

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5238817号

(P5238817)

(45) 発行日 平成25年7月17日(2013.7.17)

(24) 登録日 平成25年4月5日(2013.4.5)

(51) Int.Cl.		F I			
G06F 11/30	(2006.01)		G06F 11/30	305D	
G06F 15/173	(2006.01)		G06F 15/173	640L	
G06F 9/52	(2006.01)		G06F 9/46	475Z	

請求項の数 23 (全 20 頁)

(21) 出願番号	特願2010-529340 (P2010-529340)	(73) 特許権者	508364093
(86) (22) 出願日	平成20年10月8日 (2008.10.8)		オーエムエックス テクノロジー エービー
(65) 公表番号	特表2011-501283 (P2011-501283A)		ー
(43) 公表日	平成23年1月6日 (2011.1.6)		OMX TECHNOLOGY AB
(86) 国際出願番号	PCT/EP2008/063454		スウェーデン国 ストックホルム エスー
(87) 国際公開番号	W02009/050084		105 78, トゥルヴァクトスヴェー
(87) 国際公開日	平成21年4月23日 (2009.4.23)		ゲン 15
審査請求日	平成23年10月4日 (2011.10.4)		Tullvaktsvagen 15,
(31) 優先権主張番号	11/907, 963		S-10578 Stockholm S
(32) 優先日	平成19年10月18日 (2007.10.18)		WEDEN
(33) 優先権主張国	米国 (US)	(74) 代理人	100076428
			弁理士 大塚 康德
		(74) 代理人	100112508
			弁理士 高柳 司郎

最終頁に続く

(54) 【発明の名称】 データ処理システムにおけるループチェック機構

(57) 【特許請求の範囲】

【請求項 1】

いくつかの入力チャンネル(26, 28, 30)及び出力チャンネル(32, 34)を用いて、データ処理システム(10)内のいくつかの他のデータ処理モジュール(14, 16; 70, 72, 74; 68, 72, 76)と通信することが可能な第1のデータ処理モジュール(12; 68; 74)における、様々なシステム状態(S0, S1)に従う正しいデータ処理を保証する方法であって、

少なくとも1つの入力チャンネル(26, 28)上で、古い状態(S0)から新しい状態(S1)への変更を示す状態メッセージ(S)を受信する工程(36)と、

前記新しい状態を有する全ての入力チャンネル(26, 28)上でのデータの受信をブロックする工程(38)と、

データ処理ループ(L)のセット中のそれぞれのループに少なくとも1つのループチェックメッセージ(C)を送信する工程であって、前記セットは、結びつけられた入力チャンネル(30)及び出力チャンネル(32)を介して前記第1のデータ処理モジュール(12; 68; 74)が接続されている全てのループを含み、当該入力チャンネル(30)には前記古い状態(S0)が適用されている工程と、

前記セット中のそれぞれのループについて、前記第1のデータ処理モジュール(12)において前記ループチェックメッセージ(C)が受信されるか又は前記結びつけられた入力チャンネルが前記新しい状態を示す状態メッセージを受信しており、かつ前記第1のデータ処理モジュール(12; 68; 74)の他の全ての入力チャンネル(26, 28)が前記

10

20

新しい状態を示す状態メッセージ (S) を受信している場合に、前記ブロックを除去する工程 (4 4) と、

を含むことを特徴とする方法。

【請求項 2】

前記第 1 のデータ処理モジュール (1 2 ; 6 8 ; 7 4) の前記入力チャネルと前記出力チャネルとの少なくとも一方 (2 6 , 2 8 , 3 0 , 3 2 , 3 4) が接続されているエンティティを示す、前記第 1 のモジュールに関連するシステム設定データを読み取る工程 (4 8) と、

前記第 1 のデータ処理モジュールから発するデータ処理経路を判定するために、他のデータ処理モジュール (1 4 ; 7 2 ; 6 8) につながるいくつかの出力チャネルを判定する工程 (5 0) と、

データ処理経路において後続するデータ処理モジュール (1 2) につながる出力チャネル (3 2) について、

前記後続するデータ処理モジュール (1 4) に関連するシステム設定データを読み取るさらなる工程 (5 4) と、

前記経路において前記後続するデータ処理モジュール (1 4) の次のデータ処理モジュール (1 6) が存在するか否かを判定するさらなる工程 (5 6) と、

前記次のデータ処理モジュールを前記後続するデータ処理モジュールとするさらなる工程 (6 0) と、

前記次のデータ処理モジュールが前記第 1 のデータ処理モジュールとなる (6 0) か、又は次のデータ処理モジュールが存在しなくなる (5 6) まで、前記経路に次のデータ処理モジュールが存在するか否かを判定する工程 (5 6) を繰り返すさらなる工程と、

前記経路において遭遇した前記モジュールの 1 つが前記第 1 のデータ処理モジュール (6 0) である場合に、前記第 1 の処理モジュール (1 2) と関連するループが存在すると判定するさらなる工程 (6 4) と、

前記第 1 のデータ処理モジュール (1 2) と関連する前記ループに関するデータを格納するさらなる工程 (6 6) と、

を実行する工程 (5 2) と、

によって、前記第 1 のデータ処理モジュール (1 2 ; 6 8 ; 7 4) が接続されているそれぞれのデータ処理ループ (L) を判定する工程をさらに含むことを特徴とする、請求項 1 に記載の方法。

【請求項 3】

データ処理経路は分岐を含んでもよく、

前記方法は、分岐のデータ処理モジュール (7 0 , 6 8) を含むデータ処理モジュールの順序 (6 8 , 7 2 , 7 0 , 6 8) が、前記経路において同じ順序で以前に通過されている場合に、分岐点 (7 2) より後の前記分岐を除去する工程をさらに含むことを特徴とする、請求項 2 に記載の方法。

【請求項 4】

前記ループチェックメッセージが、前記ループにおいて通過するそれぞれのデータ処理モジュールを示すことを特徴とする、請求項 1 乃至 3 の何れか 1 項に記載の方法。

【請求項 5】

前記第 1 のデータ処理モジュール (1 2) と関連する前記ループに備えられた他のデータ処理モジュール (1 4 , 1 6) に関するループデータを収集及び格納する工程と、

それぞれのデータ処理モジュールが、前記格納されたループデータに基づいて、全てのループ内の次のデータ処理モジュールに対して前記ループチェックメッセージを与える又は転送する工程と、をさらに含み、

前記ループデータは、前記他のデータ処理モジュールが備えられているループに関するデータを含み、

ループ内のそれぞれのデータ処理モジュールは、他のデータ処理モジュールが備えられている前記ループに関する前記ループデータを含む

10

20

30

40

50

ことを特徴とする、請求項 1 乃至 3 の何れか 1 項に記載の方法。

【請求項 6】

それぞれのループに対して 1 つの前記ループチェックメッセージ (C) が送られることを特徴とする、請求項 1 乃至 5 の何れか 1 項に記載の方法。

【請求項 7】

それぞれのループに対して N 個の前記ループチェックメッセージが送られ、当該 N はループに接続された前記データ処理モジュールへの入力の数、又は前記ループに接続された前記データ処理モジュールの出力の数、に依存することを特徴とする、請求項 1 乃至 5 の何れか 1 項に記載の方法。

【請求項 8】

ループに接続された前記データ処理モジュールへの入力の数、又は前記ループに接続された前記データ処理モジュールの出力の数 + 1、のうち最小のほうに前記 N が等しいことを特徴とする、請求項 7 に記載の方法。

【請求項 9】

K + 1 回のループチェックが行われ、当該 K は必須のセットにおけるリンクの数であり、当該 K はループに接続された前記データ処理モジュールへの入力チャンネルの数よりも小さく、当該 K は当該ループに接続された前記データ処理モジュールの出力チャンネルの数よりも小さいことを特徴とする、請求項 1 乃至 5 の何れか 1 項に記載の方法。

【請求項 10】

全ての出力チャンネル上で新しい状態を示す状態メッセージを送信する工程をさらに含むことを特徴とする、請求項 1 乃至 9 の何れか 1 項に記載の方法。

【請求項 11】

様々なシステム状態 (S0, S1) に従う正しいデータ処理を保証するための、データ処理システム (10) に備えられる第 1 のデータ処理モジュール (12) を含むコンピューティングデバイスであって、

前記第 1 のデータ処理モジュール (12) は、

いくつかの入力チャンネル (26, 28, 30) へのアクセスと、

いくつかの出力チャンネル (32, 34) へのアクセスと、

状態制御ユニット (22) とを備え、

前記状態制御ユニット (22) は、

古い状態 (S0) から新しい状態 (S1) への変更を示す状態メッセージ (S) が受信されている全ての入力チャンネル (26, 28) 上でのデータの受信をブロックし、

データ処理ループ (L) のセット中のそれぞれのループに少なくとも 1 つのループチェックメッセージ (C) を送信し、ここで前記セットは、結びつけられた入力チャンネル (30) 及び出力チャンネル (32) を介して前記第 1 のデータ処理モジュール (12) が接続されている全てのループを含み、当該入力チャンネル (30) には前記古い状態 (S0) が適用されており、

前記セット中のそれぞれのループについて、前記第 1 のデータ処理モジュール (12) において前記ループチェックメッセージ (C) が受信されるか又は前記結びつけられた入力チャンネルが前記新しい状態を示す状態メッセージを受信しており、かつ前記第 1 のデータ処理モジュール (12) の他の全ての入力チャンネル (26, 28) が前記新しい状態を示す状態メッセージ (S) を受信している場合に、前記ブロックを除去する

ように構成されていることを特徴とする、コンピューティングデバイス。

【請求項 12】

ループ判定ユニット (23) をさらに備え、

前記ループ判定ユニット (23) は、

前記第 1 のデータ処理モジュール (12; 68; 74) の前記入力チャンネルと前記出力チャンネルとの少なくとも一方 (26, 28, 30, 32, 34) が接続されているエンティティを示す、前記第 1 のモジュールに関連するシステム設定データを読み取る工程と、

前記第 1 のデータ処理モジュールから発するデータ処理経路を判定するために、他のデ

10

20

30

40

50

ータ処理モジュール(14)につながるいくつかの出力チャネルを判定する工程と、
データ処理経路において後続するデータ処理モジュール(12)につながる出力チャネル(32)について、

前記後続するデータ処理モジュール(14)に関連するシステム設定データを読み取るさらなる工程と、

前記経路において前記後続するデータ処理モジュール(14)の次のデータ処理モジュール(16)が存在するか否かを判定するさらなる工程と、

前記次のデータ処理モジュールを前記後続するデータ処理モジュールとするさらなる工程と、

前記次のデータ処理モジュールが前記第1のデータ処理モジュールとなる(60)か、又は次のデータ処理モジュールが存在しなくなるまで、前記経路に次のデータ処理モジュールが存在するか否かを判定する工程を繰り返すさらなる工程と、

前記経路において遭遇した前記モジュールの1つが前記第1のデータ処理モジュールである場合に、前記第1の処理モジュール(12)と関連するループが存在すると判定するさらなる工程と、

前記第1のデータ処理モジュール(12)と関連する前記ループに関するデータを格納するさらなる工程と、

を実行する工程と、

によって、前記第1のモジュールが接続されているそれぞれのデータ処理ループ(L)を判定するように構成されていることを特徴とする、請求項11に記載のコンピューティングデバイス。

【請求項13】

データ処理経路は分岐を含んでもよく、

前記ループ判定ユニット(23)は、分岐のデータ処理モジュール(70, 68)を含むデータ処理モジュールの順序(68, 72, 70, 68)が、前記経路において同じ順序で以前に通過されている場合に、分岐点(72)より後の前記分岐を除去するようにさらに構成されていることを特徴とする、請求項12に記載のコンピューティングデバイス。

【請求項14】

前記ループチェックメッセージが、前記ループにおいて通過するそれぞれのデータ処理モジュールを示すことを特徴とする、請求項11乃至13の何れか1項に記載のコンピューティングデバイス。

【請求項15】

前記第1のデータ処理モジュール(12)は、

前記第1のデータ処理モジュール(12)と関連する前記ループに備えられた他のデータ処理モジュール(14, 16)に関するループデータを収集及び格納し、

前記格納されたループデータに基づいて、全てのループ内の次のデータ処理モジュールに対して前記ループチェックメッセージを与える

ようにさらに構成されており、

前記ループデータは、前記他のデータ処理モジュールが備えられているループに関するデータを含み、

ループ内のそれぞれのデータ処理モジュールは、前記ループチェックメッセージの送信を行うために、他のデータ処理モジュールが備えられている前記ループに関する前記ループデータを含む

ことを特徴とする、請求項11乃至13の何れか1項に記載のコンピューティングデバイス。

【請求項16】

前記状態制御ユニット(22)は、それぞれのループに対して1つの前記ループチェックメッセージ(C)を送るよう構成されていることを特徴とする、請求項11乃至15の何れか1項に記載のコンピューティングデバイス。

10

20

30

40

50

【請求項 17】

前記状態制御ユニット(22)は、それぞれのループに対してN個の前記ループチェックメッセージを送るように構成され、当該Nはループに接続された前記データ処理モジュールへの入力の数、又は前記ループに接続された前記データ処理モジュールの出力の数、に依存することを特徴とする、請求項11乃至15の何れか1項に記載のコンピューティングデバイス。

【請求項 18】

ループに接続された前記データ処理モジュールへの入力の数、又は前記ループに接続された前記データ処理モジュールの出力の数+1、のうち最小のほうに前記Nが等しいことを特徴とする、請求項17に記載のコンピューティングデバイス。

10

【請求項 19】

K+1回のループチェックが行われ、当該Kは必須のセットにおけるリンクの数であり、当該Kはループに接続された前記データ処理モジュールへの入力チャネルの数よりも小さく、当該Kは当該ループに接続された前記データ処理モジュールの出力チャネルの数よりも小さいことを特徴とする、請求項11乃至15の何れか1項に記載のコンピューティングデバイス。

【請求項 20】

前記状態制御ユニット(22)は、全ての出力チャネル上で新しい状態を示す状態メッセージを送信するようにさらに構成されていることを特徴とする、請求項11乃至19の何れか1項に記載のコンピューティングデバイス。

20

【請求項 21】

様々なシステム状態(S0, S1)に従う正しいデータ処理を保証するためのデータ処理システム(10)であって、

いくつかのデータ処理モジュール(12, 14, 16)を備え、

少なくとも1つの前記データ処理モジュール(12)は、請求項11乃至20の何れか1項に従うコンピューティングデバイスとして備えられていることを特徴とする、データ処理システム。

【請求項 22】

取引システムであることを特徴とする、請求項21に記載のデータ処理システム(10)。

30

【請求項 23】

データ処理システム(10)内の第1のデータ処理モジュール(12)の状態制御ユニット(22)を提供することによって、様々なシステム状態(S0, S1)に従う正しいデータ処理を保証するためのコンピュータプログラムであって、

前記コンピュータプログラムは、データ媒体上のプログラムコードを含み、

前記プログラムコードは、前記データ媒体がコンピュータへと入れられたときに、

古い状態(S0)から新しい状態(S1)への変更を示す状態メッセージ(S)が受信されている全ての入力チャネル(26, 28)上でのデータの受信をブロックし、

データ処理ループ(L)のセット中のそれぞれのループに少なくとも1つのループチェックメッセージ(C)を送信し、ここで前記セットは、結びつけられた入力チャネル(30)及び出力チャネル(32)を介して前記第1のデータ処理モジュール(12)が接続されている全てのループを含み、当該入力チャネル(30)には前記古い状態(S0)が適用されており、

40

前記セット中のそれぞれのループについて、前記第1のデータ処理モジュール(12)において前記ループチェックメッセージ(C)が受信されるか又は前記結びつけられた入力チャネルが前記新しい状態を示す状態メッセージを受信しており、かつ前記第1のデータ処理モジュール(12)の他の全ての入力チャネル(26, 28)が前記新しい状態を示す状態メッセージ(S)を受信している場合に、前記ブロックを除去する

ように構成されていることを特徴とする、コンピュータプログラム。

【発明の詳細な説明】

50

【技術分野】

【0001】

本発明はコンピュータシステムに関する。本発明は、特に、種々のシステム状態に従ってデータを適切に処理することを保証する方法、演算装置、データ処理システム及びコンピュータプログラムコードに関する。

【背景技術】

【0002】

取引システム等の複雑なコンピュータシステムにおいて、種々の機能性を提供する種々のデータ処理モジュールがデータを処理し且つそのような処理済みデータを互いに配信することは周知である。その後、そのようなモジュールは、データを処理して別のモジュールに配信し、且つ他のモジュールが処理した別のデータを受信してもよい。

10

【0003】

そのようなシステムにおいて、システムが動作される方法の変化又は1つ以上のモジュールの構成変化のいずれかのために、種々の状態を提供することは更に周知である。取引システムにおいて、ある状態は取引の終了を知らせ、別の状態は構成変化を知らせる状態であってもよい。新しい状態が導入される際、新しい状態は、新しい状態に先行する状態で発生した旧データ上ではなく、新データに適用されることが重要である。これが考慮されない場合、システムの操作性は妨害される可能性がある。状態の更新は、高速であると同時にデータ損失があってはならない。

【0004】

20

特に、データ処理システムが複雑であり且つ処理済みデータを互いに提供する複数の相互接続モジュールを有する場合、これはそれほど容易に実現できない。

【発明の概要】

【0005】

本発明の第1の態様は、いくつかの入力チャンネル及び出力チャンネルを用いて、データ処理システム内のいくつかの他のデータ処理モジュールと通信することが可能な第1のデータ処理モジュールにおける、様々なシステム状態に従う正しいデータ処理を保証する方法であって、

少なくとも1つの入力チャンネル上で、古い状態から新しい状態への変更を示す状態メッセージを受信する工程と、

30

前記新しい状態を有する全ての入力チャンネル上でのデータの受信をブロックする工程と、

データ処理ループのセット中のそれぞれのループに少なくとも1つのループチェックメッセージを送信する工程であって、前記セットは、結びつけられた入力チャンネル及び出力チャンネルを介して第1のデータ処理モジュールが接続されている全てのループを含み、当該入力チャンネルには前記古い状態が適用されている工程と、

前記セット中のそれぞれのループについて、前記第1のデータ処理モジュールにおいて前記ループチェックメッセージが受信されるか又は前記結びつけられた入力チャンネルが前記新しい状態を示す状態メッセージを受信しており、かつ前記第1のデータ処理モジュールの他の全ての入力チャンネルが前記新しい状態を示す状態メッセージを受信している場合に、前記ブロックを除去する工程と、

40

を含む方法に関する。

【0006】

本発明の第2の態様は、様々なシステム状態に従う正しいデータ処理を保証するための、データ処理システムに備えられる第1のデータ処理モジュールを含むコンピューティングデバイスであって、

前記第1のデータ処理モジュールは、
いくつかの入力チャンネルへのアクセスと、
いくつかの出力チャンネルへのアクセスと、
状態制御ユニットとを備え、

50

前記状態制御ユニットは、

古い状態から新しい状態への変更を示す状態メッセージが受信されている全ての入力チャンネル上でのデータの受信をブロックし、

データ処理ループのセット中のそれぞれのループに少なくとも1つのループチェックメッセージを送信し、ここで前記セットは、結びつけられた入力チャンネル及び出力チャンネルを介して前記第1のデータ処理モジュールが接続されている全てのループを含み、当該入力チャンネルには前記古い状態が適用されており、

前記セット中のそれぞれのループについて、前記第1のデータ処理モジュールにおいて前記ループチェックメッセージが受信されるか又は前記結びつけられた入力チャンネルが前記新しい状態を示す状態メッセージを受信しており、かつ前記第1のデータ処理モジュールの他の全ての入力チャンネルが前記新しい状態を示す状態メッセージを受信している場合に、前記ブロックを除去する

10

ように構成されているコンピューティングデバイスに関する。

【0007】

本発明の第3の態様は、第2の態様に係るコンピューティングデバイスを含むデータ処理システムに関する。

【0008】

本発明の第4の態様は、データ処理システム内の第1のデータ処理モジュールの状態制御ユニットを提供することによって、様々なシステム状態に従う正しいデータ処理を保証するためのコンピュータプログラムであって、

20

前記コンピュータプログラムは、データ媒体上のプログラムコードを含み、

前記プログラムコードは、前記データ媒体がコンピュータへと入れられたときに、

古い状態から新しい状態への変更を示す状態メッセージが受信されている全ての入力チャンネル上でのデータの受信をブロックし、

データ処理ループのセット中のそれぞれのループに少なくとも1つのループチェックメッセージを送信し、ここで前記セットは、結びつけられた入力チャンネル及び出力チャンネルを介して前記第1のデータ処理モジュールが接続されている全てのループを含み、当該入力チャンネルには前記古い状態が適用されており、

前記セット中のそれぞれのループについて、前記第1のデータ処理モジュールにおいて前記ループチェックメッセージが受信されるか又は前記結びつけられた入力チャンネルが前記新しい状態を示す状態メッセージを受信しており、かつ前記第1のデータ処理モジュールの他の全ての入力チャンネルが前記新しい状態を示す状態メッセージを受信している場合に、前記ブロックを除去する

30

ように構成されているコンピュータプログラムに関する。

【0009】

データ処理経路という表現は、複数のノードを含む経路を範囲に含むことを意図する。このとき、データはノードにおいて処理され、処理される経路において後続するノードへと送られる。またこのようなノードは、データ処理モジュールの形で提供される。

【0010】

データ処理経路との表現は、ループを形成する、上述の種類の経路を範囲に含むことを意図する。

40

【0011】

本発明の1つの利点は、状態変化がデータ処理モジュールを介して確定的且つ制御された方法で伝播することである。本発明によると、損失するか又は誤った状態で処理されるデータはないことが保証される。状態の更新は更に高速である。また、本発明により、動作中に競合状態のリスクを冒すことなく、システムが非線形データ処理のために設計されることが可能となる。

【0012】

尚、本明細書中で使用される場合の用語「備える(comprises/comprising)」は、記載される特徴、ステップ又は構成要素の存在を特定するものとして解釈され、1つ以上の他の

50

特徴、ステップ、構成要素又はそれらの集合の存在又は追加を除外するものではない。

【 0 0 1 3 】

次に、添付の図面に関連して本発明を更に詳細に説明する。

【図面の簡単な説明】

【 0 0 1 4 】

【図 1】複数のデータ処理モジュールを含み、有利な取引システムである本発明に係るコンピュータシステムを概略的に示す図である。

【図 2】種々のシステム状態に従ってデータの適切な処理を保証する方法における複数の方法ステップを示すフローチャートである。

【図 3】本発明に従ってデータ処理ループを判定する複数の方法ステップを示すフローチャートである。

10

【図 4】本発明の 1 つの変形例の原理に従って送出される複数のループチェックメッセージを例示するために使用される複数の相互接続ノードの第 1 の例を概略的に示す図である。

【図 5】データ処理ループの判定を例示するために使用される複数の相互接続ノードの第 2 の例を概略的に示す図である。

【図 6】第 1 のノードから開始するデータ処理ループを判定するために第 2 の例のノードがどのように処理されるかを例示する木を概略的に示す図である。

【図 7】第 4 のノードから開始するデータ処理ループを判定するために第 2 の例のノードがどのように処理されるかを例示する木を概略的に示す図である。

20

【図 8】本発明に係る方法がどのように実行されるかを例示するために、第 2 の例の相互接続ノードと、そのノードのうちの 2 つに新しい状態メッセージを送出するメッセージルータと、を概略的に示す図である。

【発明を実施するための形態】

【 0 0 1 5 】

以下の説明において、本発明を完全に理解するために、限定するためではなく説明する目的で特定のアーキテクチャ、インタフェース、技術等の特定の詳細について説明する。しかし、これらの特定の詳細から逸脱する他の実施形態において本発明が実施されてもよいことは、当業者には明らかだろう。他の例において、周知の装置、回路及び方法の詳細な説明は、不必要な詳細で本発明の説明を不明確にすることがないように省略される。

30

【 0 0 1 6 】

図 1 は、本発明に係る簡略コンピュータシステム 10 を概略的に示す図である。以下、システム 10 は取引システムに関連して説明される。しかし、本発明は取引システムに限定されず、任意の種類のコンピュータシステムにおいて使用されてもよいことが理解されるべきである。

【 0 0 1 7 】

システム 10 において、複数のデータ処理モジュール 12、14 及び 16 が存在する。各モジュールは、システムに関連する複数の関係者に対して複数の処理アクティビティを提供する異なる種類のモジュールである。取引システムにおける関係者は、例えば取引業者、証券取引所、取引機器が備えられた会社、当局、情報供給業者等のエンティティであってもよい。図 1 において 3 つのモジュールが示される。第 1 のモジュール 12 は市場モジュールであってもよく、第 2 のモジュール 14 は情報処理モジュールであってもよく、第 3 のモジュール 16 はトランザクションモジュールであってもよい。これらのモジュールは、実行されるいくつかの計算について、論理的な意味で、破線の円で示されるループ L 内で互いに接続されているかもしれない。これは、モジュールの実際の相互接続は別の方法で、例えばバスを使用して提供されてもよいが、処理されるデータは、ループが形成される順序でこれらのモジュールを通過することを意味する。

40

【 0 0 1 8 】

これらのモジュールは本発明に従って提供されうるモジュールの例にすぎず、更に多くのモジュールがあってもよいことが理解されるべきである。種々の関係者に対してアクテ

50

ィビティを実行する場合、これらのモジュールは、データを処理し且つ更なる処理のためにそのような処理済みデータを更に転送する。

【 0 0 1 9 】

第 1 のモジュール 1 2 は、複数の入力チャンネル 2 6、2 8 及び 3 0 へアクセスできる。本明細書において、第 1 の入力チャンネル 2 6 及び第 2 の入力チャンネル 2 8 はメッセージルータ 1 8 に接続される。これらのチャンネルは、種々の取引命令及び他の金融ビジネス命令であってもよい、システム 1 0 の外部からのデータを受信するために提供される。従って、システム外部で発生するデータは、本発明のこの変形例においてメッセージルータ 1 8 を通過する。第 1 のモジュール 1 2 が第 3 のモジュール 1 6 からの処理済みデータを受信する第 3 のチャンネル 3 0 が更に存在する。第 1 のモジュール 1 2 は、複数の出力チャンネル 3 2 及び 3 4 へ更にアクセスできる。処理が行なわれない場合、あるいは状態に依存する処理が行なわれない場合、第 1 の出力チャンネルは第 2 の処理モジュール 1 4 につながり、第 2 の出力チャンネル 3 4 は別のモジュール又は別のエンティティにつながってもよい。更に第 2 のモジュール 1 4 及び第 3 のモジュール 1 6 は、同様に入力チャンネル及び出力チャンネルを提供されてもよいことが本明細書において理解されるべきである。しかし、それらのうちの殆どは、本明細書をより良く説明するために本明細書では省略されている。しかし、図 1 において、第 3 のモジュール 1 6 の入力チャンネルに接続される第 2 のモジュール 1 4 上の 1 つの出力チャンネルと、第 1 のモジュール 1 2 の第 3 の入力チャンネル 3 0 に接続される第 3 のモジュール 1 6 の 1 つの出力チャンネルとが存在する。

【 0 0 2 0 】

第 1 のモジュール 1 2 は、入力チャンネル 2 6、2 8 及び 3 0 に接続されるデータ処理部 2 0 を更に含み、例えば第 1 の出力チャンネル 3 2 及び第 2 の出力チャンネル 3 4 上に出力するためにデータを処理するように構成される。第 1 のモジュール 1 2 は、データ処理部 2 0 に接続されている状態制御部 2 2 を更に含む。この状態制御部 2 2 は、システム全体に対するシステム構成データ即ち種々のモジュールに対するシステム構成データを含むシステム構成データ記憶装置 2 4 に更に接続される。第 2 のモジュール 1 4 及び第 3 のモジュール 1 6 もまた、通常、システム構成データ記憶装置 2 4 からのデータを受信できるような状態制御部を含むことが本明細書において理解されるべきである。また、第 1 のデータ処理モジュール 1 2 に提供されるループ判定部 2 3 が存在する。ループ処理判定部 2 3 は、システム構成データ記憶装置 2 4 に更に接続される。

【 0 0 2 1 】

モジュールは、コンピュータ、サーバ又は任意の他の演算デバイス等のような演算デバイスに提供される。システムの大きさに依存して、1 つ以上のモジュールが同一のコンピュータに存在してもよいし、1 つ以上のモジュールが異なるコンピュータに提供されてもよい。

【 0 0 2 2 】

図 1 において、古い状態から新しい状態への変化を示す状態メッセージ S が示される。これは、図 1 において $S=S1$ として示され、S1 は新しい状態である。更にメッセージ S は、メッセージルータ 1 8 から第 1 のモジュール 1 2 に提供されるものとして示される。また、第 1 のモジュール 1 2 から第 2 のモジュール 1 4 に送出されるものとして示されるループチェックメッセージ C が存在する。これらのメッセージが処理される方法はすぐに説明される。最後に、第 3 のモジュール 1 6 と第 1 のモジュール 1 2 との接続は、古い状態 S0 にあるものとして示される（破線で示される）。

【 0 0 2 3 】

取引システムのようなシステム 1 0 は種々の状態を有する。ある状態は取引が進行中であることを示す状態であってもよく、別の状態は取引が終了していることを示す状態であってもよい。更に、取引される証券の定義、特定の証券の定義、証券に対する種々の制限等であってもよい、種々のモジュールにより使用される参照データが更新されたことを、別の状態は示してもよい。全てのそれらの状態は、データ処理に対して異なる影響を有する。これは、ある状態におけるデータ処理が、通常、別の状態におけるデータ処理とは異

なることを意味する。図1のループLが示すように、第1の処理モジュール12により処理されたデータは処理のために第2のモジュール14に送出されてもよく、引き続いて第2のモジュール14は更なる処理済みデータを第3のモジュール16に供給する。最後に、第3のモジュールは、第2のモジュール14からのデータを処理し、処理のために第1のモジュール12に供給する。従って、これらの種々の種類の処理は互いにリンクされる。ここで第1の処理部12が新しい状態S1を受信する場合、第3の処理モジュール16により供給される「旧データ」は、何かがなされない限り、適用されるべき古い状態S0に従ってではなく、新しい状態に従って処理されるだろう。本発明の目的はこの問題を解決することである。

【0024】

10

取引システム等のシステムに提供される複数の種類のデータ処理ループが存在し、データ処理モジュールはそのようなループにおいてノードとして動作する。

【0025】

ループの1つの種類は、データフローにおける強いループ(SLDF)である。そのようなループにおいて、データは、出力チャンネル上で第1のノードを出発し、複数のノードを通過し、ここでデータは処理されてもよく、かつデータは第1のノードに再入し、ここでデータは再び処理されてもよく、最終的に同じ出力チャンネル上でこのノードを出発する。

【0026】

データフローにおける弱いループ(WLDF)は、データが、出力チャンネル上で第1のノードを出発し、複数のノードを通過し、ここでデータは処理されてもよく、かつデータは第1のノードに再入する場合のループである。しかしながらこの場合、データは、第1のノードを同じ出力チャンネル上で出発する必要はない。

20

【0027】

以下、SLDFは図1のシステムに存在しないと仮定する。

【0028】

次に、種々のシステム状態に従ってデータの適切な処理を保証する方法における複数の方法ステップを示すフローチャートである図2を更に参照して、本発明の原理に係るシステム10の基本的な機能性を説明する。

【0029】

30

最初に、システム全体は古い状態S0にしたがってデータを処理しているものと仮定する。次に、周知の原理に従って、種々のモジュール12、14、16において処理が実行される。しかし、ある特定の時点において、新しい状態である状態S1がメッセージルータ18により導入される。この状態は、システム10の外部からデータを受信する全てのデータ処理モジュールに対して送出される、新しい状態メッセージS1を使用することにより知らされる。図1の例から分かるように、このメッセージを受信する唯一のモジュール、即ち第1のデータ処理モジュール12が存在する。そのような状態メッセージは、図1に提供される例において、第1の入力チャンネル26及び第2の入力チャンネル28から状態制御部22に転送される。そのような状態メッセージは、1つの入力チャンネル上のみで、又はさらに多くの入力チャンネル上で受信されてもよいことが本明細書において理解されるべきである。従って、本発明に係る方法は、第1のデータ処理モジュール12が、第1の入力チャンネル26及び第2の入力チャンネル28を介して新しい状態S1を示す1つ以上の状態メッセージSを受信するステップ36から開始する。ステップ38において、これらの状態メッセージは状態制御部22に転送される。第1のデータ処理モジュールは最初には、新しい状態S1を有する入力チャンネル26、28上のデータをブロックする。そして第1のデータ処理モジュールは、旧データを空にするよう、すなわち内部に有するあらゆるデータを古い状態S0に従って処理及び出力するようデータ処理部20に指示する。本発明の第1の変形例に従って、状態制御部22は、ループチェックメッセージCを第1のデータ処理部が接続されている複数のループに送出する。第1のデータ処理モジュール12は、関連する入力チャンネル及び出力チャンネルでそのようなループに接続されている。第1のデータ

40

50

処理モジュール 12 が接続されるデータ処理ループが判定される方法は、すぐに説明される。状態制御部 22 はそして、ステップ 40 において、少なくとも 1 つのループチェックメッセージ C をそのようなデータ処理ループの集合に送出する。本発明に従うとここでこの集合は、第 1 のデータ処理モジュール 12 の関連する入力チャネルが古い状態 S0 を有するループのみを含む。従って、これらのループに関連する入力チャネルに対しては依然として古い状態 S0 が適用されている。図 1 の例から分かるように、状態制御部 22 は、メッセージ C をそのようなループ L の一部である第 1 の出力チャネル 32 に送出するのみである。このメッセージ C は、本発明の 1 つの変形例において C(S1, loop, 1) として送出される。ここで、S1 は新しい状態を示し、ループは該当するループを示し、1 は、ループ内で通過されたリンクの数を示すループカウンタであり、どのモジュールが次にメッセージを受信するのかが判定するために使用される。本明細書において、このメッセージ C は、開始ノードで開始し開始ノードで終了するループに属する全てのモジュール又は全てのノードのリストを更に含み、ここでは開始ノードは第 1 のモジュールである。メッセージ C は第 2 のモジュール 14 である次のノードにより処理され、第 2 のモジュール 14 は同様に旧データを空にした後ループチェックメッセージ C を転送する。リンクカウンタは、メッセージが C(S1, loop, 2) と見えるように増分される。このように、ループの全ノードはループチェックメッセージ C を処理し、最終的にループチェックメッセージ C は、通常は第 3 の入力チャネル 30 を介して再び第 1 のモジュール 12 に到達するだろう。状態制御部 22 は、ステップ 42 においてループチェック条件が満たされることを待つ。この待機中、データ処理部 20 は、旧データを空にしている、即ち古い状態 S0 に従ってデータを処理及び出力しているはずである。ループチェック条件は、ループに関連する入力チャネルのそれぞれについて、ループチェックメッセージ C、又は新しい状態 S1 を示す状態メッセージ S のいずれかが受け取られたことである。その後状態制御部 22 は、全ての他の入力チャネル上で新しい状態を示す状態メッセージが存在するのを待つようデータ処理部 20 に指示してもよい。これが行なわれるとステップ 44 において状態制御部 22 は、第 1 のモジュール 12 について状態を新しい状態 S1 に変化させ、入力チャネルのブロックを除去する。その後状態制御部 22 は、ステップ 46 において、全ての出力チャネル上で新しい状態を示す状態メッセージを送出する。上述のように、第 1 のモジュール 12 が、チェック中に全ての入力チャネル上で、即ち第 3 のチャネル 30 上でも新しい状態を示す状態メッセージを受信する場合、状態制御部 22 は、状態を直接変化させてループチェックを中断してもよい。

【0030】

チェックを開始したモジュールが入力チャネル上で新しい状態を示す状態メッセージを受信する場合、そのモジュールは、状態を変化させる準備ができていないとしても新しいチェック手順を開始するべきではない。そのモジュールは、それに到達できる旧データがそれ以上ないことを保証するために、ループから十分なチェックメッセージが受信されてくるまで単に待機する必要がある。これと同一の手順は、システム内の他のモジュールにより更に実行される。

【0031】

新しい状態を有し、且つ別のノードで発生するループチェックメッセージを入力チャネル上で別のノードから受信するモジュールは、このメッセージを削除する必要がある。

【0032】

W L D F が可能である場合に適用される本発明の第 2 の変形例に従うと、状態制御部は、全ての入力チャネル上で新しい状態が存在することを示す状態メッセージが、第 1 のモジュールとループを全く共有しないモジュールからやってくるまで、ループチェックメッセージを送出して待機する。その後状態制御部は、旧データを空にしてループチェックメッセージを送出する。従って、図 1 の例において、新しい状態が第 1 の入力チャネル 26 及び第 2 の入力チャネル 28 上で同時に送出されなかった場合に待機が生じる。同様に、ループの他のモジュールは、ループのループチェックメッセージを転送する前に、ループの一部ではない全ての入力チャネルが新しい状態を受信するのを待つ必要がある。W L D

10

20

30

40

50

Fの場合、第1のモジュールが属し且つこれを介して新しい状態を示す状態メッセージがまだ全く受信されていない全てのループは、N回チェックされる必要があることが更に不可欠である。ここで、NはL又はM+1の最小値であり、Lはループが考慮されるモジュールに対する入力チャネルの数であり、Mは同じループに属するこのモジュールの出力チャネルの数である。また、新しいチェックの集合は、前のチェックの集合が完了するまでさらに実行されることができない。この場合、第1のモジュールは、入力チャネル上の新しい状態を待つ必要はなく、十分なループチェックメッセージを受信次第、旧データを空にして新しい状態に入ってもよい。

【0033】

ループ内のノードのデータを含む伝達されるループメッセージの代わりとして、全てのノード、即ちノードにおけるデータ処理モジュールが、ループ内の他のノードが一部をなすループの知識を有することが可能である。すると上述のループチェックメッセージは、伝達される必要があるノードの情報を全く含む必要がない。その代わりに、受信ノードは、この事前記憶された情報に基づいてループにおいて次のノードを判定する。また、この判定はシステムがセットアップされる際に1回だけ行なわれてもよく、その結果は状態が実際に変化する際に使用される。

【0034】

次に、本発明に従ってデータ処理ループを判定するための、第1の処理モジュール12のループ判定部23において実行される複数のステップを示すフローチャートを示す図3に関連して、ループにおけるノードとして存在する他のデータ処理モジュールについての判定がどのように判定されるかを説明する。本発明のいくつかの変形例において、このループ判定部23は状態制御部22と組み合わされてもよい。更に他のデータ処理モジュールもまた、このノード判定機能を搭載してもよいことが本明細書において理解されるべきである。システム全体に対して1つのループ判定部が中枢として提供されてもよいことが更に理解されるべきである。

【0035】

ループ判定は、システムがセットアップされている際に有利に提供される。実際のループ判定は、ステップ48においてループ判定部23が第1のモジュール12のシステム構成データを読み取ることから開始する。そのようなシステム構成データは、システム構成記憶装置24において有利に提供される。通常、全てのモジュールはこの記憶装置24へアクセスでき、するとこの記憶装置24は全てのモジュールのシステム構成データを含む。ここで、システム構成データは、第1のモジュールに対する全ての入力チャネル及び第1のモジュールからの全ての出力チャネル、並びに他のデータ処理モジュール等のような他のどのエンティティにこれらのチャネルが接続されるのかに関するデータを含む。このデータからループ判定部23は、ステップ50において最初に他のデータ処理モジュールに対してどの接続が存在するのかを判定する。そのようなデータは、ノード間の接続を示す複数の要素を含むいわゆる隣接行列Aとして提供されてもよい。そのような行列の列及び行の双方は、システム内のノードと、2つのノード間に接続が存在するか否かを示す。そのような接続が存在しない場合は要素 $a(i, j)=0$ であり、存在する場合は $a(i, j)=1$ である。ループを検出するために、ループ判定部23は、ステップ50において接続行列Aを調査することにより第1のモジュールが接続される全ての他のモジュールを検出する。これらの接続は、ループであってもよいが、終端を有する経路のような他の種類の経路であってもよい、複数の経路を構成する。ステップ52において作成されるそのような経路毎に、ループ判定部23は、ステップ54において経路において後続するモジュールのシステム構成データを読み取る。ステップ56においてこの経路がそれ以上データ処理モジュールを有さない場合経路は終了しており、ステップ58において経路はループがないものとして判定される。しかし、次のモジュールに後続するモジュールへの接続が経路に存在する場合、即ちステップ56において経路が終了しない場合、ループ判定部23はこの後続モジュールを調査する。ステップ60においてこの後続モジュールが第1のモジュールである場合、ループ判定部23は、ステップ64において経路が実際にループであること

10

20

30

40

50

を判定し、ステップ66においてループデータを格納する。ループデータは、一般に、そのようなループに含まれるデータ処理モジュール及びデータ処理モジュールのチャネル、及びどのモジュールでそのループが始まるかに関するデータを含む。このデータは、システム構成データ記憶装置24に有利に格納されてもよい。また、このデータは、第1のデータ処理モジュールにローカルに格納されてもよい。ステップ60において後続モジュールが第1のモジュールではなかった場合、ステップ62において次のモジュールは後続モジュールにされ、ステップ54においてこのモジュールのシステム構成データがチェックされる。これは、経路がループであると判定されるか又はループではないと判定されるまで繰り返される。従って、ループチェックメッセージを送出する状態制御部は、このモジュールがどのループに接続されているのかを学習するために、システム構成データ記憶装置24を介して、対応するデータ処理モジュールのループを識別する適用可能なループデータの場所を探し出す。

10

【0036】

分岐が存在すること、即ち1つのノードは2つ以上の他のノードに接続することが更に可能であることが本明細書において理解されるべきである。従って、経路は、分岐点において別の経路から分岐される部分を含んでもよい。

【0037】

本発明に係るデータ処理モジュールとして提供される複数の相互接続ノードの第1の例を概略的に示す図4を参照して、ループ毎にいくつかのループチェックメッセージが必要なことを示す一例を説明する。

20

【0038】

図4において、第1のノード68は、第2のノード70の入力チャネルに接続する出力チャネルを有し、さらに第2のノード70は第3のノード72の入力チャネルに接続される出力チャネルを有する。第3のノード72は2つの出力チャネルを有し、第1の出力チャネルは第1のノード68の第1の入力チャネルに接続され、第2の出力チャネルは第4のノード74の入力チャネルに接続される。最後に第4のノード74は、第1のノード68の第2の入力チャネルに接続される出力チャネルを有する。本明細書において、第1のノード68で発生する2つの経路が存在する。従って、動作する1つのループは、第1のノード68、第2のノード70、第3のノード72及び第1のノード68から構成される。動作する第2のループは、第1のノード68、第2のノード70、第3のノード72、第4のノード74及び第1のノード68から構成される。本明細書において、これら双方のループがチェックされる。しかし、旧データは、第2のノード70へのリンク上を第1のノード68から出発し、第3のノード72と第1のノード68との間のリンクを介して戻っているかもしれない。(SLDFが存在しないという仮定の下)旧データは第4のノード74に再入力できず且つこの第4のノード74と第1のノード68とのリンクを介して通り過ぎることができないため、第1のループ上でもう1度チェックが行なわれるか又は2つのループの第3のノード72において旧データが存在するかもしれないという疑いが存在すれば十分である。

30

【0039】

一般的な例において、WLDFが除外できない場合、チェックされる回数は上述のようにN回である。ここで上述のように $N = \min(L, M+1)$ である。これは、最初にループがチェックされる時に、旧データはどちらのループからも来ることができ、且つ旧データが最後にノードを通過した時に使用したのと同じ出力チャネルを介して第1のノード、即ち開始ノードから出ていくことができるためである。2回目に回る時は、旧データは同一の出力チャネルを通過して第1のノードを出ることができない。従って、M回通過した後は、旧データは1つの出力チャネルを通過して第1のノードから出ることができ、旧データが再び第1のノードに戻らないことを保証するために更にチェックする必要がある。入力チャネルLの数がM+1未満である場合、第1のチェックが行なわれた際に1つの入力チャネルを通過して旧データが到着したならば旧データは他の時には同一チャネルを通過して入ることはできないため、同様の推論によりL回チェックすれば十分である。

40

50

【 0 0 4 0 】

非常に多くのチェックが必要な理由は、ネットワーク個別メッセージが通過する経路を認識する方法がないからである。しかし、特定の例において、「ボトルネック」を識別することにより、設定されるチェックの集合が実行される回数を減らすことが可能であり、最小限の必須の集合は1つ又は少数のリンクを含む。この場合、集合に存在するリンクの数Kに比べてもう1回多く必須の集合をチェックする必要がある。しかし、殆ど接続を有さないネットワークの場合、一般に第1の方法が好まれる。

【 0 0 4 1 】

次に、図5、図6及び図7を参照して、ループの識別方法の一例を説明する。図5は、データ処理ループの判定を例示するために使用される複数の相互接続ノードの第2の例を概略的に示す図である。図6は、第1のノードから開始するデータ処理ループを判定するために第2の例のノードがどのように処理されるかを例示する木を概略的に示す図である。図7は、第4のノードから開始するデータ処理ループを判定するために第2の例のノードがどのように処理されるかを例示する木を概略的に示す図である。

【 0 0 4 2 】

図5の相互接続されたノードは図4の相互接続されたノードに類似する。ここでの差異は、第4のノード74の入力チャンネルに接続される出力チャンネルを有する第5のノード76が存在すること、第1のノード68がその出力チャンネルを第3のノード72の入力チャンネルに接続させること、第3のノード72が第2のノード70の入力チャンネルに接続される第1の出力チャンネルを有すること、並びに第2のノード70が第1のノード68の第1の入力チャンネルに接続される出力チャンネルを有することである。その他の全ての点において、図5は図4と同一である。

【 0 0 4 3 】

図5のノードに対する隣接行列Aは、以下のように見える。

【数1】

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

【 0 0 4 4 】

本明細書において、第1の行は第1のノードがデータを送出する他のノードを示し、第2の行は第2のノードがデータを送出する他のノードを示し、第3の行は第3のノードがデータを送出する他のノードを示し、第4の行は第4のノードがデータを送出する他のノードを示し、第5の行は第5のノードがデータを送出する他のノードを示す。第1の列は第1のノードがデータを受信する他のノードを示し、第2の列は第2のノードがデータを受信する他のノードを示し、第3の列は第3のノードがデータを受信する他のノードを示し、第4の列は第4のノードがデータを受信する他のノードを示し、第5の列は第5のノードがデータを受信する他のノードを示す。これは、図5のノードの場合、第1の行には第3の列に対する設定が1つのみ存在することを意味し、これは第1のノードが第3のノードにデータを送出することを示す。第2の行には第1の列に対する設定のみが存在し、これは第2のノードが第1のノードにデータを送出することを示す。第3の行には第2の列及び第4の列に対する設定のみが存在し、これは第3のノードが第2のノード及び第4のノードにデータを送出

することを示す。第4の行には第1の列に対する設定のみが存在し、これは第4のノードが第1のノードにデータを送出することを示す。最後に、第5の行には第4の列に対する設定のみが存在し、これは第5のノードが第4のノードにデータを送出することを示す。

【0045】

本明細書において、ループ検出方法は、最初に、第1のノード68から開始する全てのノードを検出するように適用される。図5のノードからは図6に示す木が取得される。第1のノード68はループ検出が開始される開始ノードであるため、図6において右手側に示される。この第1のノード68からは第3のノード72へのリンクが1つ存在する。第3のノード72から2つのリンクが存在する。1つ目のリンクは第2のノード70にリンクし、引き続いて第1のノード68につながる。2つ目のリンクは第4のノード74にリンクし、引き続いて第1のノード68につながる。この木を作成するために、第1のノード68から開始し、第3のノード72への唯一の可能な経路に沿って進む。この第3のノード72からは2つの選択が可能である。即ち、第2のノード70に進むか又は第4のノード74に進むかのいずれかである。第2のノード70からは第1のノード68にのみ進むことができ、開始点に戻るにより第1のループが識別された。従って、分岐はここで終了する。代わりに第4のノード74に進む場合、この場合も第1のノード68にのみ進むことができ、第2のループが識別される。未完の分岐が存在しないため、探索が終了し且つ第1のノード68から開始する全ての可能なループを示す木が取得される。

【0046】

同一のループ検出方法は、第4のノード74から開始する全てのループ上に適用される、こうして図7の木が取得される。この木を作成するために、第4のノード74から開始し、そこから第1のノード68への唯一の可能な経路を進む。第1のノード68からは第3のノード72で終了することのみが可能である。第3のノード72からは2つの経路が可能である。即ち、1つ目の分岐は第4のノード74につながり、2つ目の分岐は第2のノード70につながる。第4のノード74が開始点であるため、第1の分岐は第3のループを識別する。他の分岐から、第2のノード70は第1のノード68につながり、且つ第1のノード68は第3のノード72につながる事が分かる。第3のノードにおいて、第2のノード70への第3の分岐及び第4のノード74への第4の分岐の可能性が存在する。第3の分岐において、第2のノード72からは第1のノード68にのみ進むことができる事が明らかである。しかし、第1のノード68に到達するために、第1のノード68、第3のノード72、第2のノード70及び第1のノード68という順番は2回現れる。これは、同一のサブループを2回通過したことを意味する。従って、本発明に従って、木の最後の分岐点の後、即ち第3のノード72と第2のノード74との間でこの分岐を切断できる。第4の分岐は開始点である第4のノード74につながり、こうして第4のループが検出される。更なる分岐は存在しないため、探索が終了して木が完成する。

【0047】

また、より大きなループ、即ち第4のループには、第4のノード74、第1のノード68、第3のノード72及び第4のノード74から構成される第3のループと同一であるサブループを構成するリンクが存在することに留意する。従ってより小さな第3のループは別個にチェックされる必要はない。これは、第4のノード74、第1のノード68、第3のノード72、第2のノード70、第1のノード68、第3のノード72及び第4のノード74から構成されるより大きな第4のループがチェックされる際にチェックされるためである。

【0048】

同一のループチェック原理が同様に全ての他のノードに適用されるが、本明細書では更に詳細には説明しない。

【0049】

第2の例の相互接続ノード、並びに第1のノード68及び第5のノード76に接続されるメッセージルータ18を概略的に示す図8に関連して、新しい状態が導入される際に起こることについて説明する。本明細書において、メッセージルータ18は、新しい状態S1

10

20

30

40

50

を示すメッセージを第1のノード68及び第5のノード76に送出する。ここでは、システムにW L D Fが存在しないことが更に仮定される。本発明の基本原理は、第1のノード68が全ての入力チャンネル上に新しい状態メッセージを有するかを、第1のノード68がまずチェックすることである。第1のノード68は、自身が新しい状態メッセージを有さないことと、新しい状態を示す状態メッセージが欠けている入力チャンネルが、第1のノード68がループを共有するノードからやってきていることとに気付くと、第1のノード68は自身の旧データを空にしてこれらのループの各々に対してループチェックメッセージを送出する。この場合、第1のノード68は、ループチェックメッセージC(S1, loop1, 1)を第3のノード72に送出し、別のループチェックメッセージC(S1, loop2, 1)を更に第3のノード72に送出する。これらのメッセージのうちの1つ目は第1のループに送出され、2つ目は第2のループに送出される。これらのループは上記のように識別された。第5のノード76は、自身が全ての入力チャンネル上に新しい状態メッセージS1を有するかをチェックして、有することを認識する。その後第5のノード76は、状態を新しい状態S1に変化させ、新しい状態メッセージを第4のノード74に送出する。第1のノード68からループチェックメッセージC(S1, loop1, 1)及びC(S1, loop2, 1)を受信した第3のノード72は、第3のノード72がループを全く共有しないノードからやってくる入力チャンネルを全く有さない。従って、第5のノード76はどんな新しい状態メッセージをも待つ必要はない。従って、第3のノード72は、ループチェックメッセージの1つであるC(S1, loop1, 2)を第2のノードに転送し、他のループチェックメッセージC(S1, loop2, 2)を第4のノードに転送する。ここで、ループカウンタは双方のループチェックメッセージにおいて増分されている。一方、第4のノード74は入力チャンネルのうちの1つのチャンネル上に存在する新しい状態メッセージを有するため、第4のノード74は全ての入力チャンネル上に新しい状態メッセージを有するかをチェックする。第4のノード74は新しい状態メッセージを全ての入力チャンネル上には有さないが、残りの入力チャンネルが第4のノード74とループを共有するノードからやってくるため、第4のノード74はループチェックメッセージC(S1, loop4, 1)を第1のノード68に送出する。また、第2のノード70は、ループカウンタを増分させてC(S1, loop1, 3)メッセージを第1のノード68に転送する。第4のノード74が第3のノード72からループチェックメッセージC(S1, loop2, 2)を取得すると、第4のノード74は、第4のノードがループを全く共有しないノードからやってくる全ての入力チャンネル上に新しい状態メッセージを有することを認識する。その後第4のノード74は、ループチェックメッセージC(S1, loop2, 3)を第1のノード68に送出する。第1のノード68は、第4のノード74で発生するループチェックメッセージを更に受信し、そのメッセージをC(S1, loop4, 2)として第3のノード72に転送する。第1のノード68は、第2のノード70からのループチェックメッセージC(S1, loop1, 3)を更に受信し、第1のループがチェックされたことを確認する。次に第1のノード68は、第4のノード74からループチェックメッセージC(S1, loop2, 3)を受信し、第2のループがチェックされたことを更に確認する。従って、第1のノード68は全てのノードのチェックを完了し、第1のノード68がループを全く共有しないノードからやってくる全ての入力チャンネル上に新しい状態メッセージを有することを認識する。そしてW L D Fが存在しないため、ここでこのノードについてのチェックを完了する。従って、第1のノード68は新しい状態S1を身につけ、新しい状態S1を示す新しい状態メッセージを第3のノード72に送出する。第3のノード72は、C(S1, loop4, 3)ループチェックメッセージを第2のノード70に更に転送し、そして新しい状態メッセージを受信すると新しい状態S1を身につけ、新しい状態S1を示す新しい状態メッセージを第2のノード70及び第4のノード74に送出する。第2のノード70は、ループチェックメッセージ(S1, loop4, 4)を第1のノード68に転送する。このメッセージが第1のノード68に到達すると、第1のノード68は既に新しい状態S1にあるため、このメッセージは削除される。第2のノード70は、第3のノード72から新しい状態メッセージを受信し、さらにこの新しい状態S1を身につけ、新しい状態メッセージを第1のノード68に送出する。第1のノード68は既に新しい状態S1を身につけているため、この新しい状態メッセージを削除する。

10

20

30

40

50

第４のノード７４は、第３のノード７２から新しい状態メッセージを受信し、新しい状態Ｓ１を身につけ、新しい状態メッセージを第１のノード６８に送出する。その後第１のノード６８は、既に問題の状態を身につけているため、この状態メッセージを削除する。

【００５０】

その後、それ以上メッセージは送出されず、ネットワークは更新された。

【００５１】

代わりとしては、ループの一部ではない全ての入力チャネルがノードが新しい状態を受信することを待つ必要がある。これは、ＷＬＤＦが存在しない場合に当てはまる。するとノードは迅速に進行し、自身の旧データを空にし、ループチェックメッセージを送出する。この場合、各ループに送出されるただ１つのループチェックメッセージが更に存在する。

10

【００５２】

本発明は複数の利点を有する。本発明によると、状態変化は、データ処理モジュールを介して確定的且つ制御された方法で伝播する。本発明によると、損失されるか又は誤った状態で処理されるデータはないことが保証される。状態の更新は更に高速である。また、本発明はさらに、動作中に競合状態のリスクを冒すことなく、システムを非線形データ処理のために設計することを可能とする。

【００５３】

上述されたように、データ処理部、状態制御部及び判定部を含む本発明のデータ処理モジュールは、コンピュータ技術を使用して実現される。従って、それらは、機能を実行するコンピュータプログラムコードと共に１つ以上のプロセッサによって実現されてもよい。

20

【００５４】

現在最も実用的かつ好適であると考えられる実施形態に関連して本発明を説明したが、本発明は開示される実施形態に限定されず、逆に種々の変形例及び均等な構成を範囲に含むことを意図することが理解されるべきである。例えばメッセージルータは、入力チャネルに提供されることに限定されず、状態変化を知らせる他の方法が備えられてもよい。また、ループチェック部及びループ判定部は同一部に提供されてもよい。上述のように、システム全体に対して１つのループ判定部が代わりに提供されてもよく、この１つのループ判定部はシステム全体のループ判定を担う。従って、本発明は以下の請求の範囲によってのみ限定される。

30

【図 1】

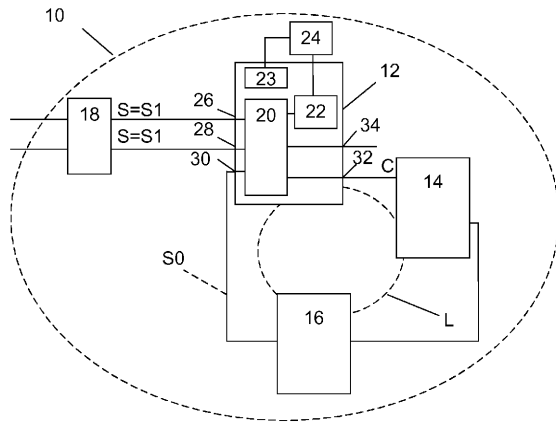


FIG. 1

【図 2】

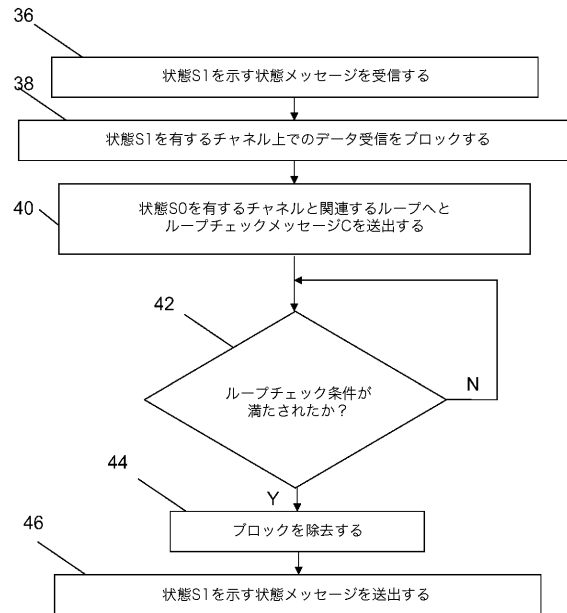


FIG. 2

【図 3】

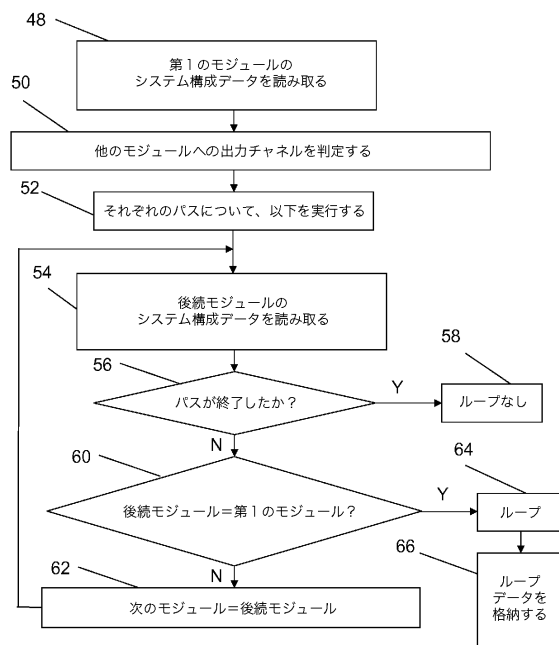


FIG. 3

【図 4】

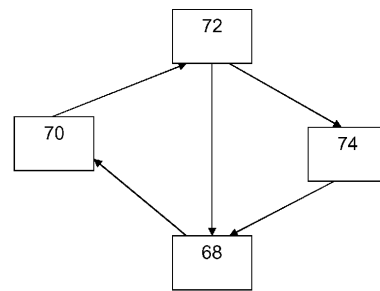


FIG. 4

【図 5】

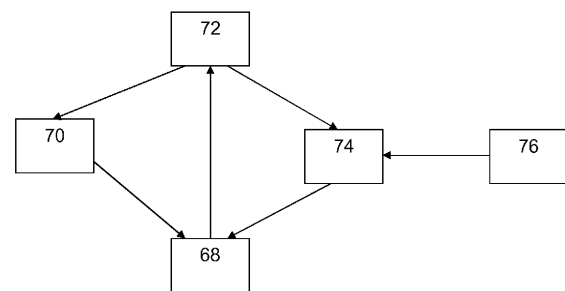


FIG. 5

【圖 7】

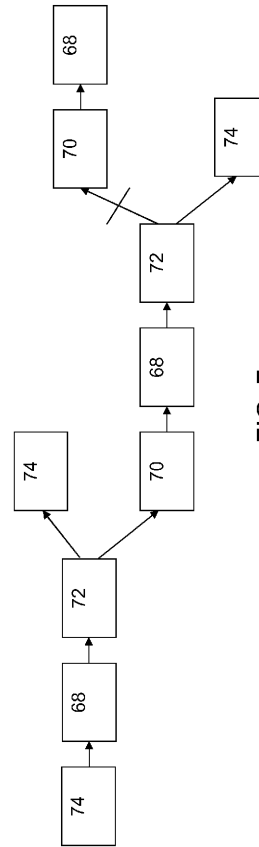


FIG. 7

```

graph TD
    72[72] --> 70[70]
    72[72] --> 74[74]
    70[70] --> 68[68]
    74[74] --> 68[68]
    76[76] --> 74[74]
    18[18] -- "S=S1" --> 68[68]
    18[18] -- "S=S1" --> 76[76]
  
```

FIG. 8

フロントページの続き

- (74)代理人 100115071
弁理士 大塚 康弘
- (74)代理人 100116894
弁理士 木村 秀二
- (74)代理人 100130409
弁理士 下山 治
- (72)発明者 ヤール, ヘンリク
スウェーデン国 ストックホルム エス - 1 1 3 5 3 , マルクヴァルドスガタン 3 , 1 ティアール エグ
- (72)発明者 ローセンルンド, マグヌス
スウェーデン国 ストックホルム エス - 1 1 7 3 8 , ベルイスンドス ストランド 4 2 , 2 ティアール
- (72)発明者 グリーン, トーマス, イー.
アメリカ合衆国 ニューヨーク州 1 0 0 0 2 2 , ニューヨーク, 3 0 0 イースト 5 9 ス , アpartment 2 0 0 2
- (72)発明者 エリクソン, ヨハン
スウェーデン国 ストックホルム エス - 1 1 3 5 6 , ビルゲル ヤールスガタン 1 0 9 エー, 1 ティアール エグ

審査官 多賀 実

- (56)参考文献 特開 2 0 0 6 - 0 7 2 7 9 7 (J P , A)
特開 2 0 0 6 - 2 5 9 8 6 3 (J P , A)
特開平 0 5 - 1 5 1 1 7 6 (J P , A)

- (58)調査した分野(Int.Cl. , D B 名)
G 0 6 F 9 / 5 2
G 0 6 F 1 1 / 3 0
G 0 6 F 1 2 / 0 0
G 0 6 F 1 5 / 0 0
G 0 6 F 1 5 / 1 7 3
G 0 6 F 1 5 / 8 2