(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6674904号
(P6674904)
(45)発行日　令和2年4月1日(2020.4.1)　(24)登録日　令和2年3月11日(2020.3.11)

(51)Int.Cl.　F I
G06F 9/50 (2006.01)　G06F 9/50 120A
G06F 9/48 (2006.01)　G06F 9/48 370
G06F 8/00 (2018.01)　G06F 8/00

請求項の数 37　(全 17 頁)

(21)出願番号　特願2016-567355 (P2016-567355)
(86)(22)出願日　平成27年5月22日(2015.5.22)
(65)公表番号　特表2017-522630 (P2017-522630A)
(43)公表日　平成29年8月10日(2017.8.10)
(86)国際出願番号　PCT/US2015/032193
(87)国際公開番号　WO2015/183738
(87)国際公開日　平成27年12月3日(2015.12.3)
審査請求日　平成30年3月15日(2018.3.15)
(31)優先権主張番号　62/004,406
(32)優先日　平成26年5月29日(2014.5.29)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　14/470,501
(32)優先日　平成26年8月27日(2014.8.27)
(33)優先権主張国・地域又は機関
米国(US)

(73)特許権者　509123208
アビニシオ　テクノロジー　エルエルシー
アメリカ合衆国　０２４２１　マサチューセッツ州　レキシントン　スプリング　ストリート　２０１
(74)代理人　100079108
弁理士　稲葉　良幸
(74)代理人　100109346
弁理士　大貫　敏史
(74)代理人　100117189
弁理士　江口　昭彦
(74)代理人　100134120
弁理士　内藤　和彦

最終頁に続く

(54)【発明の名称】ワークロード自動化およびデータ系統分析

(57)【特許請求の範囲】
【請求項１】
コンピュータシステムによって、複数のジョブの実行順序を示すジョブ依存性情報と、データ系統情報との間のリンクを識別することにより、前記ジョブ依存性情報と前記データ系統情報との結合に基づく依存性情報を判断することと、
前記依存性情報に基づいて、前記データ系統情報により指定されるデータソースにおける前記ジョブ依存性情報により指定されるジョブの実行予定についての変更の影響を判断することであって、
前記ジョブ依存性情報に含まれる、前記ジョブのジョブスケジューリング情報を識別するクエリを受信することと、
前記ジョブスケジューリング情報とデータ系統情報との間の前記リンクを識別することと、
前記ジョブスケジューリング情報に基づいて、前記データ系統情報により指定される前記データソースへの影響を識別することと、
によって前記変更の影響を判断することと、
を含む、コンピュータにより実施される方法。
【請求項２】
ワークロード自動化システムに関連付けられたワークロードリポジトリから、スケジューリング情報を取得することと、
前記スケジューリング情報を変換することと、

データ記憶に前記スケジューリング情報を記憶させることであって、前記データ記憶が、前記データ系統情報を記憶していることと、により、ジョブ依存性情報を取得すること、を更に含む、請求項１に記載の方法。

【請求項３】

前記データ系統情報および前記ジョブ依存性情報によって参照されるデータソースを識別することにより、リンクを生成することを更に含む、請求項１に記載の方法。

【請求項４】

前記データソースを識別することが、前記データ系統情報および前記ジョブ依存性情報内の同一の名前によって参照されるデータソースを識別することを含む、請求項３に記載の方法。

【請求項５】

前記データソースを識別することが、ユニフォームリソースロケータを使用してデータソースを識別することを含む、請求項３に記載の方法。

【請求項６】

前記データソースを識別することが、データベース、テーブルスペース、およびテーブル名を使用して、リレーショナルデータベーステーブルを識別することを含む、請求項３に記載の方法。

【請求項７】

前記ジョブ依存性情報および前記データ系統情報によって参照される実行可能プログラムを識別することにより、リンクを生成することを更に含む、請求項１に記載の方法。

【請求項８】

前記実行可能プログラムを識別することが、前記実行可能プログラムに提供されるパラメータに少なくとも部分的に基づいて、実行可能プログラムを識別することを含む、請求項７に記載の方法。

【請求項９】

前記実行可能プログラムを識別することが、永続的なデータ記憶上の前記実行可能プログラムの位置に基づいて、実行可能プログラムを識別することを含む、請求項７に記載の方法。

【請求項１０】

前記判断された影響を指定する報告を生成することを更に含む、請求項１に記載の方法。

【請求項１１】

前記実行予定についての変更により前記データソースが影響を受ける時に、１以上の警報を生成することを更に含む、請求項１に記載の方法。

【請求項１２】

前記判断された影響が、前記実行予定についての変更により前記データソースの中の特定のデータセットが不正確になりそうなことを明示する時に、１以上の警報を生成することを更に含む、請求項１に記載の方法。

【請求項１３】

１つまたは複数のコンピュータによる実行時に、前記１つまたは複数のコンピュータに動作を行わせるコンピュータプログラム命令を記憶した非一時的コンピュータ記憶媒体であって、前記動作が、

複数のジョブの実行順序を示すジョブ依存性情報と、データ系統情報との間のリンクを識別することにより、前記ジョブ依存性情報と前記データ系統情報との結合に基づく依存性情報を判断することと、

前記依存性情報に基づいて、前記データ系統情報により指定されるデータソースにおける前記ジョブ依存性情報により指定されるジョブの実行予定についての変更の影響を判断することであって、

前記ジョブ依存性情報に含まれる、前記ジョブのジョブスケジューリング情報を識別するクエリを受信することと、

前記ジョブスケジューリング情報とデータ系統情報との間の前記リンクを識別することと、
前記ジョブスケジューリング情報に基づいて、前記データ系統情報により指定される前記データソースへの影響を識別することと、
によって前記変更の影響を判断することと、
を含む、非一時的コンピュータ記憶媒体。
【請求項１４】
ワークロード自動化システムに関連付けられたワークロードリポジトリから、スケジューリング情報を取得することと、
前記スケジューリング情報を変換することと、
データ記憶に前記スケジューリング情報を記憶させることであって、前記データ記憶が、前記データ系統情報を記憶していることと、により、ジョブ依存性情報を取得することを更に含む、請求項１３に記載の媒体。
【請求項１５】
前記データ系統情報および前記ジョブ依存性情報によって参照されるデータソースを識別することにより、リンクを生成することを更に含む、請求項１３に記載の媒体。
【請求項１６】
前記データソースを識別することが、前記データ系統情報および前記ジョブ依存性情報内の同一の名前によって参照されるデータソースを識別することを含む、請求項１５に記載の媒体。
【請求項１７】
前記データソースを識別することが、ユニフォームリソースロケータを使用してデータソースを識別することを含む、請求項１５に記載の媒体。
【請求項１８】
前記データソースを識別することが、データベース、テーブルスペース、およびテーブル名を使用して、リレーショナルデータベーステーブルを識別することを含む、請求項１５に記載の媒体。
【請求項１９】
前記ジョブ依存性情報および前記データ系統情報によって参照される実行可能プログラムを識別することにより、リンクを生成することを更に含む、請求項１３に記載の媒体。
【請求項２０】
前記実行可能プログラムを識別することが、前記実行可能プログラムに提供されるパラメータに少なくとも部分的に基づいて、実行可能プログラムを識別することを含む、請求項１９に記載の媒体。
【請求項２１】
前記実行可能プログラムを識別することが、永続的なデータ記憶上の前記実行可能プログラムの位置に基づいて、実行可能プログラムを識別することを含む、請求項１９に記載の媒体。
【請求項２２】
前記判断された影響を指定する報告を生成することを更に含む、請求項１３に記載の媒体。
【請求項２３】
前記実行予定についての変更により前記データソースが影響を受ける時に、１以上の警報を生成することを更に含む、請求項１３に記載の媒体。
【請求項２４】
前記判断された影響が、前記実行予定についての変更により前記データソースの中の特定のデータセットが不正確になりそうなことを明示する時に、１以上の警報を生成することを更に含む、請求項１３に記載の媒体。
【請求項２５】
１つまたは複数のコンピュータと、

前記１つまたは複数のコンピュータによる実行時に、前記１つまたは複数のコンピュータに動作を行わせるように実行可能な命令を記憶する、１つまたは複数の記憶装置と、を備えるシステムであって、前記動作が、

複数のジョブの実行順序を示すジョブ依存性情報と、データ系統情報との間のリンクを識別することにより、前記ジョブ依存性情報と前記データ系統情報との結合に基づく依存性情報を判断することと、

前記依存性情報に基づいて、前記データ系統情報により指定されるデータソースにおける前記ジョブ依存性情報により指定されるジョブの実行予定についての変更の影響を判断することであって、

前記ジョブ依存性情報に含まれる、前記ジョブのジョブスケジューリング情報を識別するクエリを受信することと、

前記ジョブスケジューリング情報とデータ系統情報との間の前記リンクを識別することと、

前記ジョブスケジューリング情報に基づいて、前記データ系統情報により指定される前記データソースへの影響を識別することと、

によって前記変更の影響を判断することと、を含むシステム。

【請求項２６】

ワークロード自動化システムに関連付けられたワークロードリポジトリから、スケジューリング情報を取得することと、

前記スケジューリング情報を変換することと、

データ記憶に前記スケジューリング情報を記憶することであって、前記データ記憶が、前記データ系統情報を記憶することと、により、ジョブ依存性情報を取得することを更に含む、請求項２５に記載のシステム。

【請求項２７】

前記データ系統情報および前記ジョブ依存性情報によって参照されるデータソースを識別することにより、リンクを生成することを更に含む、請求項２５に記載のシステム。

【請求項２８】

前記データソースを識別することが、前記データ系統情報および前記ジョブ依存性情報内の同一の名前によって参照されるデータソースを識別することを含む、請求項２７に記載のシステム。

【請求項２９】

前記データソースを識別することが、ユニフォームリソースロケータを使用してデータソースを識別することを含む、請求項２７に記載のシステム。

【請求項３０】

前記データソースを識別することが、データベース、テーブルスペース、およびテーブル名を使用して、リレーショナルデータベーステーブルを識別することを含む、請求項２７に記載のシステム。

【請求項３１】

前記ジョブ依存性情報および前記データ系統情報によって参照される実行可能プログラムを識別することにより、リンクを生成することを更に含む、請求項２５に記載のシステム。

【請求項３２】

前記実行可能プログラムを識別することが、前記実行可能プログラムに提供されるパラメータに少なくとも部分的に基づいて、実行可能プログラムを識別することを含む、請求項３１に記載のシステム。

【請求項３３】

前記実行可能プログラムを識別することが、永続的なデータ記憶上の前記実行可能プログラムの位置に基づいて、実行可能プログラムを識別することを含む、請求項３１に記載のシステム。

【請求項３４】

前記判断された影響を指定する報告を生成することを更に含む、請求項２５に記載のシステム。
【請求項３５】
前記実行予定についての変更により前記データソースが影響を受ける時に、１以上の警報を生成することを更に含む、請求項２５に記載のシステム。
【請求項３６】
前記判断された影響が、前記実行予定についての変更により前記データソースの中の特定のデータセットが不正確になりそうなことを明示する時に、１以上の警報を生成することを更に含む、請求項２５に記載のシステム。
【請求項３７】
複数のジョブの実行順序を示すジョブ依存性情報と、データ系統情報との間のリンクを識別することにより、前記ジョブ依存性情報と前記データ系統情報との結合に基づく依存性情報を判断する手段と、
前記依存性情報に基づいて、前記データ系統情報により指定されるデータソースにおける前記ジョブ依存性情報により指定されるジョブの実行予定についての変更の影響を判断する手段であって、
前記ジョブ依存性情報に含まれる、前記ジョブのジョブスケジューリング情報を識別するクエリを受信する手段と、
前記ジョブスケジューリング情報とデータ系統情報との間の前記リンクを識別する手段と、
前記ジョブスケジューリング情報に基づいて、前記データ系統情報により指定される前記データソースへの影響を識別する手段と、
によって前記変更の影響を判断する手段と、を備えるシステム。
【発明の詳細な説明】
【技術分野】
【０００１】
優先権出願
本出願は、２０１４年5月２９日に出願された「ＷＯＲＫＬＯＡＤ　ＡＵＴＯＭＡＴＩＯＮ　ＡＮＤ　ＤＡＴＡ　ＬＩＮＥＡＧＥ　ＡＮＡＬＹＳＩＳ」と題する米国仮特許出願第６２／００４，４０６号、および、２０１４年8月２７日に出願された「ＷＯＲＫＬＯＡＤ　ＡＵＴＯＭＡＴＩＯＮ　ＡＮＤ　ＤＡＴＡ　ＬＩＮＥＡＧＥ　ＡＮＡＬＹＳＩＳ」と題する米国特許出願第１４／４７０，５０１号に基づき、優先権を主張する。両者の内容全体が、参照により本明細書に組み込まれる。
【背景技術】
【０００２】
ワークロードの自動化は、概して、一般にジョブをセットアップするプロセスを指し、そのように人間との対話処理なしで完結するように実行されることができる。全ての入力パラメータが、スクリプト、コマンドライン引数、ワークフロー自動化システム、制御ファイル、またはジョブ制御言語によって事前定義されている。ジョブは、利用可能な処理リソースと、予め定義された依存性に基づいてスケジュールされる。
【０００３】
データ系統は、データの発生元、およびデータがどこに移動し、データが経時的にどのように変化するかを説明する。この用語は、多様なプロセスを通じて進むにつれて、データがどうなるかを説明することもできる。データ系統は、情報がどのように使用されるかを分析し、特定の目的に合った情報の主要ビットを追跡する取り組みに役立つことができる。
【発明の概要】
【０００４】
概略的な態様１では、方法は、ジョブ依存性情報を取得する動作を含み、ジョブ依存性情報が、複数のジョブの実行順序を指定する。方法は、データ記憶と変換との間の依存関

係を識別するデータ系統情報を取得する動作も含み、少なくとも１つの変換が、第１のデータ記憶からデータを受け取り、第２のデータ記憶に対してデータを生じさせる。方法は、ジョブ依存性情報とデータ系統情報との間にリンクを生成する動作も含む。方法は、ジョブ依存性情報、生成されたリンク、およびデータ系統情報に基づいて、複数のアプリケーションのうちのアプリケーションの実行予定についての変更の影響を判断する動作も含む。

【０００５】

本態様の他の実施形態は、対応するコンピュータシステム、装置、および１つまたは複数のコンピュータ記憶装置に記録されたコンピュータプログラムを含み、それぞれが、方法の動作を実行するように構成される。１つまたは複数のコンピュータのシステムは、作動中にシステムに動作を実行させるシステムにインストールされた、ソフトウェア、ファームウェア、ハードウェア、またはそれらの組合せを有することによって、特定の動作を実行するように構成され得る。１つまたは複数のコンピュータプログラムは、データ処理装置による実行時に装置に動作を実行させる命令を含むことによって、特定の動作を実行するように構成され得る。

【０００６】

方法は、態様１による態様２を含み、態様２では、ジョブ依存性情報を取得することが、ワークロード自動化システムに関連付けられたワークロードリポジトリから、スケジューリング情報を取得することと、スケジューリング情報を変換することと、データ記憶にスケジューリング情報を記憶させることであって、データ記憶が、データ系統を記憶していることと、を含む。方法は、態様１または２による態様３を含み、態様３では、リンクを生成することが、データ系統情報およびジョブ依存性情報によって参照されるデータソースを識別することを含む。方法は、態様１、２または３による態様４を含み、態様４では、データソースを識別することが、データ系統情報およびジョブ依存性情報内の同一の名前によって参照されるデータソースを識別することを含む。方法は、態様１、２、３または４による態様５を含み、態様５では、データソースを識別することが、ユニフォームリソースロケータを使用してデータソースを識別することを含む。方法は、態様１、２、３、４または５による態様６を含み、態様６では、データソースを識別することが、データベース、テーブルスペース、およびテーブル名を使用して、リレーショナルデータベーステーブルを識別することを含む。方法は、態様１、２、３、４、５または６による態様７を含み、態様７では、リンクを生成することが、ジョブ依存性情報およびデータ系統情報によって参照される実行可能プログラムを識別することを含む。方法は、態様１、２、３、４、５、６または７による態様８を含み、態様８では、実行可能プログラムを識別することが、実行可能プログラムに提供されるパラメータに少なくとも部分的に基づいて、実行可能プログラムを識別することを含む。方法は、態様１、２、３、４、５、６、７または８による態様９を含み、態様９では、実行可能プログラムを識別することが、永続的なデータ記憶上の実行可能プログラムの位置に基づいて、実行可能プログラムを識別することを含む。方法は、態様１、２、３、４、５、６、７、８または９による態様１０を含み、態様１０では、影響を判断することが、ジョブスケジューリングデータを識別するクエリを受信することと、ジョブスケジューリングデータとデータ系統情報との間のリンクを識別することと、スケジューリングデータに基づいて、データ系統情報への影響を識別することと、を含む。

【発明の効果】

【０００７】

本明細書に記載される主題の特定の実施形態は、以下の利点のうちの１つまたは複数を実現するように実装可能である。データ処理システムの全体像を考察することができる。データ依存性を説明するデータ系統情報を、スケジューリング依存性を説明するワークフロー自動化情報と結合することができる。データ系統情報またはジョブスケジューリング情報のいずれかの変更の影響を判断することができる。これにより、技術プロセスの開発者や管理者が、ワークフローをより効率的かつ中断の少ない方法で監視し、調整すること

に役立てることができる。全体として、プロセス障害、リソース消費、およびデータ処理期間のそれぞれが、それによって減少され得る。

【図面の簡単な説明】

【０００８】

【図１】例示的なスケジューリング図である。

【図２】例示的なデータ系統図である。

【図３】スケジューリング情報とデータ系統情報を統合する、例示的なシステムである。

【図４】データ分析技術が使用され得るデータ処理システムの例を示す。

【図５】スケジューリング情報への変更の影響を識別する、例示的な処理のフローチャートである。

【図６】相互に関連付けられた、依存性情報およびデータ系統情報の例を示す。

【発明を実施するための形態】

【０００９】

ジョブ依存性情報およびデータ系統情報は、結合されて事業の状態の全体像を提供することができる。従来、ジョブ依存性情報およびデータ系統情報は、異なる情報システムおよびデータベースにまたがって細分化されている。ジョブ依存性情報は、異なるジョブまたはタスクの実行の間に確立されている順序を説明する。データ系統情報は、データソースおよびデータシンクが、事業全体にわたってどのように関係しているかを説明する。ユーザは、特定のジョブが遅延しそうかどうか、または、どの報告もしくはデータシンクが影響を受けそうか、などの疑問に答えたいと望むことがある。本明細書で説明するシステムは、これらの全く異なるデータソースを統合する。

【００１０】

ジョブ依存性情報は、ワークロード自動化プログラムまたはジョブスケジューリングプログラムから取得されることができる。ワークロード自動化プログラムまたはジョブスケジューリングプログラムは、複雑な依存性を伴うワークロードタイプの様々なセットを調整する。概して、ジョブ依存性情報は、異なるタスクが実行されるべき順序を定義する。アプリケーションのスケジューリングは、典型的には、データの依存性を考慮に入れておらず、それは、本来データを意識したものではない。アプリケーションのスケジューリングは、単に、異なるタスクが実行し得る順序を定めているにすぎない。この順序付けは、データ依存性に基づく可能性があるが、リソース割り当て、全体の実行時間、および他の効率の最適化に基づく可能性もある。ワークロード自動化システムにおけるタスクは、データフローグラフ、Ｊａｖａプログラム、ファイル転送コマンド、ビジネススイートソフトウェア統合、ウェブサービスアクセス、メッセージング、または任意の他の実行可能なプロセスを含んでもよい。ユーザは、スケジュール変更の影響、例えば、ジョブが遅延するかどうか、を判断したい場合がある。ワークロード自動化システムは、システム内で定義されている詳細を見るように、その機能が制限されている。

【００１１】

一方、データ系統情報は、データがシステムによって処理される順序を識別する。概して、データ系統情報は、データの発生元、および、データ処理アプリケーションの間にデータがどこに移動するか、またはデータがどのように変換されるかを含む、データのライフサイクルを説明する。データ系統情報は、多様なプロセスによってデータが変換されるにつれて、データに何が起こるかを説明する。概して、データ系統情報の分析は、情報がどのように使用されるかを識別し、特定の目的に合った情報の主要部分を追跡するために使用される。ジョブ依存性情報をデータ系統リポジトリに統合することによって、プロセスおよびデータのより堅牢な視界を展開することができる。

【００１２】

ジョブ依存性情報をデータ系統リポジトリに統合することによって、プロセスおよびデータのより堅牢な視界を展開することができる。

【００１３】

ジョブ依存性情報は、ワークロード自動化ツールから抽出されることができ、データ系

統情報は、データ系統ツールから抽出されることができる。情報は、共に結合され、以降のアクセスのために、共通のリポジトリに記憶されることができる。

【００１４】

図１は、ジョブ「Ｐｓｉ」についての例示的なスケジューリング図１００である。スケジューリング図１００は、ワークロード自動化システムにおける、スケジューリング動作の例である。スケジューリング図は、ジョブ間のジョブ依存性を示している。多くの実装において、コンポーネント間により大きな相互関係を有する、より複雑な図が存在する。現在のスケジューリング図１００は、例示の目的で使用される。ジョブは、後続ジョブの開始前に、先行ジョブが完了している必要がある、階層的順序で定義される。この図では、ジョブは、有向矢印で接続されている。矢印は、先行ジョブから後続ジョブに向かっている。例えば、「スクリプト　コマンド　１」ジョブ１０２は、「データベース」ジョブ１０６または「ファイル監視」ジョブ１１０が開始し得る前に完了しなければならない。同様に、「データベース」ジョブ１０６および「ファイル監視」ジョブ１１０は、「ＦＴＰ」ジョブ１１２が開始し得る前に完了しなければならない。「実行」ジョブ１１４は、「ファイル監視」ジョブ１１０が完了した後で、開始し得る。最後に、「モニタ完了」ジョブ１１６は、「ＦＴＰ」ジョブ１１２および「実行」ジョブ１１４が完了した後にのみ、実行することができる。

【００１５】

ワークロード自動化システムは、様々なジョブについての情報を収集する。例えば、「スクリプト　コマンド　１」ジョブ１０２は、ジョブを定義し、説明する属性１０４を有する。この例では、属性１０４は、実行されるジョブの種類を示すジョブタイプと、ジョブの名前を示すジョブ名と、スクリプトの位置を定義するファイルパスと、実行されるスクリプトの名前を示すファイル名と、スクリプトを実行すべきユーザ名を示す実行者と、ジョブの現在の状態（例えば、保留中、実行中、完了済み、失敗）を示す状態と、現在のジョブが完了後にのみ実行できるジョブを示す後続と、実行されるステップを定義するスクリプトと、を含む。

【００１６】

他の種類のジョブは、異なる属性を含んでもよい。例えば、「データベース」ジョブ１０６は、属性１０８を有する。これらの属性は、ＳＱＬコマンド（ここでは、「select us.order, us.order_amount from ne_production」）、先行ジョブのリスト（ここでは、スクリプト　コマンド　１）、および後続ジョブのリスト（ここでは、ＦＴＰ）を含むが、限定はされない。

【００１７】

同様に、「実行」ジョブ１１４は、実行されるべきプログラムの名前、例えば、「ＴｒａｎｓｆｏｒｍＡ．ｅｘｅ」を識別するパラメータ１１８を含んでもよい。「ＦＴＰ」ジョブ１１２は、ファイルおよびファイル転送動作の宛先を識別するパラメータ１２０を含んでもよい。例えば、パラメータ１２０は、Ｂｒａｚｉｌ　Ｆｅｅｄ．ｄａｔファイルが、ｓｅｒｖｅｒ．ｃｏｍに転送予定であることを識別する。本明細書において、識別されるパラメータは、単なる例示にすぎない。他のパラメータは、ジョブスケジューリング情報によって定義され、含まれ得る。

【００１８】

ワークロード自動化システムは、図示していない、他のジョブに関連してジョブＰｓｉをスケジューリングしてもよい。例えば、ジョブＰｓｉは、ジョブＺｅｔａ（または、図示していない、何らかの他のジョブ）の後に行われるようにスケジューリングされてもよい。ワークロード自動化システムは、ジョブ間のスケジュールを、リソース管理、依存性の報告、利用可能時間、優先度、または他の制約に基づいて判断してもよい。

【００１９】

図２は、例示的なデータ系統図２００である。データ系統は、概して、データの発生元、ならびにデータがどこへ移動するか、およびデータがどのように変換され処理されるかを含む、データライフサイクルとして定義される。この用語は、多様なプロセスを通じて

進むにつれて、データがどうなるかを説明することもできる。データ系統は、情報がどのように使用されるかを分析し、特定の目的に合った情報の主要ビットを追跡する取り組みに役立つことができる。概して、データ系統図は、データソース、データシンクおよび変換の間の関係性を示す図である。各変換は、１つまたは複数のデータソース（例えば、入力データ）を含み、１つまたは複数のデータシンクにデータ（例えば、出力データ）を生じさせることができる。データ系統情報内のそれぞれのデータソース、データシンク、および変換は、本明細書では、まとめてデータ系統要素と呼ばれるものとする。

【００２０】

この例では、データソース「Ｕ．Ｓ．Ｆｅｅｄ」２０２は、「変換Ａ」２０４の変換にデータを提供する。変換Ａ２０４は、「Ｕ．Ｓ．Ｆｅｅｄ」２０２によって提供されたデータに対して動作を実行し、「中間データセット１」２０６のデータ記憶に結果を記憶させる。データソース「Ｍｅｘｉｃｏ　Ｆｅｅｄ」２０８および「Ｂｒａｚｉｌ　Ｆｅｅｄ」２１４は、「変換Ｃ」２１０の変換にデータを提供する。「変換Ｃ」２１０の変換は、「Ｍｅｘｉｃｏ　Ｆｅｅｄ」２０８および「Ｂｒａｚｉｌ　Ｆｅｅｄ」２１４によって提供されたデータに対して動作を実行し、「中間データセット２」２１２のデータ記憶に結果を記憶させる。データソースは、例えば、単層ファイル、リレーショナルデータベース、オブジェクトデータベース、または、コンピュータシステムにデータを記憶させるための任意の他の機構であってもよい。例えば、「Ｂｒａｚｉｌ　Ｆｅｅｄ」２１４は、「Ｂｒａｚｉｌ　Ｆｅｅｄ．ｄａｔ」などのファイルであってもよい。変換は、データを操作することが可能な実行可能プログラムであってもよい。例えば、仮想マシン内で実行されるＪａｖａプログラム、実行ファイル、データフローグラフなどであってもよい。例えば、「変換Ａ」２０４の変換は、「ＴｒａｎｓｆｏｒｍＡ．ｅｘｅ」という名前の実行ファイルであってもよい。

【００２１】

「中間データセット１」２０６のデータ記憶および「中間データセット２」２１２のデータ記憶は、「変換Ｂ」２１６の変換にデータを提供する。「変換Ｂ」２１６の変換は、「中間データセット１」２０６から提供されたデータを使用し、「中間データセット２」２１２は、「出力報告」２１８のデータ記憶に結果を記憶させる。

【００２２】

データ系統に記憶されている情報は、データの異なる部分が、データの他の部分にどのように影響を与えるかを識別することができる。例えば、「Ｕ．Ｓ．Ｆｅｅｄ」データソースは、注文、および注文毎の額を含んでもよい。「変換Ａ」２０４は、地域に基づいて、例えば、ニューイングランド、東部諸州、南部、中西部、平原州で発生した注文などによって、データを集約してもよい。データ系統情報は、「Ｕ．Ｓ．Ｆｅｅｄ」２０２からの額のフィールドが、「中間データセット１」２０６の「地域合計」フィールドに集約されると識別してもよい。

【００２３】

スケジューリングデータをデータ系統データと結合することによってのみ得られる、いくつかの情報がある。例えば、図１の「データベース　ジョブ」１０６が、図２の「Ｕ．Ｓ．Ｆｅｅｄ」２０２を生成する場合において、その後ジョブ１０６が遅延し、または実行に失敗する場合、出力報告２１８は、遅延するか、または不正確なものとなる。データ系統情報およびジョブスケジューリング情報の両方を検討することなしに、これらの関係性が生じることはない。上記の例を参照すると、「ジョブＺｅｔａ」（ジョブＰｓｉに先行するものとして上述された）が遅延する場合、出力報告２１８が遅延するか、または不正確なものとなる恐れがあるため、さらに関係性がより複雑になる可能性がある。

【００２４】

図３は、スケジューリング情報およびデータ系統情報を統合する例示的システムである。スケジューリングリポジトリ３０２ａ～ｂからのデータは、データ系統リポジトリ３０６にインポートされることができる。スケジューリングリポジトリ３０２ａ～ｂは、ワークロード自動化システム、例えば、ＣＯＮＴＯＬ－Ｍ、ＴＩＶＯＬＩ、ＴＷＳ　ＡＵＴＯ

ＳＹＳ、ＣＡ－７などに関連付けられたデータリポジトリであってもよい。ワークロード自動化システムのそれぞれについてのデータは、異なるフォーマットで記憶されてもよい。変換コンポーネント３０４ａ～ｂは、スケジューリングリポジトリ３０２ａ～ｂに記憶されたデータを、結合されたリポジトリ３０６の記憶用の共通データフォーマットに変換するために使用され得る。いくつかの実装では、変換コンポーネントは、例えば、データフローグラフの計算環境で実行するデータフローグラフであってもよい。

【００２５】

データ系統リポジトリ３１４からのデータは、結合されたリポジトリに記憶されることもできる。データ系統情報は、リポジトリ内に挿入される前に、変換コンポーネント３１６によって変換されてもよい。例えば、データのデータタイプは、リポジトリの予定されるデータフォーマットに従うために、あるタイプから別のタイプに変換されてもよい。さらに、データ構造は、例えば、データをジョブスケジューリング情報と効率的に統合するために、データ系統データ構造を単純化することを含み、変更されてもよい。

【００２６】

ジョブスケジューリング情報は、収集され、結合されたリポジトリに統合されることができる。ジョブスケジューリング情報は、結合されたリポジトリで受け入れ可能なフォーマットに修正されることができる。例えば、データの特定のフィールドのフォーマットが、変更されてもよい。異なるデータオブジェクト間の関係性は、機能的に同一または異なる形式に変更されてもよい。ジョブスケジューリング情報を統合することは、以前の大量のワークロードスケジューリングデータから、古いまたは期限を経過した情報を識別すること、および上書きすること、または保管することを含むことができる。ジョブ依存性情報とデータ系統情報とが、共に結合されリンクされる。ジョブ依存性情報は、情報に関連付けられた属性またはパラメータに基づいて、データ系統情報にリンクされてもよい。例えば、ジョブおよびデータ系統要素は、同一の実行ファイル（例えば、上述の「TransforｍA.exe」）を参照してもよい。実行ファイルは、完全修飾識別子に基づいて識別されることができる。完全修飾識別子は、完全パスを含んでもよく、即ち、コンピュータ、およびハードドライブなどの永続的な記憶装置上の位置が、識別されてもよい。完全修飾識別子は、実行ファイルに提供される任意のパラメータを含んでもよい。同様に、ジョブおよびデータ系統要素は、同一のデータ記憶を参照してもよい。例えば、上記の図では、図１のＦＴＰジョブ１１２およびＢｒａｚｉｌ　Ｆｅｅｄデータソース２１４は、「Ｂｒａｚｉｌ　Ｆｅｅｄ．ｄａｔ」ファイルを参照する。データソースは、完全修飾識別子に基づいて識別され得る。例えば、完全修飾識別子は、単層ファイルを識別する完全パスもしくはユニフォームリソースロケータ（URL）、または、サーバ、データベース、テーブルスペース、リレーショナルデータベース内のテーブル名を識別する情報であってもよい。これらの、または他の共通の要素が識別される時、プロセスは、ジョブ依存性情報とデータ系統情報との間にリンクを生成することができる。

【００２７】

いくつかの実装では、ジョブスケジューリング情報は、定期的に（例えば、週１回、１日１回、１時間１回、など）、結合されたリポジトリ３０６に統合されることができる。いくつかの実装では、ジョブスケジューリング情報に対する変更が、統合プロセスのトリガとなって、ほぼリアルタイムで情報を統合してもよい。例えば、データベースのトリガは、変更が検出された時に統合プロセスを開始してもよい。あるいは、ジョブスケジューリングシステムにおけるコールバック機構が、統合プロセスを開始させてもよい。

【００２８】

結合されたリポジトリ３０６は、グラフベースのアプリケーションの開発および実行、ならびにグラフベースのアプリケーションと他のシステム（例えば、他のオペレーティングシステム）との間のメタデータの交換を支援するように設計された、拡張性のあるオブジェクト指向データベースシステムであることが好ましい。結合されたリポジトリ３０６は、ドキュメンテーション、レコードフォーマット（例えば、テーブル内のレコードのフィールドおよびデータタイプ）、変換機能、グラフ、ジョブ、ならびにモニタリング情報

を含む、あらゆる種類のメタデータのための記憶システムである。

【００２９】

結合されたリポジトリ３０６は、コンピューティングシステムによって処理されるべき実際のデータを表す、データオブジェクトを記憶することもできる。

【００３０】

結合されたリポジトリ３０６に記憶されたデータ系統情報およびジョブ依存性情報の結合は、そうでなければ利用できない、報告および情報を生成するために使用されることができる。これら２つのデータのソースを共に結合することによって、そうでなければ利用できない、ジョブの全体像を見ることが可能となる。例えば、データの結合は、「ジョブが遅延する場合に、任意の所与のデータセットに対して何を意味するか」という問いに対する答えを提供するために使用されることができる。ジョブは、データセット１に直接影響を与えないかもしれないが、ワークロード自動化システムにおけるスケジューリング指示子のために、間接的に影響を与えるかもしれない。

【００３１】

監査および報告システム３０８は、特定のデータセットが、影響を受けようとしている時に、警報を出すことができる。例えば、業務では、特定のデータセットが不正確になりそうな時に警報を出すことを望むかもしれない。

【００３２】

情報処理システム３１０は、ユーザ３１２にグラフィカルユーザインターフェースを提示することができ、ユーザが、上述のリンクに基づいて、ジョブスケジューリング情報とデータ系統情報との間をナビゲートすることを含み、スケジューリング情報および／またはデータ系統の詳細をドリルダウンし、検討することができるようにする。

【００３３】

図４は、データ分析技術が使用され得る、データ処理システム４００の例を示す。システム４００は、データソース４０２を含み、データソース４０２は、例えば、ワークロード自動化システムのデータリポジトリを含む、記憶装置、またはオンラインデータストリームへの接続などの１つまたは複数のデータのソースを含むことができる。各データ記憶は、様々なフォーマットのうちのいずれか（例えば、データベーステーブル、スプレッドシートファイル、単層テキストファイル、またはメインフレームによって使用されるネイティブフォーマット）で、データを記憶し、または提供することができる。実行環境４０４は、前処理モジュール４０６と、実行モジュール４１２とを含む。実行環境４０４は、ＵＮＩＸオペレーティングシステムのあるバージョンなどの、適切なオペレーティングシステムの制御下で、例えば、１つまたは複数の汎用コンピュータ上に提供されてもよい。例えば、実行環境４０４は、局所的（例えば、対称型マルチプロセッシング（SMP）コンピュータなどの、マルチプロセッサシステム）、もしくは局所分散（例えば、クラスタとして連結された複数のプロセッサ、もしくは大規模並列処理（MPP）システム）のいずれか、またはリモートもしくはリモート分散（例えば、ローカルエリアネットワーク（LAN）および／または広域ネットワーク（WAN）を介して連結された複数のプロセッサ）、またはそれらの任意の組合せで、複数の中央処理装置（CPU）またはプロセッサコアを使用するコンピュータシステムの構成を含む、マルチノード並列コンピューティング環境を含むことができる。

【００３４】

変換モジュール４０６は、データソース４０２からデータを読み出し、データを正準形式に変換し、データ記憶４１６に情報を記憶させる。データソース４０２を提供する記憶装置は、実行環境４０４に対して局所的（例えば、実行環境４０４を提供するコンピュータに接続された記憶媒体（例えば、ハードドライブ４０８）に記憶されている）であってもよく、または、実行環境４０４に対してリモート（例えば、リモート接続（例えば、クラウドコンピューティングインフラストラクチャによって提供される）を介して、実行環境４０４を提供するコンピュータと通信関係にある、リモートシステム（例えば、メインフレーム４１０）上で提供されている）であってもよい。

【００３５】
分析モジュール４１２は、データ系統情報と結合された、変換モジュール４０６によって生成された、記憶された情報を使用して、結合されたデータの分析を、結合されていなければ可能ではない方法で実行する。例えば、ジョブのスケジュール変更は、ジョブによって直接影響を受けるもの以外のデータ記憶に強い影響を与える。ジョブは、他のジョブに影響を与える可能性があり、それらのジョブのそれぞれが、データソースに影響を与える可能性がある。いくつかのシナリオでは、データソースに対する変更は、同様に、追加のジョブに影響を与える可能性がある。記憶された情報は、データ記憶システム４１６に記憶されてもよい。データ記憶システム４１６は、ユーザ４２０と対話処理を行う分析システム４１８にもアクセス可能である。ユーザ４２０は、結合されたデータのドリルダウン分析を実行することが可能である。

【００３６】
分析システム４１８および実行環境４０４は、いくつかの実装において、有向リンク（作業要素、即ち、データの流れを表す）によって頂点間が接続された、頂点（データ処理コンポーネントまたはデータセットを表す）を含むデータフローグラフとして、計算アプリケーションを実行するシステムを使用して設計されている。例えば、そのような環境は、参照により本明細書に組み込まれる、「Ｍａｎａｇｉｎｇ　Ｐａｒａｍｅｔｅｒｓ　ｆｏｒ　Ｇｒａｐｈ－Ｂａｓｅｄ　Ａｐｐｌｉｃａｔｉｏｎｓ」と題した米国特許出願公開第２００７／００１１６６８号に、より詳細に記載されている。そのようなグラフベースの計算を実行するシステムは、参照により本明細書に組み込まれる、「ＥＸＥＣＵＴＩＮＧ　ＣＯＭＰＵＴＡＴＩＯＮＳ　ＥＸＰＲＥＳＳＥＤ　ＡＳ　ＧＲＡＰＨＳ」と題する米国特許第５，９６６，０７２号に記載されている。このシステムに従って作られるデータフローグラフは、グラフコンポーネントによって表される個々のプロセスに出入りする情報を取得し、プロセス間で情報を移動し、およびプロセスに対する実行順序を定義する方法を提供する。このシステムは、プロセス間の通信方法を任意の利用可能な方法（例えば、グラフのリンクに従う通信パスが、ＴＣＰ／ＩＰもしくはＵＮＩＸドメインソケットを使用することができ、共有メモリを使用して、プロセス間でデータを渡すことができる）から選択するアルゴリズムを含む。

【００３７】
変換モジュール４０６は、異なる形式のデータベースシステムを含む、データソース４０２を具体化し得る様々なタイプのシステムからデータを受信することができる。データは、空値を含む可能性がある、それぞれのフィールド（「属性」または「カラム」とも呼ばれる）についての値を有するレコードとして編成されてもよい。データソースからデータを読み出す時、変換モジュール４０６は、典型的には、そのデータソース内のレコードを説明する、いくつかの初期フォーマット情報で開始される。場合によっては、データソースのレコード構造は、最初は既知でなくてもよく、その代わりに、データソースまたはデータの分析後に判断されてもよい。レコードについての初期情報は、例えば、固有値を表すビットの数字、レコード内のフィールドの順序、および、ビットによって表される値の型（例えば、文字列、符号付き／符号なし整数）を含むことができる。

【００３８】
図５は、スケジューリング情報への変更の影響を識別する、例示的なプロセス５００のフローチャートである。プロセスは、プロセスを実行するコンピュータシステムによって実行されてもよい。

【００３９】
データ系統情報が、取得５０２されることができる。データ系統情報は、上述したデータ記憶から取得されてもよい。データ系統情報は、データ記憶と変換との間の依存関係を識別することができる。変換は、１つのデータ記憶からデータを受け取り、別のデータ記憶にデータを生じさせることができる。

【００４０】
ジョブ依存性情報が、取得５０４されることができる。ジョブ依存性情報は、上記で論

じたプロセスを通じて、取得され得る。ジョブ依存性情報は、複数のジョブの実行順序を指定することができる。

【００４１】

ジョブ依存性情報およびデータ系統情報の要素のうちの少なくもいくつかの間のリンクが、識別されることができる。リンクは、直接であってもよい（例えば、ジョブが、変換を実行５０６させてもよい）。リンクは、間接であってもよい（例えば、ジョブが、データフローグラフを実行させてもよく、その場合に、データフローグラフが変換を含む）。リンクは、ジョブスケジューリング情報およびデータ系統情報によって参照されるファイルおよびデータ記憶に基づいて、判断されてもよい。

【００４２】

データ記憶上の複数のアプリケーションのうちのアプリケーションの実行予定についての変更の影響が判断５０８され得る。影響は、ジョブ依存性情報、リンク、およびデータ系統情報に基づいて判断されてもよい。例えば、ユーザは、少なくとも１つのジョブ、実行可能プログラムまたはデータ記憶を識別するクエリを提示してもよい。例えば、特定のジョブ、実行可能プログラム、もしくはデータソースが利用可能でない場合、または特定のジョブが、失敗もしくは、時間通りに実行されない場合に、ユーザは、影響を判断することを希望してもよい。代替的に、または追加的に、ジョブが失敗したこと、または時間通りの完了に失敗したことを、ワークロード自動化システムが識別してもよい。例えば、ジョブ自体によってハンドリングできないエラーが処理中に発生する時に、ジョブが失敗することがある。例えば、期限が経過する時にも、ジョブが失敗することがある。

【００４３】

プロセスは、識別されたジョブ、実行可能プログラムまたはデータ記憶に従属しているジョブの全てを識別することができる。プロセスは、識別されたジョブと、従属関係にあるジョブとの間のリンク、およびデータ系統要素を識別することができる。データ系統要素は、データソース、データシンク、およびデータ変換の間の関係性、または関係性の一部を説明する、データ要素であってもよい。識別されたジョブおよび従属関係にあるジョブにリンクされているデータ系統要素は、ジョブに従属しているデータ系統要素を判断するために使用されることができる。即ち、データ系統要素に続いてアクセスされるデータ系統要素の全てが、リンクによって識別されることができる。

【００４４】

プロセスは、再帰的に適用されることができる。例えば、データ系統要素が一度識別されると、追加のリンクが、データ系統要素を再度追加のジョブに関係付ける。追加のジョブは、同様に、再度追加のデータ系統要素にリンクしてもよい。

【００４５】

例えば、図６は、ジョブ依存性情報およびデータ系統情報の結合に基づいて判断され得る依存性情報の単純化した例を示す。ジョブ６００は、２つのサブジョブ、（１）生成ｄａｉｌｙ　ｓａｌｅｓ．ｄａｔジョブ６０２、および（２）ＦＴＰ　ｄａｉｌｙ　ｓａｌｅｓ．ｄａｔジョブ６０４を含む。ｄａｉｌｙ　ｓａｌｅｓ　ｆｉｌｅ．ｄａｔは、データ系統情報によって識別されるものとして、データフローグラフ６０６によって使用される。この例では、ＦＴＰ　ｄａｉｌｙ　ｓａｌｅｓ．ｄａｔジョブは、破線６１０で示されるように、ｄａｉｌｙ　ｓａｌｅｓ．ｄａｔデータソース６０８にリンクされている。ｉｎｐｕｔ　ｆｉｌｅ．ｄａｔ６０８は、集約変換６１２によって、図示されていない他のデータと集約される。集約変換６１２は、データソース、ｑｕａｒｔｅｒｌｙ．ｄａｔ６１４を生成する。別のジョブ６１６は、ｑｕａｒｔｅｒｌｙ．ｄａｔのファイルの生成を監視する、ファイル監視ｑｕａｒｔｅｒｌｙ．ｄａｔジョブ６１８を含む。この関係性に基づいて、ｑｕａｒｔｅｒｌｙ．ｄａｔのデータソースと、ファイル監視ｑｕａｒｔｅｒｌｙ．ｄａｔジョブ６１８とは、破線６２０で表すように、結合されたリポジトリにおいてリンクされる。生成１０－Ｋ情報ジョブ６２２は、ｑｕａｒｔｅｒｌｙ．ｄａｔのファイルを使用して、ＳＥＣ用の１０－Ｋ情報を生成する。

【００４６】

結合されたスケジュール依存性情報およびデータ系統情報を使用すること、ならびに再度追加のスケジュール依存性情報にリンクすることによって、生成ｄａｉｌｙ　ｓａｌｅｓジョブと生成１０－Ｋ情報ジョブがジョブ依存性情報によってリンクされていなくても、システムは、ｄａｉｌｙ　ｓａｌｅｓ．ｄａｔファイル生成の問題が、１０－Ｋの生成に伴う遅延という結果を引き起こし得るということを判断することができる。

【００４７】

上述のデータ統合および分析アプローチは、適切なソフトウェアを実行するコンピューティングシステムを使用して実行され得る。例えば、ソフトウェアは、１つまたは複数のプログラムされた、またはプログラム可能なコンピューティングシステム（分散、クライアント／サーバ、もしくはグリッドなどの、様々なアーキテクチャのものであり得る）上で実行する、１つまたは複数のコンピュータプログラム内の手続きを含んでもよい。それぞれのコンピューティングシステムは、少なくとも１つのプロセッサ、少なくとも１つのデータ記憶システム（揮発性および／または不揮発性メモリおよび／または記憶素子を含む）、少なくとも１つのユーザインターフェース（少なくとも１つの入力デバイスまたはポートを用いて入力を受信するための、および、少なくとも１つの出力デバイスまたはポートを用いて出力を提供するための）を含む。ソフトウェアは、例えば、データフローグラフの設計、構成、および実行に関するサービスを提供する、より大きなプログラムの１つまたは複数のモジュールを含んでもよい。プログラムのモジュール（例えば、データフローグラフの要素）は、データリポジトリ内に記憶されたデータモデルに従ったデータ構造または他の編成済みデータとして実装されてもよい。

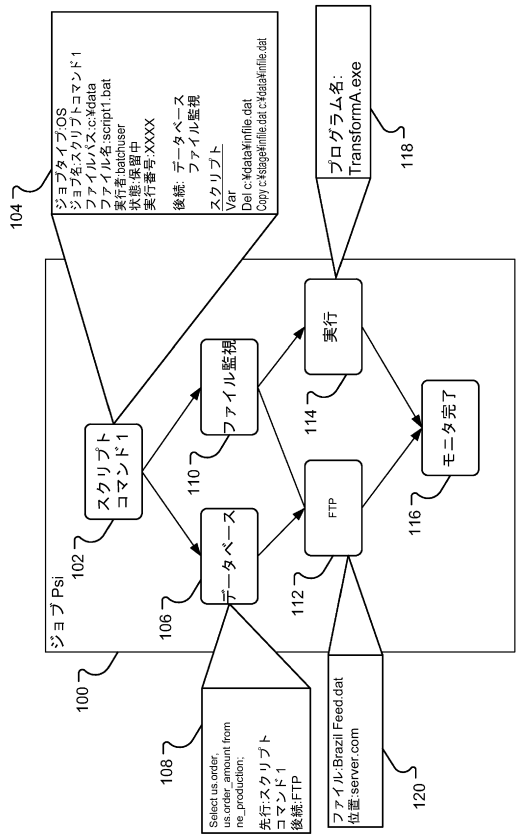
【００４８】

ソフトウェアは、ＣＤ－ＲＯＭもしくは他のコンピュータ可読媒体（例えば、汎用もしくは専用コンピューティングシステムもしくはデバイスによって可読である）などの、有形の非一時的媒体において提供されてもよく、またはネットワークの通信媒体を介して、ソフトウェアが実行されるコンピューティングシステムの有形の非一時的媒体に（例えば、伝播信号において符号化されて）配信されてもよい。処理のいくつかもしくは全ては、専用コンピュータ上で実行されてもよく、または、コプロセッサもしくはフィールドプログラマブルゲートアレイ（FPGA）もしくは専用の特定用途向け集積回路（ASIC）などの、専用ハードウェアを使用してもよい。処理は、ソフトウェアによって指定された計算の異なる部分が、異なる計算要素によって実行される、分散方式で実装されてもよい。そのようなコンピュータプログラムのそれぞれは、記憶デバイス媒体が、本明細書で説明した処理を実行するためにコンピュータによって読み出される時にコンピュータを構成および動作させるための、汎用または専用プログラム可能なコンピュータによってアクセス可能な、記憶デバイスのコンピュータ可読記憶媒体（例えば、ソリッドステートメモリもしくは媒体、または磁気媒体もしくは光学式媒体）上に記憶され、またはダウンロードされるのが好ましい。本発明のシステムは、有形の非一時的媒体として実装され、そのように構成された媒体が、コンピュータを、特定の予め定義された方法で動作させて、本明細書で説明した１つまたは複数の処理ステップを実行させる、コンピュータプログラムで構成されるものと考えられてもよい。

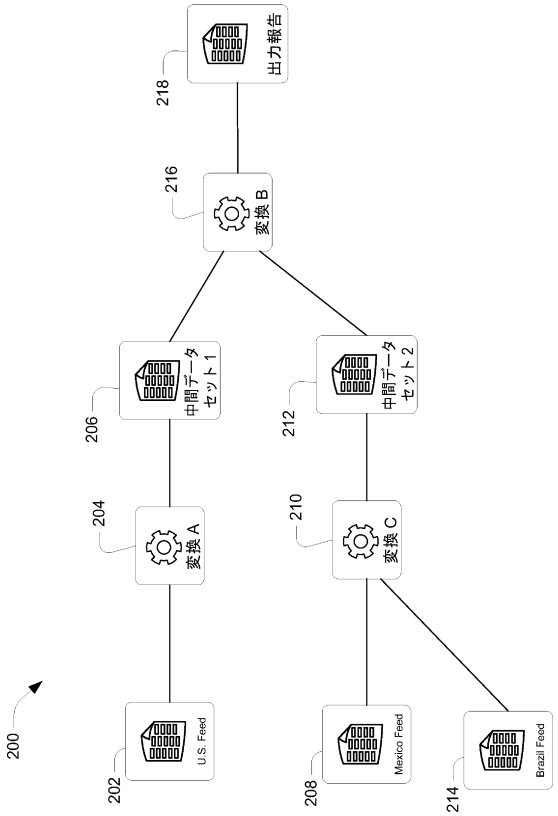
【００４９】

いくつかの本発明の実施形態について説明した。それにも関わらず、前述の説明は、以下の特許請求の範囲によって定義される本発明の範囲を例証することを意図するものであり、限定することを意図するものではないと理解されるべきである。したがって、他の実施形態もまた、以下の特許請求の範囲内である。例えば、本発明の範囲から逸脱することなく、様々な修正が行われ得る。さらに、上述したステップのうちのいくつかは、独立した順序であってもよく、したがって、説明した順序とは異なる順序で実行されることができる。

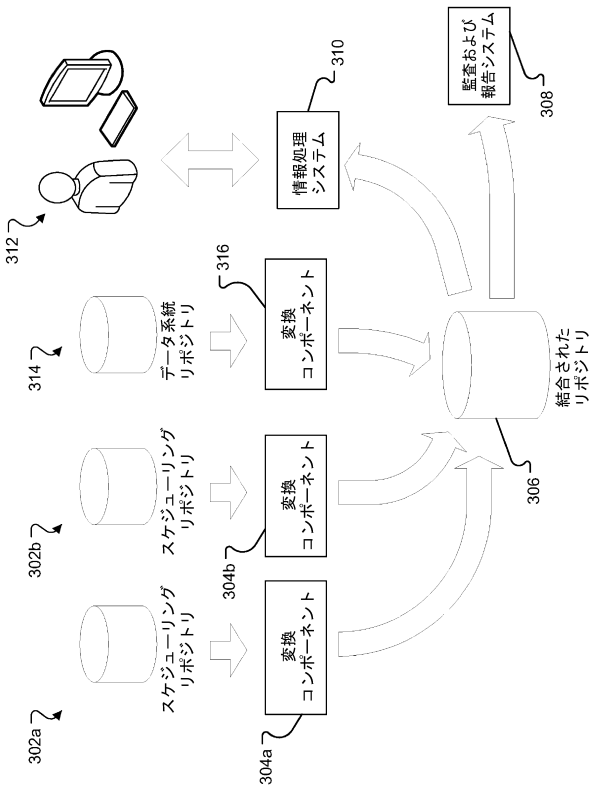
【図１】


ジョブタイプ:OS
ジョブ名:スクリプトコマンド1
ファイルパス:c:¥data
ファイル名:script1.bat
実行者:batchuser
状態:保留中
実行番号:XXXX
後続: データベース
ファイル監視
スクリプト
Var
Del c:¥data¥infile.dat
Copy c:¥stage¥infile.dat c:¥data¥infile.dat
104
プログラム名:
TransformA.exe
118
ジョブ Psi
100
スクリプト
コマンド1
102
データベース
106
ファイル監視
110
FTP
112
実行
114
モニタ完了
116
Select us.order.
us.order_amount from
ne_production;
先行:スクリプト
コマンド1
後続:FTP
108
ファイル:Brazil Feed.dat
位置:server.com
120

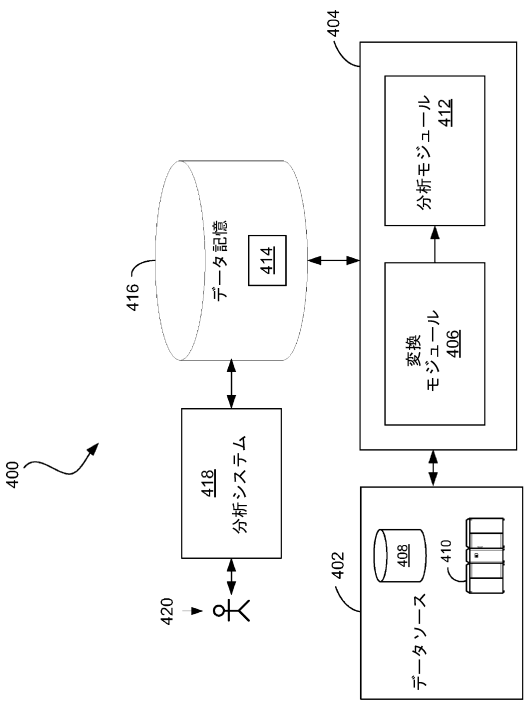
【図２】


200
出力報告
218
変換 B
216
中間データ
セット1
206
中間データ
セット2
212
変換 A
204
変換 C
210
U.S. Feed
202
Mexico Feed
208
Brazil Feed
214

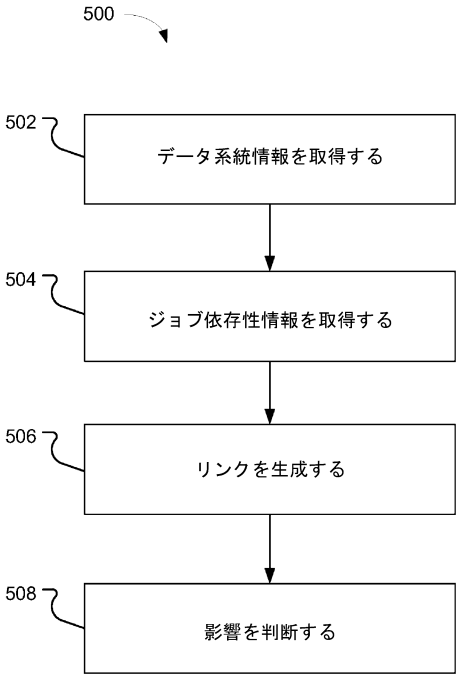
【図３】


312
情報処理
システム
310
監査および
報告システム
308
データ系統
リポジトリ
314
変換
コンポーネント
316
スケジューリング
リポジトリ
302b
変換
コンポーネント
304b
スケジューリング
リポジトリ
302a
変換
コンポーネント
304a
結合された
リポジトリ
306

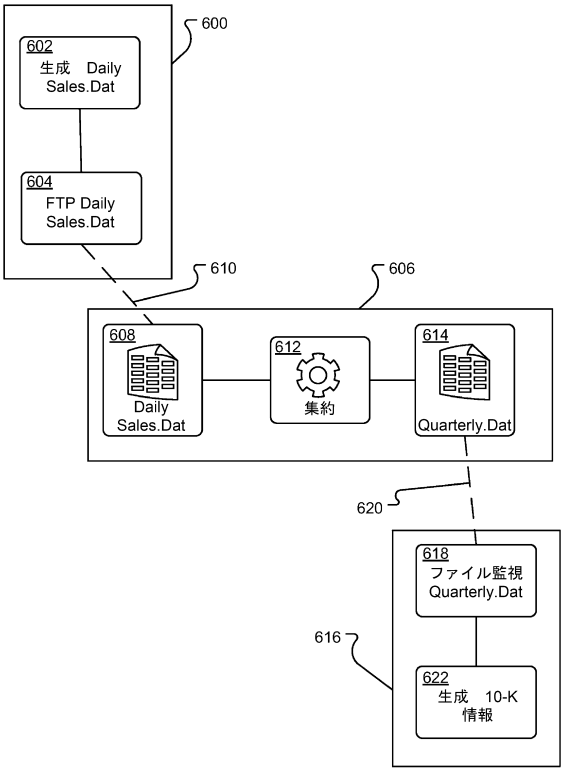
【図４】


400
データ記憶
414
416
418
分析システム
420
404
分析モジュール
412
変換
モジュール
406
データソース
408
410
402

【図５】


500
502
データ系統情報を取得する
504
ジョブ依存性情報を取得する
506
リンクを生成する
508
影響を判断する

【図６】


600
602
生成 Daily Sales.Dat
604
FTP Daily Sales.Dat
610
606
608
Daily Sales.Dat
612
集約
614
Quarterly.Dat
620
618
ファイル監視 Quarterly.Dat
616
622
生成 10-K 情報

フロントページの続き

(72)発明者　ウォルフソン，ハリー　マイケル
アメリカ合衆国，マサチューセッツ州　０２４７４，アーリントン，メルローズ　ストリート　３６
(72)発明者　ゴウルド，ジョエル
アメリカ合衆国，マサチューセッツ州　０２４７４，アーリントン，リー　テラス　２７
(72)発明者　イエラカリス，アンソニー
アメリカ合衆国，マサチューセッツ州　０２４６０，ニュートン，ワイルドウッド　アベニュー　６７
(72)発明者　ウェイクリング，ティム
アメリカ合衆国，マサチューセッツ州　０１８１０，アンドーバー，アボット　ストリート　１１

審査官　加藤　優一

(56)参考文献　米国特許出願公開第２００９／０２４１１１７（ＵＳ，Ａ１）
特開２００９－１６３５６６（ＪＰ，Ａ）
特開２００１－０２２６９５（ＪＰ，Ａ）
特開２００６－２６８５０９（ＪＰ，Ａ）
特表２００８－５０７００８（ＪＰ，Ａ）
特開２０００－０６６９３１（ＪＰ，Ａ）
特開平０６－０８３５９８（ＪＰ，Ａ）
特開２００２－３４２３１７（ＪＰ，Ａ）
特開２００９－２６８１０３（ＪＰ，Ａ）
特開２００６－１２００２１（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　　９／４５５－　９／５４
Ｇ０６Ｆ　１２／００
Ｇ０６Ｆ　　８／００