

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6419081号  
(P6419081)

(45) 発行日 平成30年11月7日 (2018. 11. 7)

(24) 登録日 平成30年10月19日 (2018. 10. 19)

(51) Int. Cl.

F I

G 0 6 F 8/35 (2018. 01)

G 0 6 F 8/35

G 0 6 F 8/41 (2018. 01)

G 0 6 F 8/41 1 3 0

請求項の数 46 (全 33 頁)

(21) 出願番号 特願2015-545913 (P2015-545913)  
 (86) (22) 出願日 平成25年12月9日 (2013. 12. 9)  
 (65) 公表番号 特表2016-510442 (P2016-510442A)  
 (43) 公表日 平成28年4月7日 (2016. 4. 7)  
 (86) 国際出願番号 PCT/US2013/073899  
 (87) 国際公開番号 W02014/093232  
 (87) 国際公開日 平成26年6月19日 (2014. 6. 19)  
 審査請求日 平成28年12月9日 (2016. 12. 9)  
 (31) 優先権主張番号 61/735, 451  
 (32) 優先日 平成24年12月10日 (2012. 12. 10)  
 (33) 優先権主張国 米国 (US)  
 (31) 優先権主張番号 61/751, 814  
 (32) 優先日 平成25年1月11日 (2013. 1. 11)  
 (33) 優先権主張国 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】 トランスフォーム生成システム

(57) 【特許請求の範囲】

【請求項 1】

データを変換するための規則セットをコード化する、1つ以上のデータ処理装置によって実行される方法であって、

実行ケースのシーケンスの内部の少なくとも1つの実行ケースが、1つ以上のトリガ条件と、前記1つ以上のトリガ条件がすべて満たされると生成される出力の仕様とを含む前記実行ケースのシーケンスを含む規則セットを受信することと、

前記規則セット内の1つ以上の実行ケースに対応する行のシーケンスを含む制御構造を生成することであって、各行が、1つ以上のトリガ条件のシーケンスと、対応する実行ケースの出力を規定する情報とを含み、前記生成された制御構造が、前記トリガ条件の1つが失敗したときに異なる行で継続するように処理を誘導するように構成され、前記生成された制御構造が、前記制御構造内の前記トリガ条件の少なくとも1つについて、前記少なくとも1つのトリガ条件が失敗したときに、前記制御構造が、前記失敗したトリガ条件を有する行の前記シーケンス内の少なくとも1つの残りの行をスキップさせるように処理を誘導するように構成されたことと、

前記制御構造を記憶又は送信することと、

を含む、方法。

【請求項 2】

入力データを受信することと、

前記制御構造を用いて決定したシーケンス内の前記入力データに照らしてトリガ条件を

検査することと、

前記制御構造によって規定された出力に基づいてデータを記憶又は送信することと、  
をさらに含む、請求項 1 に記載の方法。

【請求項 3】

前記行の少なくとも 1 つが、前記対応する実行ケースのトリガ条件を除外し、前記除外されたトリガ条件が実行ケースの前記シーケンス内の前記対応する実行ケースより前の実行ケース内で発生する、請求項 1 に記載の方法。

【請求項 4】

行内のトリガ条件の前記シーケンスが、各々が前記規則セットの一意的なトリガ条件のリスト内のあるトリガ条件へ処理を誘導するコードの部分のシーケンスである、請求項 1 に記載の方法。

10

【請求項 5】

行内の前記出力を規定する前記情報が、前記規則セットの一意的な出力のリスト内の出力式へ処理を誘導するコードの部分である、請求項 1 に記載の方法。

【請求項 6】

データ処理中に前記シーケンス内のトリガ条件が失敗したときに処理が誘導される異なる行に基づいて行のトリガ条件の前記シーケンスをソートすることをさらに含む、請求項 1 に記載の方法。

【請求項 7】

前記トリガ条件の実行時間に基づいて行のトリガ条件の前記シーケンスをソートすることをさらに含む、請求項 1 に記載の方法。

20

【請求項 8】

入力データを受信することと、

前記制御構造を用いて決定したシーケンス内の前記入力データに照らしてトリガ条件を検査することと、

前記入力データで前記トリガ条件を実行するのにかかる時間に基づいて一意的なトリガ条件のリスト内のトリガ条件の前記実行時間を更新することと、

前記更新された実行時間に基づいて前記制御構造内の行のトリガ条件へのポインタをソートすることと、

をさらに含む、請求項 7 に記載の方法。

30

【請求項 9】

トリガ条件の失敗率に基づいて行のトリガ条件の前記シーケンスをソートすることをさらに含む、請求項 1 に記載の方法。

【請求項 10】

入力データを受信することと、

前記制御構造を用いて決定したシーケンス内の前記入力データに照らしてトリガ条件を検査することと、

前記入力データ内のレコードによって前記トリガ条件が満たされるか否かに基づいて一意的なトリガ条件のリスト内のトリガ条件の失敗率を更新することと、

前記更新された失敗率に基づいて前記制御構造内の行のトリガ条件へのポインタをソートすることと、

をさらに含む、請求項 9 に記載の方法。

40

【請求項 11】

前記制御構造の行が、前記行のすべての前記トリガ条件が満たされると次に処理する前記制御構造の異なる行へ処理を誘導するコードの部分をさらに含む、請求項 1 に記載の方法。

【請求項 12】

前記規則セットが、グラフィカルユーザインターフェイスを介して規定される、請求項 1 に記載の方法。

【請求項 13】

50

前記規則セット内の実行ケースの少なくとも2つのトリガ条件が組み合わされて、前記制御構造内の単一のトリガ条件で表される、請求項1に記載の方法。

【請求項14】

前記規則セット内の異なる実行ケースの少なくとも2つの出力が組み合わされて、前記制御構造内の行内の単一の出力式で表される、請求項1に記載の方法。

【請求項15】

前記制御構造が、ノードが前記制御構造の行内の前記トリガ条件及び出力式に対応する非周期的有向グラフである、請求項1に記載の方法。

【請求項16】

データを変換するための規則セットをコード化するソフトウェアを記憶したコンピュータ可読媒体であって、前記ソフトウェアが、コンピューティングシステムに、

実行ケースのシーケンスの内部の少なくとも1つの実行ケースが、1つ以上のトリガ条件と、前記1つ以上のトリガ条件がすべて満たされると生成される出力の仕様とを含む前記実行ケースのシーケンスを含む規則セットを受信させ、

前記規則セット内の1つ以上の実行ケースに対応する行のシーケンスを含む制御構造を生成させ、各行が、1つ以上のトリガ条件のシーケンスと、対応する実行ケースの出力を規定する情報とを含み、前記生成された制御構造が、入力データを変換する将来の処理中に、前記トリガ条件の1つが失敗したときに異なる行で継続するように処理を誘導するように構成され、前記生成された制御構造が、前記制御構造内の前記トリガ条件の少なくとも1つについて、前記少なくとも1つのトリガ条件が失敗したときに、前記制御構造が、前記失敗したトリガ条件を有する行の前記シーケンス内の少なくとも1つの残りの行をスキップさせるように処理を誘導するように構成され、

前記制御構造を記憶又は送信させる命令を含む、媒体。

【請求項17】

コンピューティングシステムに、

入力データを受信させ、

前記制御構造を用いて決定したシーケンス内の入力データに照らしてトリガ条件を検査させ、

前記制御構造によって規定された出力に基づいてデータを記憶又は送信させる命令を含む、請求項16に記載の媒体。

【請求項18】

前記行の少なくとも1つが、前記対応する実行ケースのトリガ条件を除外し、前記除外されたトリガ条件が実行ケースのシーケンス内の前記対応する実行ケースより前の実行ケース内で発生する、請求項16に記載の媒体。

【請求項19】

行内のトリガ条件の前記シーケンスが、各々が前記規則セットの一意的なトリガ条件のリスト内のあるトリガ条件へ処理を誘導するコードの部分のシーケンスである、請求項16に記載の媒体。

【請求項20】

行内の前記出力を規定する前記情報が、前記規則セットの一意的な出力のリスト内の出力式へ処理を誘導するコードの部分である、請求項16に記載の媒体。

【請求項21】

コンピューティングシステムに、

データ処理中に前記シーケンス内のトリガ条件が失敗したときに処理が誘導される異なる行に基づいて行のトリガ条件の前記シーケンスをソートさせる命令を含む、請求項16に記載の媒体。

【請求項22】

コンピューティングシステムに、

前記トリガ条件の実行時間に基づいて行のトリガ条件の前記シーケンスをソートさせる命令を含む、請求項16に記載の媒体。

10

20

30

40

50

## 【請求項 2 3】

コンピューティングシステムに、  
入力データを受信させ、

前記制御構造を用いて決定したシーケンス内の前記入力データに照らしてトリガ条件を検査させ、

前記入力データで前記トリガ条件を実行するのにかかる時間に基づいて一意的なトリガ条件のリスト内のトリガ条件の実行時間を更新させ、

前記更新された実行時間に基づいて前記制御構造内の行のトリガ条件へのポインタをソートさせる命令を含む、請求項 2 2 に記載の媒体。

## 【請求項 2 4】

コンピューティングシステムに、

前記トリガ条件の失敗率に基づいて行のトリガ条件の前記シーケンスをソートさせる命令を含む、請求項 1 6 に記載の媒体。

## 【請求項 2 5】

コンピューティングシステムに、  
入力データを受信させ、

前記制御構造を用いて決定したシーケンス内の前記入力データに照らしてトリガ条件を検査させ、

前記入力データ内のレコードによって前記トリガ条件が満たされるか否かに基づいて一意的なトリガ条件のリスト内のトリガ条件の失敗率を更新させ、

前記更新された失敗率に基づいて前記制御構造内の行のトリガ条件へのポインタをソートさせる命令を含む、請求項 2 4 に記載の媒体。

## 【請求項 2 6】

前記制御構造の行が、前記行のすべてのトリガ条件が満たされると、次に処理する前記制御構造の異なる行へ処理を誘導するコードの部分を含み、請求項 1 6 に記載の媒体。

## 【請求項 2 7】

前記規則セットがグラフィカルユーザインターフェイスを介して規定される、請求項 1 6 に記載の媒体。

## 【請求項 2 8】

前記規則セット内の実行ケースの少なくとも 2 つのトリガ条件が組み合わせられて、前記制御構造内の単一のトリガ条件で表される、請求項 1 6 に記載の媒体。

## 【請求項 2 9】

前記規則セット内の異なる実行ケースの少なくとも 2 つの出力が組み合わせられて、前記制御構造の行内の単一の出力式で表される、請求項 1 6 に記載の媒体。

## 【請求項 3 0】

前記制御構造が、ノードが前記制御構造の前記行内の前記トリガ条件及び出力式に対応する非周期的有向グラフである、請求項 1 6 に記載の媒体。

## 【請求項 3 1】

データを変換するための規則セットをコード化するコンピューティングシステムであって、

実行ケースのシーケンスの内部の少なくとも 1 つの実行ケースが、1 つ以上のトリガ条件と、前記 1 つ以上のトリガ条件がすべて満たされると生成される出力の仕様とを含む前記実行ケースのシーケンスを含む規則セットを受信するように構成された入力デバイス又はポートと、

前記規則セット内の 1 つ以上の実行ケースに対応する行のシーケンスを含む制御構造を生成する手段であって、各行が、1 つ以上のトリガ条件のシーケンスと、対応する実行ケースの出力を規定する情報とを含み、前記生成された制御構造が、入力データを変換する将来の処理中に、前記トリガ条件の 1 つが失敗したときに異なる行で継続するように処理を誘導するように構成され、前記生成された制御構造が、前記制御構造内の前記トリガ条

10

20

30

40

50

件の少なくとも1つについて、前記少なくとも1つのトリガ条件が失敗したときに、前記制御構造が、前記失敗したトリガ条件を有する行の前記シーケンス内の少なくとも1つの残りの行をスキップさせるように処理を誘導するように構成された手段と、

前記制御構造を記憶するように構成されたデータ記憶システムと、  
を含む、コンピューティングシステム。

【請求項32】

入力データを受信するように構成された入力デバイス又はポートと、

前記制御構造を用いて決定したシーケンス内の前記入力データに照らしてトリガ条件を検査することを含む動作を実行するように構成された少なくとも1つのプロセッサと、

前記制御構造によって規定された出力に基づいてデータを送信するように構成された出力デバイス又はポートとを含む、請求項31に記載のシステム。

10

【請求項33】

前記行の少なくとも1つが、前記対応する実行ケースのトリガ条件を除外し、前記除外されたトリガ条件が、実行ケースの前記シーケンス内の前記対応する実行ケースより前の実行ケース内で発生する、請求項31に記載のシステム。

【請求項34】

行内のトリガ条件の前記シーケンスが、各々が前記規則セットの一意的なトリガ条件のリスト内のあるトリガ条件へ処理を誘導するコードの部分のシーケンスである、請求項31に記載のシステム。

【請求項35】

行内の前記出力を規定する前記情報が、前記規則セットの一意的な出力のリスト内の出力式へ処理を誘導するコードの部分である、請求項31に記載のシステム。

20

【請求項36】

データ処理中に前記シーケンス内のトリガ条件が失敗したときに処理が誘導される異なる行に基づいて行のトリガ条件の前記シーケンスをソートする手段を含む、請求項31に記載のシステム。

【請求項37】

前記トリガ条件の実行時間に基づいて行のトリガ条件の前記シーケンスをソートする手段を含む、請求項31に記載のシステム。

【請求項38】

入力データを受信するように構成された入力デバイス又はポートと、

動作を実行するように構成された少なくとも1つのプロセッサであって、前記動作が、

前記制御構造を用いて決定したシーケンス内の前記入力データに照らしてトリガ条件を検査することと、

前記入力データでトリガ条件を実行するのにかかる時間に基づいて一意的なトリガ条件のリスト内のトリガ条件の前記実行時間を更新することを含む少なくとも1つのプロセッサと、

前記更新された実行時間に基づいて前記制御構造内の行のトリガ条件へのポインタをソートする手段と、

を含む、請求項37に記載のシステム。

40

【請求項39】

トリガ条件の失敗率に基づいて行の前記トリガ条件の前記シーケンスをソートする手段を含む、請求項31に記載のシステム。

【請求項40】

前記コンピューティングシステムが、

入力データを受信するように構成された入力デバイス又はポートと、

複数の動作を実行するように構成された少なくとも1つのプロセッサであって、前記複数の動作が、前記制御構造を用いて決定したシーケンス内の前記入力データに照らしてトリガ条件を検査することと、前記入力データ内のレコードによって前記トリガ条件が満たされるか否かに基づいて一意的なトリガ条件のリスト内のトリガ条件の失敗率を更新する

50

こととを含む、少なくとも1つのプロセッサと、

前記更新された失敗率に基づいて前記制御構造内の行のトリガ条件へのポインタをソートする手段と、

をさらに含む、請求項39に記載のシステム。

【請求項41】

前記制御構造の行が、前記行のすべての前記トリガ条件が満たされると次に処理する前記制御構造の異なる行へ処理を誘導するコードの部分を含む、請求項31に記載のシステム。

【請求項42】

前記規則セットが、グラフィカルユーザインターフェイスを介して規定される、請求項31に記載のシステム。

【請求項43】

前記規則セット内の実行ケースの少なくとも2つのトリガ条件が組み合わされて、前記制御構造内の単一のトリガ条件で表される、請求項31に記載のシステム。

【請求項44】

前記規則セット内の異なる実行ケースの少なくとも2つの出力が組み合わされて、前記制御構造の行内の単一の出力式で表される、請求項31に記載のシステム。

【請求項45】

前記制御構造が、ノードが前記制御構造の前記行内の前記トリガ条件及び出力式に対応する非周期的有向グラフである、請求項31に記載のシステム。

【請求項46】

前記制御構造を生成する前記手段は、少なくとも1つのプロセッサを含む、請求項31から45のいずれか一項に記載のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

(関連出願の相互参照)

本出願は、2013年8月2日出願の米国特許出願第13/958,037号の優先権を主張し、これは2013年1月11日出願の米国仮出願第61/751,814号及び2012年12月10日出願の米国仮出願第61/735,451号の優先権を主張し、参照によりその全体が本明細書に組み込まれる。

【0002】

本明細書は、規則セットに基づくデータのトランスフォームを生成するシステムに関する。

【0003】

複雑な演算は、多くの場合、演算のコンポーネントがグラフの頂点に関連付けられ、コンポーネント間のデータフローがグラフのリンク(アーク、エッジ)に対応する有向グラフ(「データフローグラフ」と呼ばれる)によるデータフローとして表現することができる。コンポーネントは、1つ以上の入力ポートでデータを受信し、データを処理し、1つ以上の出力ポートからデータを提供するデータ処理コンポーネントと、データフローのソース又はシンクの役割を果たすデータセットコンポーネントとを含んでもよい。そのようなグラフベースの演算を実施するシステムは、米国特許第5,966,072号「EXECUTING COMPUTATIONS EXPRESSED AS GRAPHS」に記載されている。

【発明の概要】

【0004】

一般的な態様1では、データを変換するための規則セットをコード化する、1つ以上のデータ処理装置によって実行される方法は、その内部の少なくとも1つの実行ケースが、1つ以上のトリガ条件と、1つ以上のトリガ条件がすべて満たされると生成される出力の仕様とを含む実行ケースのシーケンスを含む規則セットを受信することと、規則セット内の1つ以上の実行ケースに対応する行のシーケンスを含む制御構造を生成することであっ

10

20

30

40

50

て、各行が、1つ以上のトリガ条件のシーケンスと、対応する実行ケースの出力を規定する情報とを含むこととを含み、生成された制御構造が、入力データを変換する将来の処理中に、トリガ条件の1つが失敗したときに異なる行で継続するように処理を誘導するように構成され、生成された制御構造が、制御構造内のトリガ条件の少なくとも1つについて、少なくとも1つのトリガ条件が失敗したときに、制御構造が、行のシーケンス内の少なくとも1つの行をスキップさせるように処理を誘導するように構成されることと、制御構造を記憶又は送信することとを含む。

【0005】

入力データを受信することと、制御構造を用いて決定したシーケンス内の入力データに照らしてトリガ条件を検査することと、制御構造によって規定された出力に基づいてデータを記憶又は送信することとをさらに含む、態様1に記載の態様2。

10

【0006】

行の少なくとも1つが、対応する実行ケースのトリガ条件を除外し、除外されたトリガ条件が実行ケースのシーケンス内の対応する実行ケースより前の実行ケース内で発生する、態様1～2のいずれか1項に記載の態様3。

【0007】

行内のトリガ条件のシーケンスが、各々が規則セットの一意的なトリガ条件のリスト内のあるトリガ条件へ処理を誘導するコードの部分のシーケンスである、態様1～3のいずれか1項に記載の態様4。

【0008】

20

行内の出力を規定する情報が、規則セットの一意的な出力のリスト内の出力式へ処理を誘導するコードの部分である、態様1～4のいずれか1項に記載の態様5。

【0009】

データ処理中にシーケンス内のトリガ条件が失敗したときに処理が誘導される異なる行に基づいて行のトリガ条件のシーケンスをソートすることをさらに含む、態様1～5のいずれか1項に記載の態様6。

【0010】

トリガ条件の実行時間に基づいて行のトリガ条件のシーケンスをソートすることをさらに含む、態様1～6のいずれか1項に記載の態様7。

【0011】

30

入力データを受信することと、制御構造を用いて決定したシーケンス内の入力データに照らしてトリガ条件を検査することと、入力データでトリガ条件を実行するのにかかる時間に基づいて一意的なトリガ条件のリスト内のトリガ条件の実行時間を更新することと、更新された実行時間に基づいて制御構造内の行のトリガ条件へのポインタをソートすることとをさらに含む、態様1～7のいずれか1項に記載の態様8。

【0012】

トリガ条件の失敗率に基づいて行のトリガ条件のシーケンスをソートすることをさらに含む、態様1～8のいずれか1項に記載の態様9。

【0013】

入力データを受信することと、制御構造を用いて決定したシーケンス内の入力データに照らしてトリガ条件を検査することと、入力データ内のレコードによってトリガ条件が満たされるか否かに基づいて一意的なトリガ条件のリスト内のトリガ条件の失敗率を更新することと、更新された失敗率に基づいて制御構造内の行のトリガ条件へのポインタをソートすることとをさらに含む、態様1～9のいずれか1項に記載の態様10。

40

【0014】

制御構造の行が、行のすべてのトリガ条件が満たされると次に処理する制御構造の異なる行へ処理を誘導するコードの部分とをさらに含む、態様1～10のいずれか1項に記載の態様11。

【0015】

規則セットが、グラフィカルユーザインターフェイスを介して規定される、態様1～1

50

1のいずれか1項に記載の態様12。

【0016】

規則セット内の実行ケースの少なくとも2つのトリガ条件が組み合わされて、制御構造内の単一のトリガ条件で表される、態様1～12のいずれか1項に記載の態様13。

【0017】

規則セット内の異なる実行ケースの少なくとも2つの出力が組み合わされて、制御構造内の行内の単一の出力式で表される、態様1～13のいずれか1項に記載の態様14。

【0018】

制御構造が、ノードが制御構造の行内のトリガ条件及び出力式に対応する非周期的有向グラフである、態様1～14のいずれか1項に記載の態様15。

10

【0019】

一般的な態様16では、データ処理装置と、データ処理装置に結合されたメモリとを含むシステムである。データ処理装置によって実行されると、データ処理装置に、その内部の少なくとも1つの実行ケースが1つ以上のトリガ条件と、1つ以上のトリガ条件がすべて満たされると生成される出力の仕様とを含む実行ケースのシーケンスを含む規則セットを受信することを含む動作を実行させる命令がその上に記憶されたメモリ。動作は、規則セット内の1つ以上の実行ケースに対応する行のシーケンスを含む制御構造を生成することをさらに含んでもよく、各行は、1つ以上のトリガ条件のシーケンスと、対応する実行ケースの出力を規定する情報とを含み、生成された制御構造は、入力データを変換する将来の処理中に、トリガ条件の1つが失敗したときに異なる行で継続するように処理を誘導するように構成され、生成された制御構造は、制御構造内のトリガ条件の少なくとも1つについて、少なくとも1つのトリガ条件が失敗したときに、制御構造が行のシーケンス内の少なくとも1つの行をスキップさせるように処理を誘導するように構成される。動作は、制御構造を記憶又は送信することをさらに含んでもよい。

20

【0020】

動作が、入力データを受信することと、制御構造を用いて決定したシーケンス内の入力データに照らしてトリガ条件を検査することと、制御構造によって規定された出力に基づいてデータを記憶又は送信することとをさらに含む、態様16に記載の態様17。

【0021】

行の少なくとも1つが、対応する実行ケースのトリガ条件を除外し、除外されたトリガ条件が実行ケースのシーケンス内の対応する実行ケースより前の実行ケース内で発生する、態様16～17のいずれか1項に記載の態様18。

30

【0022】

行内のトリガ条件のシーケンスが、各々が規則セットの一意的なトリガ条件のリスト内のあるトリガ条件へ処理を誘導するコードの部分のシーケンスである、態様16～18のいずれか1項に記載の態様19。

【0023】

行内の出力を規定する情報が、規則セットの一意的な出力のリスト内の出力式へ処理を誘導するコードの部分である、態様16～19のいずれか1項に記載の態様20。

【0024】

40

動作が、データ処理中にシーケンス内のトリガ条件が失敗したときに処理が誘導される異なる行に基づいて行のトリガ条件のシーケンスをソートすることをさらに含む、態様16～20のいずれか1項に記載の態様21。

【0025】

動作が、トリガ条件の実行時間に基づいて行のトリガ条件のシーケンスをソートすることをさらに含む、態様16～21のいずれか1項に記載の態様22。

【0026】

動作が、入力データを受信することと、制御構造を用いて決定したシーケンス内の入力データに照らしてトリガ条件を検査することと、入力データでトリガ条件を実行するのにかかる時間に基づいて一意的なトリガ条件のリスト内のトリガ条件の実行時間を更新する

50



ことと、更新された実行時間に基づいて制御構造内の行のトリガ条件へのポインタをソートすることとをさらに含む、態様 16 ~ 22 のいずれか 1 項に記載の態様 23。

【0027】

動作が、トリガ条件の失敗率に基づいて行のトリガ条件のシーケンスをソートすることとをさらに含む、態様 16 ~ 23 のいずれか 1 項に記載の態様 24。

【0028】

動作が、入力データを受信することと、制御構造を用いて決定したシーケンス内の入力データに照らしてトリガ条件を検査することと、入力データ内のレコードによってトリガ条件が満たされるか否かに基づいて一意的なトリガ条件のリスト内のトリガ条件の失敗率を更新することと、更新された失敗率に基づいて制御構造内の行のトリガ条件へのポインタをソートすることとをさらに含む、態様 16 ~ 24 のいずれか 1 項に記載の態様 25。

10

【0029】

制御構造の行が、行のすべてのトリガ条件が満たされると次に処理する制御構造の異なる行へ処理を誘導するコードの部分とをさらに含む、態様 16 ~ 25 のいずれか 1 項に記載の態様 26。

【0030】

規則セットがグラフィカルユーザインターフェイスを介して規定される、態様 16 ~ 26 のいずれか 1 項に記載の態様 27。

【0031】

規則セット内の実行ケースの少なくとも 2 つのトリガ条件が組み合わせられて、制御構造内の単一のトリガ条件で表される、態様 16 ~ 27 のいずれか 1 項に記載の態様 28。

20

【0032】

規則セット内の異なる実行ケースの少なくとも 2 つの出力が組み合わせられて、制御構造の行内の単一の出力式で表される、態様 16 ~ 28 のいずれか 1 項に記載の態様 29。

【0033】

制御構造が、ノードが制御構造の行内のトリガ条件及び出力式に対応する非周期的有向グラフである、態様 16 ~ 29 のいずれか 1 項に記載の態様 30。

【0034】

一般的な態様 31 では、処理デバイスによって実行可能な命令であって、それが実行されると、処理デバイスに、その内部の少なくとも 1 つの実行ケースが 1 つ以上のトリガ条件と、1 つ以上のトリガ条件がすべて満たされると生成される出力の仕様とを含む実行ケースのシーケンスを含む規則セットを受信することを含む動作を実行させる命令を含むソフトウェアを記憶したコンピュータ可読記憶媒体。動作は、規則セット内の 1 つ以上の実行ケースに対応する行のシーケンスを含む制御構造を生成することとをさらに含んでもよく、各行は、1 つ以上のトリガ条件のシーケンスと、対応する実行ケースの出力を規定する情報とを含み、生成された制御構造は、入力データを変換する将来の処理中に、トリガ条件の 1 つが失敗したときに異なる行で継続するように処理を誘導するように構成され、生成された制御構造は、制御構造内のトリガ条件の少なくとも 1 つについて、少なくとも 1 つのトリガ条件が失敗したときに、制御構造が行のシーケンス内の少なくとも 1 つの行をスキップさせるように処理を誘導するように構成される。動作は、制御構造を記憶又は送信することとをさらに含んでもよい。

30

40

【0035】

動作が、入力データを受信することと、制御構造を用いて決定したシーケンス内の入力データに照らしてトリガ条件を検査することと、制御構造によって規定された出力に基づいてデータを記憶又は送信することとをさらに含む、態様 31 に記載の態様 32。

【0036】

行の少なくとも 1 つが、対応する実行ケースのトリガ条件を除外し、除外されたトリガ条件が、実行ケースのシーケンス内の対応する実行ケースより前の実行ケース内で発生する、態様 31 ~ 32 のいずれか 1 項に記載の態様 33。

【0037】

50

行内のトリガ条件のシーケンスが、各々が規則セットの一意的なトリガ条件のリスト内のあるトリガ条件へ処理を誘導するコードの部分のシーケンスである、態様 3 1 ~ 3 3 のいずれか 1 項に記載の態様 3 4。

【 0 0 3 8 】

行内の出力を規定する情報が、規則セットの一意的な出力のリスト内の出力式へ処理を誘導するコードの部分である、態様 3 1 ~ 3 4 のいずれか 1 項に記載の態様 3 5。

【 0 0 3 9 】

動作が、データ処理中にシーケンス内のトリガ条件が失敗したときに処理が誘導される異なる行に基づいて行のトリガ条件のシーケンスをソートすることをさらに含む、態様 3 1 ~ 3 5 のいずれか 1 項に記載の態様 3 6。

10

【 0 0 4 0 】

動作が、トリガ条件の実行時間に基づいて行のトリガ条件のシーケンスをソートすることをさらに含む、態様 3 1 ~ 3 6 のいずれか 1 項に記載の態様 3 7。

【 0 0 4 1 】

動作が、入力データを受信することと、制御構造を用いて決定したシーケンス内の入力データに照らしてトリガ条件を検査することと、入力データでトリガ条件を実行するのにかかる時間に基づいて一意的なトリガ条件のリスト内のトリガ条件の実行時間を更新することと、更新された実行時間に基づいて制御構造内の行のトリガ条件へのポインタをソートすることとをさらに含む、態様 3 1 ~ 3 7 のいずれか 1 項に記載の態様 3 8。

【 0 0 4 2 】

20

動作が、トリガ条件の失敗率に基づいて行のトリガ条件のシーケンスをソートすることをさらに含む、態様 3 1 ~ 3 8 のいずれか 1 項に記載の態様 3 9。

【 0 0 4 3 】

動作が、入力データを受信することと、制御構造を用いて決定したシーケンス内の入力データに照らしてトリガ条件を検査することと、入力データ内のレコードによってトリガ条件が満たされるか否かに基づいて一意的なトリガ条件のリスト内のトリガ条件の失敗率を更新することと、更新された失敗率に基づいて制御構造内の行のトリガ条件へのポインタをソートすることとをさらに含む、態様 3 1 ~ 3 9 のいずれか 1 項に記載の態様 4 0。

【 0 0 4 4 】

制御構造の行が、行のすべてのトリガ条件が満たされると次に処理する制御構造の異なる行へ処理を誘導するコードの部分をさらに含む、態様 3 1 ~ 4 0 のいずれか 1 項に記載の態様 4 1。

30

【 0 0 4 5 】

規則セットが、グラフィカルユーザインターフェイスを介して規定される、態様 3 1 ~ 4 1 のいずれか 1 項に記載の態様 4 2。

【 0 0 4 6 】

規則セット内の実行ケースの少なくとも 2 つのトリガ条件が組み合わされて、制御構造内の単一のトリガ条件で表される、態様 3 1 ~ 4 2 のいずれか 1 項に記載の態様 4 3。

【 0 0 4 7 】

規則セット内の異なる実行ケースの少なくとも 2 つの出力が組み合わされて、制御構造の行内の単一の出力式で表される、態様 3 1 ~ 4 3 のいずれか 1 項に記載の態様 4 4。

40

【 0 0 4 8 】

制御構造が、ノードが制御構造の行内のトリガ条件及び出力式に対応する非周期的有向グラフである、態様 3 1 ~ 4 4 のいずれか 1 項に記載の態様 4 5。

【 0 0 4 9 】

制御構造が複数の処理デバイス上で並列に実行されるトランスフォームの一部である、態様 1 ~ 1 5 のいずれか 1 項に記載の態様 4 6。

【 0 0 5 0 】

制御構造が複数の処理デバイス上で並列に実行されるトランスフォームの一部である、態様 1 6 ~ 3 0 のいずれか 1 項に記載の態様 4 7。

50

## 【 0 0 5 1 】

制御構造が複数の処理デバイス上で並列に実行されるトランスフォームの一部である、態様 3 1 ~ 4 5 のいずれか 1 項に記載の態様 4 8。

## 【 0 0 5 2 】

一態様では、一般に、規則セットに基づいてトランスフォームを生成する方法は、その内部の少なくとも 1 つの実行ケースが 1 つ以上のトリガ条件と、 1 つ以上のトリガ条件がすべて満たされると生成される出力の仕様とを含む実行ケースのシーケンスを含む規則セットを受信することを含む。方法は、規則セット内の 1 つ以上の実行ケースに対応する行のシーケンスを含む制御構造を生成することをさらに含んでもよく、各行は、 1 つ以上のトリガ条件のシーケンスと、対応する実行ケースの出力を規定する情報とを含み、生成された制御構造は、入力データを変換する将来の処理中に、トリガ条件の 1 つが失敗したときに異なる行で継続するように処理を誘導するように構成され、生成された制御構造は、制御構造内のトリガ条件の少なくとも 1 つについて、少なくとも 1 つのトリガ条件が失敗したときに、制御構造が行のシーケンス内の少なくとも 1 つの行をスキップさせるように処理を誘導するように構成される。方法は、制御構造を記憶又は送信することをさらに含んでもよい。

10

## 【 0 0 5 3 】

一般に、本明細書に記載する主題の一態様は、データ処理装置と、データ処理装置に結合されたメモリとを含むシステム内に具現化できる。データ処理装置によって実行されると、データ処理装置に、その内部の少なくとも 1 つの実行ケースが 1 つ以上のトリガ条件と、 1 つ以上のトリガ条件がすべて満たされると生成される出力の仕様とを含む実行ケースのシーケンスを含む規則セットを受信することを含む動作を実行させる命令がその上に記憶されたメモリ。動作は、規則セット内の 1 つ以上の実行ケースに対応する行のシーケンスを含む制御構造を生成することをさらに含んでもよく、各行は、 1 つ以上のトリガ条件のシーケンスと、対応する実行ケースの出力を規定する情報とを含み、生成された制御構造は、入力データを変換する将来の処理中に、トリガ条件の 1 つが失敗したときに異なる行で継続するように処理を誘導するように構成され、生成された制御構造は、制御構造内のトリガ条件の少なくとも 1 つについて、少なくとも 1 つのトリガ条件が失敗したときに、制御構造が、行のシーケンス内の少なくとも 1 つの行をスキップさせるように処理を誘導するように構成される。動作は、制御構造を記憶又は送信することをさらに含んでもよい。

20

30

## 【 0 0 5 4 】

一般に、本明細書に記載する主題の一態様は、処理デバイスによって実行可能な命令であって、それが実行されると、処理デバイスに、その内部の少なくとも 1 つの実行ケースが 1 つ以上のトリガ条件と、 1 つ以上のトリガ条件がすべて満たされると生成される出力の仕様とを含む実行ケースのシーケンスを含む規則セットを受信することを含む動作を実行させる命令を含むソフトウェアを記憶したコンピュータ可読記憶媒体内に具現化できる。動作は、規則セット内の 1 つ以上の実行ケースに対応する行のシーケンスを含む制御構造を生成することをさらに含んでもよく、各行は、 1 つ以上のトリガ条件のシーケンスと、対応する実行ケースの出力を規定する情報とを含み、生成された制御構造は、入力データを変換する将来の処理中に、トリガ条件の 1 つが失敗したときに異なる行で継続するように処理を誘導するように構成され、生成された制御構造は、制御構造内のトリガ条件の少なくとも 1 つについて、少なくとも 1 つのトリガ条件が失敗したときに、制御構造が行のシーケンス内の少なくとも 1 つの行をスキップさせるように処理を誘導するように構成される。動作は、制御構造を記憶又は送信することをさらに含んでもよい。

40

## 【 0 0 5 5 】

一般に、本明細書に記載する主題の一態様は、その内部の少なくとも 1 つの実行ケースが 1 つ以上のトリガ条件と、 1 つ以上のトリガ条件がすべて満たされると生成される出力の仕様とを含む実行ケースのシーケンスを含む規則セットを受信するように構成された入力デバイス又はポートを含むシステム内に具現化できる。システムは、規則セット内の 1

50

つ以上の実行ケースに対応する行のシーケンスを含む制御構造を生成する手段を含んでもよく、各行は、1つ以上のトリガ条件のシーケンスと、対応する実行ケースの出力を規定する情報とを含み、生成された制御構造は、入力データを変換する将来の処理中に、トリガ条件の1つが失敗したときに異なる行で継続するように処理を誘導するように構成され、生成された制御構造は、制御構造内のトリガ条件の少なくとも1つについて、少なくとも1つのトリガ条件が失敗したときに、制御構造が行のシーケンス内の少なくとも1つの行をスキップさせるように処理を誘導するように構成される。システムは、制御構造を記憶するように構成されたデータ記憶システムを含んでもよい。

【0056】

一般に、本明細書に記載する主題の一態様は、その内部の少なくとも1つの実行ケースが、1つ以上のトリガ条件と、1つ以上のトリガ条件がすべて満たされると生成される出力の仕様とを含む実行ケースのシーケンスを含む規則セットを受信するように構成された入力デバイス又はポートを含むシステム内に具現化できる。システムは、規則セット内の1つ以上の実行ケースに対応する行のシーケンスを含む制御構造を生成することを含む動作を実行するように構成された少なくとも1つのプロセッサを含んでもよく、各行は、1つ以上のトリガ条件のシーケンスと、対応する実行ケースの出力を規定する情報とを含み、生成された制御構造は、入力データを変換する将来の処理中に、トリガ条件の1つが失敗したときに異なる行で継続するように処理を誘導するように構成され、生成された制御構造は、制御構造内のトリガ条件の少なくとも1つについて、少なくとも1つのトリガ条件が失敗したときに、制御構造が行のシーケンス内の少なくとも1つの行をスキップさせるように処理を誘導するように構成される。システムは、制御構造を送信するように構成された出力デバイス又はポートを含んでもよい。

【0057】

態様は、以下の特徴の1つ以上を含んでもよい。入力データを受信し、制御構造を用いて決定したシーケンス内の入力データに照らしてトリガ条件を検査することができる。制御構造によって規定される出力に基づくデータを記憶又は送信できる。行の少なくとも1つは対応する実行ケースについてトリガ条件を除外することができ、除外されたトリガ条件は実行ケースのシーケンス内の対応する実行ケースに先立つ実行ケース内で発生する。行内のトリガ条件のシーケンスは、各々が規則セットの一意的なトリガ条件のリスト内のトリガ条件へ処理を誘導するコードの部分のシーケンスであってもよい。行内の出力を規定する情報は、規則セットの一意的な出力のリスト内の出力へ処理を誘導するコードの部分であってもよい。行のトリガ条件のシーケンスは、データの処理の実行中にシーケンス内のトリガ条件が失敗したときに処理が誘導される異なる行に基づいてソートすることができる。行のトリガ条件のシーケンスは、トリガ条件の実行時間に基づいてソートすることができる。一意的なトリガ条件のリスト内のトリガ条件の実行時間は、入力データでトリガ条件を実行するのにかかる時間に基づいて更新できる。制御構造内の行のトリガ条件へのポインタは、更新された実行時間に基づいてソートできる。行のトリガ条件のシーケンスは、トリガ条件の失敗率に基づいてソートできる。一意的なトリガ条件のリスト内のトリガ条件の失敗率は、トリガ条件が入力データ内のレコードによって満たされるか否かに基づいて更新できる。制御構造内の行のトリガ条件へのポインタは、更新された失敗率に基づいてソートできる。制御構造の行は、行のトリガ条件がすべて満たされると次に処理される制御構造の異なる行へ処理を誘導するコードの部分を含んでもよい。規則セットは、グラフィカルユーザインターフェイスによって規定できる。規則セット内の実行ケースの少なくとも2つのトリガ条件を組み合わせ、制御構造内の単一のトリガ条件で表すことができる。規則セット内の異なる実行ケースの少なくとも2つの出力を組み合わせ、制御構造の行内の単一の出力で表すことができる。制御構造は、ノードが制御構造の行内のトリガ条件及び出力に対応する非周期的有向グラフであってもよい。

【0058】

態様は、以下の利点の1つ以上を含んでもよい。

【0059】

10

20

30

40

50

幾つかの実施態様は、規則セットに基づいてトランスフォームのレコード当たりの処理時間を低減できる。幾つかの実施態様は、規則セットに基づいてトランスフォームの編集及び起動時間を低減できる。幾つかの実施態様は、規則セットに基づいてトランスフォームの編集にメモリ使用率を低減できる。幾つかの実施態様は、とりわけ、航空機、自動車、コンピュータ、建物、又はその他のインフラストラクチャなどの物理的エンティティを表すデータのより効率的な処理を提供できる。幾つかの実施態様は、大量のデータを扱うユーザの認識の負担を低減できる。例えば、ユーザは、大量のデータ（例えば、数百万又は数十億件のレコード）の処理をより容易に指定でき、それによって、ユーザはより容易にデータ処理を理解でき、規則セットの仕様の効率的な構造を発見する作業について心配することなく、特定のアプリケーションについてのユーザのドメインの情報を適用することができる。

10

【0060】

本発明のその他の特徴及び利点は、以下の説明と請求の範囲から明らかになるう。

【図面の簡単な説明】

【0061】

【図1】例示的なデータフローグラフの概略図である。

【図2】図2A～図2Bは、スプレッドシートベースの規則の入力のための例示的な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3A】規則セットの一意的なトリガ条件のリストの一例を示す。

【図3B】規則セットの一意的な出力のリストの一例を示す。

20

【図4】図4A～図4Bは、非周期的有向グラフとして表される規則セットに基づくトランスフォームの例示的な制御構造を示す。

【図5】表示テーブルとして表される規則セットに基づくトランスフォームの例示的な制御構造を示す。

【図6】グラフベースの演算を実行するシステムのブロック図である。

【図7】規則セットに基づくトランスフォームを生成し実行する例示的なプロセスのフローチャートである。

【図8】規則セットに基づくトランスフォームを実行する例示的なプロセスのフローチャートである。

【発明を実施するための形態】

30

【0062】

グラフベースの演算を用いて大量のデータセットを処理することができる。例えば、クレジットカードの発行会社は、グラフベースの演算を用いて数百万枚のクレジットカードの取引データを処理し、アワードポイント（award point）を発行し、アワードポイントの償還の提供で提示する提携製品を選択することができる。別の例では、航空会社は、グラフベースの演算を用いて数百万人の航空便の旅客の航空マイレージアカウントを頻繁に更新できる。別の例では、銀行は、グラフベースの演算を用いて様々な情報源から得た消費者データを処理して、特定の消費者の利用可能なデータに依存する最大量までローンのローン承認を生成することができる。別の例では、航空会社は、グラフベースの演算を用いて航空機の一団の保守を追跡し制御することができる。別の例では、レンタカー会社は、グラフベースの演算を用いて車の一団の保守を追跡し制御することができる。別の例では、オンラインサービスプロバイダは、グラフベースの演算を用いてウェブサーバの1つ以上のクラスタの保守及び／又は負荷平衡を追跡し制御することができる。別の例では、市は、道路交通データに基づいて道路交通信号灯器を制御することができる。別の例では、無線ネットワークオペレータは、グラフベースの演算を用いてパーソナル無線通信デバイス（例えば、スマートフォン又はタブレットデバイス）の無線ネットワークアクセスと帯域幅割り当てとを制御することができる。これらは、グラフベースの演算の用途のごく少数の例であり、その他多数の用途が可能である。

40

【0063】

グラフベースの演算は、入力データ内のセット内のレコード（又はその他のデータ要素

50

）に適用されるトランスフォームに対応する１つ以上のコンポーネントを含んでもよい。一般に、トランスフォームは、入力レコードを処理して１つ以上の出力レコードを生成（例えば、作成又は更新）する。例えば、クレジットカード発行会社は、クレジットカードのアカウントレコードに関連付けられた取引レコードを処理して提案されたクレジットカード取引の取引承認又は拒否レコードを生成する。トランスフォームが入力レコードを処理して出力レコードを生成する方法の詳細は複雑であり、特定のアプリケーションに関する大量のドメイン情報に依存する。ソフトウェア開発又はコーディングの情報がほとんどないユーザ（例えば、オペレータ又は開発者）が特定のアプリケーションについて有する情報に基づいてトランスフォームを容易に構成できるようにすることは有用であろう。理解しやすいインターフェイスを介して規定された規則に基づいて効率的なトランスフォームを生成するシステムは、そのようなユーザの認識の負担を低減する助けになり得る。

10

**【 0 0 6 4 】**

幾つかの実施態様では、ユーザは、スプレッドシートベースのグラフィカルユーザインターフェイス（GUI）を介して潜在的に複雑なトランスフォームを構成できる。例えば、ユーザは、各実行ケースのスプレッドシートベースのグラフィカルユーザインターフェイス内に行を作成することで、実行ケースを含む規則セットを規定できる。各々の実行ケースは、入力データ及び１つ以上の出力に照らして検査できるトリガ条件と呼ばれる１つ以上の条件を含んでいてもよい。実行ケースのトリガ条件がすべて入力データによって満たされると、その実行ケースの１つ以上の出力を生成することができる。例えば、トリガ条件及び出力は、ソフトウェアコーディング技術を欠くがドメイン情報を有し、スプレッドシートに慣れたユーザが規則セットを規定することを可能にするスプレッドシートベースのGUIのカラムに対応してもよい。幾つかの実施態様では、スプレッドシートベースのGUI以外のフォーマットを用いて規則セットを（例えば、データ操作言語（DML）で）規定することができる。

20

**【 0 0 6 5 】**

ユーザが規則セットを規定すると、規則セットに基づいて入力データレコードを処理できる１つ以上のトランスフォームを生成できる。規則セットは、複数の実行ケース内で発生するトリガ条件及び／又は出力を含んでもよい。トランスフォームは、トランスフォームを編集して入力データレコードに適用するために必要なメモリ及び処理時間を低減する規則セット内の冗長性を活用する形で生成できる。例えば、複数の実行ケース内でトリガ条件が発生した場合、入力データがそのトリガ条件を満たさないとトランスフォームはそのトリガ条件に照らして残りの実行ケースを検査することをスキップすることができる。規則セット内の冗長性を活用することで、低減されたメモリ及びプロセッサ周期を用いて大規模な入力データセットについてトランスフォームを実行することができる。トランスフォームは、一部をそのトランスフォームの実行フローを制御する制御構造としてコード化できる。制御構造は、行と呼ばれる、規則セットの実行ケースに各々が対応するトリガ条件及び出力の論理的なグループ化を含んでもよい。トランスフォームの実行中、トリガ条件が成立したと評価されると、行内のトリガ条件のシーケンス内の次のトリガ条件が評価される。入力データで評価されるときに行内の最後のトリガ条件が成立すると、少なくとも一部が行によって規定される対応する出力が生成される。入力データで評価されるときにトリガ条件が失敗する（例えば、トリガ条件が満たされない）と、制御構造は異なる実行ケースに対応する異なる行へ処理を誘導する。例えば、トリガ条件が失敗すると、実行フローはジャンプによって幾つかの実行ケースをスキップして制御構造内の行のシーケンス上の次の順番ではない行内のトリガ条件の評価を開始する。幾つかの実施態様では、制御構造は、一意的なトリガ条件及び一意的な出力のリストを参照するため、これらのトリガ条件及び出力は、規則セット内で多くの回数が発生してもトランスフォームのコード化内に１回のみ記憶すればよい。

30

40

**【 0 0 6 6 】**

幾つかの実施態様では、トリガ条件及び規則セットの仕様内で発生するトリガ条件への参照を対応するトランスフォームの制御構造から除外してもよい。例えば、同じトリガ条

50

件が、トリガ条件の現在のインスタンスが評価される前に必ず評価される規則セット内の別の場所で発生するときには、トリガ条件を制御構造の行から除外してもよい。行からトリガ条件を除外することで、対応するトランスフォームのメモリ使用率と処理時間とを低減できる。

#### 【 0 0 6 7 】

幾つかの実施態様では、制御構造の行内の又は制御構造の行によって参照されるトリガ条件を、トリガ条件のパラメータに基づいてソートすることができる。このソーティングを実行して、トランスフォームが適用されるレコードの平均処理時間を低減できる。例えば、トリガ条件の実行時間、評価時にトリガ条件が失敗するとスキップされる行数（例えば、ジャンプサイズ）、又は以前にトランスフォームで処理された入力データの失敗の周期に基づいてトリガ条件のシーケンスをソートすることができる。幾つかの実施態様では、制御構造内の行のシーケンスをソートしてトリガ条件のジャンプサイズを増加させることができる（例えば、実行ケースと共通のトリガ条件とを一緒にグループ化することで）。幾つかの実施態様では、行及び／又は行のシーケンス内のトリガ条件のシーケンスを変更することで、以前に処理したデータに類似した統計情報を有する大規模なデータセットについて対応するトランスフォームを実行するのに必要な処理時間を低減できる。

#### 【 0 0 6 8 】

本明細書では、「規則セット」という用語は、各々が1つ以上の実行ケースで構成された1つ以上の規則の集合を指す。規則内の実行ケースは、入力データの処理中に実行ケースが評価される順序を決定できる配列を有してもよい。「実行ケース」は、1つ以上の出力に関連付けられた1つ以上のトリガ条件の集合である。「トリガ条件」は、実行ケースのその他の任意のトリガ条件と共に、実行ケースが始動（fire）するか否かを決定するために使用される条件である。ある実行ケースのすべてのトリガ条件が入力データによって満たされると決定されると、実行ケースが「始動」する。すなわち、実行ケースの1つ以上の出力が生成される。出力は、規則セットの一部として規定される一定のパラメータに基づいて生成されるという意味で静的であり、又は、出力は、部分的に入力データ値及び／又は中間結果値に基づいて生成されるという意味で動的である。規則は、単一始動又は多重始動である。単一始動規則では、そのトリガ条件のすべてを満たした第1の実行ケースのみが始動する。多重始動規則では、規則を構成するすべての実行ケースが検査され、それぞれのトリガ条件のすべてが満たされるすべての実行ケースについて出力が生成される。規則セット内の多重規則は、シーケンス内のデータフローに適用できる。例えば、規則セットの第1の規則に基づくトランスフォームは、複数の入力データソースからレコードを取り出して出力レコードを生成できる。これらの出力レコードは、次いで規則セットの第2の規則に基づく第2のトランスフォームへ入力レコードとして渡され、出力レコードの第2のセットを生成できる。

#### 【 0 0 6 9 】

図1は、1つ以上のトランスフォームを含む例示的なデータフローグラフ100の概略図を示す。データは、データのフローを処理するデータフローグラフ100のデータ処理コンポーネントのシーケンスを通して1つ以上のデータソースから1つ以上のデータシンクへ渡される。データフローグラフ内の様々なデータ処理コンポーネントのいずれも、別々の処理デバイス上で実行中のプロセスによって実施でき、又は複数のデータ処理コンポーネントを単一の処理デバイス上で実行中の1つ以上のプロセスによって実施することもできる。幾つかの実施態様では、入力データレコードは、到着時に（例えば、クレジットカード取引要求に回答して）連続的に処理できる。幾つかの実施態様では、データは、データフローグラフ100によって処理される入力データレコードのセットを識別するバッチ単位で処理できる。

#### 【 0 0 7 0 】

データフローグラフ100によるデータバッチの処理は、ユーザ入力又はタイマの満了などのその他の何らかのイベントによって開始できる。データバッチの処理が開始されると、1つ以上の入力データソースから入力データレコードが読み出される。例えば、デー

10

20

30

40

50

タ記憶コンポーネント 1 1 0 によって表されるようなコンピュータ可読記憶デバイスに記憶された 1 つ以上のファイルから入力データを読み出すことができる。データ記憶コンポーネント 1 1 2 によって表されるようなサーバ上で実行中のデータベースから入力データレコードを読み出すこともできる。ジョインコンポーネント 1 2 0 は、シーケンス内の複数のデータソースからデータ（例えば、レコード）を読み出し、入力データを離散的なワークユニットのシーケンスに編成する。ワークユニットは、例えば、入力レコードに基づく所定のフォーマットに記憶されたレコードを表していてもよく、又は、例えば、処理する取引を表していてもよい。ワークユニット（例えば、レコード）は、シーケンス内でデータフローグラフ内の次のコンポーネントへ渡される。

【 0 0 7 1 】

例示的なデータフローグラフ 1 0 0 はまた、トランスフォームコンポーネント 1 3 0 及び 1 4 0 を含む。トランスフォームコンポーネント 1 3 0 によって実行されるトランスフォームは、単一始動規則に基づいている。すなわち、各ワークユニット（例えば、ジョインプロセスの入力データのレコード）について 1 つの実行ケースのみが始動する。トランスフォームコンポーネント 1 3 0 は、次のデータ処理コンポーネント、この例ではトランスフォームコンポーネント 1 4 0 へ渡される出力レコードを生成する。

【 0 0 7 2 】

トランスフォームコンポーネント 1 4 0 によって実行されるトランスフォームは、多重始動規則に基づいている。トランスフォームコンポーネント 1 4 0 によって生成される出力データレコードは、入力ワークユニットについて始動した実行ケースの各々に対応する出力値のリストを含んでもよい。

【 0 0 7 3 】

例えば、トランスフォームコンポーネント 1 3 0 は、様々なデータソースからのクレジットカード取引レコードに対応するジョインプロセス 1 2 0 からの入力データレコードを処理できる。トランスフォームコンポーネント 1 3 0 は、レコード内に反映された取引の結果としてのクレジットカードに割り当てられるアワードポイントの量を反映する出力レコードを生成できる。この例では、トランスフォームコンポーネント 1 4 0 は、次に、クレジットカードアカウントに割り当てられたアワードポイントを処理して、対応するクレジットカードアカウントの保有者に提示される 1 つ以上の提供製品を生成することができる。

【 0 0 7 4 】

トランスフォームコンポーネント 1 3 0 及びトランスフォームコンポーネント 1 4 0 は、トランスフォームコンポーネント 1 3 0 に対応する単一始動規則及びトランスフォームコンポーネント 1 4 0 に対応する多重始動規則の両方を含む規則セットに対応するより大規模なコンポーネント 1 5 0 としてグループ化することができる。

【 0 0 7 5 】

ワークユニットはデータフローグラフのデータ処理コンポーネントを通して進行するため、各ワークユニットに関連する結果出力レコードは、データシンク 1 7 0 へ移送される前にそれらが蓄積されるデータキュー 1 6 0 へ渡される。データシンク 1 7 0 は、例えば、ワークユニット又はワークユニットに基づく何らかの蓄積された出力を記憶するデータ記憶コンポーネントであってもよく、あるいは、データシンク 1 7 0 は、ワークユニットが公開される行列又は最終的な結果を受信する何か別のタイプのシンクであってもよい。幾つかの実施態様では、バッチ内のすべてのワークユニットがデータシンク 1 7 0 へ移送されるとバッチ処理が終了する。この時点で、データフローグラフ内のコンポーネントを終了することができる。

【 0 0 7 6 】

図 2 A は、スプレッドシートベースの規則の入力のための例示的な GUI 2 0 0 を示す。GUI 2 0 0 は、クレジットカードアカウントと提携する 1 つ以上のレコード内で利用可能な取引データに基づいて、クレジットカードアカウントの「アワードポイント」値を決定する単一始動規則をユーザが規定することで構成される。GUI 2 0 0 は、規則の 5

10

20

30

40

50



つの規則ケースを規定する5つの行を含む。ユーザはG U I 2 0 0を用いて、規則の実行ケースのトリガ条件及び出力を規定し、ユーザがデバッグ又はトランスフォームの調整を容易にするために使用できる規則に基づいてトランスフォームのテストランから得たメタデータを表示する。例えば、G U I 2 0 0は、図6のアプリケーションスペシャリスト環境6 2 2を介してユーザに提示できる。幾つかの実施態様では、G U I 2 0 0を介して規定された規則セットは、図6の実行環境6 0 4のネットワークインターフェイスを介して受信できる。

#### 【0 0 7 7】

第1のカラム2 0 4は、クレジットカードアカウントの平均月次請求額を反映する入力データレコード内の変数に適用されるトリガ条件を規定する。第1のカラムの3行目の下向きの矢印2 0 6は、第3の実行ケースの第1のトリガ条件が第2の実行ケースの第1のトリガ条件に対するトリガ条件と同じであることを示す。幾つかの実施態様では、ユーザは、下向きの矢印アイコンを手動で選択してスプレッドシートベースのG U I 2 0 0の1つ以上のセル内に下向きの矢印を挿入して、下向きの矢印の始点の真上のトリガ条件と同じ下向きの矢印が通過する実行ケースのトリガ条件を規定できる。幾つかの実施態様では、ユーザは、スプレッドシートベースのG U I 2 0 0の隣接するセル内に同じトリガ条件を手動で入力でき、G U I 2 0 0は、トリガ条件が同じであることを自動的に認識し、この反復を示す下向きの矢印を生成することができる。

#### 【0 0 7 8】

G U I 2 0 0の第2のカラム2 0 8は、クレジットカードアカウントがこれまで有効であった年数を反映するクレジットカードアカウントの入力データレコードから導出された変数に適用されるトリガ条件を規定する。また、カラム2 0 8は、各々が1対の一致するトリガ条件を示す2つの下向きの矢印2 1 0を含む。

#### 【0 0 7 9】

G U I 2 0 0の第3のカラム2 1 4は、ある実行ケースのすべてのトリガ条件が評価され満たされると判断されたときに生成される出力を規定する。カラム2 1 4は、また、第1の2つの実行ケースが同じ出力を有することを示す下向きの矢印2 1 6を含む。ユーザは、G U I 2 0 0を介して、これを単一始動規則2 1 8であると規定しているため、実行ケースは始動すると分かる（例えば、実行ケースのすべてのトリガ条件が入力データのワークユニットによって満たされる）まで1回に1つずつ評価できる。実行ケースが始動すると、その実行ケースについて規定された出力が生成され、この規則に基づくトランスフォームがワークユニットの処理を完了する。この時点で、トランスフォームは、データフロー内の次のワークユニットの処理を開始するか又は終了する。

#### 【0 0 8 0】

G U I 2 0 0の最後の行は、そのすべてのトリガ条件のセルが「a n y」2 3 0というキーワードに設定されている。これは、対応するトリガ条件が存在しないか又は同等にこれらのトリガ条件が常に真に評価されるということを示す。この第5の行はトリガ条件を有さず、最後に評価されるため、この第5の実行ケースの対応する出力はデフォルト出力として規定される。

#### 【0 0 8 1】

第4のカラム2 2 0は、ユーザが規則のデバッグ又は調整を容易にするために使用できる規則に基づくトランスフォームのトレーステストランから得たメタデータを表示する。一般に、少なくとも3つの異なる動作モード、すなわち、生産モード、レコードテストモード、及びファイルテストモードの規則セットに基づいて異なるトランスフォームを生成できる。生産モードトランスフォームは、規則セットの論理を実施し、それを追加のコードをほとんど用いずに（あるとしても）入力データに適用する。レコードテストモードトランスフォームをコード化することで、個々のワークユニット（例えば、アカウント、取引、航空機、車、コンピュータ、モバイルデバイス、建物などを表す入力データレコード）のトランスフォームの実行によってステップングが可能になる。規則セットの必須の論理のコード化に加えて、レコードテストモードトランスフォームは、詳細なロギングメ

10

20

30

40

50

ッセージ（例えば、各入力フィールド、出力、トリガ条件の結果（真又は偽）、ルックアップキー、ルックアップフィールドの値と、どの実行ケースが始動したか、並びに幾つかの中間パラメータを反映する）を生成するコードを含んでもよい。ファイルテストモードトランスフォームをコード化して、トランスフォームを大規模なバッチテストデータに適用し、多数のワークユニットのテスト結果を要約したログをレビューすることができる。規則セットの必須の論理のコード化に加えて、ファイルテストモードトランスフォームは、ロギングメッセージ（例えば、各実行ケースが始動した回数及び／又は各トリガ条件が評価され、成立し、及び／又は失敗した回数を反映する）を生成するコードを含んでもよい。この例では、カラム 2 2 0 は、各々の実行ケースについて、その間に数千のワークユニット（例えば、クレジットカードアカウントのレコード）が処理されたファイルテストモードトランスフォームの実行中に実行ケースが始動した回数のカウントを表示する。

10

#### 【 0 0 8 2 】

図 2 B は、スプレッドシートベースの規則の入力のための例示的な GUI 2 5 0 を示す。GUI 2 5 0 は、図 2 A の例示的な GUI 2 0 0 内に規定された規則とクレジットカードアカウントに関連付けられたその他のデータとに基づくトランスフォームを用いて決定された「アワードポイント」値に基づくクレジットカードアカウント保有者へ提示する提供製品のリストを生成する多重始動規則をユーザが規定することで構成されている。GUI 2 5 0 は、規則の 5 つの規則ケースを規定する 5 つの行を含む。ユーザは GUI 2 5 0 を用いて、規則の実行ケースのトリガ条件及び出力を規定し、ユーザがデバッグ又はトランスフォームの調整を容易にするために使用できる規則に基づいてトランスフォームのテストランから得たメタデータを表示する。例えば、GUI 2 5 0 は、図 6 のアプリケーションスペシャリスト環境 6 2 2 を介してユーザに提示できる。幾つかの実施態様では、GUI 2 5 0 を介して規定された規則セットは、図 6 の実行環境 6 0 4 のネットワークインターフェイスを介して受信できる。

20

#### 【 0 0 8 3 】

第 1 のカラム 2 5 4 は、クレジットカードアカウントの「アワードポイント」を反映する変数に適用されるトリガ条件を規定する。これらの「アワードポイント」値を、図 2 A の GUI 2 0 0 内に規定された規則に基づくトランスフォームでデータフロー内のレコードに設定し書き込むことができる。第 1 のカラムの 2 ~ 3 行目の下向きの矢印 2 5 6 は、第 2 及び第 3 の実行ケースの第 1 のトリガ条件が第 1 の実行ケースの第 1 のトリガ条件に対するトリガ条件と同じであることを示す。

30

#### 【 0 0 8 4 】

GUI 2 0 0 の第 2 及び第 3 のカラム 2 5 8 は、クレジットカードのタイプとクレジットカードアカウントの保有者の居住国の人口を反映するクレジットカードアカウントに関連付けられた入力データレコードから導出されたその他の変数に適用されるトリガ条件を規定する。また、カラム 2 5 8 は、各々が 1 対の一致するトリガ条件を示す下向きの矢印を含む。

#### 【 0 0 8 5 】

GUI 2 5 0 の第 4 のカラム 2 6 4 は、ある実行ケースのすべてのトリガ条件が評価され、満たされると判断されたときに生成される出力を規定する。ユーザは、GUI 2 5 0 を介して、これを多重始動規則 2 6 8 であると規定しているため、実行ケースはすべて評価できる。実行ケースが始動すると、その実行ケースについて規定された出力が生成され、現在のワークユニットの出力値のリストにこの出力を付け加えることができる。すべての実行ケースが評価されると、トランスフォームは、データフロー内の次のワークユニットの処理を開始するか又は終了する。

40

#### 【 0 0 8 6 】

第 5 のカラム 2 7 0 は、ユーザがトランスフォームのデバッグ又は調整を容易にするために使用できる規則に基づくトランスフォームのトレーステストランから得たメタデータを表示する。この例では、カラム 2 7 0 は、各々の実行ケースについて、その間に数千のワークユニット（例えば、クレジットカードアカウントのレコード）が処理されたファ

50

イルテストモードトランスフォームの実行中に実行ケースが始動した回数のカウントを表示する。

【 0 0 8 7 】

図 3 A は、規則セットの一意的なトリガ条件のリスト 3 0 0 の一例を示す。幾つかの実施態様では、1 つ以上のトランスフォームが規則セットに基づいて生成されると、トランスフォームがその内部で発生する各々の実行ケースのトリガ条件の複製コピーを記憶するのに使うはずだったメモリを節約するために、トランスフォーム生成プロセスの一部が、制御構造が参照できる一意的なトリガ条件のリストを生成している。幾つかの実施態様では、規則仕様に基づいてトランスフォームを生成していてもよい、リスト 3 0 0 を含む規則の一部は、規則をトランスフォームコード化で使用する技術バックエンド名に規定するアプリケーションスペシャリストによって使用されるデータ名を変換することを含んでもよい。例えば、変換動作は、可変名の固定キー又はその他のマッピングに基づいて実行できる。この例では、規則セットは、図 2 A の G U I 2 0 0 を介して規定された規則（規則 1）及び図 2 B の G U I 2 5 0 を介して規定された規則（規則 2）から構成される。

【 0 0 8 8 】

一意的なトリガ条件のリストは、規則セット内で 1 回又は複数回発生する各トリガ条件の単一のコピーを含んでもよい。この例では、各トリガ条件は、データ操作言語（DML）式としてコード化される。トリガ条件の DML コード化が図 3 A の第 1 のカラム 3 1 0 内に示されている。トリガ条件のその他のコード化フォーマットも可能である（例えば、C、C++、Java、又は Cobol コード）。

【 0 0 8 9 】

幾つかの実施態様では、一意的なトリガ条件のリストはまた、トランスフォーム内のトリガ条件の発生逆ルックアップを容易にする利用ポインタのリストを含んでもよい。例えば、図 3 A の第 2 のカラム 3 2 0 は、規則 1 及び規則 2 を含む規則セットの利用ポインタのリストを示す。各ポインタは、rule\_\_id（規則 1 又は規則 2）、row\_\_id（例えば、特定の実行ケースに対応する）、及び column\_\_id（例えば、実行ケース内のトリガ条件シーケンス位置に対応する）を識別する数字の三重項である。この例では、第 1 の 5 つの一意的なトリガ条件が規則 1 内で発生し、次の 7 つの一意的なトリガ条件が規則 2 内で発生する。幾つかの実施態様では、利用ポインタは、一意的なトリガ条件のリストに含まれない。

【 0 0 9 0 】

幾つかの実施態様では、リスト 3 0 0 は、リスト 3 0 0 内の式によって規定される複雑な演算の結果を再利用し、トランスフォーム実行中に入力データが同じであると認識される同じ入力に基づく結果の再演算を回避できるように、この複雑な演算の結果をキャッシュするデータ構造を含んでもよい。

【 0 0 9 1 】

一意的なトリガ条件のリスト 3 0 0 は、様々なフォーマット又はデータ構造（例えば、リンクされたリスト又は索引付きアレイ）内に記憶できる。この例では、一意的なトリガ条件のリスト 3 0 0 は索引付きアレイとして記憶され、トランスフォームの制御構造内の索引によるトリガ条件の参照に基づくトリガ条件のルックアップが容易になる。

【 0 0 9 2 】

図 3 B は、規則セットの一意的な出力のリストの一例を示す。幾つかの実施態様では、規則セットに基づいて 1 つ以上のトランスフォームが生成されると、トランスフォームがその内部で発生する各々の実行ケースの出力の複製コピーを記憶するのに使うはずだったメモリを節約するために、トランスフォーム生成プロセスの一部が、制御構造が参照できる一意的な出力のリストを生成している。この例では、規則セットは規則 1 及び規則 2 から構成される。

【 0 0 9 3 】

一意的な出力のリストは、規則セット内で 1 回又は複数回発生する各出力の単一のコピーを含んでもよい。この例では、各出力は、DML 式としてコード化される。出力の DM

10

20

30

40

50

Lコード化が図3Bの第1のカラム360内に示されている。出力のその他のコード化フォーマットも可能である(例えば、C、C++、Java、又はCobolコード)。

【0094】

幾つかの実施態様では、一意的な出力のリストはまた、トランスフォーム内の出力の発生逆ルックアップを容易にする利用ポインタのリストを含んでもよい。例えば、図3Bの第2のカラム370は、規則1及び規則2を含む規則セットの利用ポインタのリストを示す。この例では、第1の4つの一意的な出力が規則1内で発生し、次の5つの一意的な出力が規則2内で発生する。幾つかの実施態様では、利用ポインタは一意的な出力のリストに含まれない。

【0095】

一意的な出力のリスト350は、様々なフォーマット又はデータ構造(例えば、リンクされたリスト又は索引付きアレイ)内に記憶できる。この例では、一意的な出力のリスト350は、一意的なトリガ条件のリスト300と共に索引付けされる(例えば、ディスジョイント索引値間隔で)索引付きアレイとして記憶される。

【0096】

トランスフォーム生成プロセスの一部は、トランスフォームの実行フローを制御する制御構造の生成であり、一意的なトリガ条件のリスト及び/又は一意的な出力のリストを参照できる。本明細書では、「制御構造」という用語は、様々なコード化フォーマットを指し、二重に索引付けされた二次元アレイに限定されない。例えば、図4A及び図4Bの非周期的有向グラフ及び図5に示す二重にリンクされたリストは、トランスフォームの実行フローを制御する制御構造の例である。制御構造は、実行ケースに対応する行を有する。トランスフォームの制御構造の場合、「行」という用語は、1つ以上のトリガ条件と、行内のすべてのトリガ条件が満たされると決定されると、行の出力が実行される1つ以上の出力との論理的なグループ化を指す。ここでは、「行」という用語は、表示テーブルの水平のサブセットに限定されない。

【0097】

図4A~図4Bは、非周期的有向グラフとして表される規則セットに基づくトランスフォームの例示的な制御構造400を示す。この例では、トランスフォームがそれに基づく規則セットは、規則1及び規則2を含む。この例では、トランスフォームは、データフローグラフ100のコンポーネント150内で実施されるか、又は同等に、トランスフォームは、コンポーネント140内で実施される規則2に基づく第2のトランスフォームに連続するコンポーネント130内で実施される規則1に基づく第1のトランスフォームとして実施できる。図4A及び図4Bでは、図3Aに関連して記載したように、各ノードは、利用ポインタによってラベルを与えられる(rule\_id、row\_id、column\_id)。

【0098】

制御構造400内のノードは、トリガ条件又は出力に対応する。トリガ条件に対応するノードは、ノードから出現する2つのエッジを有する。これらの2つのエッジのうち1つは、ノードの対応するトリガ条件が入力データに適用されたときに真であると決定されるとたどられる。これらの2つのエッジのうち2番目は、ノードの対応するトリガ条件が入力データに適用されたときに偽であると決定されるとたどられる。ある出力に対応するノードは、対応する出力が生成された後で常にたどられるノードから出現する1つのエッジを有する。制御構造400内の行は、以前のトリガ条件ノードから出現する「真」のエッジによって連続的に接続される1つ以上のトリガ条件ノードのシーケンスを含んでもよい。行のシーケンスの最後のトリガ条件ノードは、行の1つ以上の出力ノードの第1のノードにその「真」のエッジによって接続できる。行内の出力ノードから出現するエッジは、行内の追加の出力ノードに接続できる。行内の最後の出力ノードから出現するエッジは、異なる実行ケース又は規則に対応する別の行内のノードに接続できるか、又は実行フローをワークユニットのトランスフォーム処理の終り448へ誘導することができる。同様に、トリガ条件ノードから出現する「偽」のエッジは、異なる実行ケース又は規則に対応す

10

20

30

40

50

る別の行内のノードに接続できるか、又は実行フローをワークユニットのトランスフォーム処理の終り448へ誘導することができる。

#### 【0099】

制御構造400は、トランスフォームの実行フローの始点402をコード化する。トランスフォームは、第1の実行ケース404の第1のトリガ条件を評価することで開始する。

#### 【0100】

幾つかのケースでは、「偽」のエッジによって、実行フローは現在の行に隣接していない行にジャンプし、そうすることで、幾つかの実行ケースの評価をスキップする。この行のスキップによって複雑さが低減し、トランスフォームによるワークユニット（例えば、アカウント、取引、航空機、車、コンピュータ、モバイルデバイス、建物などを表す入力データレコードに対応する）の処理中のトランスフォームの処理時間が低減する。

10

#### 【0101】

この例では、トリガ条件ノード420を含むノードの幾つかは、それらに接続するエッジを有さない。ノードに接続するエッジがないことは、対応するトリガ条件又は出力をトランスフォームが立脚する規則セットの論理で処理する必要がないという事実を反映している。その結果、これらの非接続ノード及びそれらに対応するトリガ条件は、制御構造400から除外することができる。不要なトリガ条件の除外を図4Bの制御構造450に示す。

#### 【0102】

20

幾つかのケースでは、制御構造内で複数のノードを組み合わせて単一ノードとして表すことができる。例えば、ノード454によって参照されるトリガ条件とノード456によって参照されるトリガ条件との論理積の評価を誘導する単一のトリガ条件ノードを作成することで、ノード454及び456を組み合わせたことができる。両方のトリガ条件が真の場合、出力ノード458が次に処理され、これらのトリガ条件のいずれかが偽の場合、ノード460のトリガ条件が次に処理されるため、このことが可能である。このようにして、規則セット内の実行ケースの制御構造内450内で2つのトリガ条件を組み合わせて単一のトリガ条件によって表すことができる。組み合わせられたノードがこれらのトリガ条件のインスタンスのみに対応する場合、トリガ条件を一意的なトリガ条件のリスト内の単一のエントリ内に組み合わせることもできる。同様に、出力ノード470及び472に対応する出力は常に一緒に生成されるため、出力ノード470及び472を組み合わせることができる。この例では、出力ノード470及び472を組み合わせることで、それらに対応する行をトランスフォームが立脚する規則セットの2つの実行ケースに対応する単一の行に組み合わせることができる。これらのノード組み合わせ技術は、生成し実行すべき制御フローのコードの量を低減することでトランスフォームのメモリ使用率と処理時間とを低減できる。

30

#### 【0103】

図5は、表示テーブルとして表される規則セットに基づくトランスフォームの例示的な制御構造500を示す。制御構造500は、例えば、二重にリンクされたリスト（例えば、各々の行が行のトリガ条件及び出力のリンクされたリストを含む行のリンクされたリスト）として記憶できる。この例では、制御構造500は、規則1及び規則2を含む規則セットに基づいてトランスフォームの実行フローを制御する。この例では、制御構造500は、一意的なトリガ条件のリスト300内のトリガ条件と、一意的な出力のリスト350内の出力とを参照する。

40

#### 【0104】

図5の第1のカラム510は、制御構造500の行の索引（例えば、実行ケース番号）を示す。この索引を用いて、制御構造内の行を参照して、不要な処理を回避するための実行フロー内のジャンプを容易にすることができる。第2のカラム520は、各行のトリガ条件のシーケンスを示す。例えば、制御構造500は、一意的なトリガ条件のリスト300内のトリガ条件を参照し、結果（例えば、成立又は失敗／真又は偽）に基づいてトリガ

50

条件が評価された後にトランスフォームの処理をさらに誘導するトリガ条件のコードの部分を含んでもよい。トリガ条件が失敗した場合、トランスフォームの処理は制御構造 5 0 0 内の行のシーケンス内の現在の行から 2 行以上離れていてもよい異なる行へ誘導される。例えば、第 3 行内のコード 5 2 2 の部分は、一意的なトリガ条件のリスト 3 0 0 内の第 3 のトリガ条件を評価するように処理を誘導し、トリガ条件が失敗した場合、コード 5 2 2 の部分は、処理を制御構造 5 0 0 の第 5 の行へ誘導し、4 番目の行をスキップする。

#### 【 0 1 0 5 】

制御構造内のある行についてリストされたトリガ条件のすべてが成立すると、処理は行の出力へ誘導される。図 5 の第 4 のカラム 5 3 0 は、一意的な出力のリスト 3 5 0 内の出力への参照を示す。最終的に、制御構造 5 0 0 内のある行の出力の実行後に、処理を制御構造 5 0 0 の別の行で継続するか又は現在のワーク項目の処理を終了するように処理を誘導することができる。図 5 の第 3 のカラム 5 4 0 は、制御構造の行の索引（例えば、実行ケース番号）によって参照される制御構造 5 0 0 内の他の行への参照を示す。

#### 【 0 1 0 6 】

図 6 は、トランスフォーム生成技術が使用できる例示的なデータ処理システム 6 0 0 を示す。システム 6 0 0 は、各々がデータを様々な記憶フォーマット（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、平文テキストファイル、又はメインフレームが使用する固有のフォーマット）のいずれかに記憶できる、記憶デバイス又はオンラインデータストリームなどの 1 つ以上のデータソースを含んでもよいデータソース 6 0 2 を含む。実行環境 6 0 4 は、トランスフォーム生成モジュール 6 0 6 及び実行モジュール 6 1 2 を含む。実行環境 6 0 4 は、UNIX オペレーティングシステムなどの好適なオペレーティングシステムの制御下で 1 つ以上の汎用コンピュータ上でホストできる。例えば、実行環境 6 0 4 は、複数の中央処理装置（CPU）、ローカル側の（例えば、SMP コンピュータなどのマルチプロセッサシステム）又はローカル側に分散された（例えば、クラスター又は MPP に結合された複数のプロセッサ）、又はリモートの、又はリモート側に分散された（例えば、ローカルエリアネットワーク（LAN）及び/又はワイドエリアネットワーク（WAN）を介して結合された複数のプロセッサ）、又はそれらの任意の組み合わせを用いたコンピュータシステムの構成を含む多重ノード並列コンピューティング環境を含んでもよい。

#### 【 0 1 0 7 】

トランスフォーム生成モジュール 6 0 6 は、その少なくとも 1 つが 1 つ以上のトリガ条件と、1 つ以上のトリガ条件がすべて満たされると生成される出力の仕様とを含む実行ケースのシーケンスを含む規則セットを受信する。規則セットはまた、1 つ以上の出力を含むがトリガ条件を欠くか、又は同等に、常に真に評価される（例えば、例示的な規則 1 のデフォルト実行ケース）トリガ条件を有する実行ケースも含むことができる。例えば、規則セットは、アプリケーションスペシャリスト環境 6 2 2 を介して実行環境 6 0 4 にアクセスするユーザ 6 2 6 によって規定され得る。幾つかの実施態様では、アプリケーションスペシャリスト環境は、ユーザ 6 2 6 に規則セットを規定する GUI を提供するリモートコンピューティングデバイス上で実行されるクライアントソフトウェアを含む。例えば、アプリケーションスペシャリスト環境 6 2 2 を介して、ユーザは、図 2 A 及び図 2 B に関連して記載したように、スプレッドシートベースの GUI で規則セットを規定できる。幾つかの実施態様（図示せず）では、開発環境 6 1 8 及びアプリケーションスペシャリスト環境 6 2 2 を組み合わせて、データフローグラフと、それらのデータフローグラフ内にインスタンス化されたトランスフォームの規則セット仕様との両方を編集する単一のユーザ又はユーザグループからアクセスすることができる。

#### 【 0 1 0 8 】

トランスフォーム生成モジュール 6 0 6 は、受信した規則セットに基づいて 1 つ以上のトランスフォームを生成する。1 つ以上のトランスフォームの実行フローを制御する制御構造を生成することができる。制御構造は、規則セット内の実行ケースに対応する行を含んでもよい。各行は、実行ケースの 1 つ以上のトリガ条件のシーケンスと、出力を規定す

10

20

30

40

50

る情報とを含んでもよい。トリガ条件の幾つかは、データを変換する処理中にトリガ条件が失敗したときに現在の行よりも2行以上下側の異なる行で継続するように処理を誘導し、したがって幾つかの実行ケースの評価をスキップしてトランスフォームに必要な処理時間を低減することができる。

#### 【0109】

制御構造は、生成されたトランスフォームをコード化する任意のその他のデータと共に記憶又は送信できる。例えば、制御構造を含む生成されたトランスフォームをデータ記憶システム616内に記憶できる。幾つかの実施態様(図示せず)では、トランスフォーム生成モジュール606は、アプリケーションスペシャリスト環境622の一部として実施でき、制御構造を含む生成されたトランスフォームをコード化するデータは、アプリケーションスペシャリスト環境622を実行しているリモートコンピューティングデバイスから実行環境604へ送信することができる。

10

#### 【0110】

実行モジュール612は、トランスフォーム生成モジュール606によって生成された1つ以上のトランスフォームを用いて、入力データレコードを処理し、送信又は記憶のための出力データレコードを生成する。グラフベースの演算内の1つ以上のトランスフォームが生成されると、実行モジュール612はトランスフォームを含む演算を入力データに適用する。実行モジュール612は、データソース602からデータを読み出し、実行環境604がアクセス可能なデータ記憶システム616内に記憶できる出力データレコード614を生成する。例えば、データ記憶システム616は、バージョン制御アプリケーションを実行しているデータベースサーバ及び/又はサーバを含んでもよい。

20

#### 【0111】

幾つかの実施態様では、実行モジュールはまた、入力データレコードの処理中に評価されるトリガ条件の実行時間及び/又は結果を記録する。例えば、トランスフォーム生成モジュール606はこれらのログを用いて、制御構造内のトリガ条件の順序を変更することでトランスフォームを更新できる。

#### 【0112】

データソース602を提供する記憶デバイスは、実行環境104のローカル側にあってもよく、例えば、実行環境604を実行してい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608)上に記憶されていてもよく、又は、実行環境604のリモート側にあってもよく、例えば、実行環境604を実行しているコンピュータと通信中のリモートシステム(例えば、メインフレーム610)上にリモート接続によってホストされていてもよい。

30

#### 【0113】

データ記憶システム616は、開発者620がトランスフォームに対応するコンポーネントを含むグラフベースの演算を作成し管理できる開発環境618にアクセスできる。これらのトランスフォームの動作は、グラフベースの演算が適用されるアプリケーションの特殊な情報を有するユーザ(例えば、ユーザ626)によって構成可能である。幾つかの実施態様では、開発環境618は、頂点の間の有向リンク(ワーク要素のフローを表す)によって接続された頂点(コンポーネント又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アプリケーションを開発するシステムである。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ている「Managing Parameters for Graph-Based Applications」と題された米国公開第2007/0011668号に詳述されている。そのようなグラフベースの演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ている米国特許第5,566,072号「EXECUTING COMPUTATIONS EXPRESSED AS GRAPHS」に記載されている。このシステムに従って作成されたデータフローグラフは、グラフコンポーネントによって表される個々のプロセスへ入出力する情報を獲得し、情報をプロセス間で移動し、プロセスの実行順序を定義する方法を提供する。このシステムは、プロセス間通信方法(例えば、グラフのリンクによる通信経路は、TCP/IP又はUNIXドメインソケットを使用できるか、又は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

40

50

る)を選択するアルゴリズムを含む。

【0114】

実行モジュール612は、様々な形式のデータベースシステムを含む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、ヌル値を含んでもよいそれぞれのフィールド(「属性」又は「カラム」とも呼ばれる)の値を有するレコードとして編成できる。データソースから最初にデータを読み出すときに、実行モジュール612は、通常、そのデータソース内のレコードに関する何らかの初期フォーマット情報から開始する。状況によっては、データソースのレコード構造は当初は不明であって、データソースの分析後に決定できることがある。レコードに関する初期情報は、ビットによって表される明確な値、レコード内のフィールドの順序、及び値のタイプ(例えば、文字列、符号付き/符号なし整数)を表すビット数を含んでもよい。

10

【0115】

図7は、例示的なトランスフォーム生成及び実行プロセス700のフローチャートを示す。例えば、プロセス700は、図6の実行環境604によって実行できる。

【0116】

プロセス700は、規則セットが702で受信されると開始する。規則セットは、実行ケースのシーケンスを含んでもよい。規則セット内の実行ケースは、1つ以上のトリガ条件と、実行ケースの1つ以上のトリガ条件がすべて満たされると生成される出力の仕様とを含んでもよい。幾つかの実施態様では、規則セットは、規則セットを受信する処理デバイスのローカル側に接続されたハードウェア(例えば、コンピュータモニタ及びキーボード及び/又はマウス)を含むユーザインターフェイス(例えば、テキストファイルエディタ、スプレッドシートベースのGUI、又はその他のタイプのGUI)を介して受信される。例えば、規則セットは、図6の実行環境604のユーザインターフェイスを介して受信できる。幾つかの実施態様では、規則セットはサーバによってリモート処理デバイスからネットワークインターフェイスを介して受信される。例えば、規則セットは、アプリケーションスペシャリスト環境622を実行しているリモート処理デバイスから実行環境604のネットワークインターフェイスを介して受信できる。

20

【0117】

制御構造を含むトランスフォームが、受信された規則セットに基づいて704で生成される。制御構造を用いてトランスフォームを入力データに適用するときにトランスフォームの実行フローを制御できる。制御構造は、生成されたトランスフォームの他の部分(例えば、一意的なトリガ条件のリスト及び/又は一意的な出力のリスト)を参照できる。制御構造は、様々なフォーマットでコード化できる。制御構造フォーマットの例は、とりわけ、実行環境内の1つ以上のプロセッサのために編集した実行可能ファイル、コンピュータ言語インタプリタ又は実行時のコンパイラによって編集可能なテキストを含むテキストファイル、解釈又は編集が可能なコードの部分を含むテキストレコードの二重索引付き(2次元)アレイ、図4A及び図4Bの非周期的有向グラフ、及び図5に示す二重にリンクされたリストである。

30

【0118】

生成された制御構造は、規則セット内の1つ以上の実行ケースに対応する行を含んでもよい。制御構造内の行は、1つ以上のトリガ条件のシーケンスと実行ケースの出力を規定する情報とを含んでもよい。制御構造の行は、規則セット内の1つ以上のトリガ条件と実行ケースの1つ以上の出力とに対応する実行制御フローコードの論理的なグループ化であってもよい。幾つかの実施態様では、制御構造の行は、トランスフォームの実行を誘導して処理デバイスにトリガ条件を検査させるか、又は出力を生成させるコードの部分のシーケンス(例えば、リンクされたリストとしてソートされた)を含む。トリガ条件のコード部分はまた、異なるトリガ条件に又は現在のトリガ条件の評価結果に基づく出力に対応するコードの別の部分へトランスフォームの実行を誘導することもできる。制御構造内のトリガ条件は、データ変換プロセス中に失敗すると、行のシーケンス内の現在の行から2行以上離れた異なる行で継続するように処理を誘導することができる。このようにして、ト

40

50



リガ条件及び／又は幾つかの行の出力の実行をスキップしてワークユニット（例えば、入力データレコード）の処理時間を低減することができる。

【 0 1 1 9 】

幾つかの実施態様では、行のトリガ条件のシーケンスを、シーケンス内のトリガ条件がデータ処理中に失敗したときに処理が誘導される異なる行の行番号に基づいてソートすることができる。例えば、行のトリガ条件のシーケンスを、失敗状況で制御構造内の大きいジャンプを引き起こすトリガ条件を行のトリガ条件のシーケンス内の前の方に置き、一方、失敗状況で制御構造内の小さいジャンプを引き起こすトリガ条件を行のトリガ条件のシーケンス内の後の方に置くことで、トランスフォーム生成プロセス中にソートすることができる。

10

【 0 1 2 0 】

幾つかの実施態様では、実行ケースのシーケンス内の対応する実行ケースの直前に実行ケース内で発生する対応する実行ケースのトリガ条件を除外する形で制御構造の行が 7 0 4 で生成される。除外によって、トランスフォームの編集及び／又は実行に必要なメモリ要件を低減できる。

【 0 1 2 1 】

幾つかの実施態様では、規則セットの一意的なトリガ条件のリストもトランスフォームの一部として 7 0 4 で生成される。規則 1 及び規則 2 を含む規則セットの一意的なトリガ条件のリスト 3 0 0 は、生成可能な一意的なトリガ条件のリストの一例である。例えば、行内のトリガ条件のシーケンスがコードの部分のシーケンスとして記憶されている場合、コードの部分の 1 つは規則セットの一意的なトリガ条件のリスト内にコード化されたトリガ条件へ処理を誘導することができる。

20

【 0 1 2 2 】

幾つかの実施態様では、規則セットの一意的な出力のリストもトランスフォームの一部として 7 0 4 で生成される。規則 1 及び規則 2 を含む規則セットの一意的な出力のリスト 3 5 0 は、生成可能な一意的な出力のリストの一例である。例えば、行内の出力がコードの部分として記憶されている場合、コードの部分は規則セットの一意的な出力のリスト内の出力へ処理を誘導することができる。

【 0 1 2 3 】

幾つかの実施態様では、制御構造の行は、行のすべてのトリガ条件が満たされ、行の出力が生成されたときに次に処理する制御構造の異なる行へ処理を誘導するコードの部分を含み、さらに含む。

30

【 0 1 2 4 】

例えば、図 6 の実行環境 6 0 4 内で実行されているトランスフォーム生成モジュール 6 0 6 によって、制御構造を含むトランスフォームを 7 0 4 で生成できる。

【 0 1 2 5 】

制御構造を含む生成されたトランスフォームは、7 0 6 で記憶及び／又は送信できる。幾つかの実施態様では、トランスフォームはメモリデバイス（例えば、ランダムアクセスメモリ）内に記憶され、トランスフォームを入力データに適用する実行モジュールへ渡される。例えば、図 6 の実行モジュール 6 1 2 によってアクセスできる実行環境 6 0 4 の一部である揮発性メモリデバイス内に、トランスフォーム生成モジュール 6 0 6 によって、トランスフォームを 7 0 6 で記憶できる。幾つかの実施態様では、不揮発性メモリを含むデータ記憶デバイス（例えば、バージョン制御アプリケーションを実行しているデータベースサーバ又はサーバ）内にトランスフォームを記憶することができる。例えば、データ記憶システム 6 1 6 内にトランスフォームを 7 0 6 で記憶できる。幾つかの実施態様では、トランスフォームをリモートデバイスへ送信できる（例えば、電子通信ネットワークを介して）。例えば、アプリケーションスペシャリスト環境内で実行されているトランスフォーム生成モジュールから入力データへ適用するためのリモート実行環境へトランスフォームを 7 0 6 で送信できる。

40

【 0 1 2 6 】

50

トランスフォームが生成され、トランスフォームを実行する処理システムが利用可能になると、トランスフォームを入力データに適用することができる。例えば、トランスフォームを実施する1つ以上のコンポーネント（例えば、データフローグラフ100のコンポーネント130及び140）を含むデータフローグラフを実行している処理システムから、トランスフォームにアクセスすることができる。1つ以上のデータソース（例えば、データソース602）から入力データを708で受信できる。幾つかの実施態様では、入力データを事前処理して（例えば、ジョインプロセスを実施するコンポーネント120によって）、トランスフォームを実施する1つ以上のコンポーネントへ渡すデータフローのワークユニットを作成することができる。各々のワークユニットは受信した入力データに基づいて準備されるため、トランスフォームへ渡すことができる。幾つかの実施態様では、ワークユニットのグループがバッチ単位でトランスフォームへ渡される。例えば、図6の実行環境604のネットワークインターフェイスを介して入力データを708で受信することができる。

10

#### 【0127】

トランスフォームが710で実行されて、受信した入力データが処理される。幾つかの実施態様では、新しい入力データを処理するときにトランスフォームが実行時にさらに解釈及び/又は編集される。例えば、図8に関連して記載するプロセス800を用いて、トランスフォームを710で実行し、入力データに適用できる。トランスフォームを入力データに適用することは、制御構造を用いて決定されたシーケンス内の入力データに照らしてトリガ条件を検査することを含んでもよい。例えば、図6の実行環境604内で実行されている実行モジュール606によって、トランスフォームを710で実行できる。

20

#### 【0128】

トランスフォームの実行は、712で利用可能な入力データがなくなるまで継続できる。入力データへのトランスフォームの適用の結果を反映するデータを714で記憶できる（例えば、データシンクに書き込まれた出力データレコードとして）。記憶された結果データは、制御構造によって規定された出力に基づいて生成されたものであってもよい。例えば、図6のデータ記憶システム614内に、実行モジュール612によって結果を記憶することができる。幾つかの実施態様（図示せず）では、制御構造によって規定された出力に基づく結果データが（例えば、アプリケーションスペシャリスト環境622へ）、電子通信ネットワーク上で（例えば、実行環境604のネットワークインターフェイスを介して）送信される。

30

#### 【0129】

図8は、規則セットに基づくトランスフォームを実行する例示的なプロセス800のフローチャートである。例えば、プロセス800は、図6の実行環境604上で実行されている実行モジュール612によって実行できる。

#### 【0130】

プロセス800は、入力データに適用されるトランスフォームの制御構造を検索すること802を含む。幾つかの実施態様では、制御構造は、トランスフォームを実施するデータフローグラフ内のコンポーネントがデータフロー内のワークユニットへ渡されるときに制御構造が802で検索される。幾つかの実施態様では、メモリデバイス（例えば、ランダムアクセスメモリ）から制御構造が802で検索される。幾つかの実施態様では、不揮発性メモリを含むデータ記憶デバイス（例えば、バージョン制御アプリケーションを実行しているデータベースサーバ又はサーバ）から制御構造が802で検索される。幾つかの実施態様では、制御構造は、実行時にインタプリタ及び/又はコンパイラへ渡されて、実行のための制御構造が準備される。

40

#### 【0131】

ワークユニットの入力データに照らして、第1のトリガ条件が810で検査される。例えば、トリガ条件をコード化するDML式を解釈し実行して、ワークユニットに関連付けられたレコード内の任意の基準入力データフィールドにアクセスし、トリガ条件の論理を適用することでアクセスしたデータを試験することができる。この評価の結果は、成立又

50

は失敗（真又は偽）であってもよい。

【0132】

トリガ条件の評価の結果は記録できる（例えば、試験、デバッグ、又は最適化のために）。幾つかの実施態様では、トリガ条件の実行時間（例えば、マイクロ秒又はプロセッサ周期単位で測定した）を記録できる。入力データに適用されるときトリガ条件の失敗率又は実行時間に関するデータを用いて、将来のレコードのために平均処理時間を低減する努力の中で制御構造を動的に更新することができる。

【0133】

ワークユニットの入力データによってトリガ条件が815で満たされない（例えば、結果が失敗か又は偽である）場合、トランスフォームの実行を、トリガ条件に関連付けられた制御フローコード（例えば、トリガ条件を参照するか又は含むコードの部分）に部分的に基づいて、制御構造の異なる行へ誘導することができる。制御構造に従って、実行は、制御構造の異なる行へ820でジャンプすることができる。例えば、幾つかの失敗条件によって、実行は、制御構造の行のシーケンス内の現在の行から2行以上離れた異なる行へ820でジャンプすることができる。このようにして、幾つかの行のトリガ条件及び/又は出力の実行をスキップしてワークユニットの処理時間を低減できる。次いで、新しい行の次のトリガ条件を810で検査できる。幾つかの実施態様（図示せず）では、トランスフォームの実行は、トリガ条件（例えば、デフォルトの実行ケースに対応する）がない行へ、又は制御構造の終了へ直接ジャンプすることができる。

【0134】

ワークユニットの入力データによってトリガ条件が815で満たされる（例えば、結果が成立か又は真である）場合、トランスフォームの実行を、制御構造の現在の行内の次の要素へ誘導することができる。行825のトリガ条件のシーケンス内により多くのトリガ条件がある場合、行内の次のトリガ条件が810で検査される。そうでない場合、行の1つ以上の出力を830で生成できる。

【0135】

例えば、出力をコード化するDML式を解釈し実行して、ワークユニットに関連付けられたレコード内の任意の基準入力データフィールドにアクセスし、及び/又は出力の論理を適用して、810で1つ以上の出力レコードを生成できる。結果として得られる出力レコードは、完全に新しくてもよく、又は既存のレコードを更新又は拡張して追加のフィールド又はその他のデータを含むことができる。

【0136】

現在の行の出力が830で生成された後で、トランスフォームの実行を追加の実行ケースに対応する異なる行へ誘導できる。幾つかの実施態様では、行は、トランスフォームの実行を制御構造内の異なる行へ誘導するポイントを含む。835で処理する実行ケースがさらにある場合、制御構造によって、トランスフォームの実行は制御構造の異なる行へ840でジャンプする。例えば、多重始動規則の場合、追加の実行ケースに対応する追加の行を処理する必要がある。トランスフォームが多重規則を有する規則セットに対応する場合、制御構造によって、トランスフォームの実行は異なる規則に対応する制御構造の異なる行へ840でジャンプする。次いで、新しい行の次のトリガ条件が810で検査される。

【0137】

835で、これ以上実行ケースを、すなわち、行を処理する必要がないときには、処理された入力データのログ情報に基づいて、制御構造を動的に更新することができる。例えば、行のトリガ条件のシーケンス内のトリガ条件を、部分的にトリガ条件の平均失敗率又は実行時間に関する新しいログ情報に基づいて850でソートできる。

【0138】

幾つかの実施態様では、入力データでトリガ条件を実行するのにかかる時間に基づいて一意的なトリガ条件のリスト内のトリガ条件の実行時間の測定値を更新できる。行のトリガ条件のシーケンス内のトリガ条件を、部分的にトリガ条件の実行時間の更新された測定

10

20

30

40

50

値に基づいて850でソートできる。幾つかの実施態様では、一意的なトリガ条件のリスト内のトリガ条件の失敗率の測定値を、トリガ条件が入力データ内の1つ以上のレコードによって満たされるか否かに基づいて更新できる。行のトリガ条件のシーケンス内のトリガ条件を、部分的にトリガ条件の失敗率の更新された測定値に基づいて850でソートできる。

#### 【0139】

将来の入力データに適用するために、制御構造の更新されたバージョンを852で記憶できる。幾つかの実施態様では、メモリデバイス(例えば、ランダムアクセスメモリ)内に更新された制御構造を852で記憶できる。幾つかの実施態様では、不揮発性メモリを含むデータ記憶デバイス(例えば、バージョン制御アプリケーションを実行しているデータベースサーバ又はサーバ)内に更新された制御構造を852で記憶できる。

10

#### 【0140】

上記のトランスフォーム生成方法は、コンピュータ上で実行されるソフトウェアを用いて実施できる。例えば、各々が少なくとも1つのプロセッサと、少なくとも1つのデータ記憶システム(揮発性及び不揮発性メモリ及び/又は記憶要素を含む)と、少なくとも1つの入力デバイス又はポートと、少なくとも1つの出力デバイス又はポートとを含む1つ以上のプログラムド又はプログラマブルコンピュータシステム(分散、クライアント/サーバ、又はグリッドなどの様々なアーキテクチャの)上で実行される1つ以上のコンピュータプログラム内にソフトウェアが手順を形成する。ソフトウェアは、例えば、データフローグラフの設計及び構成に関連するその他のサービスを提供するより大きいプログラムの1つ以上のモジュールを形成できる。グラフのノード及び要素は、コンピュータ可読媒体内に記憶されたデータ構造又はデータリポジトリ内に記憶されたデータモデルに準拠するその他の編成データとして実施できる。

20

#### 【0141】

ソフトウェアは、汎用の又は特定目的のプログラマブルコンピュータによって読み出し可能なCD-ROMなどの記憶媒体上に提供できるか、又は実行中にネットワークの通信媒体上でコンピュータの記憶媒体へ送達する(伝搬された信号内にコード化すること)ことができる。すべての機能は、特定目的のコンピュータ上で、又はコプロセッサなどの特定目的のハードウェアを用いて実行可能である。ソフトウェアは、ソフトウェアによって規定された演算の異なる部分が異なるコンピュータによって実行される分散した形で実施できる。各々のそのようなコンピュータプログラムは、好ましくは、記憶媒体又はデバイスがコンピュータシステムによって読み出されて本明細書に記載する手順を実行するときにコンピュータを構成し運用するために、汎用の又は特定目的のプログラマブルコンピュータによって読み出し可能な有形の、非一過性の記憶媒体又はデバイス(例えば、ソリッドステートメモリ又は媒体、あるいは磁気又は光媒体)上に記憶され、又はダウンロードされる。本発明のシステムは、コンピュータプログラムで構成されたコンピュータ可読記憶媒体として実施されると考えられ、そのように構成された記憶媒体はコンピュータシステムを特定の所定の形で動作して本明細書に記載する機能を実行させる。

30

#### 【0142】

以上、本発明の幾つかの実施形態について説明してきた。しかしながら、本発明の精神及び範囲を逸脱することなく、様々な変更を加えることが可能であることが理解されよう。例えば、上記のステップの幾つかは順序に依存せず、記載された順序とは異なる順序で実行することができる。

40

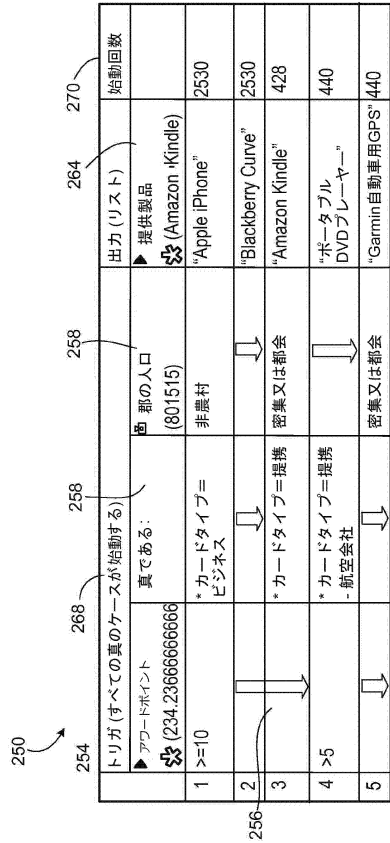
#### 【0143】

上記の説明は、例示的であって、添付の請求の範囲によって定義される本発明の範囲を限定するものではないことを理解されたい。例えば、上記の幾つかの機能ステップは、全体の処理に実質的に影響せずに別の順序で実行できる。なお、クレジットアカウントに関する特定のビジネス規則の詳細は、GUI200及びGUI250及びそれらがユーザインターフェイスを提供するトランスフォーム生成システムの機能を例示するためだけに、図2A及び図2Bの例で説明し、本明細書を通して参照している。提示された特定のビジ

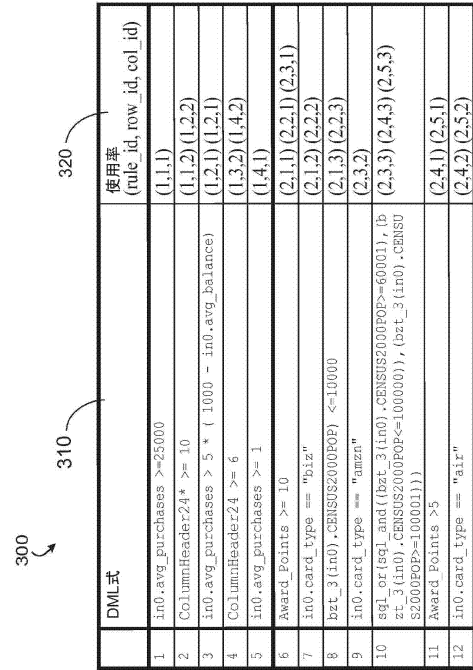
50



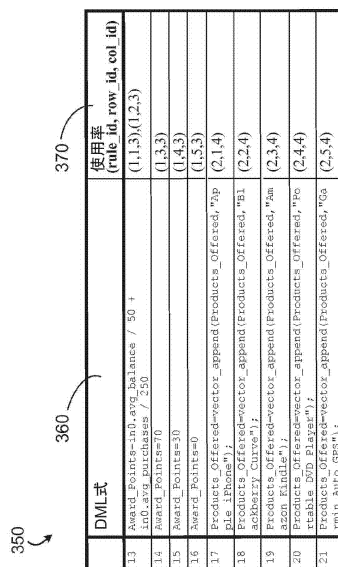
【 図 2 B 】



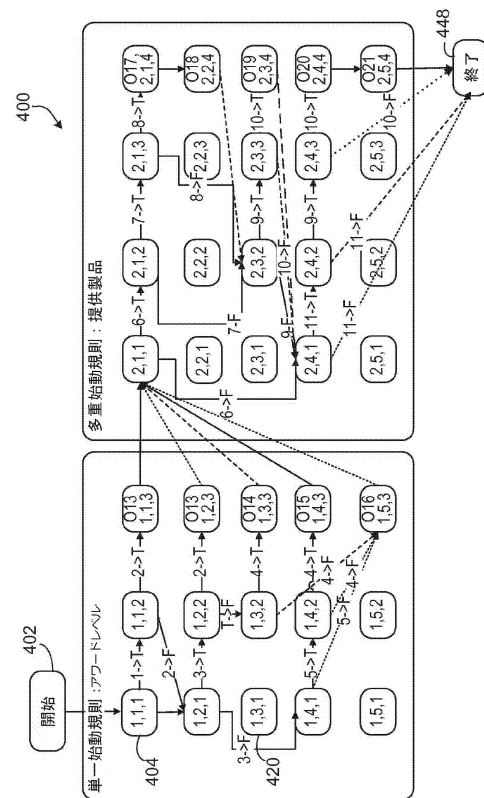
【 図 3 A 】



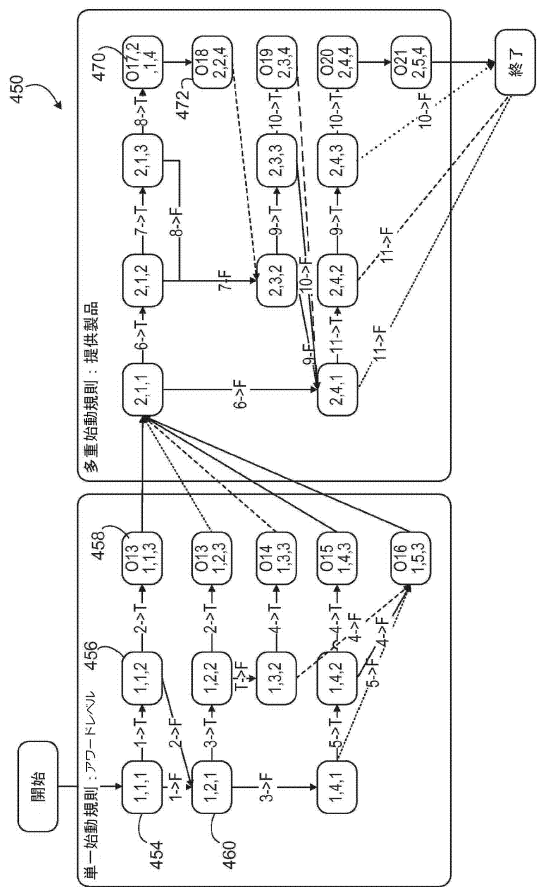
【 ㄨ 3 B 】



【 図 4 A 】



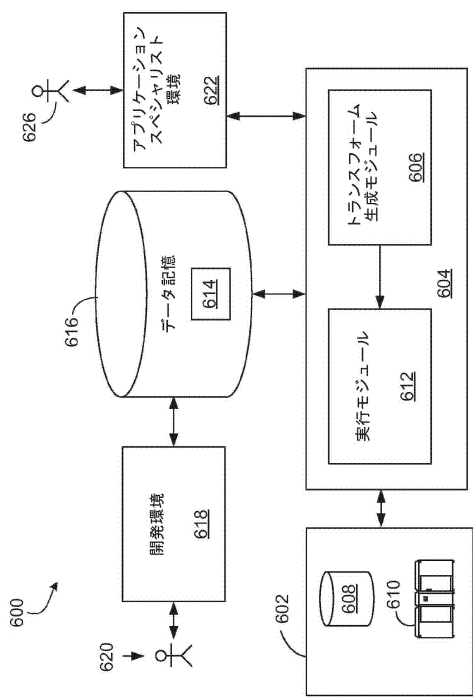
【図4B】



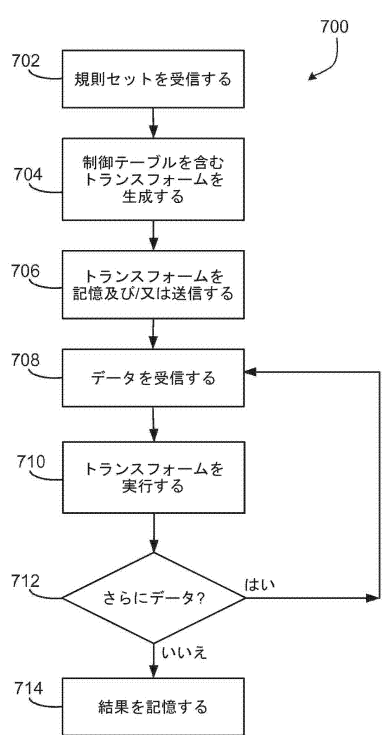
【図5】

実行ケース	トリガ	成功時の次の実行ケース	成功時の実行対象出力
ec1	Check 1 on fail goto ec2 Check 2 on fail goto ec2	ec6	13
ec2	Check 3 on fail goto ec4 Check 2 on fail goto ec3	ec6	13
ec3	Check 4 on fail goto ec5	ec6	14
ec4	Check 5 on fail goto ec5 Check 4 on fail goto ec5	ec6	15
ec5		ec6	16
ec6	Check 6 on fail goto ec9 Check 7 on fail goto ec8 Check 8 on fail goto ec8	ec7	17
ec7		ec8	18
ec8	Check 9 on fail goto ec9 Check 10 on fail goto ec9	ec9	19
ec9	Check 11 on fail goto END Check 9 on fail goto END Check 10 on fail goto END	ec10	20
ec10		END	21

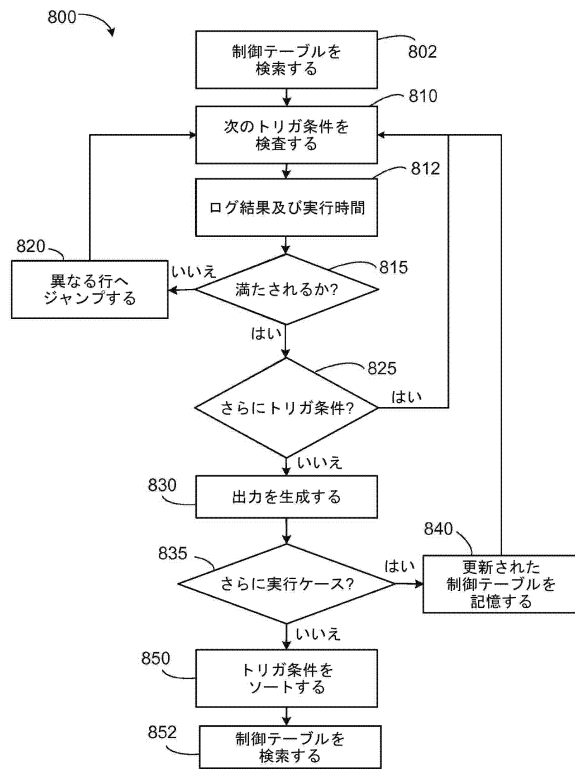
【図6】



【図7】



【図 8】





---

フロントページの続き

(31)優先権主張番号 13/958,037

(32)優先日 平成25年8月2日(2013.8.2)

(33)優先権主張国 米国(US)

(72)発明者 ストウダー, スコット

アメリカ合衆国, マサチューセッツ州 01833, ジョージタウン, ピルスベリー レーン 27

(72)発明者 ホーリー, ジョセフ スケフィントン ザ サード

アメリカ合衆国, マサチューセッツ州 02478, ベルモント, ヒルクレスト ロード 11

(72)発明者 ワイスマン, アミット

アメリカ合衆国, マサチューセッツ州 01730, ベッドフォード, レッドコート ロード 15

審査官 多賀 実

(56)参考文献 特開2008-059135(JP, A)

特開平07-036706(JP, A)

特表2010-524134(JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F8/00-8/77

G06F9/44-9/451