、これにより、ステージング層、データ統合層およびアクセス層が主な機能を収容する。一例において、ステージング層またはステージングデータベースは、1つ以上の異なるソースデータシステムの各々から抽出される生データを格納する。統合層は、ステージング層からのデータを変換し、オペレーショナルデータストア（ODS：operational data store）データベースにこの変換されたデータをしばしば格納することによって、異なるデータセットを統合する。その後、統合データは、しばしばデータウェアハウスデータベースと称されるさらに別のデータベースに移される。当該データは、（しばしば次元と称される）階層グループと、ファクト（fact）およびアグリゲートファクト（aggregate fact）とへ配され得る。アクセス層は、ユーザまたは他のシステムがデータを抽出するのを補助するために提供され得る。データウェアハウスはデータマートへ細分され得、これにより、各データマートはウェアハウスからのデータのサブセットを格納する。いくつかの実施形態において、データウェアハウジング部214は、ビジネスインテリジェンスツールと、データを抽出、変換およびリポジトリへロードするツールと、メタデータを管理および抽出するツールとを含み得る。

【0042】

マスターデータマネジメント部216は、データのマスターコピーを管理するように構成される1つ以上のハードウェアおよび/またはソフトウェア要素を示す。マスターデータマネジメント部216は、一貫してマスターデータを規定および管理するプロセス、ガバナンス、ポリシー、スタンダードおよびツールのセットを含み得る。マスターデータマネジメント部216は、マスターデータの信頼すべきソースを作成するために、重複を除去し、データを標準化し、ルールを組み込んでシステムに誤ったデータが入ることを除去するための機能を含み得る。マスターデータマネジメント部216は、データを収集、集合、マッチング、統合、品質保証、持続、組織の全体にわたって配分するためのプロセスを提供し得、これにより、情報の進行中のメンテナンスおよびアプリケーション使用において一貫性および制御を保証する。

【 0 0 4 3 】

データ同期部 2 1 8 は、データを同期させるように構成される 1 つ以上のハードウェアおよび / またはソフトウェア要素を示す。データ同期部 2 1 8 は、ソースからターゲットおよびその逆のデータの間の一貫性を確立することを提供し得る。データ同期部 2 1 8 はさらに、時間にわたるデータの連続的なハーモナイゼーションを提供し得る。

【 0 0 4 4 】

連結部 2 2 0 は、構成要素ソースからのデータの閲覧を統合するように構成される 1 つ以上のハードウェアおよび / またはソフトウェア要素を示す。連結部 2 2 0 は、複数の自律データベースシステムを単一の連結データベースへとトランスピラントにマッピングし得る。構成要素データベースは、コンピュータネットワークを介して相互に接続されてもよく、地理的に分散されてもよい。連結部 2 2 0 は、いくつかの異なるデータベースをマージすることに対する代替案を提供する。連結データベースまたは仮想データベースはたとえば、すべての構成要素データベースの合成を提供し得る。連結部 2 2 0 は、異なる構成要素データベースにおいて実際のデータ統合を提供しなくてもよく、閲覧においてのみ提供してもよい。

【 0 0 4 5 】

連結部 2 2 0 は、構成要素データベースが異種混合であっても、単一のクエリーによって、ユーザおよびクライアントが複数の非連続のデータベースにおいてデータを格納および抽出することを可能にする均一なユーザインターフェイスを提供する機能を含み得る。連結部 2 2 0 は、関連する構成要素データソースへの提出のためのサブクエリーにクエリーを分解するとともにサブクエリーの結果セットを合成する機能を含み得る。連結部 2 2 0 は、サブクエリーへの 1 つ以上のラッパー (wrapper) を、それらを適切なクエリー言語に変換するよう含み得る。いくつかの実施形態において、連結部 2 2 0 は、エクスポートスキーマおよびアクセスオペレーションの発行を通じてデータを連結部の他のメンバーに利用可能にする自律コンポーネントの集合である。

【 0 0 4 6 】

リアルタイムメッセージ部 2 2 2 は、リアルタイム制約 (たとえばイベントからシステム応答までのオペレーショナルデッドライン) に従ってメッセージサービスを提供するように構成される 1 つ以上のハードウェアおよび / またはソフトウェア要素を示す。リアルタイムメッセージ部 2 2 2 は、厳密な時間制約内における動作または応答を保証する機能を含み得る。一例において、リアルタイムメッセージ部 2 2 2 には、1 つのデータベースからいくつかのオーダおよび消費者データを取得し、ファイル中に保持されるいくつかの従業員データとそれを組合せ、その後、当該統合データをマイクロソフト S Q L サーバ 2 0 0 0 データベースにロードするタスクが課され得る。オーダは、到達する際に分析される必要があるので、リアルタイムメッセージ部 2 2 2 は、可能な限りリアルタイムに近い状態でターゲットデータベースにオーダを渡し、可能な限りワークロードを小さく維持するよう新しくかつ変更されたデータのみを抽出し得る。

【 0 0 4 7 】

情報宛先部 2 0 6 は、データを格納または消費するように構成される 1 つ以上のハードウェアおよび / またはソフトウェア要素を示す。この実施形態において、情報宛先部 2 0 6 は、データへの直接または間接のアクセスを提供し得る。この実施形態において、情報宛先部 2 0 6 は、1 つ以上のアプリケーション 2 2 4 と 1 つ以上のリポジトリ 2 2 6 とを含む。

【 0 0 4 8 】

アプリケーション 2 2 4 は、デスクトップアプリケーション、ホストされたアプリケーション、ウェブベースアプリケーション、またはクラウドベースアプリケーションのような従来のアプリケーションを示す。アプリケーション 2 2 4 は、1 つ以上の所定の目的のために、データを受け取り、処理し、かつ維持するように構成され得る。アプリケーション 2 2 4 のいくつかの例は、カスタマーリレーションシップマネジメント (C R M : customer relationship management) アプリケーション、金融サービスアプリケーション、管

10

20

30

40

50

理およびリスクコンプライアンスアプリケーション、人的資本マネジメント（HCM：human capital management）調達アプリケーション、サプライチェーンマネジメントアプリケーション、または、プロジェクトもしくはポートフォリオマネジメントアプリケーションなどを含む。アプリケーション 224 は、当該技術において公知のようなさまざまな人間が読むことが可能でありマシンが読み取り可能なフォーマットでアプリケーションデータを操作およびインポートするために構成される機能を含み得る。アプリケーション 224 はさらに、リポジトリ 226 おけるデータにアクセスするとともに、リポジトリ 226 にデータを格納し得る。

【0049】

リポジトリ 226 は、データへのアクセスを提供するように構成されるハードウェアおよび/またはソフトウェア要素を示す。リポジトリ 226 は、データの論理的および/または物理的なパーティショニングを提供し得る。リポジトリ 226 は、リポーティングおよびデータ分析をさらに提供し得る。リポジトリ 226 のいくつかの例は、データベース、データウェアハウス、またはクラウドストレージなどを含む。リポジトリは、1つ以上のアプリケーション 226 からのデータを統合することにより作成される中央レポジトリを含み得る。リポジトリ 226 に格納されたデータは、情報統合部 204 を通じてアップロードまたはインポートされ得る。データは、宛先にて利用可能になる前に付加的なオペレーションを介して渡され得る。

【0050】

データ統合システム

図 3 は、本発明の実施形態に従ったデータ統合システム 200 を実現するために使用され得るハードウェア/ソフトウェアスタックの簡略ブロック図である。図 3 は、本願明細書において開示される発明の実施形態または実現例を単に例示しているだけであり、請求の範囲に記載されるような任意の発明の範囲を限定するべきでない。当業者は、この開示および本願明細書において示される教示を通じて、図において示される実施形態または実現例に対する他の変形例、修正例および/または代替例を認識し得る。この実施形態に従ったデータ統合システム 200 において発見されるコンポーネントの一例は、カリフォルニア州レッドウッドショアズのオラクル社によって提供される製品のオラクルフュージョンミドルウェア（ORACLE FUSION Middleware）ファミリーのメンバーであるオラクルデータインテグレータ（ORACLE DATA INTEGRATOR）を含んでもよい。オラクルデータインテグレータは、セットベースのデータ統合タスクを実行するために 1つ以上のデータベースを使用する Java ベースのアプリケーションである。さらにオラクルデータインテグレータは、データを抽出し、変換されたデータをウェブサービスおよびメッセージを通じて提供し、サービス指向のアーキテクチャにおいてイベントに応答および作成する統合プロセスを作成し得る。オラクルデータインテグレータは、従来の ETL [抽出 - 変換 - ロード（extract-transform-load）] アーキテクチャではなく、ELT [抽出 - ロードおよび変換（extract-Load and Transform）] アーキテクチャに基づく。オラクルデータインテグレータのためのユーザマニュアルのコピーが、この開示に添付されており、本願明細書においてすべての目的のために参照により援用される。

【0051】

さまざまな実施形態において、データ統合システム 200 は、データ変換および統合プロセスの規定への新しい宣言型設計アプローチ（declarative design approach）を提供し、より速くより簡易な開発およびメンテナンスが得られる。したがって、データ統合システム 200 は、インプリメンテーションの詳細から宣言型ルールを分離する。データ統合システム 200 はさらに、データ変換および検証プロセスの実行のためのユニークな ELT アーキテクチャ（抽出 - ロード変換（Extract-Load Transform））を提供する。実施形態におけるこのアーキテクチャは、スタンドアロンの ETL サーバおよびプロプラエタリエンジンの必要性を除去する。その代わりに、いくつかの実施形態において、データ統合システム 200 は、RDBMS エンジンの固有のパワーをレバレッジする。

【0052】

いくつかの実施形態において、データ統合システム200は、オラクルフュージョンミドルウェアプラットフォームのような1つ以上のミドルウェアソフトウェアパッケージに統合し、ミドルウェアスタックのコンポーネントになる。図3に示されるように、データ統合システム200は、Java EEアプリケーションとしてランタイムコンポーネントを提供し得る。

【0053】

この例において、データ統合システム200の1つのコンポーネントはリポジトリ302である。リポジトリ302は、ITインフラストラクチャに関する構成情報、すべてのアプリケーションのメタデータ、プロジェクト、シナリオおよび実行ログを格納するように構成されるハードウェアおよび/またはソフトウェア要素を示す。いくつかの局面において、たとえばデベロップメント、QA、ユーザ、アクセプタンスおよびプロダクションといった、リポジトリ302の複数のインスタンスは、ITインフラストラクチャにおいて共存し得る。リポジトリ302は、メタデータおよびシナリオを交換するいくつかの分離された環境を可能にするように構成される（たとえば、デベロップメント環境、テスト環境、メンテナンス環境、およびプロダクション環境）。リポジトリ302はさらに、オブジェクトがアーカイブされバージョン番号を割り当てられるバージョン制御システムとして動作するように構成される。

【0054】

この例において、リポジトリ302は、少なくとも1つのマスターリポジトリ304および1つ以上のワークリポジトリ306から形成される。データ統合システム200内での使用のために開発または構成されるオブジェクトは、これらのリポジトリタイプの1つに格納され得る。一般に、マスターリポジトリ304は、ユーザ、プロファイルおよび権限を含むセキュリティ情報と、技術、サーバ定義、スキーマ、コンテキスト、および言語などを含むトポロジー情報と、またバージョン付けおよびアーカイブされたオブジェクトといった情報を格納する。1つ以上のワークリポジトリ306は、実際の開発されたオブジェクトを含み得る。

【0055】

いくつかのワークリポジトリは、（たとえば別個の環境を有するか、または、特定のバージョンングライフサイクルにマッチするよう）データ統合システム200において共存し得る。1つ以上のワークリポジトリ306は、スキーマ定義、データストア構造およびメタデータ、フィールドおよびカラム定義、データ品質制約、相互参照、ならびにデータリネージなどを含む、モデルについての情報を格納する。1つ以上のワークリポジトリ306はさらに、ビジネスルール、パッケージ、プロシージャ、フォルダ、ナレッジモジュール、および変数などを含むプロジェクトと、シナリオ、スケジューリング情報およびログを含むシナリオ実行とを格納し得る。いくつかの局面において、1つ以上のワークリポジトリ306は、（典型的にプロダクション目的のために）実行情報のみを含み得、実行リポジトリとして指定され得る。

【0056】

さまざまな実施形態において、リポジトリ302は、1つ以上のETLプロジェクトを格納する。ETLプロジェクトは、ソースまたはターゲットにおけるデータのデータ属性をモデリングする1つ以上のデータモデルを規定またはそうでなければ特定する。ETLプロジェクトはさらに、データを移動および変換するために、データ品質制御と、マッピングを規定することとを提供する。データの完全性制御はデータの全体的な一貫性を保証する。アプリケーションデータは、特定のソースまたはターゲットによって課された制約および宣言型ルールに必ずしも有効ではない。たとえば、オーダが顧客を伴わないことが分かった、または、オーダラインが製品を伴わないことが分かったなどである。データ統合システム200は、これらの制約違反を検出し、再利用または報告目的のためにそれらを格納するよう作業環境を提供する。

【0057】

データ統合システム200のいくつかの実施形態において、スタティック制御およびフ

10

20

30

40

50

ロー制御という2つの異なるタイプの制御が存在する。スタティック制御は、アプリケーションデータの完全性を検証するために使用されるルールの存在を示す。これらのルール（制約と称される）のうちのいくつかは、（一次キー、基準制約などを使用して）データサーバにおいて既に実現されている場合がある。データ統合システム200は、付加的な制約の定義およびチェックを、ソースにおいてそれらを直接的に宣言することなく可能にする。フロー制御は、自身の宣言型ルールを実現する変換および統合プロセスのターゲットに関する。フロー制御は、ターゲットにデータをロードする前に、これらの制約に従ってアプリケーションの入力データを検証する。フロー制御プロシージャは一般にマッピングと称される。

【0058】

ETLプロジェクトは、ランタイム環境における実行のために展開され得るパッケージへと自動化され得る。したがって、パッケージにおける異なるステップ（マッピングおよびプロシージャなど）の実行をシーケンス処理するとともに、これらのステップの各々について既存のコードを含んでいるプロダクションシナリオを作り出すことによって、データ統合フローの自動化が達成される。パッケージは典型的に、実行ダイアグラムへ組織されるステップのシーケンスから構成される。パッケージは、プロダクションのためのシナリオを生成するよう使用されるメインオブジェクトである。それらは、データ統合ワークフローを表わしており、たとえば、データストアまたはモデルに対するリバースエンジニアリングプロセスを開始し、アドミニストレータに電子メールを送信し、ファイルをダウンロードし、それを解凍し、マッピングが実行されなければならない順序を規定し、変化

【0059】

シナリオは、プロダクションにソースコンポーネント（マッピング、パッケージ、プロシージャ、変数）を配置するよう設計される。シナリオは、このコンポーネントについてコード（SQLおよびシェルなど）の生成により得られる。生成されると、ソースコンポーネントのコードが凍結され、シナリオがワークリポジトリ306の1つ以上のようなリポジトリ302の内部に格納される。シナリオは、エクスポートされ、次に、異なるプロダクション環境にインポートすることができる。

【0060】

さまざまな実施形態において、データ統合システム200は、Javaグラフィカルモジュールおよびスケジューリングエージェントによってアクセスされるモジュールの態様でリポジトリ302のまわりに組織される。グラフィカルモジュールは、リポジトリ302に格納される1つ以上の統合プロセスを設計および構築するよう使用され得る。アドミニストレータ、デベロッパおよびオペレータは、リポジトリ302にアクセスするためにデベロップメントスタジオを使用し得る。エージェントは、リポジトリ302に格納される統合プロセスに関連付けられる統合タスクのセットをスケジューリングおよび調整するために使用され得る。たとえばランタイムにおいて、デスクトップ、ウェブサービス上に展開されるか、または、ソースと通信するエージェントは、1つ以上の統合プロセスの実行を調整する。エージェントは、マスターリポジトリ304に格納されるコードを抽出し、さまざまなソースおよびターゲットシステムに接続し、全体的なデータ統合プロセスまたはシナリオを取りまとめ得る。

【0061】

この実施形態において、データ統合システム200は、上で論じたグラフィカルモジュールおよび/またはエージェントの1つ以上を含み得るデスクトップ308を含む。デスクトップ308は、パーソナルコンピュータ、ラップトップ、ネットブック、およびタブレットなどのような1つ以上のデスクトップまたはワークステーションコンピューティングデバイスを示す。デスクトップ308はJava仮想マシン(JVM)310およびオラクルデータインテグレータ(ODI: Oracle Data Integrator)スタジオ312を含む。Java仮想マシン(JVM)310は、Javaバイトコードを実行し得る仮想マシンである。JVM310はほとんどの場合、既存のオペレーティングシステム上で実行す

るよう実現されるが、ハードウェア上で直接的に実行するよう実現され得る。JVM 310は、Java バイトコードが実行され得るランタイム環境を提供し、ランタイムウェブサービス (WS) 314 およびエージェント 316 のような機能を可能にする。JVM 310は、Java クラスライブラリと、Java アプリケーションプログラミングインターフェイス (API: application programming interface) を実現する (Java バイトコードでの) スタンダードクラスライブラリのセットと、Java ランタイム環境 (JRE: Java Runtime Environment) を形成する他の要素とを含み得る。

【0062】

エージェント 316 は、リポジトリ 302 に格納された 1 つ以上の統合プロセスに関連付けられる統合タスクのセットをスケジューリングおよび調整するように構成される。たとえば、ランタイムにおいて、エージェントは、統合プロセスの実行を調整する。エージェントは、マスターリポジトリ 304 に格納されるコードを抽出し、さまざまなソースおよびターゲットシステムに接続し、全体的なデータ統合プロセスまたはシナリオを取りまとめ得る。

【0063】

図3を再び参照して、ODI スタジオ 312 はデータ統合プロジェクトを設計するよう構成されるハードウェアおよび/またはソフトウェア要素を含む。この例において、ODI スタジオ 312 は、データ統合プロジェクトを作成および管理するために使用される 4 つのグラフィカルモジュールまたはナビゲータ、すなわち、デザイナーモジュール 318、オペレータモジュール 320、トポロジモジュール 322 およびセキュリティモジュール 324 を含む。デザイナーモジュール 318 は、データストア (テーブル、ファイルおよびウェブサービスなど)、データマッピング、およびパッケージ (マッピングを含む統合ステップのセット) を規定するように構成されるモジュールである。さまざまな実施形態において、デザイナーモジュール 318 は、データ変換およびデータ完全性について宣言型ルールを規定する。したがって、プロジェクト開発メントがデザイナーモジュール 318 において発生する。さらに、デザイナーモジュール 318 において、データベースおよびアプリケーションメタデータがインポートおよび規定される。デザイナーモジュール 318 は、一実施形態において、メタデータおよびルールを使用して、プロダクションについてデータ統合シナリオまたはロードプランを生成する。一般に、デザイナーモジュール 318 は、データ完全性チェックを設計するように使用されるとともに、たとえば既存のアプリケーションまたはデータベースの自動リバースエンジニアリング、変換および統合マッピングのグラフィカル開発メントおよびメンテナンス、マッピングにおけるデータフローの可視化、自動ドキュメンテーション生成、および生成されたコードのカスタマイゼーションのような変換を構築するように使用される。

【0064】

オペレータモジュール 320 は、プロダクション統合ジョブを閲覧および管理するように構成されるモジュールである。したがって、オペレータモジュール 320 は、プロダクションにおけるデータ統合プロセスを管理および監視し、エラーカウントを有する実行ログ、処理されたロウの数、実行統計、および実行される実際のコードなどを示し得る。設計時において、デベロッパはさらに、デザイナーモジュール 318 に関連して、デバッグ目的のためにオペレータモジュール 320 を使用し得る。

【0065】

トポロジモジュール 322 は、データソースおよびエージェントへの接続を作成および管理するように構成されるモジュールである。トポロジモジュール 322 は、インフラストラクチャの物理的および論理的アーキテクチャを規定する。インフラストラクチャまたはプロジェクトアドミニストレータは、トポロジモジュール 322 を通じて、サーバと、データベーススキーマおよびカタログと、エージェントとをマスターリポジトリに登録し得る。セキュリティモジュール 324 は、ユーザおよびそれらのリポジトリ権限を管理するように構成されるモジュールである。

【0066】

10

20

30

40

50

一般に、ユーザまたはプロセスは、ソースおよびターゲット 3 2 6 について 1 つ以上のデータ統合プロセスを有するデータ統合プロジェクトを作成するよう、デザイナーモジュール 3 1 8 と相互作用する。各データ統合プロセスは、少なくとも 1 つのデータ統合タスクを含む。いくつかの実施形態において、データ統合タスクは、データのどのビットが変換され他のビットと組み合わせられるかと、当該データが実際にどのように抽出およびロードされるかの技術的な詳細とを示すビジネスルールのセットによって規定される。好ましい実施形態において、データ統合タスクは、データマッピングを構築するために宣言型のアプローチを使用して特定される。マッピングは、ターゲットと称される、1 つのデータストアにポピュレートするオブジェクトであり、1 つ以上の他のデータストアからのデータはソースとして既知である。一般に、ソースデータストアにおけるカラムは、マッピングを通じてターゲットデータストアにおけるカラムにリンクされる。マッピングは、パッケージステップとしてパッケージに加えられる。上で論じたように、パッケージはデータ統合ジョブを規定する。パッケージは、プロジェクトの下に作成され、各々がマッピングまたはプロシージャであり得るステップの組織されたシーケンスから構成される。パッケージは、1 つの入口点および複数の出口点を有し得る。

【 0 0 6 7 】

いくつかの実施形態において、新しいマッピングを作成する場合、デベロッパまたは技術的なビジネスユーザは、どのデータが統合されるかと、どのビジネスルールを使用すべきかとをまず規定するよう、デザイナー 3 1 8 と相互作用する。たとえば、デベロッパは、どのテーブルが連結されるべきか、どのフィルタが適用されるべきか、データを変換するためにどの SQL エクспレッションが使用されるべきかを特定し得る。使用される SQL の特定の言語は、コードが実行されるべきデータベースプラットフォームによって決定される。その後、別個のステップにおいて、技術スタッフは、デザイナー 3 1 8 と相互に作用して、このデータを抽出、組合せ、次いで統合するのに最も効率的な方法を選ぶ。たとえば、技術スタッフは、インクリメンタルロード (incremental load)、バルクローディングユーティリティ (bulk-loading utility)、スローリー・チェンジング・ディメンション (slowly changing dimension)、および変更データキャプチャ (changed-data capture) といった、データベースに特有のツールおよび設計技術を使用し得る。

【 0 0 6 8 】

この実施形態において、マッピングはソースおよびターゲット 3 2 6 について作成され得る。ソースおよびターゲット 3 2 6 は、1 つ以上のレガシーアプリケーション 3 2 8 と、1 つ以上のファイル / XML 文書 3 3 0 と、1 つ以上のアプリケーション 3 3 2 と、1 つ以上のデータウェアハウス (DW) と、ビジネスインテリジェンス (BI) ツールおよびアプリケーションと、エンタープライズプロセスマネージメント (EPM : enterprise process management) ツールおよびアプリケーション 3 3 4 と、(ランタイムウェブサービス 3 4 0 およびエージェント 3 4 2 を含む) 1 つ以上の JVM 3 3 6 とを含み得る。

【 0 0 6 9 】

図 4 は、本発明のさまざまな実施形態においてデータ統合シナリオが作成され得るさまざまな異種混合のデータソースを有する環境 4 0 0 のブロック図である。この例において、環境 4 0 0 は ODI スタジオ 3 1 2 およびリポジトリ 3 0 2 を含む。リポジトリ 3 0 2 は、統合シナリオ 4 0 0 を生成するのに必要とされるメタデータのすべてを含む。ユーザまたはプロセスは、データ完全性制御 4 0 2 および宣言型ルール 4 0 4 を使用して統合シナリオ 4 0 0 を作成するよう ODI スタジオ 3 1 2 と相互作用する。

【 0 0 7 0 】

オーダアプリケーション 4 0 6 は、顧客のオーダをトラッキングするためのアプリケーションを示す。「オーダアプリケーション」データモデルは、オーダアプリケーション 4 0 6 に格納されたデータと、任意のデータ完全性制御または条件とを表わすよう作成される。たとえば、「オーダアプリケーション」データモデルは、ハイパーストラクチャードクエリラングエッジデータベース (HSQLDB : Hyper Structured Query Language Database) に基づき得、SRC__CITY、SRC__CUSTOMER、SRC__ORDE

10

20

30

40

50

RS、SRC__ORDER__LINES、SRC__PRODUCTおよびSRC__REGIONという5つのデータストアを含む。

【0071】

パラメータファイル408は、販売員のリストおよび年齢のセグメンテーションから年齢の範囲を含むプロダクションシステムから発行されるフラットファイル（たとえばASCII）を示す。この例において、「パラメータ」データモデルが、フラットファイルにおけるデータを表わすために作成される。たとえば、「パラメータ」データモデルは、ファイルインターフェイスに基づき得、SRC__SALES__PERSONおよびSRC__AGE__GROUPという2つのデータストアを含み得る。

【0072】

セールスアドミニストレーションアプリケーション410はセールスをトラッキングするためのアプリケーションを示す。セールスアドミニストレーションアプリケーション410は、オーダアプリケーション406およびパラメータファイル408からのデータの変換でポピュレートされたデータウェアハウスであり得る。「セールスアドミニストレーション」データモデルは、セールスアドミニストレーションアプリケーション410に格納されるデータと、任意のデータ完全性制御または条件または変換とを表わすよう作成される。たとえば、「セールスアドミニストレーション」データモデルはハイパーストラクチャードクエリラングエッジデータベース（HSQLDB）マッピングに基づき得、TRG__CITY、TRG__COUNTRY、TRG__CUSTOMER、TRG__PRODUCT、TRG__PROD__FAMILY、TRG__REGION、およびTRG__SALEという6つのデータストアを含み得る。

【0073】

図5Aおよび図5Bは、データ統合システム200によって実行され得る従来のデータ統合処理における簡素化されたデータフローを示す。この例において、オーダアプリケーション406、パラメータファイル408、1つ以上の他の随意または付加的なソースからのデータは、セールスアドミニストレーションアプリケーション410にターゲットとされる従来のETLプロセスを通じて流れる。データ変換は別個のETLサーバ500において発生する。シナリオは専用のリソースまたはプロプラエタリリソースを必要とし、パフォーマンスがより貧弱になり、高コストを引き起こす。

【0074】

図6Aおよび図6Bは、本発明の実施形態に従った、データ統合システム200によって実行され得る次世代データ統合処理における簡素化されたデータフローを示す。この例において、オーダアプリケーション406、パラメータファイル408、1つ以上の他の随意または付加的なソースからのデータは、セールスアドミニストレーションアプリケーション410にターゲットとされるE-LTプロセスを通じて流れる。データ変換は既存のリソースをレバレッジし、パフォーマンスおよび効率がより高くなる。上述したように、先のETLシステムは、データ変換を実行するために専用および/またはプロプラエタリインフラストラクチャを必要とした。これは、部分には、未知のユーザインフラストラクチャに対応するためになされた。たとえば、どのタイプのデータベースが使用されているか知ることがなければ、先行のETLシステムは、どの変換オペレーションが所与のシステムにおいて利用可能であるか予想することができなかった。しかしながらこれは、如何なる専用および/またはプロプラエタリインフラストラクチャなしで適切なデータ変換を実行することができるユーザの既存データベースおよびサーバのようなリソースが十分に利用されないことになる。

【0075】

実施形態に従うと、本発明は、ユーザの特定のニーズに従ってユーザがデータ統合プロセスをカスタマイズすることを可能にすることによってユーザの既存のインフラストラクチャをレバレッジする。たとえば、データ統合プランが設計されると、データ統合プランは、実行単位と称される、単一のシステムによって実行可能である個別の部分に分割され得る。ひとたびデータ統合プランが複数の実行単位に分割されると、ユーザのインフラ

10

20

30

40

50

トラクチャおよびシステムリソースに基づいて、ユーザには物理的なプランが提示され得る。このプランはさらにユーザによって、どのユーザシステムがどの実行単位を実行するかを変更するようカスタマイズされ得る。たとえば、ジョインオペレーションが第1のデータベース上で実行されるプランがユーザには提示され得、ユーザは、第2のデータベースにジョインオペレーションを移動させることによって当該プランをカスタマイズし得る。

【0076】

図6Bに示されるように、これにより、先行のE T Lシステムを特徴付けたスタンドアロン変換サーバに依存しない抽出 - ロード - 変換 (E - L T : extract-load-transform) アーキテクチャが得られる。代わりに、上述したように、データ変換はユーザの既存のインフラストラクチャ上で実行され得る。E - L T アーキテクチャは、プロプラエタリ変換サーバを取得および維持することに関連付けられるコストを低減しつつユーザにさらに大きなフレキシビリティを提供する。

【0077】

図3を再び参照すると、エージェントは統合プロセスに関連付けられる統合タスクのセットをスケジューリングおよび調整するために使用され得る。たとえば、ランタイムにおいて、エージェントは、統合プロセスの実行を調整する。エージェントは、マスターリポジトリ304に格納されるコードを抽出し得、さまざまなソースおよびターゲットシステムに接続し得、全体的なデータ統合プロセスまたはシナリオを取りまとめる。さまざまな実施形態において、2つのタイプのエージェントが存在する。一例において、スタンドアロンのエージェントがエージェント316のようなデスクトップ308上にインストールされる。別の例において、アプリケーションサーバエージェントは (オラクルWebLog icサーバ上で展開されるJava EEエージェントのように) アプリケーションサーバ326上で展開され得、高いアベイラビリティ要件についてのクラスタリングのようなアプリケーションサーバレイヤーフィーチャから利益を得ることができる。さらに別の例において、エージェントは、エージェント342のように、ソースおよびターゲット326上で展開され得る。

【0078】

この実施形態において、データ統合システム200は、上で論じられるエージェントの1つ以上を含み得るアプリケーションサーバ344を含む。アプリケーションサーバ344は、1つ以上のアプリケーションサーバ、ウェブサーバ、またはホストされたアプリケーションを示す。この例において、アプリケーションサーバ344は、FMWコンソール346、サブレットコンテナ348、ウェブサービスコンテナ350、およびデータソース接続プール352を含む。

【0079】

FMWコンソール346は、サブレットコンテナ348、ウェブサービスコンテナ350およびデータソース接続プール334に係る情報のような、アプリケーションサーバ344の局面を管理するように構成される1つ以上のハードウェアおよび/またはソフトウェア要素を示す。たとえば、FMWコンソール346は、オラクルWebLog icサーバドメインを管理するために使用されるブラウザベースのグラフィカルユーザインターフェイスであり得る。FMWコンソール346は、WebLog icサーバインスタンスを構成、開始、停止し、WebLog icサーバクラスタを構成し、データベース接続性 (JDBC) およびメッセージング (JMS) のようなWebLog icサーバサービスを構成し、ユーザ、グループおよび役割を作成および管理することを含むセキュリティパラメータを構成し、Java EEアプリケーションを構成および展開し、サーバおよびアプリケーションのパフォーマンスを監視し、サーバおよびドメインのログファイルを閲覧し、アプリケーション展開記述子を閲覧し、選択されたランタイムアプリケーション展開記述子要素を編集する機能を含み得る。いくつかの実施形態において、FMWコンソール346は、プロダクションにおけるデータ統合プロセスへのアクセスをFMWコンソール346に提供するODIプラグイン354を含み、エラーカウントを有する実行ログ

、処理されたロウの数、実行統計、および実行される実際のコードなどを示し得る。

【0080】

サーブレットコンテナ348は、アプリケーションサーバ344の性能を拡張するように構成される1つ以上のハードウェアおよび/またはソフトウェア要素を示す。サーブレットは、HTMLフォームから提出されたデータを処理するまたは格納するためにほとんどの場合に使用され、データベースクエリの結果のような動的なコンテンツを提供し、適切な顧客のショッピングカートへ品物を充填するといった、ステートレスHTTPプロトコルに存在しない状態情報を管理する。サーブレットは典型的に、Javaクラスが要求に応答し得るプロトコルである、JavaサーブレットAPIに準拠するJavaEEにおけるJavaクラスである。サーブレットを展開および実行するために、サーブレットコンテナ348は、サーブレットと相互作用するウェブサーバのコンポーネントとして使用される。したがって、サーブレットコンテナ348は、ウェブサービスコンテナ350のパブリックウェブサービス356およびデータサービス358によって提供される機能と、データソース接続プール352によって提供されるデータプールへのアクセスとを拡張し得る。サーブレットコンテナ348はさらに、サーブレットのライフサイクルを管理し、特定のサーブレットにURLをマッピングし、URLリクエストが正しいアクセス権を有することを保証することを担う。

10

【0081】

この例において、サーブレットコンテナ348は、ODI SDK 362に関連付けられるJavaEEアプリケーション360と、ODIコンソール364と、JavaEEエージェント368に関連付けられるランタイムウェブサービス366とを含む。ODI SDK 362は、データ統合およびETL設計のためのソフトウェア開発キット(SDK)を提供する。ODI SDK 362は、一般的でありかつ非常に反復的である作業の自動化を可能にし、ユーザが反復のタスクをスクリプトにすることを可能にする。

20

【0082】

ODIコンソール364は、リポジトリ302へのウェブアクセスを提供するJavaエンタープライズエディション(Java EE: Java Enterprise Edition)アプリケーションである。ODIコンソール364は、プロジェクト、モデルおよび実行ログを含むDesign-Timeオブジェクトをユーザがブラウズすることを可能にするように構成される。ODIコンソール364は、ユーザがフローマップを閲覧し、すべてのデータのソースを追跡し、データを構築するのに使用される変換を理解するためにフィールドレベルにさらにドリルダウンすることを可能にし得る。さらに、エンドユーザは、ODIコンソール364を通じてシナリオ実行を開始および監視し得る。1つの局面において、ODIコンソール364は、データサーバ、物理および論理スキーマのようなトポロジーオブジェクトを閲覧および編集し、かつ、リポジトリ302を管理する能力をアドミニストレータに提供する。

30

【0083】

データシナリオ設計および開発

上で論じたように、シナリオは、ソースコンポーネント(マッピング、パッケージ、プロシージャ、変数)をプロダクションに配置するよう設計される。このコンポーネントについて、シナリオがコード(SQL、シェルなど)の発生から得られる。シナリオは、エクスポートされ、その後、異なるプロダクション環境へインポートされ得る。

40

【0084】

図7は、本発明に従った一実施形態におけるODIスタジオとデータ統合システムのリポジトリとの間の相互作用の簡略ブロック図である。図7に示される実施形態において、図3のODIスタジオ312は、プロダクションについてデータ統合シナリオ700を生成するようメタデータおよびルールを使用する。一般に、デザイナーモジュール318は、データ完全性チェックを設計するように使用されるとともに、たとえば既存のアプリケーションまたはデータベースの自動リバースエンジニアリング、変換および統合マッピングのグラフィカルデベロップメントおよびメンテナンス、マッピングにおけるデータフロ

50

ーの可視化、自動ドキュメンテーション生成、および生成されたコードのカスタマイゼーションのような変換を構築するように使用される。

【 0 0 8 5 】

図 8 は、本発明の実施形態に従った、データ統合シナリオを作成するための方法 8 0 0 のフローチャートを示す。図 8 に示された方法 8 0 0 の実現または処理は、コンピュータシステムまたは情報処理デバイスのようなロジックマシンの中央処理装置（CPU またはプロセッサ）によって実行される際にソフトウェア（たとえば命令またはコードモジュール）によって実行され得るか、電子デバイスまたは特定用途向け集積回路のハードウェアコンポーネントによって実行され得るか、または、ソフトウェアおよびハードウェア要素の組合せによって実行され得る。図 8 に示される方法 8 0 0 は、ステップ 8 1 0 において開始する。

10

【 0 0 8 6 】

さまざまな実施形態において、ユーザは O D I スタジオ 3 1 2 のデザイナーモジュール 3 1 8 とのセッションを開始し、リポジトリ 3 0 2 へ接続し得る。ユーザは 1 つ以上のユーザインターフェイスフィーチャと相互作用して、新しいデータ統合プロジェクトを作成するか、または、たとえばマスターリポジトリ 3 0 4 に格納される既存のデータ統合プロジェクトから選択し得る。一般に、デザイナーモジュール 3 1 8 は、メタデータを管理し、データ完全性チェックを設計し、変換を構築するように使用される。さまざまな実施形態において、デザイナーモジュール 3 1 8 を通じて取り扱われる主なオブジェクトはモデルおよびプロジェクトである。データモデルは、データソースまたはターゲットにおけるメタデータのすべてを含む（たとえばテーブル、カラム、制約、記述、相互参照など）。プロジェクトは、ソースまたはターゲット（たとえばマッピング、プロシージャ、変数など）についてロードおよび変換ルールのすべてを含む。

20

【 0 0 8 7 】

ステップ 8 2 0 において、1 つ以上のデータモデルが作成される。ステップ 8 3 0 において、1 つ以上のプロジェクトが作成される。図 9 は、本発明の実施形態に従った、データ統合シナリオを作成するためのユーザインターフェイス 9 0 0 のスクリーンショットである。この例において、ナビゲーションパネル 9 1 0 は、情報を表示し、プロジェクトとの相互作用のための機能を含む。ナビゲーションパネル 9 2 0 は情報を表示し、データモデルとの相互作用のための機能を含む。上で論じたように、ユーザはデータモデルを作成するだけでなく、データモデルにおけるデータについて任意のデータ完全性チェックを開発し得る。さらに、ユーザは、ソースからのデータをターゲットにロードするフローにおいてデータについてのデータ完全性および変換を提供する、プロジェクトについてのマッピングおよびプロシージャ、変数を特定し得る。ステップ 8 4 0 では、1 つ以上のデータ統合シナリオが生成される。図 8 はステップ 8 5 0 で終了する。

30

【 0 0 8 8 】

図 1 0 は、本発明の実施形態に従った、マッピングを作成するための方法 1 0 0 0 のフローチャートを示す。図 1 0 に示された方法 1 0 0 0 の実現または処理は、コンピュータシステムまたは情報処理デバイスのようなロジックマシンの中央処理装置（CPU またはプロセッサ）によって実行される際にソフトウェア（たとえば命令またはコードモジュール）によって実行され得るか、電子デバイスまたは特定用途向け集積回路のハードウェアコンポーネントによって実行され得るか、または、ソフトウェアおよびハードウェア要素の組合せによって実行され得る。図 1 0 に示される方法 1 0 0 0 は、ステップ 1 0 1 0 において開始する。

40

【 0 0 8 9 】

ステップ 1 0 2 0 において、ターゲットデータストア情報が受け取られる。たとえば、ユーザは、ターゲットデータストア情報を提供するよう、デザイナーモジュール 3 1 8 の 1 つ以上のユーザインターフェイスフィーチャと相互作用し得る。一実施形態において、ユーザは、選択されたデータモデルおよび任意の関連付けられる変換またはデータ完全性チェックの局面を視覚的に表わすマッピングまたはフローパネル上にナビゲーションパネ

50

ル 9 1 0 から、1 つ以上のデータモデルを含むターゲットデータストア情報をドラッグアンドドロップし得る。

【 0 0 9 0 】

ステップ 1 0 3 0 では、ソースデータストア情報が受け取られる。たとえば、ユーザは、ソースデータストア情報を提供するように、デザイナーモジュール 3 1 8 の 1 つ以上のユーザインターフェイスフィーチャと相互作用し得る。一実施形態において、ユーザは、選択されたデータモデルおよび任意の関連付けられる変換またはデータ完全性チェックの局面を視覚的に表わすターゲットデータストア情報の同じマッピングまたはフローパネル上にナビゲーションパネル 9 1 0 から、1 つ以上のデータモデルを含むソースデータストア情報をドラッグアンドドロップし得る。

10

【 0 0 9 1 】

さまざまな実施形態において、ソースデータストア情報およびターゲットデータストア情報は、1 つ以上のデータモデルおよび随意にオペレーションから形成され得る。オペレーションのいくつかの例は 1 つ以上のデータセットオペレーション（たとえばユニオン、ジョイン、インターセクションなど）、データ変換、データフィルタオペレーション、制約、記述、相互参照、または完全性チェックなどを含み得る。さらに別の実施形態において、これらのオペレーションのうちのいくつかは、あらかじめ構成されるとともに、デザイナーモジュール 3 1 8 において視覚的に表わされ得る。他の実施形態において、カスタムオペレーションが提供され得、オペレーションを実現するロジックおよびマッピングなどをユーザが特定することを可能にする。

20

【 0 0 9 2 】

ステップ 1 0 4 0 において、マッピング情報が受け取られる。たとえば、ユーザは、ターゲットデータストア情報にソースデータストア情報をマッピングするように、デザイナーモジュール 3 1 8 の 1 つ以上のユーザインターフェイスフィーチャと相互作用し得る。一実施形態において、ユーザは、ソースデータストア情報におけるデータ要素の属性をターゲットデータストア情報におけるデータ要素の属性に視覚的に接続し得る。これは、ソースデータストア情報およびターゲットデータストア情報におけるテーブルのカラム名をマッピングすることにより行われ得る。さらに別の実施形態において、1 つ以上の自動マッピング技術がマッピング情報を提供するために使用され得る。

30

【 0 0 9 3 】

図 1 1 は、本発明の実施形態に従った、データ統合シナリオにおけるマッピング情報を提供するためのユーザインターフェイス 1 1 0 0 のスクリーンショットである。この例において、ソースコンポーネントの属性はターゲットコンポーネントの属性にマッピングされる。

【 0 0 9 4 】

図 1 0 を再び参照して、ステップ 1 0 5 0 において、データロードストラテジーが受け取られる。データロードストラテジーは、ソースデータストア情報からの実際のデータが抽出フェーズの間にどのようにロードされるべきであるかについての情報を含む。データロードストラテジーは、デザイナー 3 1 8 のフロータブにおいて規定され得る。いくつかの実施形態において、データロードストラテジーは、マッピングの構成に依存してフローについて自動的に計算され得る。

40

【 0 0 9 5 】

たとえば、1 つ以上のナレッジモジュールが当該フローのために提案され得る。ナレッジモジュール（KM）は、異なる技術にわたって再使用可能な変換および E L T（抽出、ロード、および変換）ストラテジーを実現するコンポーネントである。1 つの局面において、ナレッジモジュール（KM）はコードテンプレートである。各 KM は、全体のデータ統合プロセスにおいて個々のタスクに専用であり得る。KM におけるコードは、ほとんど置換法で実行されるであろう形で現われ、多くの異なる統合ジョブによって一般的に使用されることを可能にする。生成および実行されるコードは、デザイナーモジュール 3 1 8 において規定される宣言型ルールおよびメタデータに由来する。この一例は、オラクルデ

50

ータベース 10g から変更データキャプチャを通じてデータを抽出し、オラクルデータベース 11g におけるパーティショニングされたファクトテーブルに変換データをロードするか、または、マイクロソフト SQL サーバデータベースからのタイムスタンプベースの抽出を作成し、このデータをテラデータエンタープライズデータウェアハウス (Teradata enterprise data warehouse) にロードすることである。

【0096】

KM の能力はそれらの再使用可能性とフレキシビリティにあり、たとえば、あるロードストラテジーが 1 つのファクトテーブルのために開発され得、その後、当該ロードストラテジーが他のすべてのファクトテーブルに適用され得る。1 つの局面において、所与の KM を使用するすべてのマッピングは、KM に対してなされる任意の変化を引き継ぐ。いくつかの実施形態において、統合ナレッジモジュール (IKM : integration knowledge module)、ロードナレッジモジュール (LKM : loading knowledge module)、およびチェックナレッジモジュール CKM (check knowledge module) といったように、5 つの異なるタイプの KM が提供され、それらの各々はソースからターゲットまで変換処理における 1 つのフェーズをカバーする。

10

【0097】

図 4 を参照して、ユーザは、環境 400 において SRC__AGE__GROUP、SRC__SALES__PERSON ファイルおよび SRC__CUSTOMER テーブルからデータを抽出する方法を規定し得る。ロードストラテジーを規定するために、ユーザは、SRC__AGE__GROUP ファイルのロードに対応するソースセットを選択し、SQL への LKM ファイルを選択し得、ファイルから SQL までのフローを実現する。1 つの局面において、LKM はリモートサーバからステージングエリアにソースデータロードすることを担う。

20

【0098】

ステップ 1060 では、データ統合ストラテジーが受け取られる。ローディングフェーズを規定した後、ユーザは、ロードされたデータのターゲットへの統合に適合すべきストラテジーを規定する。統合ストラテジーを規定するために、ユーザは、ターゲットオブジェクトを選択し、IKM SQL インクリメンタルアップデート (IKM SQL Incremental Update) を選択し得る。IKM は、最終の変換されたデータをターゲットに書き込むことを担う。IKM は、開始されると、リモートサーバのためのすべてのローディングフェーズは、たとえばすべてのリモートソースデータセットが LKM によってステージングエリアにロードされたといったように、既にそれらのタスクを行なったか、または、ソースデータストアがステージングエリアと同じデータサーバ上に存在するとみなす。

30

【0099】

ステップ 1070 では、データ制御ストラテジーが受け取られる。一般に、CKM は、データセットのレコードが規定された制約と一貫していることをチェックすることを担う。CKM は、データ完全性を維持するために使用され得、全体的なデータ品質イニシアチブに参加する。CKM は 2 つの態様で使用され得る。第 1 に、既存データの一貫性をチェックするために使用され得る。これは任意のデータストア上またはマッピング内で行われ得る。この場合、チェックされたデータは、現在データストアに存在するデータである。第 2 の場合において、ターゲットデータストアにおけるデータが、ロードされた後でチェックされる。この場合、CKM は、ターゲットへの書き込みの前に、結果得られたフロー上のターゲットデータストアの制約をシミュレートする。

40

【0100】

図 12 は、本発明の実施形態に従った、データ統合シナリオにおいてフロー情報を提供するためのユーザインターフェイス 1200 のスクリーンショットである。

【0101】

ステップ 1080 において、マッピングが生成される。図 10 はステップ 1090 において終了する。

【0102】

50

データ統合シナリオパッケージおよび展開

上で論じたように、パッケージにおいて異なるステップ（マッピングおよびプロシージャなど）の実行をシーケンス処理するとともに、これらのステップの各々について既存のコードを含んでいるプロダクションシナリオを作り出すことによって、データ統合システム200においてデータ統合フローの自動化が達成され得る。パッケージは、実行ダイアグラムへ組織されるステップのシーケンスから構成される。パッケージは、プロダクションのためのシナリオを生成するよう使用されるメインオブジェクトである。シナリオは、ソースコンポーネント（マッピング、パッケージ、プロシージャ、変数）をプロダクションに配置するよう設計される。このコンポーネントについて、シナリオがコード（SQL、シェルなど）の発生から得られる。シナリオは、エクスポートされ、その後、異なるプロダクション環境へインポートされ得る。

10

【0103】

図13は、本発明の実施形態に従った、パッケージを作成するための方法のフローチャートを示す。図13に示された方法1300の実現または処理は、コンピュータシステムまたは情報処理デバイスのようなロジックマシンの中央処理装置（CPUまたはプロセッサ）によって実行される際にソフトウェア（たとえば命令またはコードモジュール）によって実行され得るか、電子デバイスまたは特定用途向け集積回路のハードウェアコンポーネントによって実行され得るか、または、ソフトウェアおよびハードウェア要素の組合せによって実行され得る。図13に示される方法1300はステップ1310において開始する。

20

【0104】

ステップ1320において、パッケージステップ情報が受け取られる。パッケージステップ情報は、ステップ、要素、プロパティ、およびコンポーネントなどを識別する情報を含む。一例において、ユーザは、パッケージについて1つ以上のステップを作成、識別、またはそうでなければ特定するよう、デザイナーモジュール318の1つ以上のユーザインターフェイスフィーチャと相互作用し得る。一実施形態において、1つ以上のコンポーネントが選択され、図に配置される。これらのコンポーネントはパッケージにおけるステップとして現われる。

【0105】

ステップ1330において、パッケージステップシーケンス情報が受け取られる。パッケージステップシーケンス情報は、ステップについてのオーダリング、および従属性などを識別する情報を含む。ひとたびステップが作成されると、ステップがデータ処理チェーンへと順に並べられるかまたは並べ替えられる。一例において、ユーザは、パッケージの1つ以上のステップについてシーケンシングまたはオーダリングを提供するよう、デザイナーモジュール318の1つ以上のユーザインターフェイスフィーチャと相互作用し得る。データ処理チェーンは、第1のステップとして規定されたユニークステップを含み得る。一般に、各ステップは、成功または失敗のような1つ以上の終結状態を有する。失敗または成功のようないくつかの状態におけるステップの後には別のステップまたはパッケージの終了が続き得る。1つの局面において、失敗のようないくつかの状態の場合、シーケンス情報は多くの再試行を規定し得る。別の局面において、パッケージはいくつかの可能な終結ステップのみを有し得る。

30

40

【0106】

図14は、本発明の実施形態に従った、データ統合シナリオにおいてパッケージステップシーケンス情報を提供するためのユーザインターフェイスのスクリーンショットである。

【0107】

ステップ1340において、パッケージが生成される。図13はステップ1350において終了する。

【0108】

上で論じたように、データ統合フローの自動化は、パッケージにおける異なるステップ

50

(マッピングおよびプロシージャなど)の実行をシーケンス処理することにより達成され得る。次いで、パッケージのステップの各々について既存のコードを含むプロダクションシナリオのために、パッケージが作り出され得る。さまざまな実施形態において、パッケージはプロダクション環境において自動的に実行されるよう展開される。

【0109】

図15は、本発明の実施形態に従ったデータ統合シナリオを展開するための方法1500のフローチャートを示す。図15に示された方法1500の実現または処理は、コンピュータシステムまたは情報処理デバイスのようなロジックマシンの中央処理装置(CPUまたはプロセッサ)によって実行される際にソフトウェア(たとえば命令またはコードモジュール)によって実行され得るか、電子デバイスまたは特定用途向け集積回路のハードウェアコンポーネントによって実行され得るか、または、ソフトウェアおよびハードウェア要素の組合せによって実行され得る。図15に示される方法1500はステップ1510において開始する。

【0110】

ステップ1520において、統合シナリオが抽出される。一実施形態において、パッケージがリポジトリ302から抽出される。ステップ1530において、統合シナリオは1つ以上のエージェントに展開される。ステップ1540において、統合シナリオは1つ以上のエージェントによって実行される。1つの局面において、統合シナリオは、たとえばODIスタジオ312から、コマンドラインから、またはウェブサービスからといったようにいくつかの態様で実行され得る。たとえば、シナリオ実行は、上に論じられるようにオペレータモジュール320などを介して閲覧および監視され得る。図15はステップ1550において終了する。

【0111】

プロジェクトおよびセレクトタイプの使用

ほとんどのデータ統合システムにおいて、マッピングは、マップの部分形成するすべての入力および出力属性の明示的な定義を必要とする。典型的なフローベースのETLツールにおいて、コネクタが属性レベルに形成される。これにより、非常に簡潔なマッピングモデルが得られる。しかしながら、これによりさらに、巨大な数のオブジェクトが生成されるとともに、属性レベルコネクタの数によりマップを構築および維持することが煩雑になる。

【0112】

さまざまな実施形態において、データ統合システム200は、マッピングの設計およびメンテナンスを容易にするための1つ以上の技術を組み込む。コンポーネントは、すべての入力および出力属性を特定する必要なく、かつ、コンポーネントレベルコネクタがリルートされることなく、単純に既存の設計に加えられ得る。1つの局面において、アサインメントエクスプレッション(assignment expression)が上流のコンポーネントのすべてまたは部分を参照することを可能にするコンポーネントが実現される。したがって、あるタイプのコンポーネントの属性は、下流のコンポーネントに伝播され得るか、または、そうでなければ、マッピングデザイナー側に対して最小の労力で上流のコンポーネントから引き継がれる。

【0113】

1つの局面において、データ統合システム200は、属性レベル接続性を管理する必要性を最小化する。たとえば、データ統合システム200は、マッピングのコンポーネントを分類し得る(ジョイナー、セット、テーブル、フィルタなど)。カテゴリーのいくつかの例はプロジェクトおよびセレクトである。一般に、プロジェクトは、マップを通して流れるデータの形に影響するコンポーネントである。セレクトはデータのフローを制御するコンポーネントであるが、フローの形を根本的に変更しない。さまざまな実施形態では、上流のコンポーネントの属性が可視であるという点において、セレクトタイプコンポーネントはトランスピラントに構成される。プロジェクトタイプコンポーネントは、上流のコンポーネントの属性が可視ではなく、データフローの形で利用可能になったもののみ可

視であるという点で、透過ではない（opaque）であるように構成される。

【0114】

したがって、上流からのデータを閲覧することを可能にするためには、下流のコンポーネントの属性に各属性が接続されることが必要されており、さまざまな実施形態において、データ統合システム200は、ユーザが最も近いプロジェクタの属性までおよび当該属性を含む任意の上流の属性を直接的に参照することを可能にする。したがって、データ統合システム200は、マッピングの設計およびメンテナンスを非常に容易にする。データ統合システム200はさらに、既存の設計へのコンポーネントの追加、典型的には単に必要なレベルコネクタをリルートすることを簡易にする。

【0115】

図16Aおよび図16Bは、本発明に従った一実施形態におけるマッピングの簡略ブロック図である。この例において、図16Aのマッピング1600は、データソースSRC_EMPを表わすコンポーネント1610と、データソースSRC_DEPTを表わすコンポーネント1620と、データターゲットTGT_EMPDEPTを表わすコンポーネント1630とを含む。データターゲットTGT_EMPDEPTを更新するために、データソースSRC_EMPおよびSRC_DEPTについてジョインが必要とされる。JOINを表わすコンポーネント1640は、入力としてコンポーネント1610および1620に接続するとともに出力としてコンポーネント1630に接続するマッピング1600に加えられる。コンポーネント1640は（SRC_EMP.DEPTNO=SRC_DEPT.DEPTNO）といったジョインエクスプレッションを提供するように構成される。

【0116】

従来のデータ統合システムにおいて、マッピング1600は、JOINを表わすコンポーネント1640の部分形成するすべての入力および出力属性の明示的な定義を必要とする。対照的に、さまざまな実施形態において、マッピングデベロッパは、コンポーネント1640を通して流れているためコンポーネント1630に可視である、コンポーネント1610によって表わされるデータソースSRC_EMPの属性およびコンポーネント1620によって表わされるデータソースSRC_DEPTの属性からデータターゲットTGT_EMPDEPTのカラムがどのように直接的にポピュレートされるかを定義し得る。たとえば、ターゲットカラムTGT_EMPDEPT.NAMEのアサインメントは、コンポーネント1640の属性への参照の必要なく、SRC_EMP.ENAMEを参照し得る。

【0117】

図17は、本発明の実施形態に従った、コンポーネントタイプに基づいてコンポーネント属性を導き出すための方法1700のフローチャートを示す。図17に示される方法1700の実現または処理は、コンピュータシステムまたは情報処理デバイスのようなロジックマシンの中央処理装置（CPUまたはプロセッサ）によって実行される際にソフトウェア（たとえば命令またはコードモジュール）によって実行され得るか、電子デバイスまたは特定用途向け集積回路のハードウェアコンポーネントによって実行され得るか、または、ソフトウェアおよびハードウェア要素の組合せによって実行され得る。図17に示される方法1700はステップ1710において開始する。

【0118】

ステップ1720において、コンポーネントについてのコンポーネントタイプが決定される。上で論じたように、いくつかのタイプのコンポーネントは、マッピングを通して流れるデータの形に影響する一方、他のタイプのコンポーネントは、データのフローを制御するがフローの形を根本的に変更しない。ステップ1730において、下流のコンポーネントについての属性はコンポーネントタイプに基づいて決定される。たとえば、コンポーネントがデータのフローを制御するがフローの形を根本的に変更しない場合、データ統合システム200は上流のコンポーネントの属性に基づいて属性のセットを導き出し得る。ステップ1740では、導き出されたコンポーネントは、下流のコンポーネントに晒され

10

20

30

40

50

る。図 17 はステップ 1750 において終了する。したがって、ターゲットカラム TGT_EMPDEPT.NAME のアサインメントは、コンポーネント 1640 の属性への参照の必要なく、SRC_EMP.ENAME をトランスピアレントに参照し得る。

【0119】

データ統合システム 200 はさらに、既存の設計へのコンポーネントの追加、典型的には単に必要なレベルコネクタをリルートすること、を簡易にする。たとえば、フィルタコンポーネントが設計に加えられると、コンポーネントレベルコネクタの変更は、ある下流のコンポーネントの属性のアサインメントに対する変更を必要としない。

【0120】

図 18 は、本発明に従った一実施形態における、フィルタコンポーネント 1800 を有するマッピング 1600 の簡略ブロック図である。この例において、マッピング 1600 は、コンポーネント 1610 とコンポーネント 1640 との間に配置されるフィルタコンポーネント 1800 を含む。(たとえばフィルタ SRG_EMP.SAL > 3000 を使用して) マッピング 1600 へフィルタコンポーネント 1800 を加えるために、ユーザは、コンポーネント 1610 とコンポーネント 1640 との間のグラフへフィルタコンポーネント 1800 を加える必要があるだけである。コンポーネントレベルコネクタは、フィルタコンポーネント 1800 が入力としてコンポーネント 1610 に接続するとともに出力としてのコンポーネント 1640 に接続するように再リンクされる必要がある。そのような変更は、マッピング 1600 において、如何なる下流のアサインメントへの変更も必要としない。従来のフローツールでは、カラムレベルのすべては、新しいコンポーネントの導入によって再リンクされる必要がある。

【0121】

図 19 は、本発明に従った一実施形態におけるマッピング 1900 の簡略ブロック図である。この例において、マッピング 1900 は、データソースを表わすコンポーネント S1 と、データソースを表わすコンポーネント S2 と、データターゲットを表わすコンポーネント T とを含む。コンポーネント C1、C2 および C3 は、マッピングを通して流れるデータの形に影響するように構成されるか、または、データのフローを制御するがフローの形を根本的に変更しないように構成される。この例において、コンポーネント C1 はプロジェクトタイプである。したがって、コンポーネント C1 は、下流のコンポーネント(たとえばコンポーネント C3 以下)から閲覧されると、(いくつかはコンポーネント S1 の 1 つ以上の属性から宣言される) 属性の宣言されたセットを有しており、コンポーネント C3 はセレクトタイプである。したがって、コンポーネント C3 は(いくつかは、下流のコンポーネント(たとえばコンポーネント T)から閲覧されると、コンポーネント C1 および C2 の 1 つ以上の属性から導出される) 属性の導出されたセットを有する。

【0122】

図 20 は、本発明の実施形態に従った、コンポーネントを生成するための方法 2000 のフローチャートを示す。図 20 に示された方法 2000 の実現または処理は、コンピュータシステムまたは情報処理デバイスのようなロジックマシンの中央処理装置(CPU またはプロセッサ)によって実行される際にソフトウェア(たとえば命令またはコードモジュール)によって実行され得るか、電子デバイスまたは特定用途向け集積回路のハードウェアコンポーネントによって実行され得るか、または、ソフトウェアおよびハードウェア要素の組合せによって実行され得る。図 20 に示される方法 2000 はステップ 2010 において開始する。

【0123】

ステップ 2020 において、コンポーネント定義が受け取られる。たとえば、コンポーネント定義はルール、オペレーション、プロシージャ、変数、およびシーケンスなどを含み得る。ステップ 2030 において、コンポーネントタイプが受け取られる。たとえば、コンポーネントがフローにおけるデータの形を変更する場合、コンポーネントは 1 つの態様で分類され得る。コンポーネントがデータのフローを制御するがフローの形を根本的に変更しない場合、コンポーネントは別の態様で分類され得る。ステップ 2040 では、コ

ンポーメントが生成され、データ統合システム 200 によって使用され得る。図 20 はステップ 2050 において終了する。

【0124】

したがって、データ統合システム 200 はユーザがプラットフォームおよび技術に依存しない論理設計を作成することを可能にする。ユーザは、どのようにデータがソースとターゲットとの間に流れることをユーザが望むかをハイレベルで規定する論理設計を作成し得る。ユーザのインフラストラクチャを考慮して、ツールは、論理設計を分析し得、物理的設計を作成する。論理設計は、設計における各ソースおよびターゲットに対応する複数のコンポーメントと、ジョインまたはフィルタのようなオペレーションと、アクセスポイントとを含み得る。物理的設計に転送された際の各コンポーメントは、データに対してオペレーションを実行するようコードを生成する。存在する技術（たとえば SQL サーバ、オラクル、Hadoop など）と使用される言語（SQL、pig など）とに依存して、各コンポーメントによって生成されるコードは異なり得る。

10

【0125】

したがって、データ統合システムのユーザは、論理設計における各コンポーメントにて、始めから終わりまですべてのデータ属性を特定することは必要とされない。データ統合システム 200 は、論理設計を通して流れる情報を完全に宣言する必要性を回避する、プロジェクトおよびセレクトタイプのような複数のコンポーメントタイプを提供する。データ統合システム 200 は、所定のコンポーメントタイプによって表わされるオペレーションにてどの属性が必要かを決定することができる。これは設計およびメンテナンスの両方を簡素化する。

20

【0126】

結論

図 21 は、本発明の実施形態を実施するために使用され得るコンピュータシステム 2100 の簡略ブロック図である。図 21 に示されるように、コンピュータシステム 2100 は、バスサブシステム 2120 を介して多くの周辺機器と通信するプロセッサ 2110 を含む。これらの周辺機器は、メモリサブシステム 2140 およびファイルストレージサブシステム 2150 を含むストレージサブシステム 2130 と、入力デバイス 2160 と、出力デバイス 2170 と、ネットワークインターフェイスサブシステム 2180 とを含み得る。

30

【0127】

バスサブシステム 2120 は、コンピュータシステム 2100 のさまざまなコンポーメントおよびサブシステムを意図されるように互いと通信させるためのメカニズムを提供する。バスサブシステム 2120 は単一のバスとして概略的に示されるが、バスサブシステムの代替的な実施形態は複数のバスを利用してもよい。

【0128】

ストレージサブシステム 2130 は、本発明の機能を提供する基本的なプログラミングおよびデータ構造を格納するように構成され得る。本発明の機能を提供するソフトウェア（コードモジュールまたは命令）は、ストレージサブシステム 2130 に格納され得る。これらのソフトウェアモジュールまたは命令は、プロセッサ 2110 によって実行され得る。ストレージサブシステム 2130 は、さらに本発明に従って使用されるデータを格納するためのリポジトリを提供し得る。ストレージサブシステム 2130 は、メモリサブシステム 2140 とファイル/ディスクストレージサブシステム 2150 とを含み得る。

40

【0129】

メモリサブシステム 2140 は、プログラム実行の間に命令およびデータの格納のためのメインランダムアクセスメモリ（RAM）2142 と、固定された命令が格納されるリードオンリメモリ（ROM）2144 とを含む多くのメモリを含み得る。ファイルストレージサブシステム 2150 は、プログラムおよびデータファイルのための持続性（不揮発性）ストレージを提供しており、ハードディスクドライブ、関連するリムーバブル媒体を有するフロッピー（登録商標）ディスクドライブ、コンパクトディスクリードオンリメモ

50

リ(CD-ROM)ドライブ、DVD、オプティカルドライブ、リムーバブル媒体カートリッジ、および他の同様のストレージ媒体を含み得る。

【0130】

入力デバイス2160は、キーボードと、マウス、トラックボール、タッチパッドまたはグラフィックスタブレットのようなポインティングデバイスと、スキャナと、バーコードスキャナと、ディスプレイに組み込まれるタッチスクリーンと、音声認識システム、マイクロホンのような音声入力デバイスと、他のタイプの入力デバイスとを含み得る。一般に、「入力デバイス」という用語の使用は、コンピュータシステム2100に情報を入力するためのすべての可能なタイプのデバイスおよびメカニズムを含むように意図される。

【0131】

出力デバイス2170は、ディスプレイサブシステム、プリンタ、ファックスマシン、または音声出力デバイスなどのノンビジュアルディスプレイを含み得る。ディスプレイサブシステムは、陰極線管(CRT)、液晶ディスプレイ(LCD)のようなフラットパネルデバイス、または投射デバイスであり得る。一般に、「出力デバイス」という用語の使用は、コンピュータシステム2100から情報を出力するためのすべての可能なタイプのデバイスおよびメカニズムを含むように意図される。

【0132】

ネットワークインターフェイスサブシステム2180は、他のコンピュータシステム、デバイス、および通信ネットワーク2190のようなネットワークにインターフェイスを提供する。ネットワークインターフェイスサブシステム2180は、コンピュータシステム2100からデータを受け取るとともに他のシステムにデータを送信するためのインターフェイスとして機能する。通信ネットワーク2190のいくつかの例は、プライベートネットワーク、パブリックネットワーク、専用回線、インターネット、イーサネットネットワーク、トークンリングネットワーク、および光ファイバーネットワークなどである。

【0133】

コンピュータシステム2100は、パーソナルコンピュータ、ポータブルコンピュータ、ワークステーション、ネットワークコンピュータ、メインフレーム、キオスクまたは任意の他のデータ処理システムを含むさまざまなタイプのものであり得る。コンピュータおよびネットワークの絶えず変化する性質により、コンピュータシステムの好ましい実施形態を説明する目的のために、図21に示されるコンピュータシステム2100の説明は特定の例としてのみ意図される。図21に示されたシステムよりも多くまたはより少ないコンポーネントを有する多くの構成が可能である。

【0134】

図22は、実施形態に従ったデータマッピングの生成を促進するためのシステム2200の簡略ブロック図である。図22においてシステム2200に含まれるユニットは、コンピュータシステムまたは情報処理デバイスのようなロジックマシンの中央処理装置(CPUまたはプロセッサ)によって実行される際にソフトウェア(たとえば命令またはコードモジュール)において実行され得るか、電子デバイスまたは特定用途向け集積回路のハードウェアコンポーネントにおいて実行され得るか、または、ソフトウェアおよびハードウェア要素の組合せにおいて実現され得る。

【0135】

図22に示されるように、特に、データ統合システム2200は、受取部2210、判定部2220および生成部2230を含み得る。受取部2210は、論理設計の1つ以上のコンポーネントを特定する情報を受け取るように構成され得、1つ以上のコンポーネントのうちの少なくとも1つは第1のタイプである。判定部2220は、論理設計における上流のコンポーネントに基づいて、第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つの論理設計における下流のコンポーネントに可視であるデータ属性のセットを判定するように構成され得る。生成部2230は、第1のタイプである1つ以上のコンポーネントのうちの少なくとも1つの論理設計における下流のコンポーネントに可視である属性のセットを示す情報を生成するように構成され得る。

10

20

30

40

50

【 0 1 3 6 】

実施形態に従うと、第 1 のタイプである 1 つ以上のコンポーネントのうちの少なくとも 1 つの論理設計における下流のコンポーネントに可視であるデータ属性のセットを判定することは、上流のコンポーネントから可視である 1 つ以上の属性を導き出すことと、下流のコンポーネントに 1 つ以上の属性をエクスポートすることとを含み得る。

【 0 1 3 7 】

実施形態に従うと、論理設計の 1 つ以上のコンポーネントを特定する情報は、論理設計を通して流れる情報の形を変更するオペレーションを示す情報を含み得る。

【 0 1 3 8 】

実施形態に従うと、論理設計の 1 つ以上のコンポーネントを特定する情報は、論理設計を通して流れる情報のフローを制御するが論理設計を通して流れる情報の形を変更しないオペレーションを示す情報を含み得る。

10

【 0 1 3 9 】

実施形態に従うと、論理設計の 1 つ以上のコンポーネントを特定する情報は、ソースデータストアに格納されるデータの 1 つ以上の属性を有するソースコンポーネントを示す情報を含み得る。

【 0 1 4 0 】

実施形態に従うと、論理設計の 1 つ以上のコンポーネントを特定する情報は、ターゲットデータストアに格納されるデータの 1 つ以上の属性を有するターゲットコンポーネントを示す情報を含み得る。

20

【 0 1 4 1 】

実施形態に従うと、第 1 のタイプである 1 つ以上のコンポーネントのうちの少なくとも 1 つの論理設計における下流のコンポーネントに可視である属性のセットを示す情報を生成することは、下流のコンポーネントに属性のリストをエクスポートすることを含み得る。

【 0 1 4 2 】

実施形態に従うと、受取部 2 2 1 0 はさらに、論理設計へのコンポーネントまたは属性の導入または除去による論理設計における変更を受け取るように構成され得る。判定部 2 2 2 0 はさらに、論理設計における変更が、第 1 のタイプである 1 つ以上のコンポーネントのうちの少なくとも 1 つに影響するかどうかを判定するように構成され得る。論理設計における変更が、第 1 のタイプである 1 つ以上のコンポーネントのうちの少なくとも 1 つに影響するという判定に基づいて、判定部 2 2 2 0 はさらに、下流のコンポーネントに可視であるデータ属性の更新されたセットを決定するように構成され得る。

30

【 0 1 4 3 】

実施形態に従うと、データ統合システム 2 2 0 0 は保存部 2 2 4 0 をさらに含み得る。受取部 2 2 1 0 はさらに、論理設計へのコンポーネントまたは属性の導入による論理設計における変更を受け取るように構成され得る。判定部 2 2 2 0 はさらに、論理設計における変更が、第 1 のタイプである 1 つ以上のコンポーネントのうちの少なくとも 1 つに影響するかどうか判定するように構成され得る。論理設計における変更が第 1 のタイプである 1 つ以上のコンポーネントのうちの少なくとも 1 つに影響するという判定に基づいて、保存部 2 2 4 0 は下流のコンポーネントに可視であるデータ属性のセットを保存するように構成され得る。

40

【 0 1 4 4 】

実施形態に従うと、データ統合システム 2 2 0 0 はリネーム部 2 2 5 0 をさらに含み得る。受取部 2 2 1 0 はさらに、コンポーネントまたは属性をリネームする、論理設計における変更を受け取るように構成され得る。判定部 2 2 2 0 はさらに、論理設計における変更が、第 1 のタイプである 1 つ以上のコンポーネントのうちの少なくとも 1 つに影響するかどうかを判定するように構成され得る。論理設計における変更が第 1 のタイプである 1 つ以上のコンポーネントのうちの少なくとも 1 つに影響するという判定に基づいて、リネーム部 2 2 5 0 は下流のコンポーネントに可視であるデータ属性のセットをリネームする

50

ように構成され得る。

【 0 1 4 5 】

本発明の具体的な実施形態を記載してきたが、さまざまな修正例、変更例、代替的な構成、および均等例も本発明の範囲内に含まれる。記載された発明は、ある特定のデータ処理環境内のオペレーションに制限されず、複数のデータ処理環境において自由に作用する。さらに、特定の一連のトランザクションおよびステップを使用して本発明が記載されたが、本発明の範囲は記載された一連のトランザクションおよびステップに限定されるわけではないということは当業者に明らかであるはずである。

【 0 1 4 6 】

さらに、本発明をハードウェアおよびソフトウェアの特定の組合せを用いて説明したが、ハードウェアおよびソフトウェアの他の組合せも本発明の範囲内であると認識されるべきである。本発明は、ハードウェアのみで、またはソフトウェアのみで、またはその組合せを使用して実現され得る。

【 0 1 4 7 】

したがって、明細書および図面は、限定的な態様ではなく例示的な態様であるとみなされるべきである。しかしながら、添付の特許請求の範囲に記載されるより広い本発明の精神および範囲から逸脱することがなければ、追加、削減、削除、ならびに、他の修正および変更もなされてもよいということが明らかであろう。

【 0 1 4 8 】

その教示がこの開示内に示され得る 1 つ以上の発明のいずれかのさまざまな実施形態がソフトウェア、ファームウェア、ハードウェアまたはその組合せにおけるロジックの形で実現され得る。ロジックは、この開示において示された発明のさまざまな実施形態において開示され得るステップのセットを実行するために、ロジックマシンの中央処理装置（CPU またはプロセッサ）を指示するように適合される命令のセットとして、マシンアクセス可能なメモリ、マシン読み取り可能な物品、有形的なコンピュータ読取可能媒体、コンピュータ読取可能記憶媒体、または他のコンピュータ / マシン読取可能媒体に格納され得る。ロジックは、この開示に示される発明のさまざまな実施形態における方法またはプロセスを行うよう実行される際に、コードモジュールがコンピュータシステムまたは情報処理デバイスのプロセッサにより作動状態になると、ソフトウェアプログラムまたはコンピュータプログラムプロダクトの一部を形成し得る。本願明細書において提供されるこの開示および教示に基づいて、示された発明の 1 つ以上のさまざまな実施形態の開示されたオペレーションまたは機能のいずれかをソフトウェア、ファームウェア、ハードウェアまたはその組合せで実現するための他の態様、変形例、修正例、代替例および / または方法を当業者は理解するであろう。

【 0 1 4 9 】

その教示がこの開示に示され得るそれらの発明のいずれか 1 つの開示された例、実現例およびさまざまな実施形態は、当業者にこの開示の教示を妥当な明瞭さで伝えるために単に例示的である。これらの実現例および実施形態は例示的な図または特定の図を参照して記載され得る際に、記載される方法および / または特定の構造のさまざまな修正例または適合例は当業者に明らかになり得る。本願明細書において発見されるこの開示およびこれらの教示に依存し、かつ、当該教示によって技術を進歩させたすべてのそのような修正例、適合例または変形例は、その教示がこの開示内に示され得る 1 つ以上の発明の範囲内に存在すると考えられるべきである。したがって、開示内に示された発明が具体的に示される実施形態にまったく限定されないということが理解されるので、この記載および図は限定的な意味で考えられるべきでない。

【 0 1 5 0 】

したがって、上記の記載および如何なる添付の図面、説明および図は、例示的であるが限定的ではないように意図される。したがって、この開示に示される如何なる発明の範囲も、上記の記載および図に示されるそれらの実施形態を単純に参照してではなく、それらの完全な範囲または均等物とともに係属中の請求項を参照して決定されるべきである。

10

20

30

40

50

【図 1】

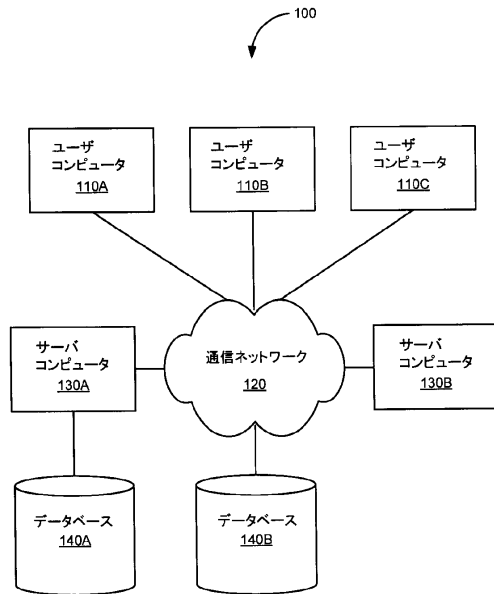


FIG. 1

【図 2】

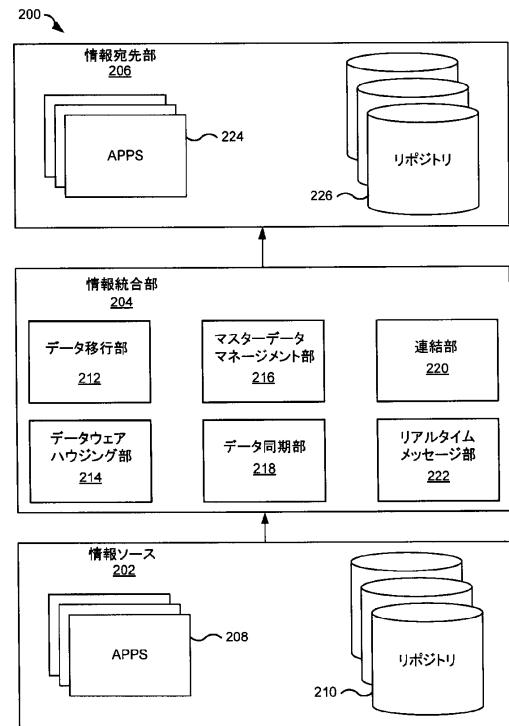


FIG. 2

【図 3】

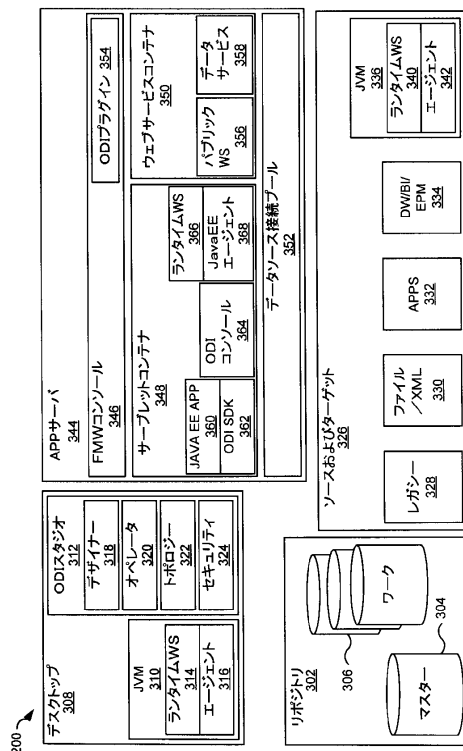


FIG. 3

【図 4】

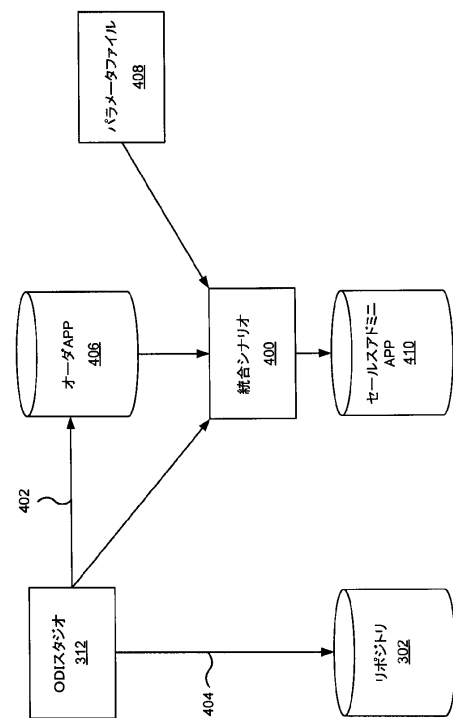


FIG. 4

【図 5 A】

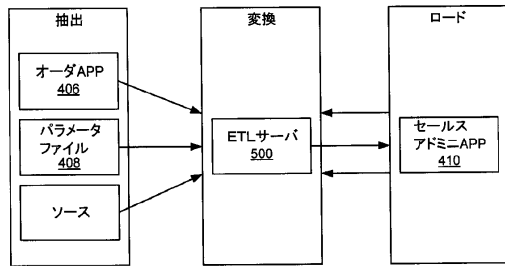


FIG. 5A

【図 6 A】

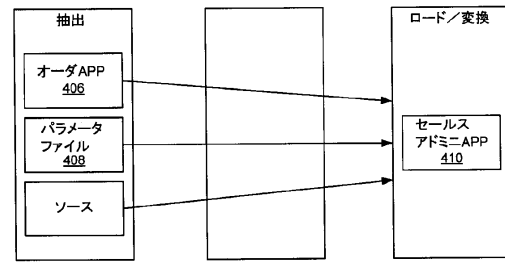


FIG. 6A

【図 5 B】

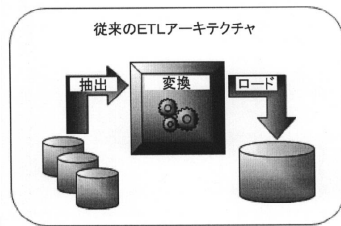


FIG. 5B

【図 6 B】

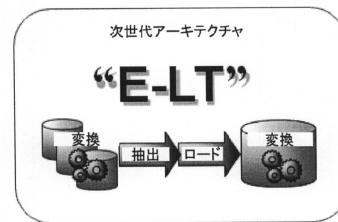


FIG. 6B

【図 7】

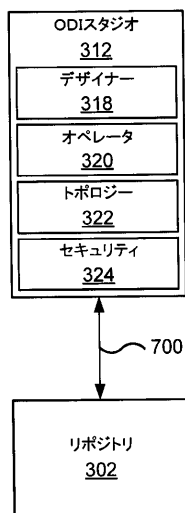


FIG. 7

【図 8】

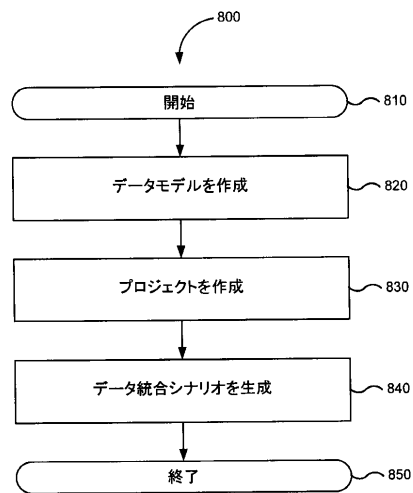


FIG. 8

【図 9】

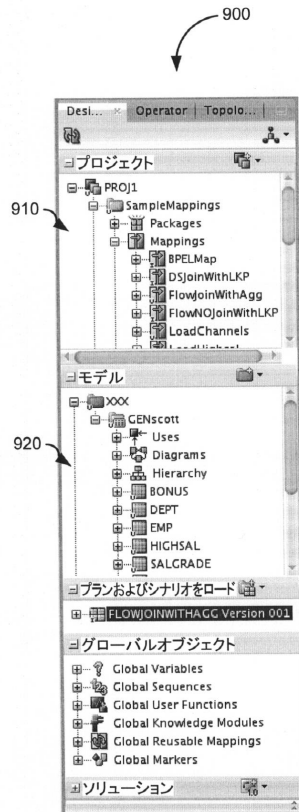


FIG. 9

【図 10】

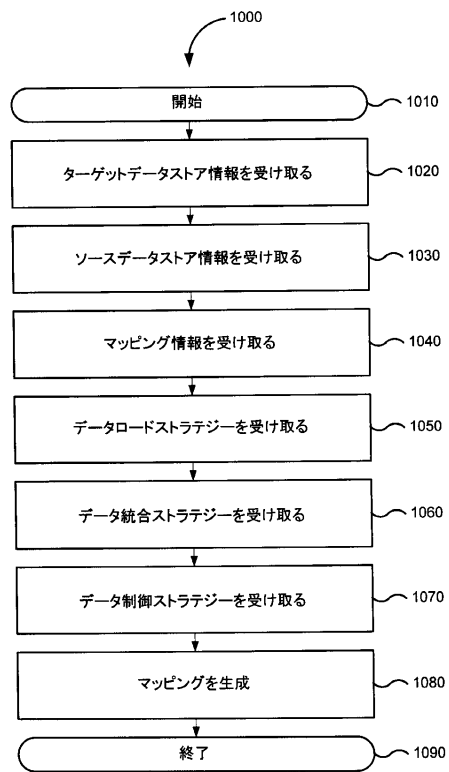


FIG. 10

【図 11】

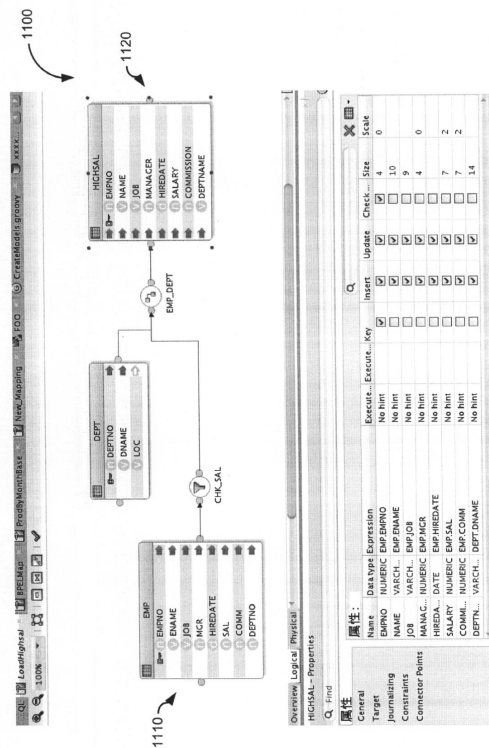


FIG. 11

【図 12】

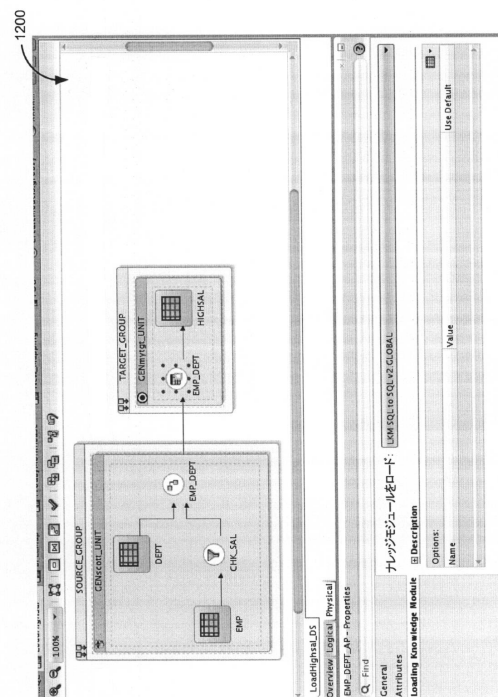


FIG. 12

【図 13】

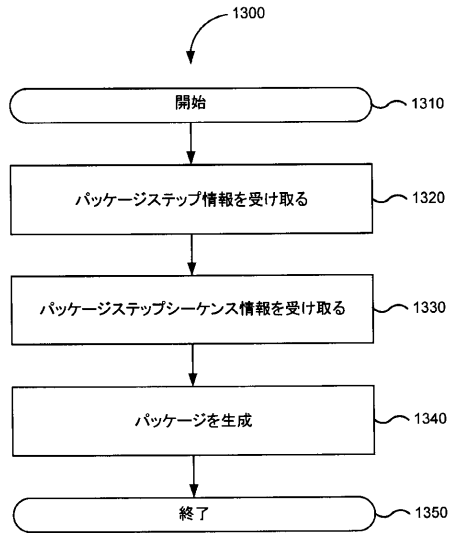


FIG. 13

【図 14】

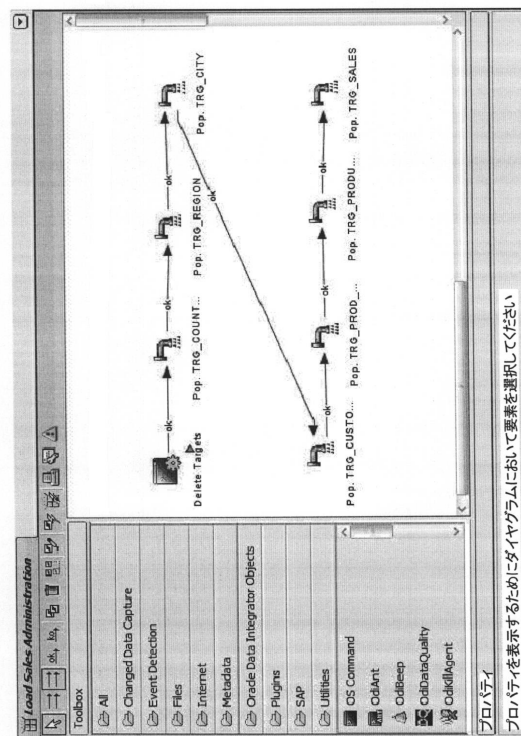


FIG. 14

【図 15】

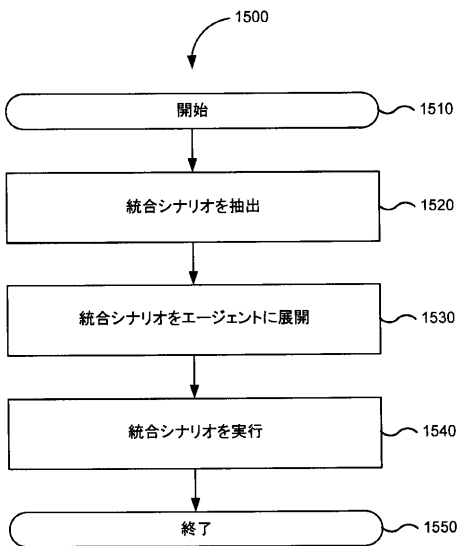


FIG. 15

【図 16 A】

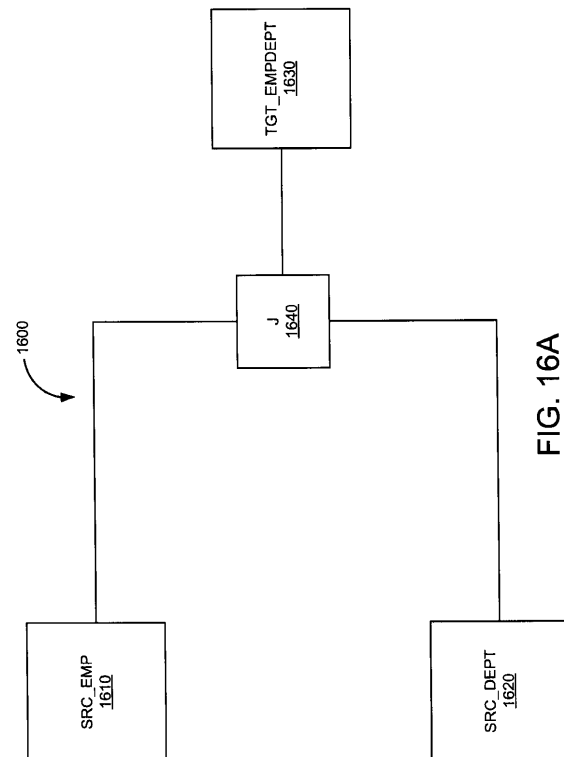


FIG. 16A

【図 16 B】

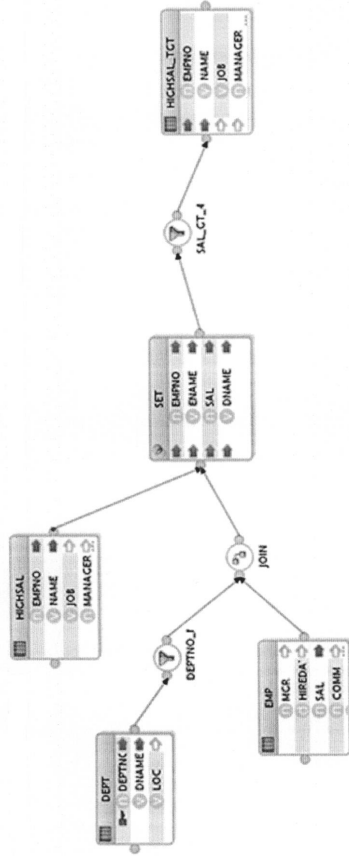


FIG. 16B

【図 17】

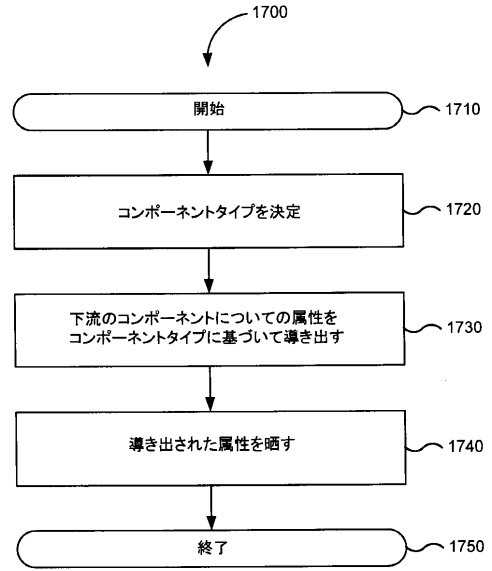


FIG. 17

【図 18】

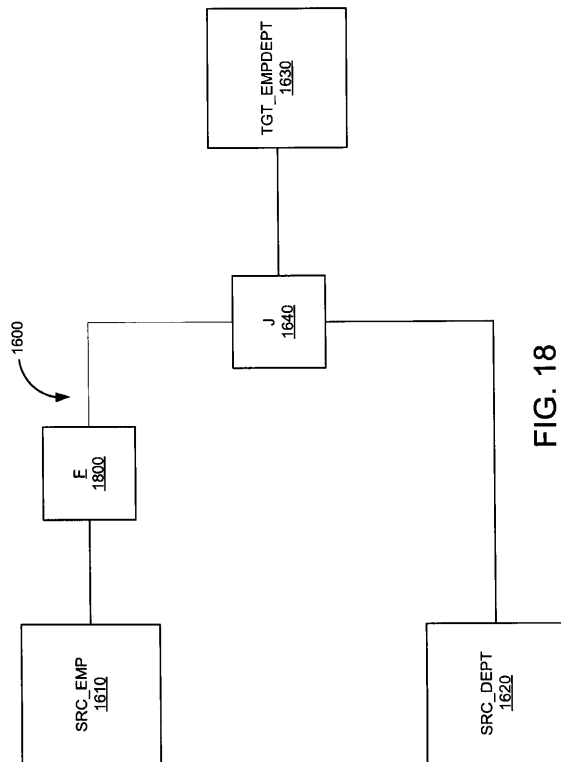


FIG. 18

【図 19】

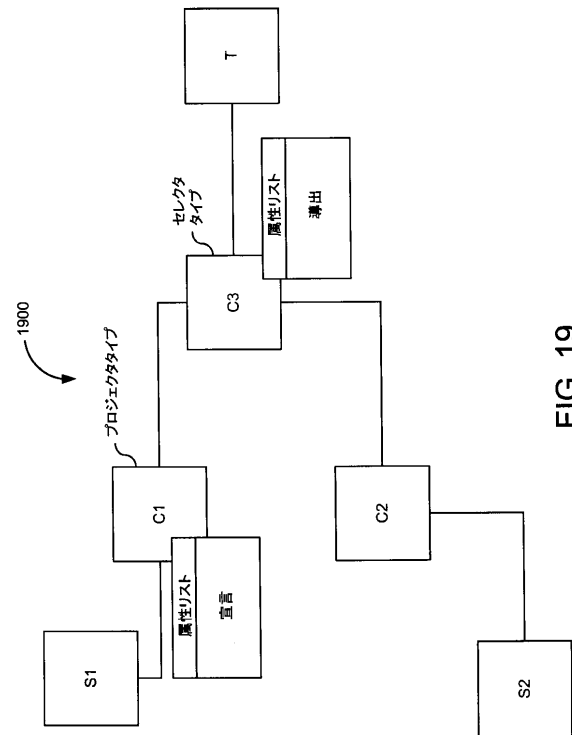


FIG. 19

【図 20】

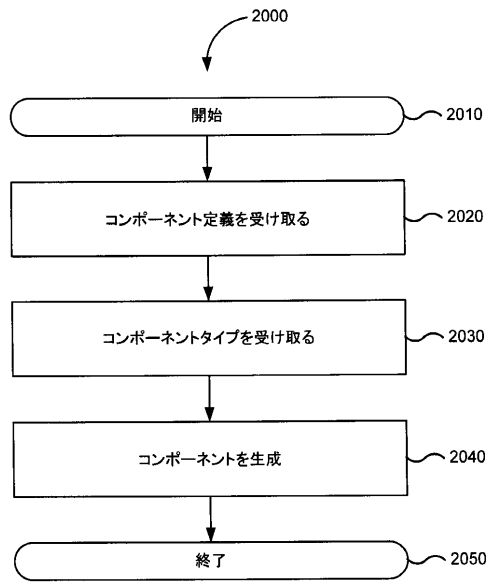


FIG. 20

【図 21】

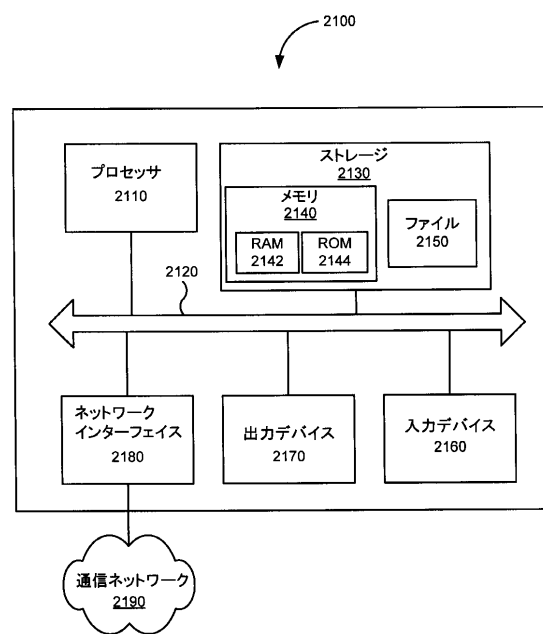


FIG. 21

【図 22】

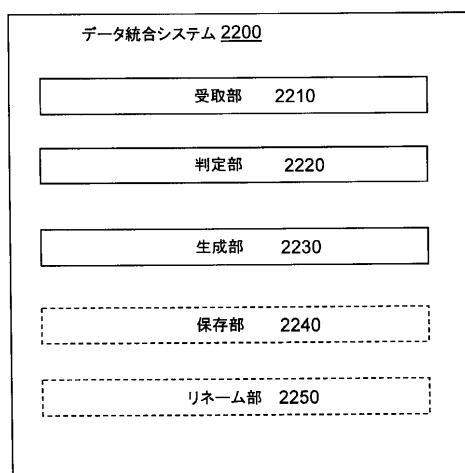


FIG. 22

フロントページの続き

- (72)発明者 アラン, デイビッド
アメリカ合衆国、9 4 0 4 0 カリフォルニア州、マウンテン・ビュー、ブラックフィールド・ウ
エイ、1 1 1 2
- (72)発明者 ラウ, クウォク・ハン・(トーマス)
アメリカ合衆国、9 4 0 2 2 カリフォルニア州、ロス・アルトス、メルセデス・アベニュー、1 0
7 0、ナンバー・2 5

審査官 大野 朋也

- (56)参考文献 米国特許出願公開第2 0 1 3 / 0 1 1 1 3 7 5 (U S , A 1)
米国特許出願公開第2 0 1 3 / 0 1 0 3 7 0 5 (U S , A 1)
特開2 0 1 1 - 0 3 4 2 4 6 (J P , A)
特開2 0 0 5 - 3 2 7 2 3 2 (J P , A)
特開2 0 1 2 - 0 2 7 6 9 0 (J P , A)
国際公開第2 0 0 7 / 0 8 3 3 7 1 (W O , A 1)
データベース技術を最大限に活用できるデータ関係基盤 ~Oracle Data Integratorのご紹介~
 , 日本オラクル株式会社 , 2 0 1 0 年1 2 月3 1 日 , p . 1 - 5 4
- (58)調査した分野(Int.Cl. , D B 名)
G 0 6 Q 1 0 / 0 0 - 9 9 / 0 0
G 0 6 F 1 2 / 0 0