

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3732113号

(P3732113)

(45) 発行日 平成18年1月5日(2006.1.5)

(24) 登録日 平成17年10月21日(2005.10.21)

(51) Int. Cl.

F I

G 0 6 F 12/00 (2006.01)

G 0 6 F 12/00 5 1 8 A

G 0 6 F 15/00 (2006.01)

G 0 6 F 15/00 3 1 0 U

請求項の数 19 (全 62 頁)

(21) 出願番号	特願2001-143203 (P2001-143203)	(73) 特許権者	595080061
(22) 出願日	平成13年5月14日(2001.5.14)		株式会社八十二銀行
(65) 公開番号	特開2002-342138 (P2002-342138A)		長野県長野市大字中御所字岡田178番地8
(43) 公開日	平成14年11月29日(2002.11.29)	(74) 代理人	100109416
審査請求日	平成13年5月24日(2001.5.24)		弁理士 鈴木 健治
		(72) 発明者	石井 雅由
			長野市大字中御所字岡田178番地8株式会社八十二銀行内
		審査官	平井 誠

最終頁に続く

(54) 【発明の名称】 トランザクション制御システム、方法及びプログラム

(57) 【特許請求の範囲】

【請求項1】

端末からネットワークを介して入力される業務処理要求を受信する受信手段と、この受信手段にて受信した業務処理要求を処理する演算手段と、この演算手段の処理結果を記録するファイルとを備え、

前記演算手段は、

所定のオンライントランザクション及び所定の派生トランザクションを並列に実行し、この各トランザクションの実行に際して、それぞれ、前記ファイルのデータ項目をロックし、処理し、前記ファイルに当該処理結果を記録し、当該記録後に前記ロックを解除するトランザクション実行部と、

前記端末から入力される業務処理要求をオンライントランザクションとして前記トランザクション実行部に処理させ、当該オンライントランザクションの処理結果を前記端末への応答として出力する制御をするオンライン応答部と、

前記派生トランザクションを前記トランザクション実行部に処理させ、各トランザクションのコミットを管理する派生トランザクション管理部とを備えたトランザクション制御システムにおいて、

前記各トランザクションの実行中に、新たなトランザクションの起動要求依頼を出力するトランザクションを派生起動トランザクションとし、当該起動要求依頼されたトランザクションを派生トランザクションとして、

前記派生トランザクションは、前記オンライントランザクションの前記処理結果に付随

10

20

して生じ、かつ、前記オンライントランザクションの応答には必要ない付随業務の全部又は一部を処理するトランザクションであって、

前記派生トランザクション管理部は、

前記派生トランザクションの起動と開始とを管理し、

前記起動要求依頼が出力された際に、当該派生トランザクションの起動処理とは別に、当該派生トランザクションの起動要求依頼に関連する派生起動情報を前記ファイルに記録し、当該派生起動情報の記録を、前記派生起動トランザクションの原子性に包む制御をすることで、派生トランザクションの起動を管理する派生起動情報格納制御機能と、

当該起動要求依頼された派生トランザクションの開始を待機せずに当該派生起動トランザクションをコミットさせる機能と、

10

前記派生起動トランザクションがコミットした際に、当該派生起動トランザクションによる前記ロックを解除することで他のオンライントランザクションによる当該ロック解除されたデータ項目へのアクセスを許容しつつ、当該起動要求依頼された派生トランザクションの処理を開始させることで、前記派生トランザクションの開始を管理する直列性管理機能とを備えたことを特徴とするトランザクション制御システム。

【請求項2】

前記派生トランザクション管理部は、前記派生起動トランザクションがコミットした後に前記派生トランザクションがアボートした場合には前記派生起動情報に基づいて当該派生トランザクションを再起動させる派生再起動制御機能とを備えたことを特徴とする請求項1記載のトランザクション制御システム。

20

【請求項3】

前記演算手段は、前記オンライン応答部及び派生トランザクション管理部によって実行が管理される各トランザクションの総量に応じて前記派生トランザクションの開始を制限させる総量制御部を備え、

前記総量制御部は、実行を待機しているトランザクションの総数を監視する実行状況監視機能と、

この実行状況監視機能によって実行待機中のトランザクション数が予め定められた数を超えたと判定された場合には、前記派生起動情報が前記ファイルに記録された状態で派生トランザクションの開始を中止する派生起動中止制御機能を備えたことを特徴とする請求項1又は2記載のトランザクション制御システム。

30

【請求項4】

前記トランザクション実行部は、

前記派生起動トランザクションの実行によって業務上の一貫性に関して中間的なデータが生じる場合には当該中間的なデータについて耐久性を持った状態で前記ファイルに記録し、

この業務処理要求の実行によって生じた業務上の一貫性に関する中間的な状態を解消して業務上の一貫性を保つための判定及び実行を前記派生トランザクションとして実行することを特徴とする請求項1、2又は3記載のオンラインシステム。

【請求項5】

端末からネットワークを介して入力される業務処理要求を受信する受信手段と、この受信手段にて受信した業務処理要求を処理する演算手段と、この演算手段の処理結果を記録するファイルとを備え、

40

この演算手段は、

所定のオンライントランザクション及び所定の派生トランザクションを並列に実行し、この各トランザクションの実行に際して、それぞれ、前記ファイルのデータ項目をロックし、処理し、前記ファイルに当該処理結果を記録し、当該記録後に前記ロックを解除するトランザクション実行部と、

前記端末から入力される業務処理要求をオンライントランザクションとして前記トランザクション実行部に処理させ、当該オンライントランザクションの処理結果を前記端末への応答として出力する制御をするオンライン応答部と、

50

前記派生トランザクションを前記トランザクション実行部に処理させ、各トランザクションのコミットを管理する派生トランザクション管理部とを備えたトランザクション制御システムの前記派生トランザクション管理部を使用して、

前記各トランザクションを管理するトランザクション制御方法であって、

前記各トランザクションの実行中に、新たなトランザクションの起動要求依頼を出力するトランザクションを派生起動トランザクションとし、当該起動要求依頼されたトランザクションを派生トランザクションとして、

前記派生トランザクションは、前記オンライントランザクションの前記処理結果に付随して生じ、かつ、前記オンライントランザクションの応答には必要ない付随業務の全部又は一部を処理するトランザクションであって、

前記起動要求依頼が出力された際に、当該派生トランザクションの起動処理とは別に、当該派生トランザクションの起動要求依頼に関連する派生起動情報を前記ファイルに記録し、当該派生起動情報の記録を、前記派生起動トランザクションの原子性に包む制御をすることで、派生トランザクションの起動を管理する工程と、

前記派生トランザクションの開始を待機せずに当該派生起動トランザクションをコミットさせる工程と、

前記派生起動トランザクションがコミットした際に、当該派生起動トランザクションによる前記ロックを解除することで他のオンライントランザクションによる当該ロック解除されたデータ項目へのアクセスを許容しつつ、当該起動要求依頼された派生トランザクションの処理を開始させることで、前記派生トランザクションの開始を管理する工程とを備えたことを特徴とするトランザクション制御方法。

【請求項6】

端末からネットワークを介して入力される業務処理要求を受信する受信手段と、この受信手段にて受信した業務処理要求を処理する演算手段と、この演算手段の処理結果を記録するファイルとを備え、

この演算手段は、

所定のオンライントランザクション及び所定の派生トランザクションを並列に実行し、この各トランザクションの実行に際して、それぞれ、前記ファイルのデータ項目をロックし、処理し、前記ファイルに当該処理結果を記録し、当該記録後に前記ロックを解除するトランザクション実行部と、

前記端末から入力される業務処理要求をオンライントランザクションとして前記トランザクション実行部に処理させ、当該オンライントランザクションの処理結果を前記端末への応答として出力する制御をするオンライン応答部と、

前記派生トランザクションを前記トランザクション実行部に処理させ、各トランザクションのコミットを管理する派生トランザクション管理部とを備えたトランザクション制御システムの前記派生トランザクション管理部に所定の工程を実現させるためのトランザクション制御プログラムであって、

前記各トランザクションの実行中に、新たなトランザクションの起動要求依頼を出力するトランザクションを派生起動トランザクションとし、当該起動要求依頼されたトランザクションを派生トランザクションとして、

前記派生トランザクションは、前記オンライントランザクションの前記処理結果に付随して生じ、かつ、前記オンライントランザクションの応答には必要ない付随業務の全部又は一部を処理するトランザクションであって、

前記派生起動トランザクションから派生トランザクションの起動要求依頼が出力された際の処理の工程として、前記派生トランザクション管理部に対して、

前記起動要求依頼が出力された際に、当該派生トランザクションの起動処理とは別に、当該派生トランザクションの起動要求依頼に関連する派生起動情報を前記ファイルに記録し、当該派生起動情報の記録を、前記派生起動トランザクションの原子性に包む制御をすることで、派生トランザクションの起動を管理する工程と、

前記派生トランザクションの開始を待機せずに当該派生起動トランザクションをコミッ

10

20

30

40

50

トさせる工程と、

前記派生起動トランザクションがコミットした際に、当該派生起動トランザクションによる前記ロックを解除することで他のオンライントランザクションによる当該ロック解除されたデータ項目へのアクセスを許容しつつ、当該起動要求依頼された派生トランザクションの処理を開始させることで、前記派生トランザクションの開始を管理する工程とを実現させることを特徴とするトランザクション制御プログラム。

【請求項7】

端末からネットワークを介して入力される業務処理要求を受信する受信手段と、この受信手段にて受信した業務処理要求を処理する演算手段と、この演算手段の処理結果を記録するファイルとを備え、

10

前記ファイルが、前記業務処理要求に関連した各種マスタDBと、所定の派生トランザクションの起動要求依頼に関連する派生起動情報を派生管理通番毎に記録する派生管理DBと、この派生管理DBのキーとなる派生管理通番毎に派生管理DBが使用中であるか否かを指示するインデックス情報を記録する派生インデックスDBとを備え、

前記演算手段が、

所定のオンライントランザクション及び前記派生トランザクションを並列に実行し、この各トランザクションの実行に際して、それぞれ、前記ファイルの前記各DBのデータ項目をロックし、処理し、前記ファイルの各DBに当該処理結果を記録し、当該記録後に前記ロックを解除するトランザクション実行部と、

前記端末から入力される業務処理要求をオンライントランザクションとして前記トランザクション実行部に処理させ、当該オンライントランザクションの処理結果を前記端末への応答として出力する制御をするオンライン応答部と、

20

前記派生トランザクションを前記トランザクション実行部に処理させ、各トランザクションのコミットを管理するトランザクション管理部とを備え、

前記各トランザクションの実行中に、新たなトランザクションの起動要求依頼を出力するトランザクションを派生起動トランザクションとし、当該起動要求依頼されたトランザクションを派生トランザクションとして、

前記派生トランザクションは、前記オンライントランザクションの前記処理結果に付随して生じ、かつ、前記オンライントランザクションの応答には必要ない付随業務の全部又は一部を処理するトランザクションであって、

30

前記トランザクション管理部が、前記派生起動トランザクションから新たな派生トランザクションの起動要求依頼が出力された際には、前記派生インデックスDBのインデックス情報を特定すると共に当該特定したインデックス情報に基づいて前記派生管理DBの使用可能な派生管理通番を特定する派生管理通番特定機能と、

この派生管理通番特定機能によって特定された派生管理通番のセグメントに、当該派生トランザクションの起動処理とは別に、当該起動要求された派生トランザクションの派生起動情報を記録し、当該派生起動情報の記録を前記派生起動トランザクションの原子性に包む制御をし、当該派生起動情報の記録後に、当該派生トランザクションを起動させる派生起動情報格納制御機能と、

当該派生起動トランザクションの各処理が正常終了した際に、当該起動要求依頼された派生トランザクションの開始を待機せずに、当該派生起動トランザクションをコミットさせる派生起動トランザクションコミット処理機能と、

40

前記派生起動トランザクションがコミットした際に、当該派生起動トランザクションによる前記ロックを解除することで他のオンライントランザクションによる当該ロック解除されたデータ項目へのアクセスを許容しつつ、前記起動された派生トランザクションが当該派生管理通番の前記派生起動情報をロック可能な状態での正常な読み出しに成功した後に、当該派生トランザクションの各処理を開始させる直列性制御機能と、

前記派生トランザクションの各処理が正常終了した際に、当該派生管理通番の前記派生インデックスDB及び前記派生管理DBを使用可に更新させると共に当該派生トランザクションをコミットさせる派生トランザクションコミット処理機能と、

50

前記派生トランザクションがアボートされた際に、前記派生管理DBに格納した派生起動情報に基づいた再起動を制御する派生再起動制御機能とを備えたことを特徴とするトランザクション制御システム。

【請求項8】

前記派生起動トランザクションが、予め定められたJOB番号群の中のいずれかのJOB番号で実行され、

前記派生インデックスDBが、前記派生起動トランザクションのJOB番号をキー値とするインデックス領域番号毎に前記インデックス情報を有することを特徴とした請求項7記載のトランザクション制御システム。

【請求項9】

前記派生インデックスDBが、前記インデックス情報を前記インデックス領域番号毎のビットマップとして保有し、

前記派生管理通番特定機能が、前記インデックス領域番号と前記ビットマップのビット位置とに基づいて前記派生管理DBの派生管理通番を特定する機能を備えたことを特徴とする請求項8記載のトランザクション制御システム。

【請求項10】

前記派生インデックスDBが、前記インデックス領域番号毎のビットマップを派生インデックスビットとして有すると共に、当該派生インデックスビットのON又はOFFに応じて当該ビットのビット位置に対応する派生管理DBの使用状態を記録し、

前記派生管理DBが、前記派生管理通番毎に当該派生管理通番の使用状態を使用中又は使用可として記録する使用中フラグを有し、

前記派生管理通番特定機能が、OFFとなっている派生インデックスビット及び当該使用可となっている派生管理通番を特定したときに当該派生インデックスビットについて使用中を示すONへと更新し、

前記派生起動情報格納制御機能が、前記派生起動情報を格納したときに当該派生管理通番の使用中フラグを「使用中」へと更新し、

前記直列性制御機能は、前記派生起動トランザクションから渡された派生管理通番の使用可フラグが使用可であった場合には、当該派生トランザクションがアボートしたとして、当該起動要求されている派生トランザクションを開始せずにアボートさせ、

派生トランザクションコミット制御機能は、当該派生トランザクションの完了時又は完了後に前記派生インデックスビット及び前記使用中フラグについて使用可へと更新することを特徴とする請求項9記載のトランザクション制御システム。

【請求項11】

前記演算手段が、前記オンライントランザクション及び前記派生トランザクションを予め定められた数の前記JOB番号を単位に並列して処理し、この演算手段の前記JOB番号へとスケジュールされる各トランザクションの要求を一時的に格納する待ち行列を備え、この待ち行列が、前記オンライントランザクションの要求を一時的に格納する一般待ち行列と、前記派生トランザクションの要求を一時的に格納する派生待ち行列とを有し、

前記演算手段が、前記一般待ち行列及び派生待ち行列で滞留しているトランザクション数を監視する滞留監視部と、この滞留監視部によって監視される各待ち行列でのトランザクションの滞留数に応じて前記派生起動トランザクション又は前記派生トランザクションの起動を制御する総量制御部とを備え、

前記滞留監視部は、前記一般待ち行列又は派生待ち行列の滞留数が予め定められたレベル以上である場合には派生中止ビットをONとし、

前記総量制御部は、前記派生中止ビットがONである場合には派生起動要求があった時には当該派生起動情報を前記派生管理DBに記録後、派生トランザクションの開始を行わない派生起動中止制御機能とを備えたことを特徴とする請求項8記載のトランザクション制御システム。

【請求項12】

前記派生起動情報格納制御機能が、前記派生管理通番特定機能によって特定された派生

10

20

30

40

50

管理通番のセグメントに前記派生起動トランザクションによる派生トランザクションの起動要求依頼に関する情報及び起動時刻を派生起動情報として格納し、

前記演算手段は、前記派生インデックスDB及び前記派生管理DBを参照して、前記インデックス情報が使用中を示し、かつ、前記起動時刻を参照して、派生管理DBに派生起動情報を格納した後予め定められた時間以上経過している派生管理通番を派生再起動要と判定する派生再起動要否監視部を備えたことを特徴とする請求項7又は8記載のトランザクション制御システム。

【請求項13】

前記演算手段は、前記派生再起動要否監視部によって再起動要と判定された派生管理通番を再起動する場合には、当該派生管理通番によって特定される派生起動情報に基づいて派生トランザクションを起動すると共に当該派生管理通番へ派生起動情報を格納した時刻を更新する派生再起動制御部とを備えたことを特徴とする請求項7又は12記載のトランザクション制御システム。

【請求項14】

端末からネットワークを介して入力される業務処理要求を受信する受信手段と、この受信手段にて受信した業務処理要求を処理する演算手段と、この演算手段の処理結果を記録するファイルとを備え、

前記ファイルが、前記業務処理要求に関連した各種マスタDBと、所定の派生トランザクションの起動要求依頼に関連する派生起動情報を派生管理通番毎に記録する派生管理DBと、この派生管理DBのキーとなる派生管理通番毎に派生管理DBが使用中であるか否かを指示するインデックス情報を記録する派生インデックスDBとを備え、

前記演算手段が、

所定のオンライントランザクション及び前記派生トランザクションを並列に実行し、この各トランザクションの実行に際して、それぞれ、前記ファイルの前記各DBのデータ項目をロックし、処理し、前記ファイルの各DBに当該処理結果を記録し、当該記録後に前記ロックを解除するトランザクション実行部と、

前記端末から入力される業務処理要求をオンライントランザクションとして前記トランザクション実行部に処理させ、当該オンライントランザクションの処理結果を前記端末への応答として出力する制御をするオンライン応答部と、

前記派生トランザクションを前記トランザクション実行部に処理させ、各トランザクションのコミットを管理するトランザクション管理部とを備えたトランザクション制御システムの前記トランザクション管理部を使用して、

前記各トランザクションを管理するトランザクション制御方法であって、

前記各トランザクションの実行中に、新たなトランザクションの起動要求依頼を出力するトランザクションを派生起動トランザクションとし、当該起動要求依頼されたトランザクションを派生トランザクションとして、

前記派生トランザクションは、前記オンライントランザクションの前記処理結果に付随して生じ、かつ、前記オンライントランザクションの応答には必要ない付随業務の全部又は一部を処理するトランザクションであって、

前記派生起動トランザクションから新たな派生トランザクションの起動要求依頼が出力された際には、前記派生インデックスDBのインデックス情報を特定すると共に当該特定したインデックス情報に基づいて前記派生管理DBの使用可となっている派生管理通番を特定する派生管理通番特定工程と、

この派生管理通番特定工程にて特定された派生管理通番のセグメントに、当該派生トランザクションの起動処理とは別に、当該起動要求依頼された派生起動トランザクションの派生起動情報を記録し、当該派生起動情報の記録を前記派生起動トランザクションの原子性に包む制御をし、当該派生起動情報の記録後に、当該派生トランザクションを起動させる派生起動情報格納制御工程と、

当該派生起動トランザクションの各処理が正常終了した際に、当該起動要求依頼された派生トランザクションの開始を待機せずに、当該派生起動トランザクションをコミットさ

10

20

30

40

50

せる派生起動トランザクションコミット処理工程と、

前記派生起動トランザクションがコミットした際に、当該派生起動トランザクションによる前記ロックを解除することで他のオンライントランザクションによる当該ロック解除されたデータ項目へのアクセスを許容しつつ、前記派生トランザクションが当該派生管理通番の前記派生起動情報をロック可能な状態での正常な読み出しに成功した後に、当該派生トランザクションの各処理を開始させる直列性制御工程と、

前記派生トランザクションの各処理が正常終了した際に、当該派生管理通番の前記派生インデックスDB及び前記派生管理DBを使用可に更新させると共に当該派生トランザクションをコミットさせる派生トランザクションコミット処理工程と、

前記派生トランザクションがアボートされた際に、前記派生管理DBに格納した派生起動情報に基づいた再起動を制御する派生再起動制御工程とを備えたことを特徴とするトランザクション制御方法。

10

【請求項15】

端末からネットワークを介して入力される業務処理要求を受信する受信手段と、この受信手段にて受信した業務処理要求を処理する演算手段と、この演算手段の処理結果を記録するファイルとを備え、

前記ファイルが、前記業務処理要求に関連した各種マスタDBと、所定の派生トランザクションの起動要求依頼に関連する派生起動情報を派生管理通番毎に記録する派生管理DBと、この派生管理DBのキーとなる派生管理通番毎に派生管理DBが使用中であるか否かを指示するインデックス情報を記録する派生インデックスDBとを備え、

20

前記演算手段が、

所定のオンライントランザクション及び前記派生トランザクションを並列に実行し、この各トランザクションの実行に際して、それぞれ、前記ファイルの前記各DBのデータ項目をロックし、処理し、前記ファイルの各DBに当該処理結果を記録し、当該記録後に前記ロックを解除するトランザクション実行部と、

前記端末から入力される業務処理要求をオンライントランザクションとして前記トランザクション実行部に処理させ、当該オンライントランザクションの処理結果を前記端末への応答として出力する制御をするオンライン応答部と、

前記派生トランザクションを前記トランザクション実行部に処理させ、各トランザクションのコミットを管理するトランザクション管理部とを備えたトランザクション制御システムの前記トランザクション管理部に所定の工程を実現させるためのトランザクション制御プログラムであって、

30

前記各トランザクションの実行中に、新たなトランザクションの起動要求依頼を出力するトランザクションを派生起動トランザクションとし、当該起動要求依頼されたトランザクションを派生トランザクションとして、

前記派生トランザクションは、前記オンライントランザクションの前記処理結果に付随して生じ、かつ、前記オンライントランザクションの応答には必要ない付随業務の全部又は一部を処理するトランザクションであって、

前記派生起動トランザクションから派生トランザクションの起動要求依頼が出力された際の処理の工程として、前記派生トランザクション管理部に対して、

40

前記オンライントランザクション又は派生トランザクションである派生起動トランザクションから新たな派生トランザクションの起動要求依頼が出力された際には、前記派生インデックスDBのインデックス情報を特定すると共に当該特定したインデックス情報に基づいて前記派生管理DBの使用可となっている派生管理通番を特定する派生管理通番特定工程と、

この派生管理通番特定工程にて特定された派生管理通番のセグメントに、当該派生トランザクションの起動処理とは別に、当該起動要求依頼された派生起動トランザクションの派生起動情報を記録し、当該派生起動情報の記録を前記派生起動トランザクションの原子性に包む制御をし、当該派生起動情報の記録後に、当該派生トランザクションを起動させる派生起動情報格納制御工程と、

50

当該派生起動トランザクションの各処理が正常終了した際に、当該起動要求依頼された派生トランザクションの開始を待機せずに、当該派生起動トランザクションをコミットさせる派生起動トランザクションコミット処理工程と、

前記派生起動トランザクションがコミットした際に、当該派生起動トランザクションによる前記ロックを解除することで他のオンライントランザクションによる当該ロック解除されたデータ項目へのアクセスを許容しつつ、前記派生トランザクションが当該派生管理通番の前記派生起動情報をロック可能な状態での正常な読み出しに成功した後に、当該派生トランザクションの各処理を開始させる直列性制御工程と、

前記派生トランザクションの各処理が正常終了した際に、当該派生管理通番の前記派生インデックスDB及び前記派生管理DBを使用可に更新させると共に当該派生トランザクションをコミットさせる派生トランザクションコミット処理工程と、

前記派生トランザクションがアボートされた際に、前記派生管理DBに格納した派生起動情報に基づいた再起動を制御する派生再起動制御工程とを実現させることを特徴とするトランザクション制御プログラム。

【請求項16】

端末からネットワークを介して入力される業務処理要求を受信する受信手段と、この受信手段にて受信した業務処理要求に応じたオンライントランザクション及び所定の派生トランザクションを予め定められた数のJOB番号を単位に並列して処理する演算手段と、この演算手段の処理結果を記録するファイルとを備え、

前記ファイルが、

前記業務処理要求に関連した各種マスタDBと、

前記派生トランザクションの起動要求依頼に関連する派生起動情報を派生管理通番毎に記録すると共に当該派生管理通番が使用中であるか又は使用可であることを示す使用中フラグを有する派生管理DBと、

この派生管理DBでの派生管理通番を前記各トランザクションのJOB番号を単位とした複数の派生管理通番から使用可の派生管理通番を特定すると共に前記派生管理通番が使用中であるか否かをビットのON又はOFFにて示す派生インデックスビットを前記JOB番号毎に複数有する派生インデックスDBとを備え、

前記演算手段が、

前記オンライントランザクション及び派生トランザクションを構成するプログラムの実行を管理する業務メイン処理部と、

この業務メイン処理部によって実行が管理されるプログラムを前記JOB番号毎のトランザクションの一部として並列に実行し、この各トランザクションの実行に際して、それぞれ、前記ファイルの前記各DBのデータ項目をロックし、処理し、前記ファイルの各DBに当該処理結果を記録し、当該記録後に前記ロックを解除する業務処理部とを備え、

前記各トランザクションの実行中に、新たなトランザクションの起動要求依頼を出力するトランザクションを派生起動トランザクションとし、当該起動要求依頼されたトランザクションを派生トランザクションとして、

前記派生トランザクションは、前記オンライントランザクションの前記処理結果に付随して生じ、かつ、前記オンライントランザクションの応答には必要ない付随業務の全部又は一部を処理するトランザクションであって、

前記業務処理部は、前記端末から入力される業務処理要求をオンライントランザクションとして処理し、当該オンライントランザクションの処理結果を前記端末への応答として出力し、

前記業務メイン処理部は、前記派生トランザクションの起動を管理する派生要求共通処理と、前記派生トランザクションの開始及び完了を管理する派生特性管理共通処理と、前記派生トランザクションの完了の一部を管理する派生DB制御処理とを実行し、

前記派生要求共通処理は、前記派生起動トランザクションの原子性に包まれる処理として、前記プログラムの実行中に新たな派生トランザクションの派生起動要求依頼が出力された際に、当該派生起動トランザクションのJOB番号に応じて前記派生インデックスD

10

20

30

40

50

Bを参照し、前記派生管理DBの派生管理通番を特定し、当該派生トランザクションの起動処理とは別に、当該派生管理通番のセグメントに当該派生起動情報を格納し、当該派生管理通番に対応する前記インデックスビット及び使用中フラグをそれぞれ使用中の状態に更新して、当該派生トランザクションを起動させることで、当該派生トランザクションの起動を管理し、業務メイン処理部は、当該起動後で当該派生トランザクションの開始を待機せずに当該派生起動トランザクションをコミットさせることで当該派生起動トランザクションのロックを解除し、当該ロック解除したデータ項目への他のトランザクションによるアクセスを許容し、

前記派生特性管理共通処理は、当該起動された派生トランザクションの原子性に包まれる処理として、当該派生トランザクションが起動され、当該派生トランザクションに与えられた派生管理通番をキー値とする派生管理DBの読み込みに成功した際に、当該派生管理通番の使用フラグが使用中である場合には、当該派生トランザクションの処理を実行させ、一方、当該使用中フラグが使用可である場合には、当該派生トランザクションをアポートすることで、当該派生トランザクションの開始を管理し、前記業務処理部は、当該派生トランザクションの処理を実行し、

10

当該派生特性管理処理は、当該派生トランザクションの原子性に包まれる処理として、当該派生トランザクションの処理が終了した後に、前記派生管理DBの使用フラグを使用可に更新して、前記派生DB制御処理を当該起動要求依頼された派生トランザクションとは別の派生トランザクションとして起動させ、この起動後に、当該派生トランザクションをコミットし、

20

前記派生DB制御処理は、前記派生DB制御処理がスケジュールされた場合に、当該派生管理通番に対応する前記派生インデックスDBの前記派生インデックスビットを使用可を示すOFFへ更新することを特徴とするトランザクション制御システム。

【請求項17】

前記派生インデックスDBが、前記派生インデックスビットを前記JOB番号を単位としたビットマップとして保有すると共に当該ビットマップのビット位置で前記派生管理通番を特定し、

前記派生DB制御処理機能が、前記派生管理通番から前記ビットマップ及びビット位置を算出することを特徴とする請求項16記載のトランザクション制御システム。

【請求項18】

30

端末からネットワークを介して入力される業務処理要求を受信する受信手段と、この受信手段にて受信した業務処理要求に応じたオンライントランザクション及び所定の派生トランザクションを予め定められた数のJOB番号を単位に並列して処理する演算手段と、この演算手段の処理結果を記録するファイルとを備え、

前記ファイルが、

前記業務処理要求に関連した各種マスタDBと、

前記派生トランザクションの起動要求依頼に関連する派生起動情報を派生管理通番毎に記録すると共に当該派生管理通番が使用中であるか又は使用可であるかを示す使用中フラグを有する派生管理DBと、

この派生管理DBでの派生管理通番を前記各トランザクションのJOB番号を単位とした複数の派生管理通番から使用可の派生管理通番を特定すると共に前記派生管理通番が使用中であるか否かを示す派生インデックスビットを前記JOB番号毎に複数有する派生インデックスDBとを備え、

40

前記演算手段が、

前記オンライントランザクション及び派生トランザクションを構成するプログラムの実行を管理する業務メイン処理部と、

この業務メイン処理部によって実行が管理されるプログラムを前記JOB番号毎のトランザクションの一部として並列に実行し、この各トランザクションの実行に際して、それぞれ、前記ファイルの前記各DBのデータ項目をロックし、処理し、前記ファイルの各DBに当該処理結果を記録し、当該記録後に前記ロックを解除する業務処理部とを備えたオ

50

ンラインシステムの前記演算手段を使用してトランザクションを管理するトランザクション制御方法であって、

前記各トランザクションの実行中に、新たなトランザクションの起動要求依頼を出力するトランザクションを派生起動トランザクションとし、当該起動要求依頼されたトランザクションを派生トランザクションとして、

前記派生トランザクションは、前記オンライントランザクションの前記処理結果に付随して生じ、かつ、前記オンライントランザクションの応答には必要ない付随業務の全部又は一部を処理するトランザクションであって、

派生起動トランザクションの起動を管理すると共に当該派生起動トランザクションによって起動要求依頼される派生トランザクションの開始を待機せずに当該派生起動トランザクションをコミットさせることで当該派生起動トランザクションのロックを解除し、当該ロック解除したデータ項目への他のトランザクションによるアクセスを許容する派生起動トランザクション管理工程と、

前記派生起動トランザクションによって起動要求依頼された派生トランザクションの開始及び管理を管理する派生トランザクション管理工程と、

この派生トランザクション管理工程にて派生トランザクションとして起動される派生DB管理工程とを備え、

前記派生起動トランザクション管理工程は、当該派生起動トランザクションの原子性に包まれる処理として、当該派生起動トランザクションのJOB番号に応じて前記派生インデックスDBを参照し、前記派生管理DBの派生管理通番を特定し、当該派生トランザクションの起動処理とは別に、当該派生管理通番のセグメントに派生起動情報を格納し、当該使用した派生管理通番に対応する前記インデックスビット及び使用中フラグをそれぞれ使用中の状態に更新して、当該派生トランザクションを起動させることで、当該派生トランザクションの起動を管理する派生要求共通処理工程を備え、

前記派生トランザクション管理工程は、当該起動された派生トランザクションの原子性に包まれる処理として、当該派生トランザクションが起動されスケジュールされ、当該派生トランザクションに与えられた派生管理通番をキー値とする派生管理DBの読み込みに成功した際に、当該派生管理通番の使用フラグが使用中である場合には当該派生トランザクションの処理を実行させ、一方、当該使用中フラグが使用可である場合には、当該派生トランザクションをアポートすることで、当該派生トランザクションの開始を管理し、

当該派生トランザクションの処理が終了した後に、前記派生管理DBの使用フラグを使用可に更新して、さらに所定の派生DB制御処理を派生トランザクションとして起動させ、この起動後に、当該派生トランザクションをコミットする派生特性管理共通処理工程を備え、

前記派生DB管理工程は、前記派生DB制御処理がスケジュールされた場合に、当該派生管理通番に対応する前記派生インデックスDBの前記派生インデックスビットを使用可を示すOFFに更新する派生DB制御処理工程を備えたことを特徴とするトランザクション制御方法。

【請求項19】

端末からネットワークを介して入力される業務処理要求を受信する受信手段と、この受信手段にて受信した業務処理要求に応じたオンライントランザクション及び所定の派生トランザクションを予め定められた数のJOB番号を単位に並列して処理する演算手段と、この演算手段の処理結果を記録するファイルとを備え、

前記ファイルが、

前記業務処理要求に関連した各種マスタDBと、

前記派生トランザクションの起動要求依頼に関連する派生起動情報を派生管理通番毎に記録すると共に当該派生管理通番が使用中であるか又は使用可であるか示す使用中フラグを有する派生管理DBと、

この派生管理DBでの派生管理通番を前記各トランザクションのJOB番号を単位とした複数の派生管理通番から使用可の派生管理通番を特定すると共に前記派生管理通番が使

10

20

30

40

50

用中であるか否かを示す派生インデックスビットを前記 J O B 番号毎に複数有する派生インデックス D B とを備え、

前記演算手段が、

前記オンライントランザクション及び派生トランザクションを構成するプログラムの実行を管理する業務メイン処理部と、

この業務メイン処理部によって実行が管理されるプログラムを前記 J O B 番号毎のトランザクションの一部として並列に実行し、この各トランザクションの実行に際して、それぞれ、前記ファイルの前記各 D B のデータ項目をロックし、処理し、前記ファイルの各 D B に当該処理結果を記録し、当該記録後に前記ロックを解除する業務処理部とを備え、

前記演算手段に所定の工程を実現させるためのトランザクション制御プログラムであって、

10

前記各トランザクションの実行中に、新たなトランザクションの起動要求依頼を出力するトランザクションを派生起動トランザクションとし、当該起動要求依頼されたトランザクションを派生トランザクションとして、

前記派生トランザクションは、前記オンライントランザクションの前記処理結果に付随して生じ、かつ、前記オンライントランザクションの応答には必要ない付随業務の全部又は一部を処理するトランザクションであって、

派生起動トランザクションの起動を管理すると共に当該派生起動トランザクションによって起動要求依頼される派生トランザクションの開始を待機せずに当該派生起動トランザクションをコミットさせることで当該派生起動トランザクションのロックを解除し、当該ロック解除したデータ項目への他のトランザクションによるアクセスを許容する派生起動トランザクション管理工程と、

20

前記派生起動トランザクションによって起動要求依頼された派生トランザクションの開始及び管理を管理する派生トランザクション管理工程と、

この派生トランザクション管理工程にて起動要求依頼された派生トランザクションとは別の派生トランザクションとして起動される派生 D B 管理工程とを前記演算手段に実現させ、

前記派生起動トランザクション管理工程の一部として、当該派生起動トランザクションの原子性に包まれる処理として、当該派生起動トランザクションの J O B 番号に応じて前記派生インデックス D B を参照し、前記派生管理 D B の派生管理通番を特定し、当該派生トランザクションの起動処理とは別に、当該派生管理通番のセグメントに派生起動情報を格納し、当該使用した派生管理通番に対応する前記インデックスビット及び使用中フラグをそれぞれ使用中の状態に更新して、当該派生トランザクションを起動させることで、当該派生トランザクションの起動を管理する派生要求共通処理工程を実現させ、

30

前記派生トランザクション管理工程の一部として、当該起動された派生トランザクションの原子性に包まれる処理として、当該派生トランザクションが起動されスケジュールされ、当該派生トランザクションに与えられた派生管理通番をキー値とする派生管理 D B の読み込みに成功した際に、当該派生管理通番の使用フラグが使用中である場合には当該派生トランザクションの処理を実行させ、一方、当該使用中フラグが使用可である場合には、当該派生トランザクションをアボートすることで、当該派生トランザクションの開始を管理し、

40

当該派生トランザクションの処理が終了した後に、前記派生管理 D B の使用中フラグを使用可に更新して、さらに所定の派生 D B 制御処理を派生トランザクションとして起動させ、この起動後に、当該派生トランザクションをコミットする派生特性管理共通処理工程を実現させ、

前記派生 D B 管理工程の一部として、前記派生 D B 制御処理がスケジュールされた場合に、当該派生管理通番に対応する前記派生インデックス D B の前記派生インデックスビットを使用可を示す O F F に更新する派生 D B 制御処理工程を実現させることを特徴とするトランザクション制御プログラム。

【発明の詳細な説明】

50

【0001】

【発明の属する技術分野】

本発明は、トランザクション制御システム及びその方法に関する。

【0002】

【従来の技術】

従来より、大規模なデータベースに対して正確で一貫性があり障害に対して確実な復旧を可能とする処理を実現する手法として、トランザクション処理が利用されている。トランザクションは、データベース（DB）に対する処理の単位として用いられる。トランザクションは、DBに一又は複数の更新を行う。このとき、トランザクションは、その性質によって、全体として成功するか又は全体として失敗するかのいずれかであることが保証される。DBへの全ての更新に成功した場合に、トランザクションは同期点(syncpoint)に達する。すなわち、トランザクションはコミット(commit)する。一般的なトランザクション管理システムでは、同期点に達したトランザクションによる更新はログに記録され、このDBへの更新は障害に対して耐久性（Durability）を有する。

10

【0003】

一方、トランザクションに含まれる処理のうち一つでも成立しなかった場合には、そのトランザクションはアボート(abort)される。トランザクションがアボートされると、そのトランザクションが行った全ての更新は取り消される。すなわち、トランザクションがアボートされると、トランザクション管理システムは、その時までに行われた全ての更新をロールバックする。このため、トランザクションは、一又は複数のプログラムを実行し、複数の更新を行っていても、全ての更新を正常に実行しコミットするか、又は、ロールバックによりなにも処理しないかのいずれかである。トランザクションは、アプリケーションにとって処理の最小単位となることから、これをトランザクションの原子性（アトミック性, Atomicity）という。

20

【0004】

トランザクションは原子性を有するから、プログラムは、DBの更新に際して一貫性・整合性（Consistency）を保つことができる。適用業務プログラム（アプリケーション、APP L）のプログラムは、他のトランザクションによって参照されるべきではない中間的なデータの算出処理と、この中間的なデータを正常なデータへと変換又は関連させる処理とを1つのトランザクションとして包み込むことで、DBの整合性、データの一貫性を保つことができる。正常なデータへ変換されたことを条件としてコミットするため、中間的なデータの状態でDBに格納され、他のトランザクションによってこの中間的なデータが参照されることはない。一方、正常なデータに変換されない場合には、トランザクションがアボートされるため、中間的なデータの算出及び更新もロールバックにより取り消される。

30

【0005】

例えば、口座振替（資金移動）は、ある口座Aからの数値の減算（出金）と、他の口座Bへの数値の加算（入金）とからなる処理である。中間的なデータは、例えばある口座Aからの数値の減算である。この数値の減算処理と加算処理とを1つのトランザクションとして包み込むことで、口座振替に成功するか、または失敗するかのいずれかであることが保証される。この場合、数値の減算のみ又は加算のみが行われることはない。すなわち、出金と入金とを1つのトランザクションとして処理すると、出金だけが行われ、入金なされないという事態は生じない。

40

【0006】

また、トランザクション間での前後関係を明確に保ち、トランザクション処理中のデータを他のトランザクションが参照できないようにすることで、複数のトランザクションを並行して処理することができる。このように、並列処理や分散処理を行うためには、トランザクション間で扱うデータを相互に隔離（Isolation）しなければならない。このデータの隔離（分離）にはロックが有効である。トランザクションが更新の可能性を持ってデータにアクセスした場合には、そのトランザクションは、そのデータ項目をロックし、必要

50

な更新を行った後に、そのロックを解除する。トランザクションは、ロックされているデータにアクセスしようとする、そのロックが他のトランザクションによって解除されるまで待機する。これにより、トランザクションの直列性を確保し、トランザクションに隔離性を与え、トランザクションの並列処理・分散処理が可能となる。

【0007】

ロックを用いることで、トランザクションの直列性を維持しながら並列処理を行うことができるが、ロックの粒度（競合の単位）が大きい場合には、ロックの解除を待機する頻度が増大し、そして待機時間が長くなる傾向となる。また、ロックを使用する例では、各トランザクションのDBの読み込み順序によっては、デッドロックが発生する。例えば、DBxとDByがあるとき、トランザクションT1がDBxを読み、続いて、DByを読み出そうとするが、このDByはトランザクションT2によってロックされているとする。この場合、トランザクションT1は、DBxをロックしたまま、DByのロックが解除されるまで待機する。トランザクションT2がDByを読み出した後、DBxを読み出そうとしていると、デッドロックが生じる。このように、トランザクションT1がT2の処理完了を待機し、トランザクションT2がT1の処理完了を待機する場合に、デッドロックが生じる。デッドロックは、すべてのトランザクションでDBの読み込み順序を同一とすることなどで回避することができる。

10

【0008】

さて、トランザクションは原子性、一貫性、隔離性、耐久性のACID特性を有する。この特性によるトランザクション処理を用いると、厳密で強固なシステムを構築することができる。従って、トランザクション処理システムは、金融機関での勘定系システムや、航空機等の運行や発券等の管理システムや、各種産業の製造工程の管理システムや、通信の制御システムなど、幅広い分野で利用されている。このようなシステムは、オンライン処理とバッチ処理とを実行する。例えば、従来の勘定系システムでは、日中はATM等の端末や行内の営業店端末からの業務処理要求をオンライントランザクションとして処理し、オンラインを停止した夜間から早朝にかけて口座振替などをバッチで処理していた。しかしながら、今日では、バッチ処理を最大限少なくし、24時間連続運用を行うことが求められている。

20

【0009】

ACID特性を有するトランザクションで実行できる業務は、比較的単純な処理に限られていた。これは、1つのトランザクションで複雑な処理の実行を図ると、ロックが多数生じ、オンライン処理の応答性を悪化させてしまうことによる。また、1つのトランザクションでは処理が不可能なデータ量などがある。従来のトランザクション処理では、一般的に、オンラインでの業務処理要求を受信して即座に実行し、直ちにコミットするようにプログラムされている。

30

【0010】

トランザクション間の直列性を管理するためのロッキングを行わなければ、応答性は向上するが、一方、DBの一貫性を厳格に管理することができなくなる。優れた設計には英知に従った妥協が存在するが、このオンラインの応答性の確保と、より複雑な処理の実現との両立に関しては、近年、種々模索されている。この対立は、トランザクションの並列処理と、厳格な原子性とをどのように調和させるのかという点と関連する。並列処理を有効に機能させることができれば、オンライントランザクションの応答性は良好に確保される。

40

【0011】

この課題に対するいくつかの手法は、例えば、“トランザクション処理システム入門”フィリップ・A・バーンスタイン、エリック・ニューカマー著、日経BP社刊(Principles of Transaction Processing, Philip Bernstein, Eric Newcomer)や、“CORBA, Encinaによるエンタープライズ・トランザクション”イアン・ゴートン著、ピアソン・エデュケーション刊(Enterprise Transaction Processing Systems: Putting the CORBA OTS, Encina++ and OrbixOTM to Work, Ian Gorton)に記載されている。

50

【0012】

ACID特性による単純なトランザクション管理を越える手法として、セーブポイント (save point) がある。これは、トランザクションの実行途中ですべてのリソースの状態を保存するものである。これにより、障害が発生した場合に、トランザクションの途中からの再開が可能となる。永続セーブポイントでは、トランザクションの原子性を抜きにして耐久性を与える。これにより、処理に長時間を要するトランザクションについて、全体をアボートするのではなく、途中からの再開を可能とする。セーブポイントを用いた手法は、例えば、特開2000-10810号公報に開示されている。一般的に、セーブポイントを利用して、そのセーブポイント時でトランザクションが取り扱っているデータは他のトランザクションから隔離されるため、ロックによる競合を減らすことはできない。

10

【0013】

複数のトランザクションを関連させる手法として、入れ子トランザクション (例えば、Transarc社のEncina TPモニタ) や、連鎖トランザクション (例えば、IBM社のMQシリーズ) がある。入れ子トランザクションは、トランザクションに親子関係を与えることで、トランザクション実行中に他のトランザクションを呼び出すことを可能とする。最初に起動されるトランザクションは親トランザクションとなる。この親トランザクションが呼び出したトランザクションは、副トランザクションとなる。副トランザクションは、親トランザクションに対してACID特性を有するが、他のトランザクションに対しては耐久性を有さない。副トランザクションが最終的にコミットするのは、親トランザクションがコミットした場合である。副トランザクションは、同じ親を持つ他の副トランザクションに対して原子的であり、また、他のトランザクションや同じ親を持つ副トランザクションから隔離される。

20

【0014】

この入れ子トランザクションモデルを用いると、振替処理という親トランザクションが、出金処理という副トランザクションと、入金処理という副トランザクションとを起動することができる。出金処理という副トランザクションがコミットすると、その結果は入金処理から参照することができる。一方、出金処理である副トランザクションのコミットによって、そのロックの管理者は親トランザクションに移る。このため、親トランザクションがコミットするまでは、その副トランザクション以外の他のトランザクションは出金処理の結果を参照することができない。従って、入金処理が何らかの理由でアボートされた場合には、親トランザクションがアボートされ、出金処理もロールバックすることができる。この入れ子トランザクションモデルは、オブジェクト指向プログラミングとの相性が良いとされている。

30

【0015】

入れ子トランザクションモデルを用いると、複雑なデータ構造や処理をクラス階層で定義し、オブジェクトのメッセージと振る舞いを定義することでシステムの開発をすることができるが、副トランザクションは実際には親トランザクションがコミットするまで耐久性が与えられないため、ロックの増大が生じる可能性がある。すなわち、入れ子トランザクションモデルでは、副トランザクションを含んだ大きなトランザクションを実行しており、その大域のトランザクションは他のトランザクションに対してACID特性を有する。従って、入れ子トランザクションは、プログラムの再利用性を高める点で役に立つが、親トランザクションを単位としたロックの競合を減少させることはない。

40

【0016】

「連鎖(chain)」という用語は、種々の文脈で用いられている。特開平10-69418号公報では、連鎖トランザクションについて段落0060以下に記述されている。これによると、連鎖トランザクションモデルでは、メンバトランザクションT3が別のメンバトランザクションT4を起動する。メンバトランザクションT4の起動処理は、起動側のメンバトランザクションT3の原子性に包まれる。従って、起動側のメンバトランザクションT3がコミットする場合のみ別のメンバトランザクションT4が起動される。起動されたメンバトランザクションT4のアボートは、トランザクションT3に波及しない。

50

【0017】

連鎖トランザクション方式による再起動の扱いは、最近にコミットされたメンバトランザクションのコミットの状態を再び確立することによって実行される。この連鎖トランザクションモデルでは、トランザクションT3のコミットを早めることができる。すなわち、トランザクションT3は、トランザクションT4のコミットを待機せずにコミットすることができる。一方、連鎖トランザクションモデルでは、大域のトランザクションはACID特性を有さない。従って、連鎖トランザクションモデルでは原子性は厳密ではなくなる。

【0018】

オンライン応答性の確保を始めとする処理速度の向上を図る手法として、高速パス（Fast Path, FP）方式がある。例えば、特開平6-195250号公報には、IBM社のIMS上に高速パスシステムを実装したシステムにて、障害時の回復のためにアクティブなデータベースの複製をバックアップシステムに連続的に維持するための手法が開示されている。FPは、特に金融機関でのオンラインシステムにて応答性を向上するために用いられており、処理速度を向上させる目的で、リソースに関して種々の制限を行っている。例えば、FPでは、IMSデータベースで一般的に使用されている一時的な記憶領域であるルックアサイド・データベース・バッファを使用しておらず、そして、1トランザクションで使用するデータベースバッファ数も制限されている。さらに、IMSでは、業務処理要求であるメッセージはメッセージ・キューに格納されるが、FPではこのキューイング・システムをバイパスして、端末からより簡易な待ち行列（回復可能ではない）を介してトランザクションプログラムに送信することができる。これらにより、オンライン応答性を高めている。

【0019】

【発明が解決しようとする課題】

近年のオンラインシステムは、24時間365日の連続稼働が求められている。従って、バッチ処理をオンライン処理と並行して処理することが望まれる。すなわち、従来のバッチ処理をオンライン処理として再構築することが要請される。一般的には、バッチ処理はDBの静的状態を前提として実行されるため、オンライン応答によってDBの内容が変化している最中に、バッチ処理そのものを実行することはできない。静的な状態を前提としていた処理を、動的な状態で実行しようとする場合には、連鎖的アボートと同様の事態が生じ得る。

【0020】

連鎖的アボート(cascading abort)は、トランザクションの一貫性、直列性が失われた場合の弊害である。例えば、DBの値がxからyへと更新されたのち、その更新された値であるyに基づいて複数のトランザクションが実行されたとする。その後、このxからyへの更新が取り消されたとする。または、yからzへと更新されたとする。すると、yを前提として成立したトランザクションは全て取り消して再実行しなければならない。トランザクションのアボートが生じると、さらに次のトランザクションのアボートを連鎖的に生じさせる。このような事態は避けなければならない。しかし、バッチ処理を日中にオンラインと並行して処理しようとする場合には、このような連鎖的アボートと同様の事態が生じ得る。例えば、前日の最終残高に基づいて行う口座振替を、前日の日中に処理する場合には、前提としていた前日の最終残高が変化する可能性がある。口座振替処理後に口座残高に異動があると、一見、連鎖的アボートが生じる事態と同様の状態となり得る。

【0021】

このように、バッチ処理を日中のオンライン処理として実行しようすると、DBが静止していない状態で種々の処理を行う必要があるため、一貫性及び直列性を厳密に管理する必要が生じる。この課題は、口座振替のみならず、種々の応用分野でバッチ処理を日中のオンライン処理として実行しようとする開発での共通した課題である。

【0022】

連鎖トランザクションモデルは、端末へ応答する処理をコミットさせておき、その後に別

のトランザクションを実行することができるため、オンラインの応答性と複雑な処理の実現とを両立させることができる。しかしながら、バッチ処理のオンライン化という点では、処理要求が集中した場合の対策や、上述した連鎖的アボートとなりえる状況への対策が必要となる。また、連鎖トランザクションモデルは標準化されておらず、すべてのトランザクション管理システムで利用できるとは限らない。

【0023】

上記特開平10-69418号公報には、連鎖トランザクションモデルを階層化して用い、大域トランザクションにACID特性を付する旨開示されているが、メンバ・トランザクションのアボートと大域トランザクションの原子性との関係については開示されていない。

10

【0024】

オンラインの応答性を確保するには、高速パス(FP)などの採用が望ましい。しかしながら、FPを用いると、データベースのバッファ数に制限があり、また、キューからの回復ができないなどの一定の制限があるため、例えば、ACID特性を有する1トランザクションで処理可能な処理量が制限されてしまう。例えば、入れ子トランザクションと同様に1つのトランザクション内で複数の処理を連動させる連動トランザクションについて、FPを採用している場合には、1トランザクションで実行可能な連動回数に制限がある。

【0025】

【発明の目的】

本発明は、係る従来例の有する不都合を改善し、特に、オンライン処理の応答性を確保しつつ複雑な処理を実現することのできるトランザクション制御システム、方法及びプログラムを提供することを、その目的とする。

20

【0026】

【課題を解決するための手段】

本発明では、派生トランザクションモデルを用いる。派生トランザクションは、主取引に付随する業務を実行する。付随業務を処理するためのトランザクションモデルを用意することで、適用業務プログラムのプログラマは、業務処理(主業務)を、オンライン応答に無関係な付随業務を除いて通常のACID特性を有するトランザクションとしてプログラムを構築し、主業務に付随する付随業務を派生トランザクションとしてプログラムを構築することができる。そして、業務処理を実行するトランザクションは、その実行中に派生トランザクションを起動要求依頼した後、派生トランザクションの処理完了を待たずに端末に応答を返す。

30

【0027】

本発明では、端末からネットワークを介して入力される業務処理要求を受信する受信手段と、この受信手段にて受信した業務処理要求を処理する演算手段と、この演算手段の処理結果を記録するファイルとを備えている。

そして、演算手段は、トランザクション実行部と、オンライン応答部と、派生トランザクション管理部とを備えている。

トランザクション実行部は、所定のオンライントランザクション及び所定の派生トランザクションを並列に実行し、この各トランザクションの実行に際して、それぞれ、前記ファイルのデータ項目をロックし、処理し、前記ファイルに当該処理結果を記録し、当該記録後に前記ロックを解除する。

40

オンライン応答部は、前記端末から入力される業務処理要求をオンライントランザクションとして前記トランザクション実行部に処理させ、当該オンライントランザクションの処理結果を前記端末への応答として出力する制御をする。

派生トランザクション管理部は、前記派生トランザクションを前記トランザクション実行部に処理させ、各トランザクションのコミットを管理する。さらに、この派生トランザクション管理部は、前記各トランザクションの実行中に、新たな派生トランザクションの起動要求依頼が出力された際には、当該起動要求依頼を出力したトランザクションを派生起動トランザクションとして、次の制御を行う。すなわち、当該派生トランザクションの

50

起動要求依頼を、当該派生起動トランザクションの原子性に包む制御をする。続いて、当該起動要求依頼された前記派生トランザクションのコミットを待機せずに前記派生起動トランザクションをコミットさせる。

本発明によるトランザクション制御システムでは、前記派生トランザクションは、前記派生起動トランザクションの実行中に起動要求依頼され、前記オンライントランザクションの前記処理結果に付随して生じ、かつ、前記オンライントランザクションの応答には必要ない付随業務の全部又は一部を処理するトランザクションである。

そして、本発明では特に、前記派生トランザクション管理部は、前記派生トランザクションの起動と開始とを管理する。

すなわち、派生トランザクション管理部は、前記起動要求依頼が出力された際に、当該派生トランザクションの起動要求依頼に関連する派生起動情報を前記ファイルに記録し、当該派生起動情報の記録を、前記派生起動トランザクションの原子性に包む制御をすることで、派生トランザクションの起動を管理する派生起動情報格納制御機能を備える。

さらに、派生トランザクション管理部は、当該起動要求依頼された派生トランザクションの開始を待機せずに当該派生起動トランザクションをコミットさせる機能を備える。

しかも、派生トランザクション管理部は、前記派生起動トランザクションがコミットした際に、当該派生起動トランザクションによる前記ロックを解除することで他のオンライントランザクションによる当該ロック解除されたデータ項目へのアクセスを許容しつつ、当該起動要求依頼された派生トランザクションの処理を開始させることで、前記派生トランザクションの開始を管理する直列性管理機能を備えた、という構成を採っている。これにより前述した課題を解決しようとするものである。

【0028】

オンライン応答部は、業務処理要求をオンライントランザクションとして処理させ、その応答を端末へ出力する。オンライントランザクションは、顧客が使用する端末の操作によって生じた業務処理要求の処理でも良いし、内部的な従来バッチ処理で行っていた処理に相当する業務処理要求でも良い。「処理結果を端末への応答として出力する」というのは、オンライントランザクションの実行の一部として端末（又は擬似的な端末）への出力が要求されている場合に応答を端末に出力するのであって、全てのトランザクションが端末に応答を返すとは限らない。

【0029】

派生トランザクションは、オンライントランザクションによる業務処理と関連した付随業務を処理する。付随業務は、オンライントランザクションのオンライン応答を行う必要のない業務であり、一般的には主業務に付随する業務であるが、主業務側が付随的で付随業務が実際に必要な処理を行う場合もある。派生トランザクションは、オンライントランザクションの実行中に、プログラムによって起動される。また、派生トランザクションがさらに別の派生トランザクションを起動するようにしても良い。派生起動管理機能は、派生トランザクションのコミットを待機せずに前記派生起動トランザクションをコミットさせる。すなわち、付随業務の完了を待たずに業務処理を完了させる。派生トランザクションがアボートした場合であっても、派生起動トランザクションが取り消されることはない。

【0030】

このように、本発明では、連鎖トランザクションモデルを、主業務となる業務処理と、付随的な業務処理との区分けに応用している。これにより、オンライン応答に必要な処理をオンライントランザクションで処理し、オンライン応答に必要なない付随処理を派生トランザクションとして処理することで、オンライントランザクションを早期にコミットさせることができる。すなわち、オンラインの応答性を確保しつつ、付随的な業務のみを事後的に派生トランザクションとして処理することで、全体として複雑な処理を行うことができる。

【0031】

また、状況によっては、派生トランザクションは、派生起動トランザクションとなることができる。この場合、付随的な業務を1トランザクションで処理することなく、派生ト

10

20

30

40

50

ンザクションの繰り返しで実行する。この派生トランザクションを繰り返し起動・実行する例では、個々の派生トランザクションは個別にコミットするため、1つのトランザクションが処理する処理量は比較的少なく、そして、コミットによって一旦ロックが解放される。このため、複雑な付随業務を行いつつシステムに高負荷をもたらすことがない。

【0032】

派生起動トランザクションと、派生トランザクションとからなる処理（派生利用トランザクション）は、ACID特性を満たさない。すなわち、派生起動トランザクションがコミットしたが、派生トランザクションがアボートされる、という状況はあり得る。一方、派生起動トランザクションがコミットしない場合には、派生トランザクションを実行しない。すなわち、付随業務が必要となる業務処理自体が未成立であるのに、付随業務のみを成立させることはしない。業務処理が成立し、付随業務が未成立となった場合には、その付随業務を処理する派生トランザクションを再起動する。これらの仕組みを実現するために、派生トランザクション管理部が、前記派生トランザクションの起動要求に関連する派生起動情報を前記ファイルに格納した後に前記派生起動トランザクションをコミットさせる派生起動情報格納制御機能と、派生起動トランザクションがアボートされる場合には前記派生トランザクションの開始又はコミットをしない制御をする直列性管理機能とを備える。また、好ましい実施形態では、派生起動トランザクションがコミットした後に前記派生トランザクションがアボートした場合には前記派生起動情報に基づいて当該派生トランザクションを再起動させる派生再起動制御機能を備えると良い。

10

【0033】

派生起動情報格納制御機能は、派生トランザクションの起動要求があった場合には当該起動要求をしている派生起動トランザクションとこの派生起動トランザクションから派生トランザクションに渡されるデータ等を派生起動情報としてファイルに格納する。この派生起動情報は、正常な処理では利用せず、派生トランザクションがアボートし派生トランザクションのみを再起動する場合に利用される。すなわち、派生再起動制御機能は、派生起動トランザクションがコミットした後に前記派生トランザクションがアボートした場合には前記派生起動情報に基づいて当該派生トランザクションを再起動させる。

20

【0034】

派生起動情報格納制御機能等を有する本発明では、一般的な障害発生等によるトランザクションのアボート時の再起動処理とは別に、派生トランザクションの再起動を制御する機能を備えている。これは、派生起動トランザクションと派生トランザクションとを一体と考えた場合にはACID特性を有しないことから、派生トランザクションのアボート時の取り扱いをより厳格に行うものである。派生トランザクションは、派生起動情報に基づいて起動される。従って、この派生起動情報を耐久性を持った状態でファイルに格納しておく、と、当初起動時と完全に同一の状態で派生トランザクションを再起動することができる。また、キューイング・モデルやログによる再起動と比較して、再起動処理の対象の発見が容易で、さらに、派生トランザクションのアプリケーションのバグなどによるアボートであっても、アプリケーションプログラムの修正後の再実行が可能であるなど、取り扱いを柔軟に定めることができる。また、派生トランザクションは付随業務を処理するため、状況によっては再起動が不要である場合も想定できる。このような再起動の可否をオペレータが判断する場合であっても、派生起動情報に派生トランザクションの起動に関する種々の情報を含めることで、再起動の可否を判断する際にオペレータの端末に表示する等の処理が可能となる。

30

40

【0035】

派生トランザクションモデルでは、派生起動トランザクションがコミットし、派生トランザクションがアボートされる事態の発生を許容することで、オンラインの応答性を確保している。従って、派生トランザクションの再起動を厳密に管理できることがシステム運用の安定性に寄与する。

【0036】

また、本発明では、演算手段が、前記オンライン応答部及び派生トランザクション管理部

50

によって実行が管理されるトランザクションの総量に応じて前記派生トランザクションの起動を制限させる総量制御部を備えるようにすると良い。これにより、派生トランザクションの滞留によってオンライン応答が必要なオンライントランザクションを遅らせることがなくなる。「派生トランザクションの起動の制限」は、例えば、派生起動トランザクションとなる業務処理要求のうち、内部的な業務処理要求についてその入力を待機させたり、また、業務処理要求を実行しつつ、派生トランザクションの起動を中止したり、さらに、派生起動トランザクションの実行をエラーとする処理などを含む。起動が中止された派生トランザクションは、トランザクションの滞留が減少するなどシステムの負荷が軽減された段階で、再起動することができる。

【0037】

派生トランザクションモデルでは、即時のオンライン応答を行わない付随業務を派生トランザクションとして実行させるため、派生トランザクションは、オンライントランザクションと比較して、実行を待機させることができる時間が長い。このため、総量制御部は、実行されているトランザクションの総量が多い場合には、派生トランザクションの起動を制限させる処理を行う。これにより、オンライン応答が必要なオンライントランザクションの処理を優先する。総量制御部を有する実施形態では、この付随的な業務を処理する派生トランザクションの起動制限により、システムの能力を最大限活用し、複雑な処理を実行しつつ、オンラインの応答性を確保することができる。これにより、バッチ処理の日中処理化を安定して、厳格に実現することができる。

【0038】

【発明の実施の形態】

以下、本発明の実施の形態について図面を参照して説明する。本発明は、3つの実施形態を有する。第1実施形態は、派生トランザクションモデルの構造と利用法とを開示する。第2実施形態では、派生トランザクションの特性をDBに対するロックを用いて実現する手法を開示する。第3実施形態では、第2実施形態のうち、例外的なデッドロックを回避するための手法を開示する。

【0039】

【第1実施形態】

図1は第1実施形態によるトランザクション制御システムの構成を示すブロック図である。図1に示す例では、トランザクション制御システムは、端末等1A、1Bからネットワーク3を介して入力される業務処理要求を受信する受信手段2と、この受信手段2にて受信した業務処理要求を処理する演算手段4と、この演算手段4による業務処理要求の処理結果を記録するファイル6とを備えている。端末1Aは、例えば顧客が使用する端末で、ATMやパーソナルコンピュータなどである。端末1Bは内部のオペレータが使用する端末で、例えば営業店端末と呼ぶ。また、コンソール端末2Aは、システムの運用管理や業務処理要求の入力を指示するコマンドを入力する端末である。業務処理要求は、端末1A、1B又は2Aから入力される。図1に示す例では、受信手段2で受信した業務処理要求は、メッセージ入力手段8を介して演算手段4に入力される。メッセージ入力手段8は、業務処理要求であるメッセージを一時的に格納する待ち行列を備えると良い。図1に示す例では、業務処理要求が通信規格等に応じて種々の形式を採用しているとしても、メッセージ入力手段8が入力するメッセージは演算手段4によって解釈可能な形式とする。また、メッセージ入力手段8は、複数のメッセージをブロッキングして一括した入力を行うようにしてもよい。

【0040】

前記演算手段4は、前記端末等1A、1Bから入力される業務処理要求をメッセージ入力手段8から入力されるメッセージに従ってオンライントランザクションとして処理させると共に、当該処理結果を前記端末への応答として出力する制御をするオンライン応答部10を備えている。また、演算手段4は、オンライントランザクション又は所定の派生トランザクションである派生起動トランザクションの実行処理の一部として起動要求され当該派生起動トランザクションと関連した付随業務を処理する派生トランザクションを管理す

10

20

30

40

50

る派生トランザクション管理部12を備えている。演算手段4は、オンライントランザクションや、派生トランザクションや、またはバッチ処理を実行するトランザクション実行部16を備えている。トランザクション実行部16は、ファイル6に格納された各種のDBに対する操作を行う。

【0041】

端末1Aから入力される現金の入金や、端末1Bから入力される融資に関する情報の登録などは、オンライントランザクションとして処理される。ここでは、オンライントランザクションの実行と並行して実行される一括処理も、オンライントランザクションと呼ぶ。例えばオンライン稼働中の口座振替処理などは、オンライントランザクションである。オンライントランザクションではないトランザクションは、例えば、夜間バッチである。

10

【0042】

派生トランザクションは、端末からの業務処理要求としてではなく、他のトランザクションのプログラムから起動要求される。派生トランザクションの起動要求をそのプログラムの実行に従って依頼するトランザクションを、ここでは派生起動トランザクションと呼ぶ。そして、派生起動トランザクションによって起動された派生トランザクション(T1)が、さらに別の派生トランザクション(T2)を起動することがある。この場合、派生トランザクション(T1)は派生起動トランザクションである。

【0043】

オンライントランザクションは、オンラインでの応答に必要な処理を行い、そして、付随処理の必要性を判定し、付随処理が必要な場合には派生トランザクションの起動要求を依頼する。派生トランザクションの起動要求依頼があった場合には、派生トランザクション管理部12の派生起動管理機能20は、前記派生トランザクションのコミットを待機せずに前記派生起動トランザクションをコミットさせる。派生起動管理機能が、派生トランザクションの完了を待たずに派生起動トランザクションをコミットさせるため、オンライントランザクションを早期に完了させ、ロックの解放を促し、そしてオンラインの応答性を良好に維持する。すなわち、業務処理要求を要求した端末1A、1Bは、付随処理の完了を待たずにオンライントランザクションの処理結果を得ることができる。派生起動管理機能20が、派生トランザクションのコミットを待機せずに派生起動トランザクションをコミットするため、派生起動トランザクションのコミットの後に派生トランザクションがアボートされる可能性がある。本実施形態では、派生トランザクションのアボートがあった場合の再起動を制御する機能を具備することで、派生起動トランザクションと派生トランザクションとを有する処理の実際の原子性と特別な直列性を確保している。

20

30

【0044】

図1を参照すると、派生トランザクション管理部12は、前記派生トランザクションの起動要求に関連する派生起動情報を前記ファイルに格納した後に前記派生起動トランザクションをコミットさせる派生起動情報格納制御機能22と、派生起動トランザクションがアボートされる場合には前記派生トランザクションの開始又はコミットをしない制御をする直列性管理機能24と、派生起動トランザクションがコミットした後に前記派生トランザクションがアボートした場合には前記派生起動情報に基づいて当該派生トランザクションを再起動させる派生再起動制御機能25とを備えている。

40

【0045】

派生起動情報は、派生起動トランザクションから渡されるデータや、付随業務の種類や、派生起動情報のファイルへの格納位置や、派生トランザクションの起動時刻(エントリー時刻)など、派生トランザクションの起動や管理に必要な情報である。派生トランザクションを実行するための派生メッセージは、この派生起動情報又はその元となる各データに基づいて作成される。派生トランザクションは派生トランザクション管理部12によってその性質及び実行が管理される。ファイル6に格納される派生起動情報は、派生トランザクションがアボートした場合の再起動に用いられる。

【0046】

派生起動情報格納制御機能22は、派生トランザクションの再起動を确实且つ容易とする

50

ために、派生起動情報を格納する。直列性管理機能24は、派生起動トランザクションがアボートされた場合に、派生トランザクションが起動又は実行される事態の発生を防止する。派生トランザクションは派生起動トランザクションがコミットした場合に開始又は実行されるため、派生トランザクションがアボートされた場合には、派生起動トランザクションは既にコミットしている。従って、派生起動トランザクションを再起動する必要なく、派生トランザクションのみを再起動すればよい。派生再起動制御機能25は、派生トランザクションの再起動要求等を受信したときに、ファイル6に格納された派生起動情報に基づいて、派生トランザクションを再起動する。

【0047】

このように、本実施形態では、派生起動トランザクションと派生トランザクションとは次のように定義される。

(1) 個別ACID

派生起動トランザクション、派生トランザクションはそれぞれACID特性を満たす。派生トランザクションは、派生起動トランザクションとして、さらに別の派生トランザクションを起動することもできる。

(2) 拡張された直列性

派生起動トランザクションがコミットした場合にのみ派生トランザクションはコミットする。派生起動トランザクションがコミットした場合に、派生トランザクションの動作が開始するとしても良い。

(3) 拡張された原子性

派生トランザクションのアボートは派生起動トランザクションに及ばない。

(2)と(3)によると、一方通行の原子性という性質が明らかになる。派生トランザクションのコミットの条件の一つは派生起動トランザクションのコミットである。一方、派生起動トランザクションは、派生トランザクションの運命に関わらずコミットする。派生起動トランザクションと派生トランザクションを一体とした派生利用トランザクションを考えると、派生利用トランザクションは原子的でない。(3)拡張された原子性の効用は、派生起動トランザクションを早期にコミットし、そのロックを解放させ、派生起動トランザクションの応答性を高める点にある。派生トランザクションをリアルタイムに近い状態で実行することができるため、従来バッチ処理で行っていたような処理を派生トランザクションとしてオンライン応答に多少遅れて、しかしほぼ同時に、他のオンライントランザクションと並行して実行することができる。

(4) 拡張された隔離性

派生起動トランザクションが扱ったデータやその結果を派生トランザクションに渡すことができる。これは、派生トランザクションの起動の仕方に依存する。派生トランザクションの結果を派生起動トランザクションが待機することはない。派生起動トランザクションから派生トランザクションへ渡されるデータは、他のトランザクションからは隔離されている。一方、派生起動トランザクションのコミットによって、そのリソースは、派生トランザクションを含めた他のトランザクションからアクセス可能となる。

(5) 拡張された一貫性 (オプション)

派生起動トランザクションはACID特性を満たすから、そのトランザクションによってデータベースの内容等はアプリケーションにとって一貫性のある状態となる。しかし、派生トランザクションモデルでは、この一貫性は、システムレベルの一貫性であって、業務レベルでの一貫性である必要はない。すなわち、派生トランザクションモデルでは、業務レベルでの一貫性を、付随業務を処理する派生トランザクションの実行で確保するようにしてもよい。派生トランザクションを活用する一例としては、業務レベルで失われる一貫性を、システムレベルでは厳格に一貫性ある状態とするために、業務レベルでの中間的なデータを耐久性のある状態で記録するデータ構造を用いることができる。例えば、端末へのオンライン応答に必要なデータと、オンライントランザクションの処理を行った結果、数秒後や次の日までに整理又は最適化しておくべきデータとがある場合には、派生起動トランザクションは、オンライン応答に必要なデータを更新し、次の日までに必要なデータ

10

20

30

40

50

については、業務レベルでの一貫性に欠ける中間的なデータとして耐久性を持って格納する。そして、派生起動トランザクションは、この中間的なデータを格納した段階でコミットする。次の日までに整理等が必要なデータについては、派生トランザクションを用いてオンライン応答の完了後に処理することができる。

【0048】

図2は、派生トランザクションの性質を説明する説明図であり、図2(A)は、派生起動トランザクションから派生トランザクションの起動要求がなされる例を示す図である。図2(A)に示すように、業務処理要求があると、オンライントランザクションは派生起動トランザクションS1として実行され、派生トランザクションを派生起動要求した上でコミットC1する。その後、派生トランザクションS2が実行され、コミットC2する。

10

【0049】

オンライントランザクション制御システムを用いて派生トランザクションを管理する方法は、端末等から入力される業務処理要求をオンライントランザクションとして実行させる工程(S1A、オンライン応答部10)と、このオンライントランザクション実行処理の一部として派生トランザクションの起動が要求された場合には当該派生トランザクションの起動に必要な派生起動情報を前記ファイル6に格納する工程(S1B、派生起動情報格納制御機能22)とを備える。さらに、派生トランザクション管理方法は、派生トランザクションの実行開始又はコミットを待機せずに当該派生トランザクションを起動したオンライントランザクションをコミットさせる工程(C1、派生起動管理機能20)と、オンライントランザクションがアボートした場合には前記派生トランザクションの開始又はコミットをさせない制御をする工程(S2A、直列性管理機能24)とを備える。

20

【0050】

図2(B)は、派生起動トランザクションが連動トランザクションである場合を示す図である。1つのトランザクション内で複数の業務を実行する仕組みを、ここでは連動と呼ぶ。連動起動業務S3Aと連動受動業務S3Bとは1つのトランザクションとして実行されるため、双方が成功するか否かの原子性が保証される。勘定系システムを例にすると、連動起動業務S3Aは、例えば電気料の引落であり、この場合、連動受動業務S3Bは、普通預金からの出金である。連動トランザクションは1つのトランザクションとして実行されるため、電気料の支払に成功する一方、その金額が普通預金から出金されていないといった一貫性のない状態は発生しない。図2(B)に示す例では、連動起動業務S3Aが連

30

【0051】

図2(C)は、派生トランザクションが自らを派生トランザクションとして起動要求する例を示す図である。本実施形態では、派生トランザクション自身が派生起動トランザクションとなることができる。また、派生トランザクションは、派生起動トランザクションを派生トランザクションとして起動することもできる。図2に示す例では、派生起動トランザクションのコミット後に派生トランザクションがアボートした場合には、前記派生起動情報に基づいて当該派生トランザクションを再起動させる。この図2に示す各態様を組み合わせて利用することで、種々の特性を有する業務処理を実現することができる。

40

【0052】

図1に示す構成及び図2に示す仕組みは、所定のプログラムを演算手段が実行することで実現することができる。演算手段4は、所定のプログラム(指令)に従って各種の演算を行うCPUを備えている。CPUは、業務処理要求(メッセージ)に従ってオンライントランザクションをスケジュール(ルーティング)すると共に、その実行結果を端末1A、1Bに戻すための指令や、派生トランザクションの性質を管理するための指令等のオンライントランザクション制御用プログラムを実行することで、図1に示すオンライン応答部

50

10や、派生トランザクション管理部12として動作する。これらのプログラムはCPUに併設されたプログラム記憶部に格納される。また、基本的な入出力を行うオペレーティングシステムや、データベースの管理やログによる復帰や、一般的なトランザクションの管理や、DBのロックなどを行うプログラム群についても、このプログラム記憶部に格納されている。

【0053】

このようなトランザクション制御プログラムは、CPU（演算手段）を動作させる指令を備える。例えば、トランザクション制御プログラムは、端末等から入力される業務処理要求をオンライントランザクションとして実行させる指令と、このオンライントランザクションの実行処理の一部として派生トランザクションの起動が要求された場合には当該派生トランザクションの起動に必要な派生起動情報を前記ファイルに格納させる指令と、派生トランザクションの実行開始又はコミットを待機せずに当該派生トランザクションを起動したオンライントランザクションをコミットさせる指令と、オンライントランザクションがアボートした場合には前記派生トランザクションの開始又はコミットさせない制御をする指令とを備えている。演算手段4は、これらの指令を実行することで、派生起動管理機能20や、派生起動情報格納制御機能22として動作する。また、トランザクション制御プログラムは、派生起動トランザクションがコミットした後に前記派生トランザクションがアボートした場合には前記派生起動情報に基づいて当該派生トランザクションを再起動させる指令を備える。その他、トランザクション制御システムや、オンラインシステムにて実現すべき各手段、各部又は各機能や、フローチャートの各工程に応じた指令を備えることで、CPUは種々の機能を有する演算手段4として動作する。

【0054】

トランザクション制御プログラムは、磁気テープ（MT）やディスク等の記録媒体11Aに格納されて搬送され、媒体読取部11によって読み取られる。記録媒体に格納されていたトランザクション制御プログラムは、媒体読取部11にて読み取られた後、プログラム記憶部に格納される。また、他のホスト装置から通信回線を経由してプログラム記憶部にプログラムを提供することもできる。

【0055】

プログラムについて、CPU（演算手段）を「動作させる指令」というときには、各指令のみでCPUを動作させる指令と、プログラム記憶部に予め格納されているオペレーティングシステム等の他のプログラムに依存して当該CPUを動作させる指令とのいずれかまたは双方を含む。ここでは、「オペレーティングシステム」は広義に解釈している。トランザクションマネージャや、データベースマネージャ等を含む。従って、IMS、FP及びSAILなども便宜上OSと考える。

【0056】

派生起動情報格納制御指令は、派生起動情報の格納先をオペレーティングシステムに引き渡す指令のみでもよい。このように、当該トランザクション制御プログラムを記憶する記録媒体11Aであって、当該プログラムをユーザへ搬送する用途の記録媒体11Aには、例えば「派生起動情報の格納先をオペレーティングシステムに引き渡す指令」のみが格納され、実際のデータの格納等はオペレーティングシステムによって行われる場合がある。これは、動作させようとするコンピュータのオペレーティングシステム等との関係で定まる。この点、通信回線を介してプログラム（指令）を提供する場合も同様である。

【0057】

このプログラムとシステム又は方法、搬送媒体等に関する説明は後述する実施形態においても同様である。特に、同一の名前が付されている機能、工程、指令のそれぞれが対応する点は、各実施形態においても同様である。例えば、トランザクション制御プログラムは、派生起動情報格納制御機能を実現するには、その機能を実現するための派生起動情報格納制御指令を備える。

【0058】

次に、オンラインシステムでの派生トランザクションの利用法を説明する。図3（A）は

10

20

30

40

50

派生トランザクションの利用法に応じた端末応答等を説明するための説明図である。端末 1 (1 A 又は 1 B) は、業務処理要求を演算手段 4 に入力する。派生トランザクションモデルでは、主業務をオンライントランザクション S 1 1 として実行し、付随業務を派生トランザクション S 1 2 として実行する。すなわち、トランザクション実行部 1 6 は、前記業務処理要求に関連した付随業務を前記派生トランザクションとして実行する。図 3 (A) に示す例では、主業務は付随業務を派生起動しているため、派生起動トランザクション (派生起動 T) である。派生起動トランザクションと派生トランザクションとは個別にコミットする。派生起動トランザクションは、付随業務の完了を待機せずに端末 1 に応答を返す。

【0059】

勘定系システムを例とすると、図 3 (A) に示す利用法としては、口座を新設した場合の名寄せ処理がある。勘定系システムで顧客を単位とした口座体系を採用する場合、営業店 2 0 1 で口座を有する顧客が営業店 2 0 2 でも口座を新設した場合には、この 2 つの口座を関連させる名寄せ処理を行う。名寄せ処理は、例えば、同一氏名で住所が同一の場合に行う。このため、新規口座開設があると、その口座氏名と同一氏名の口座を全営業店の全口座から探索しなければならない。そして、口座の開設自体は、名寄せ処理を行わなくても完了させることができる。このため、口座の開設処理をオンライントランザクションとして構築し、名寄せ処理をこの口座開設処理の付随業務として構築する。これにより、口座開設を早期にコミットさせつつ、バッチ処理を用いずに名寄せ処理を行うことができる。

【0060】

図 3 (A) に示す他の利用法として、用途に応じたデータの二重化を行う処理がある。例えば、融資に関する情報を全営業店の通し番号である融資先番号を単位として融資マスタ DB に格納しているとする。この融資マスタ DB には、融資の実行予定日を指定した予約オペレーションにより、融資先番号を単位として予約セグメントが作成される。融資の実行に本部の決裁が必要な場合には、稟議書作成オペレーションにより稟議書を作成し、本部に送付する。本部では、融資の内容に応じて決裁又は否認のオペレーションを行う。営業店では、融資の実行予定日に、本部決裁された案件をあわせて融資実行のオペレーションを行い、融資先番号で特定できる融資先の口座に入金する。

営業店では、融資実行のオペレーションを行うために、その日に実行すべき案件の一覧を照会する必要があるが、融資マスタ DB から照会を行うと、融資先番号は店別でなく全店通し番号で採番されているため、全融資先を検索しなければならない。この場合、分単位の処理時間が必要となってしまう。そこで、営業店での照会用に、融資マスタ DB とは別の稟議 DB を構築することで、予約セグメント等の二重化を行う。稟議 DB は営業店の店番単位に、実行予定日順に予約セグメントを保有するデータ構造とし、店別の照会を短時間で行えるようにした。稟議 DB への予約セグメントの登録は、店番の下に多くのセグメントが存在し、かつ、追加される場所は実行予定日によって異なり、常に最終位置とは限らないため、必ずしも短時間で終了するというわけではない。一方、主業務である予約オペレーションは短時間で終了させたい。このため、本実施形態では、予約オペレーション等を主業務とする。そして、稟議 DB への予約セグメントの登録を付随業務として派生トランザクションで実行することとした。図 3 (A) に示す例では、ステップ S 1 1 にて融資マスタ DB へ予約セグメントを登録するための予約オペレーションを実行し、付随業務を派生トランザクション S 1 2 として起動する。予約オペレーションを実行する派生起動トランザクション S 1 1 は、付随業務 S 1 2 のコミットを待たずにコミットする。これにより、予約オペレーションは短時間で終了する。付随業務 S 1 2 では、予約オペレーションに少し遅れて、稟議 DB への予約セグメントの登録処理を行う。

【0061】

図 3 (B) は、付随業務として先日付完結処理システムでの取消再カットメインを実行する例を示す説明図である。先日付完結処理システムは、従来バッチ処理で行っていた口座振替等の決済をオンライントランザクションとして日中に処理するための手法であり、そ

10

20

30

40

50

の詳細は同一出願人により別途出願されている。ここでは、その概要を説明する。口座振替等の口座処理は、勘定日を有している。先日付完結処理システムでは、現在の残高に基づいて次営業日以降（先日付）の勘定日の口座処理を実行し、その勘定日の残高を算出しておく。このため、先日付完結処理システムでは、演算日の残高と、将来の勘定日の残高とを保有する。

例えば、10日に30,000の残高があり、10日にて11日を勘定日とする電気料の引落10,000を実行し、10日の残高を30,000としたまま、11日（先日付）の残高を20,000とする。10日に現金出金等の口座処理要求（業務処理要求）があった場合には、10日の残高である30,000を参照する。一方、日付が変更された場合には、11日の残高を参照する。11日の残高はすでに電気料の引落が完了した残高と

10

【0062】

さて、電気料の引落を先日付にて処理した後、10日の口座残高に異動があったとする。例えば25,000の出金があるとする。10日の残高は30,000であるため出金可能で、出金後は5,000となる。11日の残高については、20,000から25,000減算すると、マイナス5,000となる。残高がマイナスとなるのは自動融資等の契約によらない限り、業務上ありえないため、11日の残高を不足として電気料の引落を取り消さなければならない。電気料の引落を取り消すと、11日の残高は5,000となる。続いて、この未成立であった電気料の引落がある場合に、例えば10日を勘定日として

20

【0063】

このように、先日付完結処理では、その口座残高を最終残高とした場合の先日付の決済を実行する。口座に異動があった場合には、再度先日付の決済を調整する。これにより、口座振替等の決済の夜間バッチを日中に処理することができる。この口座振替の日中処理を、ここではセンター一括処理ともいう。センター一括処理は、バッチ処理ではなく、コンソール端末2Aからのコマンド入力により開始されるオンライン処理である。

【0064】

30

オンライントランザクションによる当日の口座残高の異動と、この異動によって生じる取消、再カットとを1つのトランザクションで実行しようとする、種々の弊害が生じる。第1に、1つのトランザクションの実行時間が長くなるため、オンラインの応答性が悪化する。第2に、取消や再カットすべき口座処理要求の数が予測できないため、例えば連動回数の制限などによって実際上処理できなくなる可能性がある。このため、先日付完結処理では、派生トランザクションモデルの採用が望ましい。

【0065】

上述したように、派生トランザクションモデルでは、拡張された一貫性を用いることができる。派生起動トランザクションはACID特性を満たすから、そのトランザクションによってデータベースの内容等はアプリケーションにとって一貫性のある状態となる。しかし、この一貫性はシステムレベルの一貫性であって、業務レベルでの一貫性である必要はない。

40

【0066】

業務レベルでの一貫性に欠けるデータは、例えば、先日付での出金可能額のマイナスを許容したデータ構造である。出金可能額がマイナスである状態は、最終的には、マイナスの勘定日を迎えるまでに解消すれば良い（実際には、派生トランザクションによって数秒以内にマイナスの状態が解消される）。先日付完結処理での派生トランザクションモデルの利用方法を図3（B）に示す。上記の例で、10日に25,000の出金要求があると、主業務（オンライントランザクション）S17Aは10日の残高を参照して25,000の出金を行い、10日の残高と11日の残高を更新する。11日の残高はマイナス5,0

50

00となる。そして、口座残高に異動が生じたため、取消再カットメインを派生トランザクションとして起動する。マイナス5, 000をファイルに格納し、取消再カットメインS18Aを派生トランザクションとして起動すると、その後の処理はオンライン応答とは無関係であるため、主業務S17Aはコミットし、出金処理を完了させる。すなわち、取消再カットメインの完了を待たずに現金出金処理をコミットさせる。先日付の決済の再調整は、口座残高の異動という主業務に対する付随業務として、派生トランザクションで処理するのである。

【0067】

取消再カットメインS18Aでは、11日のマイナス5, 000という残高に基づいて、取り消すべき口座処理の特定を行う。11日に成立している口座処理が電気料引落のみであるとすると、取消再カットメイン（派生T）は、電気料引落の取消を派生トランザクションとして起動する。ステップS17Bでは、電気料引落の取消を行い、11日の残高を更新する。電気料引落の取消処理S17Bは、口座に異動が生じたため、再度取消再カットメインS18Bを派生トランザクションとして起動する。取消再カットメインS18Bは、11日の口座残高がプラスとなったため取り消しの対象はないと判定し、また未成立の取引も存在しないため処理を完了する。

10

【0068】

先日付完結処理システムでは、派生トランザクションモデルを採用し、当日の口座処理要求に関してはオンライントランザクションとして処理し、口座の異動に伴う先日付の口座処理の調整については、派生トランザクションを用いる。これにより、オンラインの良好な応答性を確保しつつ、口座振替等の日中処理を実現する。この先日付完結処理システムは、先日付の決済には決済完了性がない（勘定日を迎えるまで取り消しが可能である）ことを利用して、口座処理間の優先順位に従った取消再カットを行うことで、口座処理を優先順位に従って並べ替えることができる。

20

【0069】

図4は、優先順位に従った口座処理の入れ替えを行う取消再カットメインの構成を示す説明図である。図4に示した技術の詳細は、同一出願人による特願2001-078306号の「先日付完結処理システム及びその方法」の明細書及び図面に記述されている。ここでは、その概要を説明する。図3を参照すると、取消再カットメインは、主業務を起動し、さらにその主業務は取消再カットメインを起動している。この派生トランザクションの繰り返し起動による処理を、スパイラル処理と呼ぶ。各取消再カットメインS18A, S18BはそれぞれACID特性を満たす独立したトランザクションであるから、内部的な情報を使用して繰り返しによる連続的な処理を実現することはできない。このため、スパイラル処理では、取消再カットメインの判定及び処理の連続性を保つための派生管理フラグを用いる。

30

【0070】

図4を参照すると、派生起動トランザクションは図2（B）に示すような連動取引を利用している。連動起動処理では、口座処理の取消や、再カットを行う。この口座処理の取消と連動して、連動受動処理側では普通預金への入金や、出金を行う。例えばS31では電気料引落の取消を行い、自動振替を管理するDBに決済不能である旨を格納する。そして、連動受動処理では、その勘定日の普通預金の残高に取消金額を入金する。電気料引落の取り消しと、普通預金への取消金額の入金とは連動するため、1トランザクションとして実行される。従って、取消及び入金とは原子性を有する。この連動起動処理S31及び連動受動処理S32は、連動受動処理が成立し、その成立に応じた処理が起動処理側で完了したときに、同期点に達する（C35）。

40

【0071】

取消再カットメインは、派生管理フラグと、マイナスを許容した先日付の残高明細や、成立した取引を記録する取引明細や、未成立の取引を記録する残不足明細などのセグメントを参照して、口座を最適化（または、業務としての一貫性の回復）のためにどのような処理をすべきかを判定し、その判定結果に従って、取消処理や、修正再カット処理や、再カ

50

ット処理を派生トランザクションとして起動する。取消再カットメインは、まず、先日付の勘定日にて出金可能額がマイナスとなっているか否かを判定する。そして、マイナスとなっている場合には（分岐 B 1）、マイナスとなっている勘定日での優先順位が最も低い口座処理要求を取り消す取消処理を、派生トランザクションとして起動する。取消処理を派生起動すると、取消再カットメインは同期点に達し（C 3 1）、終了する。

【0072】

連動起動処理 S 3 1 及び連動受動処理 S 3 2 の連動によって取消が行われ、取り消された金額分口座残高が更新された後に、取消再カットメインは再度出金可能額を確認し、マイナスであれば、この時点で最も優先順位の低い口座処理要求の取消処理を派生起動し、その後コミットする（C 3 1）。この取消処理の繰り返しを、ここでは取消スパイラル S P 1 と呼ぶ。取消スパイラル S P 1 は、出金可能額のマイナスがなくなると終了する（分岐 B 2）。

10

【0073】

口座残高が増額された場合や、出金可能額のマイナスがなくなった場合には（分岐 B 2）、修正再カットの可否を判定する。修正再カットは、先日付の勘定日について、成立済みの口座処理と未成立の口座処理とを入れ替えることで、その勘定日にて成立させる口座処理を予め定められた優先順位の順序とする処理である。具体的には、未成立の口座処理要求のうち最も優先順位の高い口座処理要求を最高位の口座処理要求とし、この口座処理要求よりも優先順位が低く成立済みの口座処理要求で利用している出金額を合計し、さらに残高を加算することで修正出金可能額を算出する。最高位の口座処理要求が修正出金可能額よりも小さければ（分岐 B 3）、最高位の口座処理要求の修正再カットを実行する（S 3 4）。修正再カットを実行すると、その勘定日の残高はマイナスとなる。従って、次に起動される取消再カットメインでは、出金可能額がプラスとなるまで優先順位の低い順に成立済みの口座処理要求を取り消す。

20

【0074】

修正再カットを繰り返すと、取引明細に格納された口座処理要求の優先順位が、残不足明細に格納された口座処理要求群の優先順位よりも高くなる。この状態では、修正出金可能額の算出ができなくなるため、修正再カットは不要と判定される（分岐 B 4）。

【0075】

修正再カットができない口座処理要求が、再カットされることはない。再カットの判定では、修正再カットの対象となっていない口座処理要求の再カットを検討する。例えば、修正再カットは取消を前提とするため、口座処理の取消を行わない当日の口座処理要求の再カットや、修正再カットを行わないと判定された優先順位の低い口座処理要求の再カットの可否を判定する。再カットは、処理対象とする勘定日（例えば、演算日当日）を勘定日とする未成立の口座処理要求のうち、最も優先順位の高い口座処理要求の取引金額（出金額）と対象勘定日の出金可能額とを比較する。取引金額が出金可能額より小さければ、再カット可能であるため、その口座処理要求の再カットを派生トランザクションとして起動し、コミットする（C 3 3）。

30

【0076】

連動起動処理 S 3 7 及び連動受動処理 S 3 8 は連動して再カット処理を行い、同期点に達した後に再度取消再カットメインが起動される。出金可能額にマイナスがなく、修正再カットが完了し、再カットの対象がない場合には、取消再カットメインは処理を完了する。一般的には、残高が減少した場合には取消スパイラル S P 1 が動作し、残高が増加した場合には修正再カット及び再カットスパイラル S P 2 が動作する。取消再カットメインが、派生トランザクションを活用して口座処理の最適化を順次実行するため、先日付での取消の発生を前提とした先日付完結処理を厳格に実行することができ、さらには、優先順位による決済という複雑な処理を安定して実現することができ、また、先日付完結処理であるため、優先順位に基づいた処理を行いつつ、24時間連続稼働を行うことができる。さらに、取消再カットメインが、口座残高に異動が生じるとに口座処理要求の優先順位に基づいた最適化を実行するため、バッチ処理と比較して、口座処理要求の実行順序が限定さ

40

50

れることがなくなる。

【0077】

この口座処理の並べ替えや、先日付を用いた夜間バッチの日中処理化は、勘定系システムに関わらず、種々のオンラインシステムで利用可能である。例えば、チケットや宿泊施設の予約に関する優先順位の反映や、商品販売をインターネット等を介して行う場合の売価や商品の変更や、注文があった場合の各種の情報の更新などである。また、販売と在庫管理と需要予測と発注とを関連させたシステムでは、販売と在庫管理とを連動させたオンライントランザクションとして実行し、発注に必要な処理を派生トランザクションとして起動する。発注の有無は販売とは無関係であるため、オンライントランザクションは在庫を参照して販売を記録した時点でコミットする。派生トランザクションは、在庫数や需要予測に応じて発注の有無を判定する。この場合、販売及び在庫管理が業務処理（主業務）であり、発注が付随処理（付随業務）となる。また、販売の成立によって、配送や請求の手配が必要となる。販売をトリガとする配送や請求の手配も、派生トランザクションの起動とすることもできる。このように、オンラインシステムで応答性の確保と、複雑な処理の自動化と、処理の厳密さとを同時に達成すべき場合には、本実施形態による派生トランザクションモデルを利用することができる。

10

【0078】

本実施形態による派生トランザクションモデルを利用すると、派生起動トランザクションの実行によって業務上の一貫性に関して中間的なデータが生じる場合には当該中間的なデータについて耐久性を持った状態で前記ファイル6に格納し、この業務処理要求の実行によって生じた業務上の一貫性に関する中間的な状態を解消して業務上の一貫性を保つための判定及び実行を前記派生トランザクションとして実行する。これにより、オンラインの応答性の向上と複雑な処理の厳密な実行とを両立させることができる。

20

【0079】

図5は端末折り返し連続処理に派生トランザクションを用いた例を示す説明図である。派生トランザクションは、付随業務の処理に用いる。付随業務の特殊な例として、端末折り返し連続処理がある。端末折り返し連続処理は、端末から入力されたトランザクションの処理がシステムの各種制限の関係で端末とのデータの送受信を繰り返さなければ完成しない処理に関連する。端末とのデータの送受信を行うと、一般に演算速度よりも通信速度の方が遅いため、CPUを通信の間待機させなければならない。

30

【0080】

例えば、積立定期の出金処理では、毎月10年間の積立で120明細となるなど支払うべき明細数が多くなることがある。この場合、システム資源に制約があるため、1トランザクションで全明細を処理できるとは限らない。従って、例えば、1トランザクションで5明細処理することとする。5明細の処理が完了する毎に端末にダミー応答を送信し、端末からの継続処理要求を受信したら、また5明細処理することを繰り返す。しかし、端末とホストのデータ送受信には、一般的に2秒程度を要するため、例えば100明細処理するのには、40秒（2秒×20回）必要になる。CPUが5明細を処理するのに要する時間は、メッセージの送受信を除くと0.1秒程度である。

【0081】

このため、図5に示す例では、端末折り返し連続処理となる業務処理要求を主業務とする場合、実際に利息を含めた支払内容の算出を付随業務と考えて、派生トランザクションの繰り返しを用いることで、端末との通信によるレスポンスの悪化を防止する。すなわち、端末折り返し連続処理では、付随業務側が実際の処理を行い、主業務は端末応答を受け持つ。図5に示す例では、積立定期の出金要求を受信すると、主業務S13Aは、支払うべき明細を特定して、最初の5明細の支払処理を行う。そして、主業務S13Aは、利息を含めた支払内容を取引明細としてファイル6の取引明細に登録する。主業務S13Aは、さらに、図5に示す端末制御DBの処理中フラグをONとする。そして、主業務S13Aは、支払処理S14Aを派生トランザクションとして起動する。端末に対しては、まだ処理中であるとの情報をセットしたダミー応答を送信する。ここで、主業務S13Aは同期

40

50

点に達する。

【0082】

支払処理 S 1 4 A は、主業務 S 1 3 A から渡された支払うべき明細のキーをもとに、5 明細の支払処理を行う。利息を含めた支払内容は、上記取引明細に計上し、更新する。そして、同一の処理を行う支払処理 S 1 4 B を起動する。ステップ S 1 4 B では、ステップ S 1 4 A と同様に、支払処理 S 1 4 A から渡された支払うべき明細のキーをもとに、次の 5 明細の支払処理を行う。これを繰り返し、支払処理 S 1 5 にて支払処理が完了すると、端末制御 D B 7 の処理中フラグを O F F にする。

【0083】

支払処理は A C I D 特性を有する派生トランザクションとして繰り返し起動されている。これとは別に、端末から状況確認を受信すると、主業務 S 1 3 B は端末制御 D B 7 をチェックする。端末制御 D B 7 の処理中フラグが O N であれば、ダミー応答を端末に送信する。さらに、状況確認要求を受信し、端末制御 D B 7 の処理中フラグが O F F となっている場合には、主業務 S 1 6 は、取引明細から支払伝票への印字データを作成し、送信する。

10

【0084】

このように、端末折り返し連続処理では、トランザクション実行部 1 6 は、業務処理要求を処理する主業務トランザクションが派生トランザクションを起動し (S 1 3 A)、派生トランザクションは処理対象が無くなるまで当該派生トランザクションを繰り返し起動する (S 1 4 A, S 1 4 B, S 1 4 C, S 1 4 D, S 1 5)。一方、主業務トランザクションは前記派生トランザクションの繰り返し起動によって処理が完了するまで前記端末にダミー応答を繰り返す (ステップ S 1 3 B)。派生トランザクション (支払処理) の繰り返し起動により処理対象の処理が完了した後に、前記主業務トランザクションは前記端末に応答を返す。

20

【0085】

この図 5 に示す例では、例えば 1 0 0 明細で派生トランザクションが 1 9 回実行され、その所要時間は 1. 9 秒 (0. 1 秒 × 1 9 回) となり、ダミー応答を送信してから次の要求 (状況確認) を受信するまでの 2 秒の間に全ての支払処理は完了し、全体のレスポンス時間は 4 秒ということになる。これは、端末への応答を行う場合の 1 / 1 0 である。

【0086】

また、連続した多数の処理を行う場合に、派生トランザクションを用いることでオペレーションを簡略化し、全体的な処理速度の向上を図ることもできる。例えば、ある処理を実際に実行させるには管理者の承認が必要な業務がある。例えば、為替振込処理について、管理者の承認が必要であるとする。この場合、その処理の内容を電文 D B に登録しておき、管理者の承認を得た段階で、実際の振込を行う。承認を得たことを個別に通知すると、振込件数分のオペレーションと端末とのデータの送受信が必要となる。一方、振込全件を 1 トランザクションで実行しようとする、処理時間が長くなり、システムのリソース不足が生じる可能性もある。このため、派生トランザクションを利用する。

30

【0087】

顧客から振込依頼書を受け付けると、オペレータは、振込依頼書に基づいて振込内容登録のオペレーションを行う。この時点では、実際の振込は行わない。この振込内容登録の内容は、伝票に印字される。振込内容の登録数がある程度まとまった時点で、オペレータは管理者の承認を求める。管理者は、振込依頼書と伝票の印字内容とが一致していることを確認し、伝票に検印を行う。伝票に検印されると、オペレータは、承認通知処理を行う。この承認通知処理により、振込処理が実行される。承認通知処理は、最初の電文番号と件数を入力することで、1 オペレーションで最大 2 2 件までの振込を処理する。

40

【0088】

承認通知処理では、まず、オンライントランザクションとして、入力された最初の電文番号で電文 D B を読み込み、正当であれば入金処理を行った後に、次の電文番号をセットして、承認通知処理を派生トランザクションとして起動する。派生された派生トランザクションでは、前処理から渡された電文番号に基づいて同じ処理を行い、次の電文番号をセッ

50

トして、派生トランザクションを起動する。この図 2 (C) に示すような派生トランザクションを入力された件数に達するまで繰り返すことで、1 度のオペレーションで最大 2 2 トランザクションを実行する。これにより、端末応答を省略しつつ、システム負荷を減らすことができる。

【0089】

このように、本実施形態では、「派生トランザクション」に関して次の 2 通りの用い方を開示する。(1) オンライントランザクションを直接要求された業務とし、このオンライン時に応答する必要はないが、このオンライントランザクションの処理結果に応じて処理しなければならない業務処理を派生トランザクションとして処理する。この (1) は、従来バッチ処理で行っていた処理、又はバッチ処理をオンライン化したことにより必要となる処理を派生トランザクションとして実行することで、オンラインの応答性を確保する。(2) 端末と CPU との通信時間を減少させるために、オンライン応答自体を処理せずに、主体的な業務を派生トランザクションの繰り返しで処理する。この (2) は、端末折り返し連続処理や、承認通知処理などが該当する。

10

【0090】

次に、派生トランザクションモデルにて、CPU の負荷に応じて派生トランザクションの実行を制限する手法を説明する。派生トランザクションは、派生起動トランザクションと同期して実行される必要はない。このため、派生起動トランザクションであるオンライントランザクションのみをコミットし、一方、派生トランザクションの実行については中止することもできる。派生トランザクションの実行を中止すると、CPU の負荷は軽減され、オンライン処理の応答性を良好に維持することができる。中止された派生トランザクションは、CPU の負荷が軽減された状態で、再起動すればよい。さらに、センター一括処理など、派生トランザクションが起動される可能性があり、且つ、内部的な処理については、CPU の負荷が多い場合にその入力を中断するようにしても良い。このように、本実施形態では、付随業務によってオンラインの応答が悪化する事態を最大限防止しつつ、複雑な業務を実行している。

20

【0091】

再度図 1 を参照すると、演算手段 4 は、前記オンライン応答部 10 及び派生トランザクション管理部 12 によって実行が管理されるトランザクションの総量に応じて前記派生トランザクションの起動を制限させる総量制御部 14 を備えている。派生トランザクションの起動の制限は、例えば、派生起動トランザクションとなる業務処理要求のうち、内部的な業務処理要求についてその入力を待機 (中断) させたり、また、業務処理要求を実行しつつ、派生トランザクションの起動を中止したり、さらに、派生起動トランザクションの実行をエラーとする処理などを含む。

30

【0092】

この例では、総量制御部 14 は、実行を待機しているトランザクションの総数を監視する実行状況監視機能 26 と、この実行状況監視機能 26 によって実行待機中のトランザクション数が予め定められた第 1 レベルの数を超えたと判定された場合には派生起動トランザクションとなる業務処理要求の入力を中断させる派生起動取引入力中断制御機能 28 とを備える。また、総量制御部 14 は、実行状況監視機能 26 によって実行待機中のトランザクション数が予め定められた第 2 レベルの数を超えたと判定された場合には、前記派生起動情報がファイルに登録された状態で派生トランザクションの起動を中止する派生起動中止制御機能 30 を備えるようにしてもよい。

40

【0093】

図 1 に示す例では、実行を待機しているトランザクション (業務処理要求の要求メッセージ) の数に基づいて総量制御をする。業務処理要求の数は、例えば待ち行列に格納された要求メッセージの数である。ここでは、派生トランザクションの起動の減少を促す第 1 のレベルと、直ちに派生起動を中止する第 2 のレベルとの 2 段階の制御を行う。派生起動取引入力中断制御機能 28 は、待機中のトランザクション数が第 1 のレベルを超えたときに、例えば先日付完結処理システムでのセンター一括処理を要求する業務処理要求の入力を

50

中断する。派生起動中止制御機能 30 は、第 2 のレベルを超えたときに、実行中の派生起動トランザクションから起動要求される派生トランザクションの実行開始を中止する。これにより、CPU の負荷が大きくなる状態であっても、オンライントランザクションの応答性の悪化を防止する。

【0094】

図 6 は 2 段階の総量制御を行う総量制御処理の構成を示すフローチャートである。ここでは、第 1 のレベル<第 2 のレベルとする。まず、トランザクションの滞留又は実行状況を調査する（ステップ S 4 1）。待ち行列等に格納されたトランザクションの総数が第 1 のレベルを越えている場合には、派生起動トランザクションの入力を中断させる（ステップ S 4 2）。さらに、第 2 のレベルを越えた滞留がある場合には、現に実行している派生起動トランザクションによる新たな派生トランザクションの起動を中止させる。これは、派生起動情報をファイル 6 に格納した後、派生起動メッセージを出力しないことで行うと良い。

10

【0095】

一方、待ち行列に登録されている業務処理要求が第 1 レベルよりも少ない場合に（ステップ S 4 6）、派生トランザクションに関する中断及び中止の継続中であれば（ステップ S 4 7）、滞留数が 0 であるか否かを確認し、滞留数が 0 であれば、派生起動トランザクションとなる業務処理要求の入力再開や、派生トランザクションの再実行を制御する（ステップ S 4 8）。ステップ S 4 8 では、再開（再実行）が可能となった旨を端末に表示し、再開（再実行）のコマンド入力を促すようにしても良いし、滞留数が 0 となったことやその後一定時間が経過したことをトリガとして自動的に再開（再実行）制御を行うようにしても良い。

20

【0096】

本実施形態では、派生トランザクションの起動を中止した場合であっても、ファイル 6 に格納した派生起動情報に基づいて再起動を管理するため、派生トランザクションの再起動が確実で、且つ容易である。このように、本実施形態では、付随業務を派生トランザクションとすることで、派生起動トランザクションであるオンライントランザクションを早期に同期点に至らせることで直接にオンラインの応答性を良好に保ち、さらに、CPU の処理能力との関係で滞留しているトランザクション数が増加した場合には、内部的な派生起動トランザクションの入力の中断や、派生トランザクションの起動中止を行うことで、間接的にオンライントランザクションの応答性の悪化を防止する。これにより、予測できないような業務処理要求の集中があった場合や、種々の業務集中日であっても、自動的にオンラインの応答性を一定の水準に維持することができる。

30

【0097】

上述したように本実施形態によると、派生トランザクションモデルを活用することで、派生起動トランザクションを早期にコミットさせつつ、仮に派生トランザクションがアボートした場合であっても確実に再起動を制御できる安定性を提供できるため、オンライン応答を良好に保ちつつ複雑な処理を実現することができ、さらに、総量制御を行う場合には、CPU の能力に応じて派生トランザクションの実行数を制限して間接的にオンライントランザクションの応答性を確保することができる。これにより、例えば、従来夜間バッチ処理で行っていたような業務を日中処理に再構築することができる。ほとんどの局面で、派生トランザクションは限りなくオンライン・リアルタイムに近い処理を可能とする。例えば、勘定系システムを例にすると、残不足で未成立の口座処理がある場合に、ATM での入金を行ったとする。入金処理が派生起動トランザクションであるため、早期にコミットし、入金による残高を通帳に記帳する。この通帳記帳を行っている間に、派生トランザクションによる未成立の口座処理の再カットが完了する。このため、通帳には再カットされた後の残高を続けて印刷することができる。

40

【0098】

このように、派生トランザクションは、通常はほぼオンライン・リアルタイムに近い状態で処理され、一方、例外的に処理が長くなる場合や、CPU の負荷が大きい場合には数十

50

秒後に処理され、また、付随業務のアプリケーションに問題があってエラーとなった場合であっても、オンラインの応答を完了しつつ、アプリケーションの修正を行い、事後的に付随業務のみを再実行することもできる。このように、派生トランザクションモデルでは、1つのトランザクションで多数の処理を行う場合の弊害からプログラマとシステムとを解放することができ、特に、通常のトランザクションモデルでは実現を断念しがちな複雑な処理のシステム開発が可能となる。

【0099】

【第2実施形態】

第1実施形態にて開示したように、派生トランザクションモデルはオンラインの応答性と複雑な業務処理の実現との優れた調和をもたらす。しかし、派生トランザクションモデルや、連鎖トランザクションモデルは標準化されていないため、使用するトランザクション管理システム（TPモニタ、トランザクションマネージャ）によっては、その機構は提供されない。しかし、派生トランザクションの特性は、DBに対するロック機構を用いることで確保することができる。第2実施形態では、この派生トランザクションの特性を管理するために2つのデータベースを用いる例を開示する。この第2実施形態を適用することで、通常のトランザクションのACID特性を管理できる。すなわち、DBのロッキングによる直列性の確保を行うことができるシステムであれば、派生トランザクションモデルを構築することができる。この第2実施形態が2つのデータベースを用いる理由は種々あるが、その一つはこの2つのDBの再編成処理を不要とするためである。従って、2つのDBを用いるようにしても、システム管理上の負荷が増えることはない。

【0100】

＜派生トランザクション管理＞

図7は、本実施形態による勘定系システムを構成するトランザクション制御システム（又は、オンラインシステム）の構成の概要を示すブロック図である。トランザクション制御システムは、端末等1A、1B、…、1Gからネットワーク3A、3Bを介して入力される業務処理要求を受信する受信手段2と、この受信手段2にて受信した業務処理要求を処理する演算手段4と、この演算手段4による業務処理要求の処理結果を記録するファイル6とを備えている。受信手段2は、図7に示す例では、通信制御装置52である。

【0101】

通信制御装置52は、内部（行内）ネットワーク3Aや、対外ネットワーク3Bと接続されている。各ネットワーク3A、3Bは、物理的、論理的に多数のネットワークを含むようにしても良い。通信制御装置52は、内部ネットワーク3Aを介して、顧客とのオンライン取引に用いられるCD/ATM1Aや、営業店内に配置され営業店の担当者によって操作される営業店端末1Bや、顧客からの電話による取引を遂行するためのテレホンバンキングセンタ1Cと接続されている。

【0102】

また、通信制御装置52は、対外ネットワーク3Bを介して他の金融機関との為替等の通信を制御する全銀センタ1Dや、その他の外部センタ1Eや、ファームバンキングやインターネットバンキングを行う顧客のパーソナルコンピュータ等の顧客PC1Fや、携帯電話等の顧客携帯端末1Gと接続されている。

【0103】

各端末1A、1B、1C、1F、1Gや、センタ1D、1E等から、通信制御装置52を介してCPUに入金や出金の要求が入力される。また、通信制御装置52には、センター一括処理の開始を要求するコマンド（業務処理要求）や、アボートされた派生トランザクションの再起動を要求するコマンドを入力するコンソール端末2Aが接続されている。全銀センタ1Dは、全国銀行データ通信システムのセンタである。個人情報センタや、他行コンピュータセンタと接続されている。振込、送金等の為替取引を行う要求指令（電文）が送受信される。

【0104】

外部センタ1Eは、金融機関の業態別に構築された各種のネットワーク（地方銀行のAC

10

20

30

40

50

S, 都市銀行のBANCS, 全国キャッシュサービスのMICS等)のセンタや、ファームバンキング(FB)やエレクトロニックバンキング(EB)や、テレフォンバンキングに必要なデータの送受信を行うためのセンタ(例えば、ANSER)や、販売店のPOSシステムやクレジットカード決済用の端末との通信を行うためのセンタ(例えば、CAFISやDCS)である。この外部センタからは、通知、照会、資金移動など、金融機関窓口で可能な種々の取引を行う要求指令(電文)が入力される。

【0105】

通信制御装置52は、それぞれのプロトコルに従って要求指令を受信し、演算手段4を構成するCPUに入力する。CPUは、各端末からそれぞれの形態で入力される要求指令を予め定められた形式の口座処理要求に変換して、口座処理を行うプログラムに引き渡す。

10

【0106】

また、公共料金の振替等(外部自振)は、持ち込み媒体54にて持ち込まれる。CPUは、媒体読取部56によって読み取られた外部自振原データと他のデータベース(DB)を用いて業務処理要求を生成する。図7に示す例では、勘定系システムは、持ち込み媒体54から読み出したデータや、予め登録した自振契約や融資返済等に関するデータを記録する自振関連DB50と、この自振関連DBに格納されたデータに基づいて口座振替等の自動振替を行うため業務処理要求群である自振入力ファイルを生成する自振入力ファイル生成手段60と、この自振入力ファイル生成手段60によって生成された自振入力ファイルであるセンターカット電文をコンソール端末2Aのオペレーションに応じてトランザクション管理部64に入力するセンターカット電文入力手段62とを備えている。センターカット電文は、センター一括処理を指示する電文であり、自振の処理要求を複数ブロッキングした電文である。自振の処理要求を5つブロッキングした場合、このセンターカット電文によるトランザクションは自動振替を5つ処理する。

20

【0107】

ファイル6は、図7に示す例では、派生トランザクションの派生起動情報を派生管理通番毎に記録する派生管理DB70と、この派生管理DB70のキーとなる派生管理通番毎に派生管理DBが使用中であるか否かを指示するインデックス情報を記録する派生インデックスDB68とを備えている。また、ファイル6は、トランザクションの滞留に応じたシステムの状況を示すビット(派生起動取引中断ビット等)などを記録するシステム状況DB83を備えている。

30

本実施形態では、派生管理DB70に派生起動情報を格納する。そして、この派生起動情報の振る舞いを利用して、派生トランザクションの特性を管理する。

【0108】

派生起動トランザクションは派生起動情報を格納する。しかし、派生起動トランザクションがアボートされた場合には、ロールバックにより派生起動情報の更新がなかったこととされる。従って、派生トランザクションは、起動開始直後にこの派生起動情報の読み込みを行うことで、派生起動トランザクションがコミットしたか又はアボートしたかを知ることができる。例えば、派生起動情報に使用中フラグを含める例では、派生起動トランザクションがコミットすれば、この使用中フラグは「使用中」(例えば、キャラクターの“1”)である。一方、アボートされた場合には、派生起動トランザクションによる使用中フラグの「使用中」への更新があったとしても、その更新はロールバックされる。このため、派生トランザクションは、派生起動情報の使用中フラグが「使用可」(例えば、キャラクターの“0”)である場合には、その派生起動トランザクションがアボートされたと判定することができる。これにより、派生起動トランザクションがコミットした場合にのみ派生トランザクションがコミットするという条件を満たすことができる。

40

【0109】

また、派生トランザクションは、派生起動トランザクションの実行中に起動要求依頼される。従って、派生起動トランザクションのコミット前に、派生トランザクションがスケジュールされ、処理を開始してしまうことがある。これを防止するために、派生トランザクションは、その開始直後に派生起動情報を読みに行く。派生起動トランザクションがコミ

50

ットするまでは、この派生起動情報はロックされているため、派生トランザクションは、そのロックが解除されるまで待機する。すなわち、派生トランザクションは、その開始時に派生起動情報を読み取ることで、派生起動トランザクションのコミットを待機する。これにより、派生起動トランザクションと派生トランザクションとの直列性を管理する。

【0110】

このように、派生起動トランザクションがその一部として派生起動情報を格納し、派生トランザクションが処理の開始時に派生起動情報を読み取ることで、拡張された直列性という派生トランザクションの特性を実現することができる。

【0111】

また、派生管理DB70のみとして、派生インデックスDB68を用いないことも可能であるが、派生インデックスDB68を用いると、派生管理DB70の再編成を不要とし、データの削除等のメンテナンスを行うことなく24時間連続稼働させることができる。派生インデックスDB68は、派生管理通番の状態（使用中又は使用可）を管理する。トランザクション管理部64が使用可の派生管理通番を使用するとき、派生トランザクションの正常なコミットによって不要となった派生起動情報を消去し、新たな派生起動情報を記録する。従って、不要となった派生起動情報は次々と上書きされるため、派生管理DBの再編成処理を行う必要がなくなる。この派生起動情報を上書きする例では、派生インデックスDB等を用いて消去しても良い派生起動情報を格納したセグメントを示す派生管理通番を特定する。従って、データを格納しているという点では、派生管理DBは未使用であるとは限らない。データが格納されていても、その派生起動情報は役割を終えて消去可能な状態である。この消去可能な状態を、「使用可」という。

【0112】

図7に示す例では、ファイル6はさらに、センター一括処理を要求する電文（業務処理要求群）を生成するための自振関連DBや、取引明細や、融資に関連する情報などを格納する各種のマスタDBを備えている。

【0113】

演算手段4は、前記端末等から入力される業務処理要求をオンライントランザクションとして処理させると共に当該処理結果を前記端末への応答として出力する制御をするオンライン応答部58と、前記オンライントランザクション又は所定の派生トランザクションの実行処理の一部として起動要求される派生トランザクションを管理すると共に前記オンライントランザクションを管理するトランザクション管理部64とを備えている。

【0114】

本実施形態では特に、トランザクション管理部64が、派生インデックスDBを参照して使用可の派生管理通番を特定する派生管理通番特定機能101と、特定された派生管理通番のセグメントに派生起動情報を格納する派生起動情報格納制御機能102と、この派生起動情報の格納後に派生トランザクションをコミットさせる派生起動トランザクションコミット処理機能103と、派生起動トランザクションがアボートした場合には派生トランザクションの処理を開始させずに終了させる直列性制御機能104と、派生トランザクションをコミットさせると共に使用した派生管理通番等を使用可に更新する派生トランザクションコミット処理機能105とを備えている。また、トランザクション管理部64は、アボートされた派生トランザクションの再起動を制御する派生再起動制御機能106とを備えている。

【0115】

派生管理通番特定機能101は、オンライントランザクション又は派生トランザクションである派生起動トランザクションから派生トランザクションの起動要求依頼があったときに、前記派生インデックスDB68のインデックス情報を特定すると共に、当該特定したインデックス情報に基づいて前記派生管理DBの使用可の派生管理通番を特定する。並列に実行されている全てのトランザクションが一つのインデックス情報を読み取る仕組みとすると、このインデックス情報へのコンテンションが増大してしまう。このため、本実施形態では、派生管理通番特定機能101は、なんらかの条件に従って起動要求依頼を発行

10

20

30

40

50

した派生起動トランザクションが利用できるインデックス情報を特定する。インデックス情報を特定できると、そのインデックス情報を参照して使用可の派生管理通番を特定する。派生管理DBへのアクセスは、派生管理通番をキーとして行うため、意図しないコンテンションは発生しない。

【0116】

派生起動情報格納制御機能102は、派生管理通番特定機能101によって特定された派生管理通番のセグメントに前記派生起動トランザクションによる派生トランザクションの起動要求依頼に関する派生起動情報を格納する。派生起動情報格納制御機能102は、この派生起動情報の格納後に、派生トランザクションを起動する。

【0117】

派生起動トランザクションコミット処理機能103は、派生起動情報を与えられた派生管理通番のセグメントに格納した後で、当該派生起動トランザクションの各処理が正常終了した場合に、当該派生起動トランザクションをコミットさせる。第1実施形態と同様に、派生起動トランザクションは派生トランザクションの運命に関わらず早期に同期点に達する。そして、直列性制御機能104は、派生トランザクションの開始時に前記派生起動情報をロック可能な状態での正常な読み出しに成功した場合に当該派生トランザクションの各処理を開始させる。「正常な読み出し」は、派生起動トランザクションによって格納された状態での読み出しを意味する。すなわち、直列性制御機能104は、派生起動トランザクションがアボートされ、この派生起動トランザクションによって格納された派生起動情報がロールバックにより無かったものとされている場合には、正常な読み出しができなかったとして、派生トランザクションの起動を行わない。

【0118】

派生トランザクションコミット処理機能105は、派生トランザクションの各処理が正常終了した場合に、使用した派生管理通番について前記派生インデックスDB68及び前記派生管理DB70を使用可に更新させる。この派生インデックスDB及び派生管理DBを使用可に更新する処理と前後して、当該派生トランザクションをコミットさせる。派生インデックスDBを使用可に更新する処理は、派生トランザクションの一部として実行しても良いし、また、第3実施形態で詳述するように別のトランザクション（派生DB制御処理）にて行うようにしても良い。

【0119】

派生再起動制御機能106は、派生トランザクションがアボートされた場合に、派生管理DBに格納した派生起動情報に基づいた再起動を制御する。再起動は、自動的に行うようにしても良いし、再起動を指示する業務処理要求を受信したときに実行するようにしても良い。

【0120】

図8は、JOBと派生インデックスDB等との関係を示す説明図である。TPモニタの一種であるCICSやIMSでは、リージョン（region）と呼ばれる抽象化を提供している。リージョンは、複数の処理を実行できるアドレス空間である。リージョンは資源を保有する単位であり、アプリケーション実行時の障害はリージョン内に局所化される。業務処理要求は、リージョンにスケジュールされる。例えばリージョンを50個起動する場合には、JOB番号0から49までのトランザクションが並行して動作する。JOB番号は、予め固定された番号とする。このように、リージョン数が予め定められており、且つ、使用するJOB番号（例えば、JOB名の下三桁の数値）が予め定められた範囲内の数値である場合には、派生インデックスDB68のインデックス情報をJOB番号を単位として構築すると良い。この場合、別のリージョンで並行して実行されているトランザクションとインデックス情報の読み取りで競合することがない。リージョンの概念を用いないシステムであっても、並列して実行されるトランザクションを識別するための番号は必ず存在する。この場合、JOB番号としてトランザクションID又はこれを編集した番号を用いる。JOB番号は、並列して実行されるトランザクション又はプロセス間でユニークで、且つ、その番号が予め定められた範囲内の数値であれば良い。

10

20

30

40

50

【0121】

図8に示す例では、派生起動ランザクションは、予め定められたJOB番号群の中のいずれかのJOB番号で実行される。そして、派生インデックスDBが、前記ランザクションのJOB番号をキー値となるインデックス領域番号88として当該インデックス領域番号88毎にインデックスビットマップ68を有する。そして、図8に示す例では、派生インデックスDBは、前記インデックス情報を前記インデックス領域番号毎のビットマップとして保有している。リージョン0では、JOBFP000がスケジュールされ、実行している。JOBFP000のランザクションがインデックス情報（ビットマップ）を読み取る場合には、このJOB名の下三桁の数値である「000」をキー値として派生インデックスDBにアクセスする。すると、インデックス領域番号が「000」であるビットマップにアクセスする。このビットマップでは、先頭から「11100…」となっている。同様に、リージョン1で動作するランザクションはインデックス領域番号が「001」のビットマップにアクセスする。このビットマップは、先頭から「0100…」となっている。

10

【0122】

インデックス情報をビットマップとする例では、ビットのON（1）、OFF（0）で派生管理通番が使用中であるか否かを示し、そのビットの位置で派生管理通番を特定する。ここでは、一例として、1つのリージョンから利用できる派生管理通番を400個とする。この場合、ビットマップは、400ビットある。すなわち、1バイトを単位とする場合には長さが50となる。リージョン0を例とすると、派生管理通番0001, 0002, 0003は使用中で、それ以降が使用可である。この3つの派生管理通番が使用中であるということは、リージョン0から起動要求された3つの派生ランザクションがそれぞれコミットしていないことを意味する。

20

【0123】

派生管理通番特定機能101は、例えば、ビットマップを順に先頭から読み出し、「0」であるビットのビット位置を特定する。そして、このビット位置から派生管理通番を求める。例えば、リージョン0の場合、先頭から4番目のビットが「0」であるため、特定する派生管理通番は0004となる。リージョン1の場合、派生管理通番は0401から開始するため、ビットマップの先頭を特定した場合には、派生管理通番は0401となる。

【0124】

このように、好ましい実施形態では、派生管理通番特定機能101が、前記インデックス領域番号88と前記ビットマップのビット位置とに基づいて前記派生管理DB70の派生管理通番を特定する機能を備える。

30

【0125】

図9は、この場合の派生DBのデータ項目を例示する説明図であり、図9（A）は派生インデックスDBの項目例を示す図である。Nは例えば125、Mは400とする。派生インデックスDB68は、その項目として、インデックス領域番号（各ビットマップに付されJOB番号と同一の数値）と、ビットマップ（インデックス・ビットマップ）とを備えている。また、派生インデックスDBは、そのビットマップが特定する派生管理通番の残り個数と派生件数とを格納する。例えば、残り個数が0であれば派生ランザクションの起動を待機する等の処理を行う。図9（A）の残り個数の初期値Mは、上記の例では400である。

40

【0126】

図9（B）は派生管理DBの項目例を示す図である。派生管理DBは、キー値として派生管理通番を有する。その他の項目は派生起動情報の詳細である。派生起動ランザクションのAPPL IDと、派生ランザクションのAPPL IDとを保持することで、どの主業務がどの付随業務を派生起動したのかを記録する。派生編集番号は、入出力の形式を特定するために利用する。エントリー日付及び時刻は、アボートされた派生ランザクションの探索に利用する。使用中フラグは、派生ランザクションが派生起動ランザクションがコミットされたか否かを知るために用いる。FIFO保存とあるのは、派生起動

50

トランザクションが派生トランザクションを起動するために用意した各種のデータ内容の保存領域である。

【0127】

本明細書では、F I F Oというときには、定型入出力情報域 (Formatted Input Formatted Output) を意味する。先入れ先出し方式の略語としては用いない。F I F Oは、I M S / E S A オンライン適用業務開発／運用支援プログラムであるS A I Lによって提供される抽象化である。端末等からの全ての入力、その業務に応じてF I F Oという情報域に編集され、セットされる。このため、適用業務プログラムは、端末等から入力の形式によらず、予め定義したF I F Oでの位置に特定の項目のデータがセットされている前提で開発することができる。すなわち、F I F Oを用いることで、入出力メッセージの形式によらず、適用業務プログラムを開発することができる。入出力メッセージとF I F Oとは、編集定義を用いて関連させる。例えば、派生起動トランザクションが派生トランザクションを起動する場合、派生トランザクションの実行に必要なデータをF I F Oにセットし、派生起動要求依頼を行う。すると、S A I Lは、派生編集番号に従ってF I F Oから派生メッセージを生成する。端末からの業務処理要求の入力や、端末への出力メッセージについても同様に業務毎の編集定義を用いる。

10

【0128】

図10は、J O B番号と、派生インデックスD B 6 8のインデックス領域番号8 8と、ビット位置と、符号8 9で示す派生管理通番との関係を例示する説明図である。図10 (A) に示すように、J O B番号0 0 0で実行されているトランザクションによって読み出されるインデックス・ビットマップの番号 (インデックス領域番号) は、「0 0 0」である。図10 (A) に示す例では、ビット位置1及び5が使用中である。この場合、派生管理D Bの内容は図10 (B) に示す如くとなる。派生起動トランザクション側では、インデックス・ビットマップのビット (派生インデックスビットと呼ぶ) と、使用中フラグとは、原子性に包まれていることが望ましい。

20

【0129】

図10 (C) 及び (D) に示すように、1つのインデックス・ビットマップが特定する派生管理通番の数Mが定められていると、インデックス領域番号8 8とビット位置とに基づいて派生管理通番を特定することができる。

【0130】

図11は、派生トランザクションの要求を管理するための派生要求管理処理の一例を示すフローチャートである。図12は、派生トランザクションの特性を管理するための派生特性管理処理の一例を示すフローチャートである。図11に示す派生要求共通処理は、派生起動トランザクションの一部として実行される。図12に示す派生特性管理共通処理は、派生トランザクションの一部として、または別の処理として実行する。

30

【0131】

図11を参照すると、派生要求共通処理は、派生起動トランザクションから派生起動要求依頼を受けると、まず、そのJ O B番号 (すなわち、インデックス領域番号) をキー値として派生インデックスD Bを読み込む (ステップS 5 2)。そして、インデックス領域番号での残り個数フィールドをチェックし (ステップS 5 2)、空きがあれば、インデックス・ビットマップから空き (インデックスビットが0) のビット位置を特定する。ステップS 5 2にて空きが無い場合には、予備の派生インデックスD Bを読み込み、残り個数フィールドをチェックする (ステップS 6 2)。予備の派生管理通番まで使用中である場合には、異常終了する。

40

【0132】

空きの派生インデックスビットを特定すると、そのビット位置に基づいて派生管理通番を算出する。そして、この派生管理通番を派生起動トランザクションが動作しているリージョンのF I F Oにセットする。さらに、派生インデックスビットをO N (対応する派生管理通番が使用中) を示す「1」に更新する (ステップS 5 4)。

【0133】

50

続いて、派生管理通番をキー値として、派生管理DBを読み込み（ステップS55）、派生インデックスDB68の残り個数フィールドと、派生件数フィールドとを更新する（ステップS56）。そして、派生管理DBに派生起動情報を登録し、派生管理DB70の使用フラグを「使用中」とする（ステップS57）。続いて、派生トランザクションの起動を制御する（ステップS58）。例えば、SAILは、派生起動トランザクションのFIFFOから派生メッセージを生成し、待ち行列に登録する。続いて、派生起動トランザクションはコミットする。

【0134】

図12は、派生トランザクションの開始及び終了時等の処理を示すフローチャートである。派生特性管理共通処理では、まず、派生トランザクションがいずれかのリージョンにスケジュールされると、まず、派生起動メッセージの派生管理通番をキー値として派生管理DB70を読み込む。派生起動トランザクションがコミットしていない段階では、派生管理DB70の当該派生管理通番のセグメントはロックされているため、ステップS71では、派生起動トランザクションがコミット又はアポートし、このロックが解除されるまで待機する。

10

【0135】

派生トランザクションは、派生管理DBの内容の読み出しに成功すると、まず、使用中フラグをチェックする。使用中フラグは、派生起動トランザクションによってステップS57によって使用中を示す1に更新にされている。派生起動トランザクションがコミットした場合、この使用中フラグは「使用中」のままである。一方、派生起動トランザクションがステップS58以後にアポートした場合、ロールバックによって、派生起動トランザクションによる使用中フラグの「使用中」への更新が取り消され、使用中フラグは使用可を示す0となっている。従って、ステップS72では、使用中フラグが使用可の場合には、派生起動トランザクションがコミットした場合にのみ派生トランザクションがコミットするという派生トランザクションの性質に従って、派生トランザクションの適用業務プログラムを実行せずに終了する（ステップS74）。

20

【0136】

派生起動トランザクションが正常にコミットしている場合、使用中フラグは「使用中」であるため、派生トランザクションの適用業務プログラムを呼び出し、実行を制御する（ステップS75）。この適用業務プログラムの実行が完了すると、派生特性管理共通処理では、派生管理通番をキーとして派生管理DBを読み込む（ステップS76）。そして、使用中フラグを使用可に更新する（ステップS77）。このとき、派生インデックスDBの派生インデックスビットも使用中のままとなっている。このため、派生トランザクションの一部として、又は別のトランザクションを用いて、派生インデックスDB78の派生インデックスビットをオフに更新する。この派生インデックスビットの更新に前後して、派生トランザクションはコミットする。

30

【0137】

また、図12に示す派生特性管理共通処理によってその特性が管理される派生トランザクションが、別の派生トランザクションを起動する場合には、この派生トランザクションをコミットさせる前に、図11に示す処理を実行する。

40

【0138】

ここで、図7に示す各機能と、図11及び図12に示す各工程との関係を説明する。図7に示すトランザクション管理システムの各機能101、102、103、104、105、106は、第1実施形態の場合と同様に、CPUが、派生管理通番特定機能101を実現するための派生管理通番特定指令や、派生起動情報格納制御指令などを実行することで実現できる。

【0139】

例えば、このプログラムは、オンライントランザクション又は派生トランザクションである派生起動トランザクションから派生トランザクションの起動要求依頼があったときに前記派生インデックスDBのインデックス情報を特定させると共に当該特定したインデック

50

ス情報に基づいて前記派生管理DBの使用可の派生管理通番を特定させる派生管理通番特定指令を備えている。この指令を実行すると、CPUは派生管理通番特定機能101として動作する。図11に示す例では、この派生管理通番特定指令を実行すると、CPUはステップS51、52、S53及びS54等の工程を実現する。

【0140】

また、このプログラムは、派生管理通番特定指令に応じて特定された派生管理通番のセグメントに前記派生起動トランザクションによる派生トランザクションの起動要求依頼に関する派生起動情報を格納させると共に当該派生起動情報を格納した後に当該派生トランザクションを起動させる派生起動情報格納制御指令を備える。CPUは、この指令を実行することで派生起動情報格納機能102として動作する。図11に示す例では、ステップS54、S55、S56、S57の工程を処理する。

10

【0141】

プログラムは、さらに、派生起動情報の格納後で当該派生起動トランザクションの各処理が正常終了した場合に当該派生起動トランザクションをコミットさせる派生起動トランザクションコミット処理指令を備える。これにより、CPUは、派生起動トランザクションコミット処理機能103として動作する。ステップS11では、最後の工程である。

【0142】

また、プログラムは、前記派生トランザクションの開始時に前記派生起動情報をロック可能な状態での正常な読み出しに成功した場合に当該派生トランザクションの各処理を開始させる直列性制御指令を備える。図12に示す例では、これは、ステップS12の派生起動情報の読み取り工程S71によって、この直列性の制御を実現する。直列性制御指令を実行するCPUは、直列性制御機能104として、図12に示すステップS71からS75の処理を行う。

20

【0143】

派生トランザクション管理用プログラムはさらに、派生トランザクションの各処理が正常終了した場合に使用した派生管理通番について前記派生インデックスDB及び前記派生管理DBを使用可に更新させると共に当該派生トランザクションをコミットさせる派生トランザクションコミット処理指令を備える。この指令を実行するCPUは、派生トランザクションコミット処理機能105として、図12に示すS76、S77、S78、S79の処理を行う。第3実施形態では、デッドロックの発生を防止するために、派生トランザクションとは別のトランザクションとしてS78及びS79による派生インデックスDBの更新を行う。

30

【0144】

また、このプログラムは、派生トランザクションがアボートされた場合には前記派生管理DBに格納した派生起動情報に基づいた再起動を制御する派生再起動制御指令を備える。この派生再起動制御指令を実行するCPUは、派生再起動制御機能106として機能する。実際の再起動処理は、派生再起動制御部77が行うようにしても良い。このように、図7に示す各部や図11、図12等のフローチャートに示す各工程は、それぞれの処理を行うプログラムを実行することによって実現できる。これは、総量制御処理や(図19)、派生再起動要否判定処理(図22)や、派生再起動制御処理(図23)の各フローチャートに示す工程についても同様である。

40

【0145】

このように、図11及び図12に示す例では、派生インデックスDB68が、前記インデックス領域番号88毎のビットマップを派生インデックスビットの列として有すると共に、当該派生インデックスビットのON又はOFFに応じて当該ビットのビット位置に対応する派生管理DB70の使用状態を記録している。そして、派生管理DB70は、派生管理通番毎に当該派生管理通番の使用状態を使用中又は使用可として記録する使用中フラグを有する。

【0146】

さらに、派生管理通番特定機能101は、使用可を示すOFF(0)の派生インデックス

50

ビット及び使用可（０）の派生管理通番を特定したときに、当該派生インデックスビットについて使用中を示すＯＮ（１）へと更新する（ステップＳ５４）。続いて、派生起動情報格納制御機能１０２は、派生起動情報を格納したときに当該派生管理通番の使用フラグについて使用中（１）へと更新する（ステップＳ５７）。そして、直列性制御機能１０４は、前記派生起動トランザクションから渡された派生管理通番の使用フラグが使用可（０）であった場合には当該起動要求されている派生トランザクションを起動せずに処理を終了させ（ステップＳ７４）、派生トランザクションコミット制御機能１０５は、当該派生トランザクションの完了時又は完了後に前記派生インデックスビット及び前記使用中フラグについて使用可の状態（０）へと更新する（ステップＳ７７、Ｓ７９）。すなわち、派生インデックスビットをＯＦＦ（０）とし、使用中フラグを使用可（０）とする。

10

【０１４７】

このように、派生起動トランザクションと派生トランザクション間の派生起動情報のコンテンツンションを利用することで、派生トランザクションの性質を一般的なＤＢのロック機構を用いて実現することができる。

【０１４８】

次に、先日付完結処理システムでの取消再カットメインを例として派生トランザクション管理の一例を説明する。

トランザクションのコミットのタイミングをプログラム中に記述しない方が、そのプログラムの再利用性は高まる。このため、トランザクションの実行を管理するプログラムが、適用業務プログラム（ＡＰＰＬ）を呼び出し、実行し、正常に終了した段階でコミットを与える仕組みとすると、ＡＰＰＬを種々の態様で利用することができる。トランザクションの管理を行うプログラムを、ここでは業務メインと呼ぶ。また、ＡＰＰＬの実行形態として、端末から初回に入力された場合や、連続して繰り返し処理する場合や、連動処理として呼び出された場合や、派生トランザクションとして起動された場合や、センター一括処理として入力される場合などがある。これらの取引の態様が異なっても、プログラムの一部または全体が共通する場合がある。このため、プログラムを細分化しておき、起動の形態に応じて適切なプログラムを特定する仕組みとすると、プログラムの再利用性が高まり、全体の行数を少なくすることができる。

20

【０１４９】

図１３は、このような仕組みを提供するためのオペコードテーブルの一例を示す説明図である。業務処理要求や、派生トランザクションの内容は、オペコードで特定される。例えば、普通預金の出金処理のオペコードは０１００１である。そして、端末初回入力の場合には、プログラム０１，０２，０３をこの順に呼び出して実行する。同様に、連動処理の依頼を受けた場合には、プログラム０６，０７，０８をこの順に実行する。ＳＡＩＬは、業務処理要求（オペコードを含む）を受信すると、その業務に応じた編集定義を特定し、編集定義に基づいてＦＩＦＯを組み立てると共に、業務メインを呼び出す。業務メインは、オペコードテーブルのプログラムリストに従ってプログラム（ＡＰＰＬ）を呼び出す。

30

【０１５０】

さらに、業務メインは、派生トランザクションの起動要求依頼を受けた場合には、図１１に示す処理を実行するプログラムを呼び出す。一方、派生トランザクションの起動に際しては、派生トランザクションのオペコードによるＡＰＰＬの呼び出しの前後に、図１２に示す処理を実行する。

40

【０１５１】

図１３に示すように、普通預金出金のオペコードは、０１００１である。先日付完結処理システムでは、この普通預金出金は取り消されることがある。この取消については、すべてのオペコードについて一定値（図１３に示す例では、５００）を加算する体系としている。これにより、ある取引を取り消す場合にオペコードを探索する必要がなくなる。図１３に示すように、普通預金出金は、定積入金（０３００１）や、定期入金（０４００１）から連動起動又は派生起動なされる。

【０１５２】

50

図14は、取消再カットメインでの処理例を示す説明図であり、図14(A)は口座処理による電気料引落(内部自振)の取消を行う例を示す図で、図14(B)はその処理による残高の推移を示す図である。図14(B)に示すように、6日の残高が60,000で、7日を勘定日とした電気料の引落(自振通番12345)20,000が先日付完結処理により実行され、7日の残高が40,000となっているとする。この状態で、ATMを用いて普通預金から定期預金への入金があると、定期入金処理S21は、オペコード01001の普通預金出金S22を連動起動する。普通預金出金では、6日の残高は60,000であるため、50,000の出金が可能であることから、50,000の出金を行い、成立した旨を定期入金に伝える(連動戻り)。

【0153】

定期入金では定期預金マスタ等に記録し、ATMに定期入金が完了した旨の応答を出力する。一方、普通預金出金S22では、6日の残高の更新後に、先日付の勘定日である7日の残高も更新する。7日の残高は、マイナス10,000となる。普通預金出金S22は、7日の残高がマイナスとなったことから、7日を勘定日として取消再カットメインを起動する。取消再カットメインは、出金可能額のマイナスが解消するまで7日にて成立している口座処理を取り消す制御をする。ここでは、電気料の引落を取り消しを起動する。ステップS24では、電気料の引落の取消(オペコード02001+500=02501)を行い、普通出金の取消に連動する。普通出金の取消(オペコード01001+500=01501)は、20,000の出金の取消を行うため、7日の残高に20,000を加算する。そして、7日の口座残高に異動が生じたため、再度取消再カットメインを派生トランザクションとして起動する。取消再カットメインS26は、図4に示す処理を行い、取消再カットの必要性がないことから、なんら起動せずに処理を終了する。

【0154】

図14(A)に示すように、定期入金と普通出金は連動しており、1つのトランザクションT1である。これが、リージョン0にスケジュールされ、JOB番号が「000」とあるとする。普通出金が派生トランザクションの起動要求依頼をしたとき、トランザクションT1はコミットしていない。高速パス(FP)を用いたシステムでは、トランザクションT1のコミットの前に派生トランザクションがスケジュールされる。リージョン0は派生起動トランザクションT1によって使用されているため、派生トランザクションである取消再カットメインは、例えばリージョン1にスケジュールされる。この場合、取消再カットメインのJOB番号は「001」となる。取消再カットメインS23は、種々の判定処理を行った後に電気料引落の取消を派生トランザクションとして起動する。

【0155】

取消再カットメインS23が派生起動トランザクションT2として電気料引落の取消処理を起動させたとき、定期入金及び普通出金のトランザクションT1は既にコミットしている可能性が高い。すなわち、派生起動トランザクションT1はその付随業務である取消再カットメインである派生トランザクションT2のコミットを待機せずにコミットする。トランザクションT1がコミットしていると、リージョン0は空いているため、電気料引落の取消を行う派生トランザクションT3は、リージョン0にスケジュールされることがある。

【0156】

図15乃至図17は、この状態での派生起動情報の格納及び派生メッセージの出力や、FIFOの推移を説明するための説明図である。この例では、IBM社のMVSで動作するIMSをFPで高速化し、さらにSAILによるFIFO及び編集定義を使用し、そして、業務メインによりオペコードテーブルを用いて各トランザクションを管理する。この例では、第2実施形態で説明している派生インデックスDB及び派生管理DBを用いている。

【0157】

業務メインは、ATMから定期入金の業務処理要求を、図15の符号M1で示す入力メッセージの形式で受信する。入力メッセージM1中、Hはヘッダー、オペコードは定期入金

10

20

30

40

50

を示す04001（図13参照）、金額は50,000であり、ここでは、普通預金の店番211の口座21yyyyから出金して、210店番の口座3000xxxxxに入金する処理を要求する。IMSは、入力メッセージM1を待ち行列から取り出すと、空いているリージョン0へスケジュールし、トランザクションを開始させる。SAIL及び業務メインは、図17に示すように、このリージョン0内に、入力メッセージM1から所定の入力編集定義を用いて定期入金のFIFO86Aを展開する。図15に示すように、FIFO86Aには、予め定められた位置に、オペコードや、定期入金する店番や、入金金額等が格納される。そして、業務メインは、定期入金のオペコードテーブルを参照して所定のプログラムを呼び出し、実行させる。定期入金プログラムは、普通預金からの出金を依頼するために、オペコード01001への連動を起動する。このとき、連動編集定義に基づいて、普通出金用のFIFO86Bがセットされる。結果コードは、普通預金からの出金に成功するか否かを示すコードである。

10

【0158】

普通預金出金は、FIFOにセットされた2001/3/6を勘定日とする残高を参照し、出金が可能であるか否かを判定し、出金が可能であれば、6日の残高を10,000に更新する。そして、先日付の勘定日が存在するため、7日の残高も更新する。7日の残高がマイナスとなったため、取消再カットメインを処理する開始日（派生日付）を2001/3/7にセットして、派生起動要求依頼を業務メインに渡す。例えば、派生起動のマクロを実行する。SAILは、普通出金FIFOの内容から派生編集定義を用いて派生メッセージ（取消再カットメイン）M2を生成する。このとき、業務メインは、図11に示す処理を呼び出す。図11に示すように、呼び出された派生要求共通処理は、普通出金FIFOの内容を派生管理DBに格納する。この例では、JOB番号が「000」であるため、図17に示すようにインデックス領域1のインデックス・ビットマップを読み取り先頭が「0」であったことから、派生管理通番を1とする。そして、このエントリー日付や時刻等と共に、図15に示す普通出金FIFO86Bの内容を派生管理DB70の派生管理通番1のセグメントに記録する。普通出金FIFOが記録された状態では、図17に示すように、インデックス・ビットマップの先頭の派生インデックスビットがON（1）であり、また、派生管理通番1の使用フラグが使用中（1）となる。

20

【0159】

図15に示すように、普通預金の出金に成功すると、連動戻り編集定義を用いて、連動戻りブロックに出金に成功した旨を示す結果コード（0000）を記録する。定期入金処理は、定期入金に成功したため、定期入金FIFOから出力編集定義を用いて出力メッセージを生成し、ATMに送信させる。これにより、JOBFP000のトランザクションT1はコミットする。

30

【0160】

IMSは、派生メッセージM2をリージョン1にスケジュールし、SAILは、派生メッセージから取消再カットFIFO86Dをセットする。取消再カットメインは、派生トランザクションであるため、業務メインは、まず図12に示す処理を行う。すなわち、派生特性管理共通処理として、派生管理通番をキー値とした派生管理DBの読み込みを行う。ここでは、派生管理通番1のセグメントを読み取る。すなわち、派生管理DB70に格納された普通出金FIFO86Bを読み出す。これにより、トランザクションT1のコミットを待機すると共に、T1のアボートの有無の確認を行う。普通出金を含むトランザクションT1が正常にコミットした後、ステップS75に示すように、適用業務プログラムを呼び出す。ここでは、派生メッセージのオペコードである05001（取消再カットメイン）を構成するプログラムを呼び出す。取消再カットメインは、取消の対象を特定し、その自振通番（123456）や、取消金額等を取消再カットFIFO86Dにセットし、派生起動要求依頼を行う。業務メインは、図11に示す派生要求共通処理を実行させる。このとき、取消再カットメインはJOB番号001で動作しているため、図17に示すように、派生管理通番は401となる。このため、FIFO86Dの派生管理通番の項目を1から401へと更新し、このFIFO86Dを派生管理通番401のセグメントに格納

40

50

する。そして、派生編集定義に従って、図16に示す派生メッセージM3を生成する。

【0161】

派生メッセージM3は自振（電気料引落）の取消であり、オペコードは02501である。トランザクションT3は連動取引を用いてこの自振取消を行い、連動受動業務である普通出金取消処理は派生メッセージM4を出力させる。このとき、トランザクションT3がリージョン0にスケジュールされたとする。

【0162】

図17に示す例では、トランザクションT2の派生要求共通処理が派生メッセージを出力させた時点では、トランザクションT1がコミットしている。これは、派生起動トランザクションT1は派生トランザクションT2のコミットを待機せずに早期にコミットすることで、ここでは定期入金処理の応答性を確保していることによる。一方、派生メッセージ（自振取消）M3は、トランザクションT2がコミットする前にスケジュールされる。このため、自振取消を主業務とするトランザクションT3はリージョン0にスケジュールされた。トランザクションT3では、まず、取消再カットメインであるトランザクションT2のコミットを待機するために、派生管理通番401の派生管理情報を読み出そうとする。

10

【0163】

トランザクションT2がコミットすると、図12に示すように、業務メインは、派生特性管理共通処理（S76，S77）をトランザクションT2の一部として実行する。さらに、S78及びS79もトランザクションT2の一部として実行する例では、インデックス領域1の先頭の派生インデックスビットがOFFに更新された後に、派生トランザクションT2がコミットする。この場合、トランザクションT3が派生トランザクションを起動する時には、すでに派生管理通番1が空いている。

20

【0164】

図17に示す例では、派生インデックスDBのOFF（対応する派生管理通番が使用可）への更新を別トランザクションとして起動する。この場合、ステップS77が終了した段階でトランザクションT2はコミットする。このとき、トランザクションT3は派生管理通番401の読み出しに成功し、さらに普通預金出金取消が完了すると、再度取消再カットメインを起動しようとする。このとき、トランザクションT2によって起動された派生インデックスビットのOFFへの更新処理（派生DB制御処理）が完了していないと、インデックス領域1の先頭の派生インデックスビットは1のままである。このため、次に空いているビット位置が2の派生インデックスビットを特定する。トランザクションT3からの派生起動情報は、普通出金取消FIFO等であり、これが派生管理通番2のセグメントに格納される。

30

【0165】

トランザクションT2によって起動された派生DB制御処理が当該インデックスビットを更新すると、インデックス領域1のインデックスビットの先頭ビットは0となる。このため、次にリージョン0で実行されるトランザクションが派生トランザクションを起動する場合には、派生管理通番1が利用される。このように派生管理通番を繰り返し利用するため、本実施形態では、派生管理DB及び派生インデックスDBの再編成処理が不要である。

40

【0166】

図15乃至図17に示す例では、各リージョンは、一般トランザクションと派生トランザクションの双方を実行するものとして説明したが、図18に示すように、JOB番号（リージョン）を一般トランザクションと派生トランザクションとに別々に割り当てるようにしても良い。図18に示す例では、全てのJOBが派生トランザクションによって使用されるという可能性がなくなるため、オンラインの応答性の悪化を防止することができる。そして、業務処理要求（メッセージ）の取り扱いを容易にするために、図18に示す例では、オンラインシステムが、待ち行列13として、オンライントランザクションやセンター一括処理の実行を要求するメッセージ（電文）を一時的に記憶する一般待ち行列13A

50

と、派生トランザクションの実行を要求する派生メッセージを一時的に記憶する派生待ち行列とを備えている。2種類の待ち行列を併用することで、トランザクションの総量制御が容易となる。

【0167】

＜総量制御＞

再度図7を参照すると、オンラインシステム（または、トランザクション制御システム）は、図18に示す一般待ち行列13A及び派生待ち行列13Bで滞留しているトランザクション数を監視する滞留監視部78と、この滞留監視部78によって監視される各待ち行列でのトランザクションの滞留数に応じて前記派生起動トランザクション又は前記派生トランザクションの起動を制御する総量制御部79とを備えている。

10

【0168】

図19は、滞留監視処理の一例を示すフローチャートである。滞留監視部78は、前記一般待ち行列又は派生待ち行列の滞留数が予め定められた第1レベル以上である場合には（ステップS81、S83）、派生起動取引中断ビットをONとする（ステップS82、S84）。図19に示す例では、それぞれの待ち行列13A、13Bに格納されたメッセージ数が200を越えた場合に、派生起動取引中断ビットをONとしている。そして、前記派生待ち行列に滞留しているトランザクション数が前記第1レベル（200）よりも多い数にて予め設定された第2レベル（500）以上である場合には（ステップS85）、派生中止ビットをONとする（ステップS86）。一方、上記各ビットがONとなった後に、それぞれの待ち行列の滞留数が0となった場合には（ステップS87）、各ビットをOFFとする（ステップS88）。この滞留監視部78による監視処理は、所定の監視時間隔で、例えば30秒ごとに繰り返し実行する。滞留監視部78は、派生起動取引中断ビットや、派生中止ビットを図7に示すシステム状況DB83に格納する。

20

【0169】

図20は、派生トランザクションの実行を待機させる処理の一例を示す図表である。総量制御部79は、前記派生起動取引中断ビットがONである場合には前記派生トランザクションを起動する業務処理要求の入力を中断させる派生起動取引入力中断制御機能28を備える。派生トランザクションを起動する業務処理要求としては、例えばセンター一括処理がある。派生起動取引入力中断制御機能28は、本実施形態では、例えば、センター一括処理を要求するセンターカット電文を入力するとき（例えば、SAILのEXIT処理時）、システム状況DB83に格納された派生起動取引中断ビットを参照し、派生起動取引中断ビットがONであると、センターカット電文の入力を中断する。従って、センター一括処理自体を待機させることでシステム負荷を軽減し、且つ、センター一括処理は取消再カットメインを派生トランザクションとして起動する可能性があるため、派生トランザクションの起動を抑制することができる。

30

【0170】

また、総量制御部79は、システム状況DB83に格納された派生中止ビットがONである場合に、派生起動要求があった時には、当該派生起動情報を前記派生管理DBに登録後、派生トランザクションの起動を行わない派生起動中止制御機能30を備える。図18に示す例では、派生起動トランザクションを管理する派生起動要求共通処理は、派生起動情報を登録後、派生中止ビットがONである場合には、派生メッセージを出力しない。すると、派生起動トランザクションがコミットし、派生トランザクションがアポートされたのと同様の状態となる。すなわち、派生起動中止制御機能30は、派生メッセージの出力処理時にシステム状況DB83の派生中止ビットを参照し、派生中止ビットがONである場合には付随処理である派生トランザクションを起動しない。

40

【0171】

また、総量制御部79は、前記派生中止ビットがONである場合に業務処理要求を受信した場合であって、当該業務処理要求に滞留時停止指定がある場合には、当該業務処理要求の処理を不能としてエラー出力する滞留時エラー制御機能31を備えるようにしても良い。図13を参照すると、動作環境指定として、オペコードに続いて滞留=N o、センター

50

一括=Y e s等の指定がある。この動作環境指定で滞留=Y e sとなっているオペコードの処理を要求されたときに、派生中止ビットがONとなっている場合には、滞留時エラー制御機能31を備える例では、当該メッセージを処理不能としてエラーを出力する。

【0172】

＜派生再起動＞

次に、派生トランザクションの再起動について説明する。派生トランザクションは、アボートされた場合や、派生トランザクション実行中にシステム障害が発生した場合や、派生トランザクションの起動が中止された場合に、事後的に再起動を行う。派生トランザクションの再起動の手順を明確にすることで、派生起動トランザクションと派生トランザクションとを利用した派生利用トランザクションの実際の原子性を保つことができる。

10

【0173】

再度図7を参照すると、演算手段4は、前記派生インデックスDB68及び派生管理DB70を参照して派生インデックスビット及び使用中フラグがONで且つ派生管理DBに派生起動情報を格納した後予め定められた時間以上経過している派生管理通番を派生再起動要と判定する派生再起動要否監視部76と、派生再起動要否監視部76によって再起動要と判定された派生管理通番を再起動する場合には、当該派生管理通番によって特定される派生起動情報に基づいて派生トランザクションを起動すると共に当該派生管理通番へ派生起動情報を格納した時刻を更新する派生再起動制御部77とを備えている。

【0174】

まず、アボートの態様を説明する。派生起動トランザクションがアボートした場合、派生起動情報が一旦格納されていたとしてもロールバックによりその更新が無効となるため、派生管理通番の使用フラグがオフとなる。このため、派生トランザクションは起動しない(図12, S74)。従って、派生起動トランザクションのアボートでは派生トランザクションの再起動が必要となることはない。

20

【0175】

派生トランザクションがアボートした場合、使用中フラグを使用可に更新していたとしても、この使用可への更新はロールバックにより取り消される。従って、派生トランザクションの原子性により、派生トランザクションがアボートした場合には使用中フラグが使用中となっていることが保証される。従って、派生再起動要否監視部76は、派生インデックスビット及び使用中フラグがONで且つ派生管理DBに派生起動情報を格納した後予め定められた時間以上経過している派生管理通番を派生再起動要と判定することができる。デッドロックの発見に処理開始後の経過時間による判定が極めて有効であるのと同様に、本実施形態では、派生トランザクションのアボートの判定に派生起動情報のエントリー後の経過時間を用いることで、確実に派生トランザクションのアボートを検索する。

30

【0176】

また、派生トランザクションの終了処理にて、派生インデックスビットを派生DB制御処理にて実行する場合、この派生DB制御処理がアボートしたときには、派生トランザクションによる使用中フラグの更新は成功し、すでにコミットしているため、使用中フラグがオフで、派生インデックスビットがONとなる。この場合、派生トランザクションの付随業務の内容は正常にコミットしているため、派生トランザクションを再起動する必要はない。

40

【0177】

派生再起動制御部77は、派生再起動要否監視部76によって再起動要と判定された派生管理通番を再起動する場合には、当該派生管理通番によって特定される派生起動情報に基づいて派生トランザクションを起動すると共に当該派生管理通番での派生起動情報を格納した時刻を更新する。本実施形態では、自動的な再起動を行うのではなく、アボートした派生トランザクションの存在をオペレータに表示し、オペレータからの再起動の要求を待つ再起動処理を行う。このとき、派生再起動制御部77は、派生トランザクションの再起動時にエントリー時刻を更新する。

【0178】

50

図21は、派生トランザクションの再実行を派生管理業務により行う構成を示すブロック図である。コンソール端末2Aから派生再実行要求90を受信すると、そのコマンドを解析し、特定の派生トランザクションを再起動するか、又は、アボートした全派生トランザクションを一括して再起動するかの判定を行う。派生管理通番で指定された一件の派生トランザクションを再起動する場合、派生管理業務92は、派生管理通番から派生管理DBに格納されているFIFOを特定し、このFIFOを派生管理業務92のFIFOへセットする。そして、この派生管理業務のFIFOに基づいて派生トランザクションの起動処理を行う。派生受動業務93として、派生トランザクションが再起動される。

【0179】

全トランザクションを再起動する場合、派生管理業務92は、予め定められた数の派生トランザクションを起動すると、図2(C)に示すような形式で、派生管理業務95を派生起動する。派生管理業務95も、予め定められた数の派生トランザクションを起動すると、派生管理業務を派生トランザクションとして起動する。各派生管理業務が予め定められた数の再起動しか行わないのは、1つのJOB番号から起動できる派生トランザクションの数が派生管理通番の数である図8乃至図10に示すM（例えば、400+予備）以下に制限されるためである。

【0180】

図22は、派生再起動要否監視部76による派生再起動要否判定処理の詳細構成を示すフローチャートである。この派生再起動要否判定処理は、アボートされ、且つ再起動が必要な派生トランザクションの有無を監視する処理である。この派生再起動要否判定処理は、コンソール端末2Aからのコマンド入力により実行する。また、一定時間間隔で自動的に起動するようにしても良い。コマンド入力により実行する場合、日付変更を行う前や、また、ある処理の終了時刻の前などに実行する。また、トランザクションがアボートされるとコンソール端末2Aにその旨が表示されるため、この表示があったときに実行するようにしても良い。

勘定系システムを例とすると、例えば他の金融機関への振込処理はある時間までに完了させなければならないため、この為替の中断を指示する為替中断コマンドの入力前に、この派生再起動要否判定処理を実行することでアボートされた派生トランザクションの有無の確認を行う。例えば、上述した承認通知処理は派生トランザクションを利用するが、この派生トランザクションがアボートされていないことを確認する。また、日付変更時には再起動が必要な派生トランザクション（付随処理）が一件もないことが必要であるため、この日付変更コマンドの入力前に派生再起動要否判定処理をコマンド入力により起動する。図22を参照すると、派生再起動要否監視部76は、まず、対象インデックス領域番号を0にセットする（ステップS91）。続いて、派生インデックスDB78のインデックス・ビットマップを「読み取りのみ」のモードで読み込む（ステップS92）。これは、通常の派生起動トランザクションとのインデックス・ビットマップに関するコンテンションの発生を防止するためである。続いて、インデックスビットの位置を先頭（1番目）にセットする（ステップS93）。そして、セットされた位置のビットを読み取り、この派生インデックスビットがON（使用中）であるか否かを判定する（ステップS94）。派生インデックスビットがONであれば、派生管理DBの内容を読み取る（ステップS95）。そして、派生起動情報のエントリー時刻を参照し、派生取引起動時刻から30秒以上経過しているか否かを判定する（ステップS96）。30秒以上経過している場合には、再派生不要ビットがONか、又は派生管理DBの使用フラグが使用可（0）であるか否かを確認する（ステップS97A）。

再派生不要ビットは、例えば端末折り返し連続処理などでONとされている。派生インデックスビットが使用中で、派生管理DBの使用フラグがOFFであるのは、派生トランザクションがコミットし、派生DB制御処理がアボートした場合であるため、派生トランザクションの再起動は不要である。ステップS97Aでノーであれば、その派生管理通番で起動された派生トランザクションについて再派生が必要と判定し、コンソール端末2Aに、派生管理通番や、起動業務の識別コードや、派生受動業務のオペコードや、派生起動

10

20

30

40

50

情報のエントリー日付及び時刻や、当初の業務処理要求の入力店番や端末番号や、さらに、派生取引対象口座の店番や口座番号等を出力する（ステップS97B）。この再派生関連情報を参照して、オペレータは、当該派生管理通番で管理される派生トランザクションを再起動するか否かを判定する。

【0181】

派生インデックスビットがOFFである場合や（ステップS94、ノー）、ONであったとしても派生取引起動時刻から30秒以上経過していない場合には（ステップS96、ノー）ステップS98A、S98Bを経由して次のインデックスビット（次の派生管理通番）の判定を行う。そして、派生取引起動時刻から30秒以上経過していても、再派生不要ビットがONであったり、また、派生トランザクションがコミットしたものの派生DB制御処理がアボートした場合（派生管理DB70の使用フラグが使用可）であるときも（ステップS97A、イエス）、次のインデックスビットの判定を行う。

10

ステップS98Aでは、インデックスビット位置が最終であるか否かを確認する。最終位置でなければ、ビット位置を1つずらして対応する次の派生管理通番を処理対象にセットし（ステップS98B）、ステップS94に処理を移す。一方、インデックスビットの位置が最終であれば、次のリージョンに対応するインデックス領域番号を判定対象とするため、対象インデックス領域番号に1を加算してセットし、ステップS92に処理を移す。一方、ステップS99Aにて、すべてのインデックス領域（リージョン）の探索が完了している場合には、この派生再起動要否判定処理を終了する。

【0182】

20

図23は、アボート又は起動中止された全ての派生トランザクションを一度のコマンドで順に再起動する処理例を示すフローチャートである。図23に示す例では、派生再起動制御部77は、まず、初回起動時に対象インデックス領域番号として0をセットする（ステップS101）。続いて、対象インデックス領域番号のインデックス・ビットマップのうち、使用中のビット位置を探索する（ステップS102）。使用中のビットがある場合には、その対象インデックス領域番号とビット位置とから派生管理通番を算出し、派生管理DBに格納された内容を読み込む（ステップS103）。そして、使用中フラグがON（使用中）で、且つ派生トランザクションの起動時刻（派生起動情報のエントリー時刻）から5分以上経過しているか否かを判定する（ステップS104）。5分以上の経過がなければ、次に使用中である派生インデックスビットを探索する（ステップS102）。

30

【0183】

派生管理DBの使用フラグが「使用中」で、且つ派生起動情報のエントリーから5分以上経過している場合には、派生管理DB70に格納されている派生起動情報に基づいて派生トランザクションの起動を制御する（ステップS105）。具体的には、派生管理DB70に格納されたFIFOをセットし、派生管理DB70に格納された派生編集番号に応じて当該FIFOから派生メッセージを生成する。派生起動制御を行うと、再派生処理回数をインクリメントする（ステップS106）。そして、派生取引起動時刻を現時刻に変更して派生管理DBを更新する。続いて、次の使用中のインデックスビットを探索する（ステップS102）。対象インデックス領域番号のインデックス・ビットマップを全て確認した場合には（ステップS102）、次の派生インデックスビットについて検討するため、対象インデックス領域番号に1を加算してセットする（ステップS108）。このとき、再派生処理回数（起動した派生トランザクションの数）が一定数（例えば、80）を越えている場合には、再派生処理回数を0にクリアして、派生再起動処理を派生トランザクションとして起動する（ステップS111）。再起動された派生再起動処理では、初回起動時ではないためステップS101をパスし、セットされたインデックス領域番号のインデックス・ビットマップの派生トランザクションの再起動を検討する。

40

【0184】

ステップS108にて、再派生処理回数が一定数を越えていない場合には、全てのインデックス領域番号（インデックス・ビットマップ）の探索が完了したか否かを確認し（ステップS110）、未検討のインデックス・ビットマップが残っている場合には、対象イン

50

デックス領域番号に1を加算して、ステップS102に処理を移す。これにより、全てのインデックス・ビットマップについてアボートされた派生トランザクションの有無を探索することができる。

【0185】

上述したように第2実施形態によると、派生インデックスDBをJOB番号毎にアクセス可能とし、さらに、派生起動情報を派生管理DBに格納するため、並列に処理されているトランザクション間のコンテンション防止し、デッドロックの発生を防止し、さらに派生管理DBの再編成不要としつつ、一般的なDBのロック機構を用いて派生トランザクションの特性を管理することができる。

【0186】

10

【第3実施形態】

図12に示す派生特性管理共通処理にて、ステップS79を派生トランザクション内の処理とすると、デッドロックが発生する可能性がある。このデッドロックは、プログラムの構築の仕方によって回避することができるが、その回避のために適用業務プログラムのプログラマが留意すべき事項は、複雑である。このため、第3実施形態では、第2実施形態を拡張するものとして、派生DBを用いつつもデッドロックの発生を回避することができる仕組みを開示する。具体的には、ステップS79を派生トランザクションとは別のトランザクション（派生DB制御処理）として実行するものである。

【0187】

図24は、本実施形態の構成を示すブロック図である。図24に示すオンラインシステムは、ATM等の端末1A及び営業店端末1Bからネットワークを介して入力される業務処理要求を受信すると共に、コンソール端末2Aから入力される業務処理要求（コマンド）を受信する受信手段2と、この受信手段2にて受信した業務処理要求をオンライントランザクションとして処理する演算手段4と、この演算手段4による業務処理要求の結果を記録するファイル6とを備えている。

20

【0188】

そして、ファイル6は、前記オンライントランザクション又は所定の派生トランザクションである派生起動トランザクションの実行の一部として起動される派生トランザクションの派生起動情報を派生管理通番毎に記録すると共に当該派生管理通番が使用中であるか否かを示す使用中フラグを有する派生管理DB70と、この派生管理DB68での派生管理通番を前記トランザクションのJOB番号を単位とした複数の派生管理通番から使用可の派生管理通番を特定すると共に前記派生管理通番が使用中であるか否かを示す派生インデックスビットを前記JOB番号毎に複数有する派生インデックスDB68とを備えている。この派生インデックスDB及び派生管理DB70の構成及び役割については、上記第2実施形態と同様である。従って、好ましい例ではJOB番号からインデックス・ビットマップを特定し、このインデックス・ビットマップを参照して使用可の派生管理通番（派生インデックスビットのビット位置）を特定する。

30

【0189】

図24に示す例では、演算手段4が、前記オンライントランザクション又は派生トランザクションの種類を特定するオペコードに応じて当該トランザクションを構成するプログラムの実行を管理する業務メイン処理部100と、この業務メイン処理部100によって呼び出されたプログラムを実行する業務処理部110とを備えている。ここでは、演算手段4の構成を、実際のプログラムの配分に従って、各プログラムをトランザクションとして管理する業務メイン処理部100と、実際にプログラムを実行する業務処理部110とに機能分けをする。

40

【0190】

演算手段4では、IMS上でFP及びSAILが各種のサービスを提供しているとする。図24に示すメッセージ入力手段8（例えば、SAILの入力インタフェース）は、業務処理要求（入力メッセージ）を受信すると、入力メッセージ上の業務識別コード、媒体タイプ、入力編集番号をもとに入力編集定義テーブルを特定する。本実施形態では、業務識

50

別コードと入力編集番号とはオペコードから変換する。入力インタフェースは、入力編集定義テーブルに従って、メッセージ上のデータを指定された編集を行い、編集後のデータを指定されたF I F Oのフィールドへセットする。この編集の完了後、S A I Lは業務識別コードに対応した適用業務プログラムを呼び出す。本実施形態では、全業務に共通の業務メインと呼ばれるプログラムを呼び出す。

【0191】

業務メイン処理部100は、図25に示すように、業務メインの(1)開始処理を実行する。そして、例えば図13に示すようなオペコードテーブルに基づいて適用業務プログラムを特定し、呼び出す。図25に示す例では、業務メインは、派生起動ランザクションの一部となる派生起動A P P Lを呼び出す。すると、業務処理部110は、この派生起動A P P Lを実行する。この実行中に、業務処理部110は、派生起動要求依頼(マクロを用いる場合には、派生要求マクロ)を業務メインに入力する。

10

【0192】

業務メインは、派生起動要求依頼を受信した場合には、派生要求共通処理を呼び出す。すなわち、業務メイン処理部100は、派生起動ランザクションの実行によって派生起動要求依頼を受信した場合には派生要求共通処理を当該派生起動ランザクションの一部として実行する派生要求共通処理呼び出し機能102を備えている。派生要求共通処理は、例えば図11に示す処理である。業務処理部110は、この派生要求共通処理を実行する派生要求共通処理機能112を備える。派生要求共通処理機能112は、例えば、C P Uが、図25に示す派生要求共通処理指令112Aを実行することで実現する。

20

【0193】

派生要求共通処理機能112は、図25に示すように、派生要求共通処理が呼び出されて処理を開始すると(1)、当該派生起動A P P L111Aがスケジュールされているリージョンでの当該ランザクションのJ O B番号を特定し(2)、前記派生インデックスD Bを参照する(3)。この参照により、前記派生管理D Bの派生管理通番を特定して派生インデックスビットをO N(使用中)にする(4)。そして、ビット位置から派生管理通番を算出し、これを当該リージョン内のF I F Oにセットする(6)。そして、派生インデックスD B68の残り個数及び派生件数を更新する(7)。さらに、派生管理通番のセグメントに派生起動情報を格納し、使用中フラグを使用中に更新する(8)。そして、派生要求共通処理は、システム状況D B83に格納された派生中止ビットを参照し、派生中止ビットがO F FであればS A I L(O Sの一部)に派生起動要求(派生要求マクロ)を入力する(9)。その後、業務メインに戻る(10)。派生中止ビットがO Nであれば、派生要求マクロを入力せず、派生起動ランザクションをコミットさせる。

30

【0194】

派生要求マクロが出力されると、S A I Lの連動インタフェースは、起動業務識別コード・プログラムで指定した受動業務識別コード・編集番号から、派生編集定義テーブルを特定する。本実施形態では、受動業務識別コードと派生編集番号は派生受動処理(派生ランザクション)のオペコードから変換する。連動インタフェースは、派生編集定義テーブルに従って、指定された派生起動業務側F I F O上のデータを、指定された編集を行って派生メッセージへセットする。連動インタフェースは、派生メッセージを受信すると、入力メッセージ上の起動業務識別コード・受動業務識別コード・派生編集番号をもとに派生編集定義テーブルを特定する。連動インタフェースは、派生編集定義テーブルに従って、派生メッセージ上のデータを、指定された編集を行った後指定されたF I F Oのフィールドへセットする。この編集完了後、S A I Lは業務メインを呼び出す。これにより、派生ランザクションの処理が開始される。呼び出された業務メインは、図25に示す例では、派生特性管理共通処理指令104Aを実行する。

40

【0195】

図25に示す本実施形態では、「派生特性管理共通処理104A」は、例えば図12に示す処理のうち、派生インデックスビットを使用可に更新する処理を除いた部分である。図24及び図25に示す例では、業務メイン処理部100が派生特性管理共通処理104A

50

を実行する派生特性管理共通処理機能104を備える。

【0196】

派生特性管理共通処理機能104は、図25に示すように、派生特性管理共通処理104Aを開始すると(1)、まず、派生管理通番をキー値として派生管理DBの読み込みを試みることで(2)、派生起動トランザクションとの直列性を保つ。派生管理情報の読み込みに成功すると、使用中フラグをチェックする。使用中フラグがOFFであれば、派生起動トランザクションがアボートされたとして、派生受動APPLを呼び出さずに業務メインを終了する(4)。一方、この派生管理通番の使用フラグが使用中である場合には、当該派生トランザクションのオペコードに応じた派生受動業務プログラムを呼び出して実行させる(4)。

10

【0197】

業務処理部110は、派生受動APPLを実行し、終了すると、業務メインに処理を戻す。派生特性管理共通処理は、派生受動APPLが終了した後に、前記派生管理DBの使用フラグをオフに更新して(6)、さらに所定の派生DB制御処理を派生トランザクションとして起動させる(7)。SAILは、派生トランザクションのFIFOを参照して所定の編集定義に基づいて派生メッセージを出力する。派生メッセージがスケジュールされると、派生DB制御処理が実行される。

【0198】

前記業務処理部110は、派生インデックスビットを使用可へと更新する派生DB制御処理を派生トランザクションとはさらに別のトランザクションとして実行する派生DB制御処理機能114を備えている。派生DB制御処理機能114は、CPUが派生DB制御処理指令114Aを実行することで実現する。派生DB制御処理114Aでは、まず、派生管理通番から派生インデックスDB68のインデックス領域番号とビット位置とを算出する(2)。そして、算出したインデックス領域番号をキー値として派生インデックスDB68のインデックス・ビットマップを読み込む(3)。続いて、派生管理通番をキー値として派生管理DB70を読み込む(4)。この派生管理DBの読み込みにより、派生特性管理共通処理112Aとの直列性が維持される。そして、派生管理DBの使用フラグがOFFであるか否かを確認し、OFFの場合に派生インデックスビットをOFF(使用可)にして派生インデックスDBを更新する。

20

【0199】

図26は、派生トランザクションの一部として、派生トランザクションの完了時に派生インデックスDBの派生インデックスビットをOFFとした場合に、生じる可能性があるデッドロックの一例を示す説明図である。すなわち、派生インデックスDB68を用いて派生管理通番を特定する派生DB制御処理を派生トランザクションと別のトランザクションとしない場合に生じる可能性のあるデッドロックの態様の一例を示す図である。

30

【0200】

デッドロックは、2つのトランザクションが相互に相手方の処理の完了を待機する場合に生じる。例えば、トランザクションT1がDBxを読み込み、これをロックした後、次に派生インデックスDB68を読み込もうとする。一方、トランザクションT2が派生インデックスDB68を読み込み、これをロックした後、次にDBxを読み込もうとしたとする。トランザクションT1が派生インデックスDB68を読み込もうとしたときに、この派生インデックスDB68がトランザクションT2によってロックされている場合に、トランザクションT1とT2はデッドロックとなる。

40

【0201】

図26に示す例では、まず、JOBFP000から派生起動要求が出力され、この派生トランザクションT2はJOBFP001にスケジュールされた。トランザクションT2では、別のオペコードに連動するが、連動先ではあるDBxをキーyで読み込む。トランザクションT2は連動処理が完了してから、派生トランザクションの後処理(派生特性管理共通処理)として、派生管理DBの使用フラグをOFFとし、派生インデックスDBの派生インデックスビットをOFFとする。このとき、派生インデックスDBのキー値は、

50

インデックス領域番号であるから、トランザクション1のJOB番号である0がキー値となる。派生管理DBのキー値（派生管理通番）は、ここでは1とする。

【0202】

トランザクションT2の処理中に、トランザクションT3がJOBFP000でスケジュールされた。トランザクションT3では、別オペコードに連動するが、連動先では派生トランザクションを起動するため、派生インデックスDBをキー値を0として読み、派生管理DBの派生管理通番1のセグメントに派生起動情報を格納する。トランザクションT3では、連動処理が完了してから、DBxを読み込む。

【0203】

さて、トランザクションT2では、DBxを読み取った後、派生インデックスDBをキー値0として読み取ろうとする。トランザクション3では、派生インデックスDBを読み取った後、DBxを読みに行く。このため、トランザクションT2にて連動受動処理がDBxを読み込んでロックした後、派生インデックスDBを読み取る前のタイミングで、トランザクションT3が派生インデックスDBを読み取ると、デッドロックとなる。また、トランザクションT3が派生インデックスDBを読み取った後、連動処理が完了してDBxを読み取る前のタイミングで、トランザクションT2がDBxを読み取ると、デッドロックが生じる。

10

【0204】

デッドロックの発生を完全に防止する手法は、並列処理を行わないことであるが、これは採用できない。ロックに関しては、並列処理の直列化を維持するために、あるトランザクションは、ロックを取得するフェーズと、ロックを解放するフェーズとの2相ロックングプロトコルに従う必要がある。従って、一度ロックを解放すると、そのトランザクションはロックを取得してはならない。この2相ロックングプロトコルの要件によると、デッドロックが生じた場合には一方のトランザクションをアボートしなければならない。

20

【0205】

デッドロックを回避しつつ、並列性も維持する方法は、トランザクションのロックの順序を調整することである。例えば、全てのトランザクションが同一の順序でロックを要求すれば、デッドロックを回避することができる。しかし、これはプログラマにとって厳しい制限となる。また、トランザクションが使用する全てのロックをその開始時に取得してから、実際の処理を行うことによって、デッドロックの発生を回避することができるが、この場合、限りなく直列処理に近くなり、並列処理の利点が減少する。

30

【0206】

これらの標準的な手法でデッドロックを回避するためには、「派生トランザクションを起動する可能性がある連動受動処理に連動する前に、連動後に読むDBを連動前に読んでおく」必要がある。これにより、実行順序が同一となる。全ての連動起動処理がこの条件を満たすのであれば、図26に示すデッドロックの発生を回避することができる。しかしながら、連動起動業務の開発者が、このDBの読み込み順序を守るよう意識するのは、負担が大きく、困難である。

【0207】

このため、第3実施形態では、派生DB制御処理として、派生インデックスDBの派生インデックスビットをOFFにする処理を別トランザクションとしている。このため、派生特性管理共通処理では、派生インデックスDBの派生インデックスビットの更新を行わない。従って、1つのトランザクション内でDBxのロックと派生インデックスDBのロックとを行うことがなくなる。派生DB制御処理は、派生DB（派生管理DBと派生インデックスDB）以外のDBを読み込まないため、デッドロックは生じない。

40

【0208】

再度図25を参照して、本実施形態での派生トランザクション管理方法を説明する。派生トランザクション管理方法は、派生起動トランザクション（派生起動APPL111A）を実行すると共に、当該派生起動トランザクションによって起動要求依頼される派生トランザクション（派生受動APPL113A）のコミットを待機せずに当該派生起動トラン

50

ザクションをコミットさせる派生起動トランザクション管理工程（100A, 112A）と、派生起動APPL111Aによって起動要求依頼された派生トランザクションを管理する派生トランザクション管理工程（派生特性管理共通処理104A）と、この派生トランザクション管理工程にて派生トランザクションとして起動される派生DB管理工程（派生DB制御処理114A）とを備えている。

【0209】

前記派生起動トランザクション管理工程は、当該派生起動トランザクションの一部として、当該派生起動トランザクションのJOB番号に応じて前記派生インデックスDBを参照し、前記派生管理DBの派生管理通番を特定し、当該派生管理通番のセグメントに派生起動情報を格納し、当該使用した派生管理通番に対応する前記インデックスビット及び使用中フラグをそれぞれ使用中に更新して、当該派生トランザクションを起動させる派生要求共通処理工程（派生要求共通処理112A）を備えている。

10

【0210】

そして、派生トランザクション管理工程は、前記派生トランザクションの一部として、派生要求共通処理工程112Aにて派生トランザクションが起動されスケジュールされた場合に、当該派生トランザクションに与えられた派生管理通番をキー値として派生管理DBを読み込み、当該派生管理通番の使用フラグが使用中である場合には当該派生トランザクションのオペコードに応じた派生受動業務プログラムを実行させ、この派生受動業務プログラムが終了した後に前記派生管理DBの使用フラグをオフに更新して、さらに所定の派生DB制御処理を派生トランザクションとして起動させる派生特性管理共通処理工程（派生特性管理共通処理指令104A）を備えている。この派生特性管理共通処理工程にて、インデックス・ビットマップの更新を行わないため、図26に示すデッドロックが生じない。

20

【0211】

そして、派生DB管理工程は、前記派生DB制御処理がスケジュールされた場合に、前記派生受動業務プログラムを含む派生トランザクションを再起動するための派生起動情報を格納された派生管理DBの派生管理通番に対応する前記派生インデックスDBの前記派生インデックスビットを使用可に更新する派生DB制御処理工程（派生DB制御処理指令114A）を備える。派生DB制御処理工程では、1つのとして、インデックス・ビットマップの更新のみを行うため、他のトランザクションのロックの待機を行うことはあっても、デッドロックを生じることはない。また、派生DB制御処理による派生DB制御トランザクションは、図25に示す例では、派生特性管理共通処理104Aによって指定された派生管理通番にて派生管理DB70を読み取るため、派生特性管理共通処理104Aとの直列性を有する。すなわち、派生特性管理共通処理が正常にコミットした後に、インデックス・ビットマップを更新する。従って、並列処理によってシステム的な一貫性が失われることはない。

30

【0212】

また、図24に示すように、第3実施形態においても総量制御部14の各構成を備えるようにしても良い。第3実施形態では、システム負荷が大きい場合には、派生DB制御処理の起動を中止し、滞留の解消後に一括して派生インデックスビットをオフにする処理を行うようにしても良い。

40

【0213】

第3実施形態では、アボートされた派生トランザクションの再起動は、第2実施形態と同様である。業務メイン100A, 派生起動APPLがアボートした場合には、使用中フラグ、派生インデックスビット共に使用可のままであり、派生トランザクションの起動要求もなされない。派生要求共通処理112Aにて派生DB68, 70の更新後、派生起動要求の出力前にアボートされた場合には、使用中フラグ及び派生インデックスビットは更新されるものの、ロールバックによってこの更新がなかったものとなる。そして、派生トランザクションもスケジュールされない。

【0214】

50

派生起動要求の出力後に派生要求共通処理 1 1 2 A や、業務メイン 1 0 0 A や、派生起動 A P P L 1 1 1 A が連動受動業務の場合で連動戻り後の連動起動業務の処理がアボートした場合にも、ロールバックによって派生要求共通処理の D B への更新が無かったものとなる。しかし、派生メッセージはスケジュールされている可能性がある。この場合、派生特性管理共通処理 1 0 4 A (3) にて使用中フラグをチェックする。使用中フラグを読み込めた時には派生起動トランザクションはアボートしている。このため、ロールバックによって更新されない状態の使用中フラグを読み取ることとなる。従って、派生特性管理共通処理 1 0 4 A (3) では、使用中フラグが「使用可」であるため、派生受動 A P P L を呼び出さずに処理を終了する。この場合、派生起動トランザクションも派生トランザクションも無かったこととなる。

10

【 0 2 1 5 】

派生起動トランザクションがコミットすると、使用中フラグが「使用中」の状態に派生管理 D B に耐久性を持って記録される。派生特性管理共通処理 1 0 4 A の (1) から (5) までと、派生受動 A P P L のアボートの場合、派生管理 D B の使用中フラグは使用中のままであり、派生インデックス D B も使用中のままである。派生受動 A P P L が完了後、派生特性管理共通処理が使用中フラグをオフにした後にアボートした場合や、さらに派生トランザクションを起動するための派生要求共通処理 1 1 2 A 等の処理中にアボートした場合には、ロールバックによって派生特性管理共通処理の更新が無かったものとなる。従って、使用中フラグは使用中 (1) のままである。

【 0 2 1 6 】

20

派生特性管理共通処理がコミットした後、派生 D B 制御処理がアボートしたとする。この場合、使用中フラグはオフとなっている。しかし、派生インデックス D B の派生インデックスビットは O N のままである。このように、 (1) . 派生起動トランザクションのアボートの場合には、派生トランザクションは起動しない。 (2) . 派生トランザクションがコミットしたが、派生受動 A P P L 1 1 3 A や派生特性管理共通処理 1 0 4 A がアボートした場合には、使用中フラグ及び派生インデックスビットが共に使用中のままである。 (3) . 派生トランザクションもコミットしたが、派生 D B 制御処理 1 1 4 A がアボートした場合には、使用中フラグが使用可で、派生インデックスビットが O N の状態である。

【 0 2 1 7 】

派生トランザクションの再起動が必要となるのは、 (2) . の場合である。従って、使用中フラグが O N で、起動と一定時間が経過している処理を探索することで、再起動の要否を判定すべき派生トランザクションを特定することができる。 (3) . の派生 D B 制御処理 1 1 4 A のアボートの場合には、派生受動 A P P L 及び派生特性管理共通処理 1 0 4 A は正常にコミットしているため、派生トランザクションを再起動する必要はない。

30

【 0 2 1 8 】

上述したように本実施形態によると、派生と連動とが複雑に絡み合った状態で生じるデッドロックを派生 D B 制御処理によって回避することができるため、開発者がプログラム作成上の調整を行う必要がなくなる。

【 0 2 1 9 】**【発明の効果】**

40

本発明は、その構成によって、オンライン応答が必要なオンライントランザクションと、オンライントランザクション等に付随した処理を実行する派生トランザクションとに区分けされて実行される。このため、オンライン応答に必要な処理をオンライントランザクションとして処理し、オンライン応答に必要なない付随処理を派生トランザクションとして オンライントランザクションのコミット後に処理する。従って、オンライントランザクションを早期にコミットさせることができる。これにより、オンラインの応答性を確保しつつ、付随的な業務のみを事後的に派生トランザクションとして処理することができる。従って、従来バッチ処理で行っていたような処理について派生トランザクションを利用してオンライントランザクションと並行した処理が可能となる。このように、本発明によると、派生トランザクションモデルを用いることで、従来バッチ処理等で実現していた処理

50

や、1つのトランザクションとしては処理量が多くなる処理を実行しつつ、オンラインの応答性を確保することができる、というトランザクション制御システムを実現することができる。

【図面の簡単な説明】

【図1】図1は本実施形態によるオンライントランザクション制御システムの構成を示すブロック図である。

【図2】図2は、派生トランザクションの性質を説明する説明図であり、図2(A)は、派生起動トランザクションから派生トランザクションの起動要求がなされる例を示す図で、図2(B)は、派生起動トランザクションが連動トランザクションである場合を示す図で、図2(C)は、派生トランザクションが自らを派生トランザクションとして起動要求する例を示す図である。

10

【図3】図3は派生トランザクションの利用法に応じた端末応答等を説明するための説明図であり、図3(A)は、付随業務の場合を示す図で、図3(B)は取消再カットメインの場合を示す図である。

【図4】図4は、取消再カットメインの動作例を示す説明図である。

【図5】図5は端末折り返し連続処理に派生トランザクションを用いた例を示す説明図である。

【図6】図6は、トランザクションの総量制御の一例を示すフローチャートである。

【図7】図7は、第2実施形態の構成を示すブロック図である。

【図8】図8は、JOBと派生インデックスDB等との関係を示す説明図である。

20

【図9】図9は、派生DBのデータ項目を例示する説明図であり、図9(A)は派生インデックスDBの項目例を示す図で、図9(B)は派生管理DBの項目例を示す図である。

【図10】図10は、JOB IDと、派生インデックスDBのインデックス領域番号と、ビット位置と、派生管理通番との関係を例示する説明図であり、図10(A)はJOB ID 0000で実行されているトランザクションによって読み出される派生インデックスDBとそのビット位置及び派生管理通番の関係を例示する図で、図10(B)はこの場合の派生管理DBの内容の例を示す図で、図10(C)はJOB ID 0001で実行されているトランザクションによって読み出される派生インデックスDBとそのビット位置及び派生管理通番の関係を例示する図で、図10(D)はこの場合の派生管理DBの内容の例を示す図である。

30

【図11】図11は、派生トランザクションの要求を管理するための派生要求管理処理の一例を示すフローチャートである。

【図12】図12は、派生トランザクションの特性を管理するための派生特性管理処理の一例を示すフローチャートである。

【図13】図13は、オペコードテーブルの一例を示す図で、ここでは普通預金のオペコードテーブルを例示する。

【図14】図14は、取消再カットメインでの処理例を示す説明図であり、図14(A)は口座処理による電気料引落(内部自振)の取消を行う例を示す図で、図14(B)はその処理による残高の推移を示す図である。

【図15】図15は、図14に示した処理でのFIFO推移及び派生メッセージの例を示す説明図である。

40

【図16】図16は、図15に続くFIFO等の推移を示す説明図である。

【図17】図17は、図14及び図15に示す例での派生DBの利用例を示す説明図である。

【図18】図18は、JOBを一般トランザクションと派生トランザクションとに別々に割り当てる例を示す説明図である。

【図19】図19は、滞留監視処理の一例を示すフローチャートである。

【図20】図20は、派生トランザクションの実行を待機させる総量制御処理の例を示す図表である。

【図21】図21は、アボートされた派生トランザクションの再起動処理の一例を示すブ

50

ロック図である。

【図 2 2】図 2 2 は、派生トランザクションの再起動の可否を判定する処理の一例を示すフローチャートである。

【図 2 3】図 2 3 は、派生トランザクションの再起動処理の一例を示すフローチャートである。

【図 2 4】図 2 4 は、第 3 実施形態の構成を示すブロック図である、

【図 2 5】図 2 5 は、派生管理処理の構成を示す説明図である。

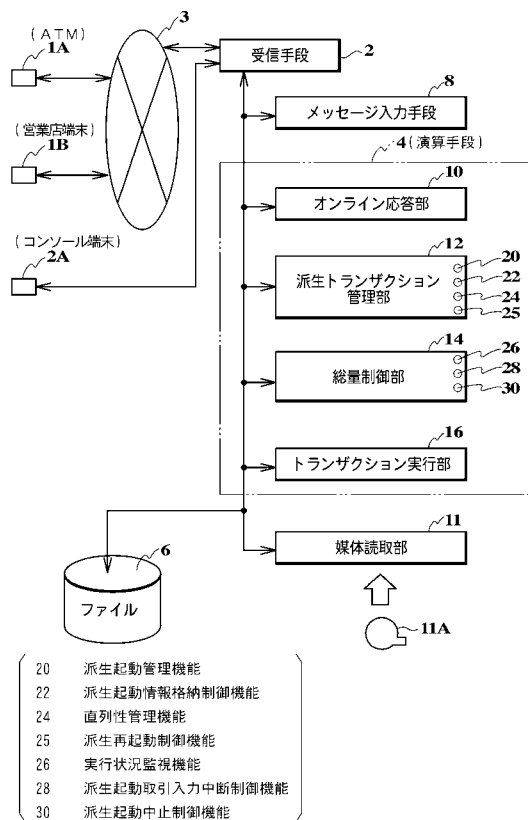
【図 2 6】図 2 6 は、派生 DB 制御処理を別トランザクションとしない場合に生じる可能性のあるデッドロックの態様の一例を示す説明図である。

【符号の説明】

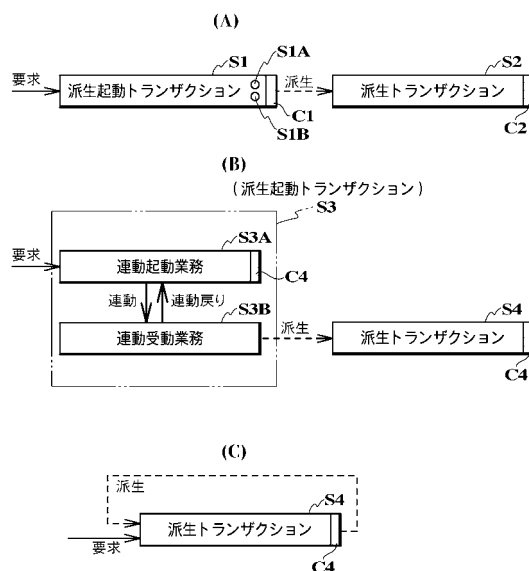
- 2 受信手段
- 4 演算手段
- 6 ファイル
- 8 メッセージ入力手段
- 10 オンライン応答部
- 12 派生トランザクション管理部
- 14 総量制御部
- 16 トランザクション実行部

10

【図 1】

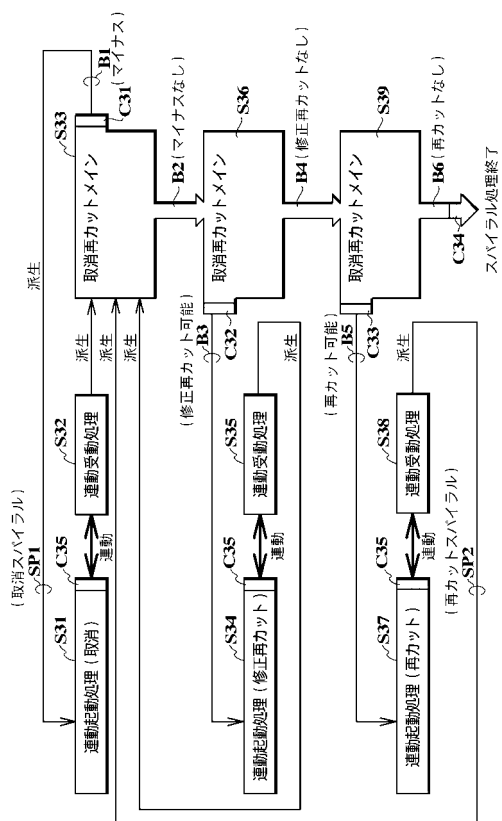


【図 2】

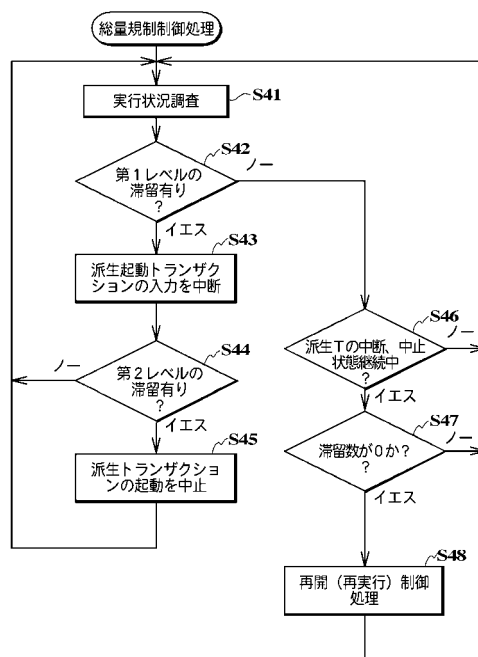


- S1A オンライントランザクション実行
- S1B 派生起動情報格納

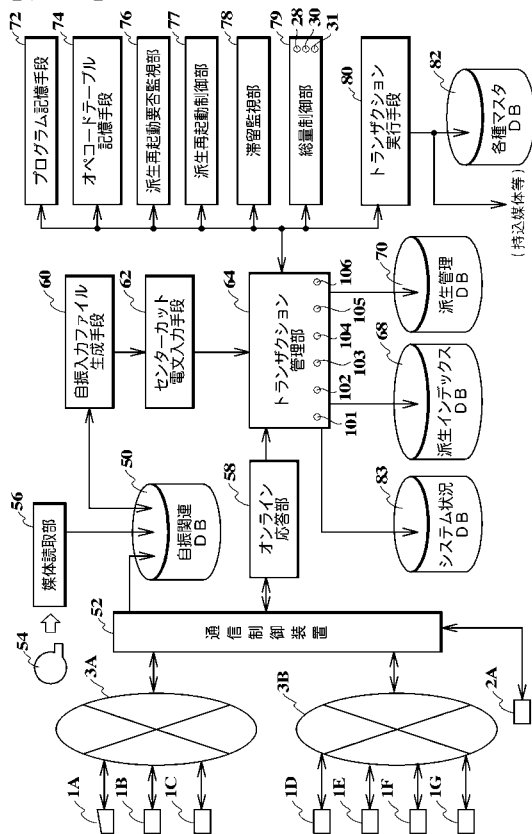
【図 4】



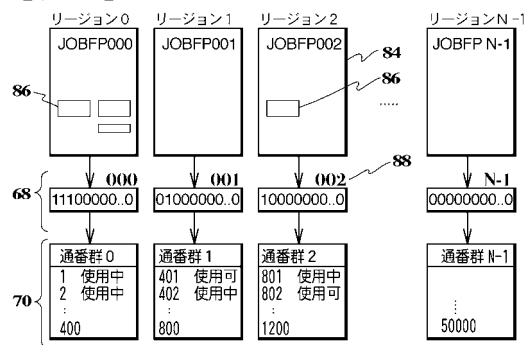
【图 6】



【図 7】



【図 8】



【図 9】

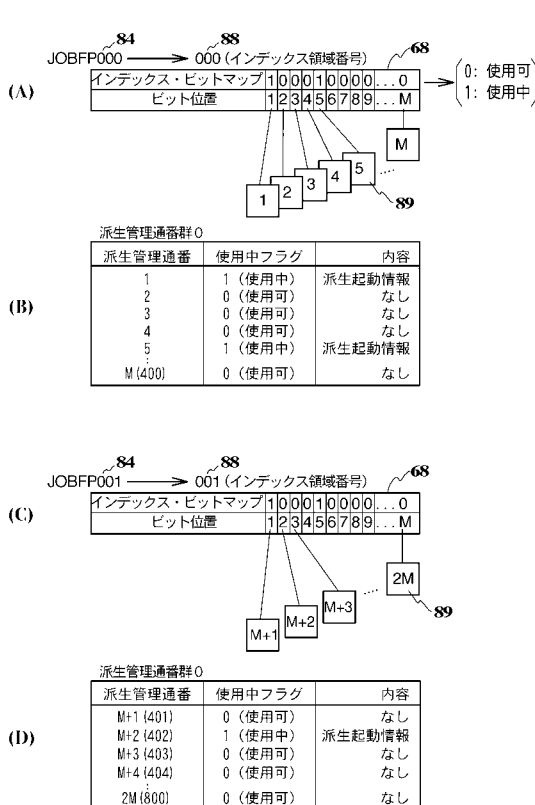
派生インデックス DB

名称	長さ	属性	初期値
インデックス領域番号 (キー値)	1	2進数	0~N-1
(A) 残り件数	4	バック10進数	M
派生件数	4	バック10進数	0
インデックス・ビットマップ	50	ビット	00000000

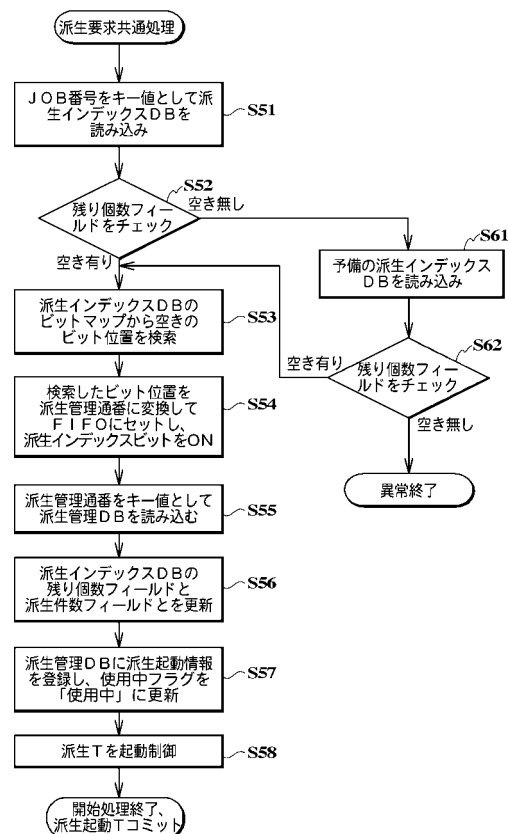
派生管理 DB

名称	長さ	属性	初期値
派生管理通番 (キー値)	4	バック10進数	1~NM
派生起動 APPL ID	4	キャラクター	ブランク
派生変動 APPL ID	4	キャラクター	ブランク
派生編集番号	4	キャラクター	ブランク
エントリー日付	4	yyyymmdd	ブランク
エントリー時刻	4	hhmmssst	ブランク
使用中フラグ	1	キャラクター	0
FIFO保存	3072	キャラクター	ブランク

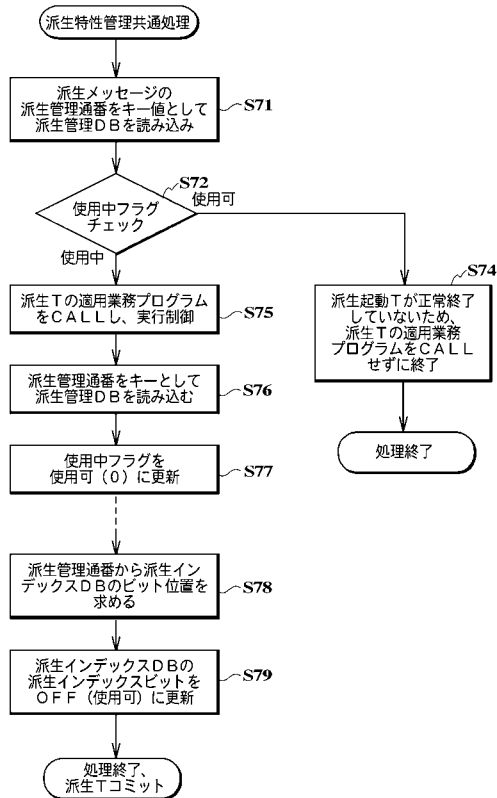
【図 10】



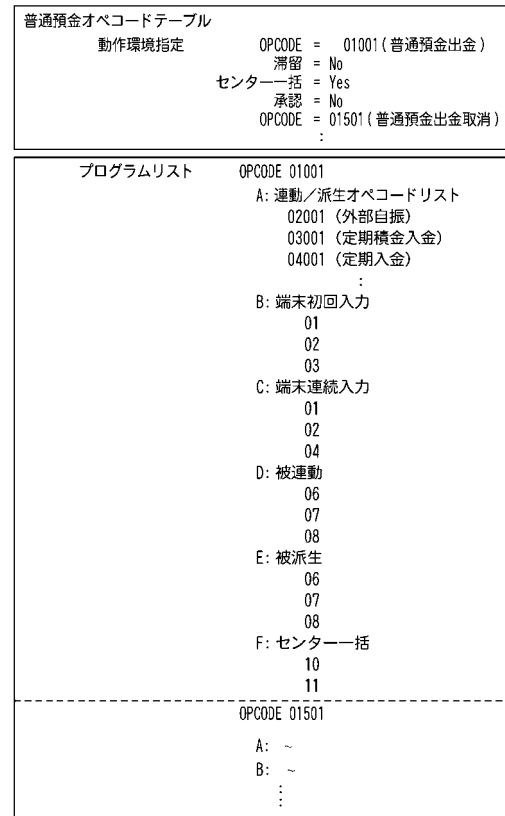
【図 11】



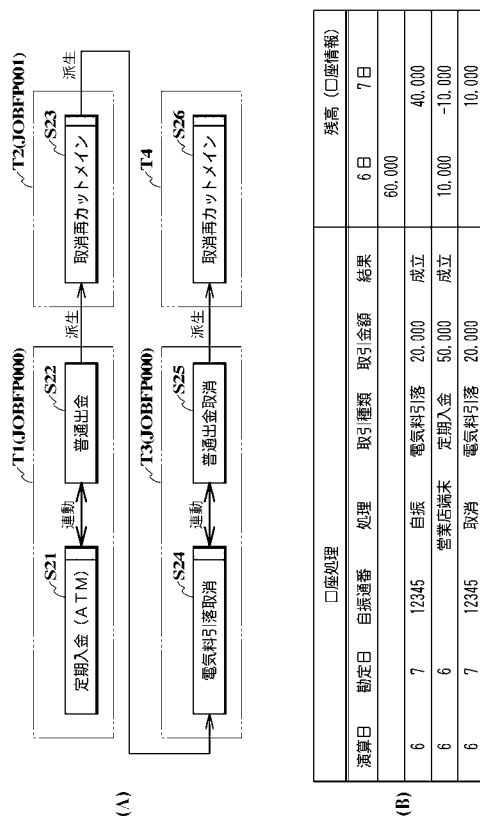
【図 1 2】



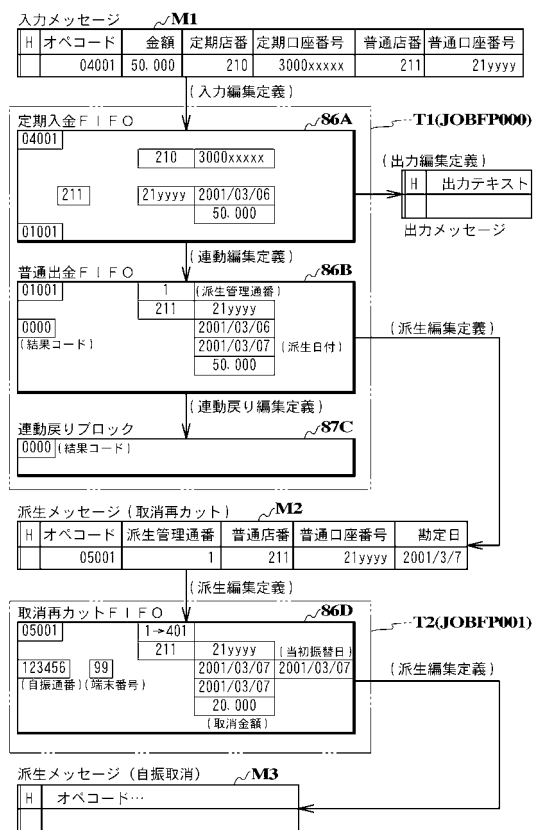
【図 1 3】



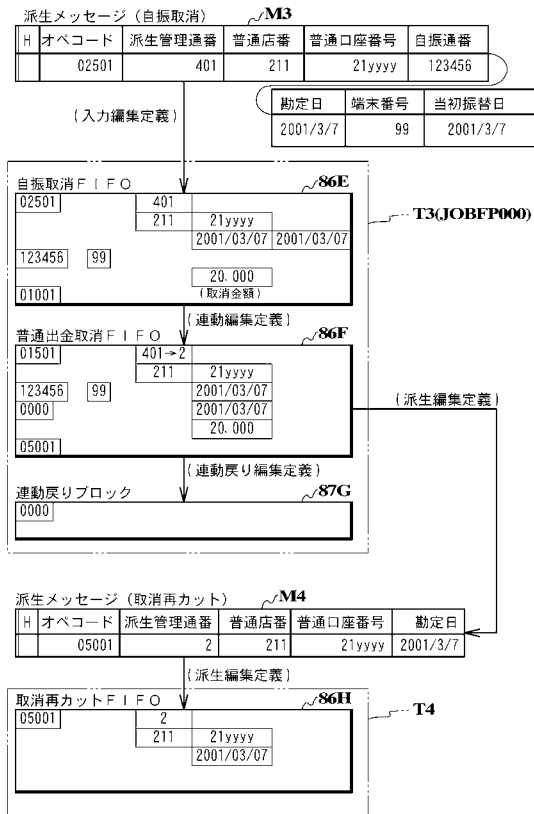
【図 1 4】



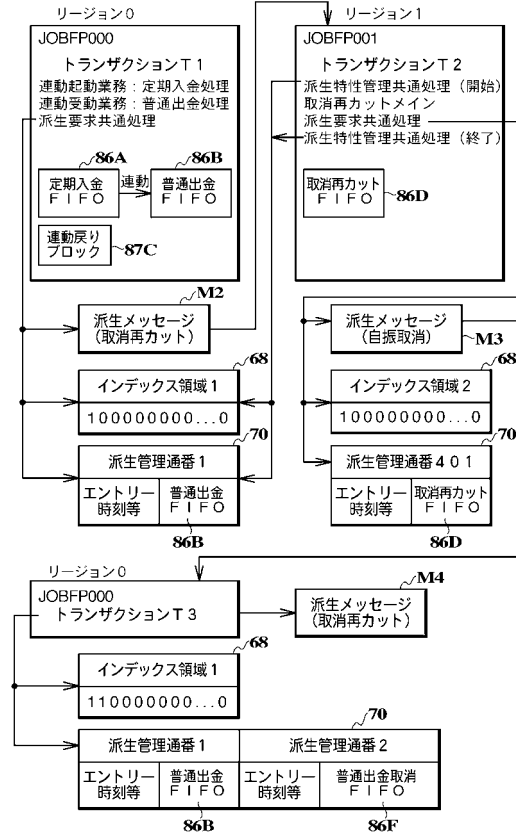
【図 1 5】



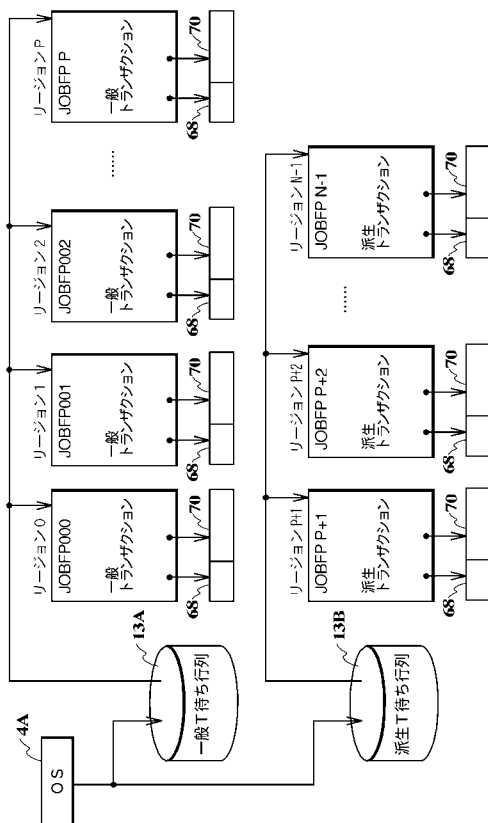
【図 16】



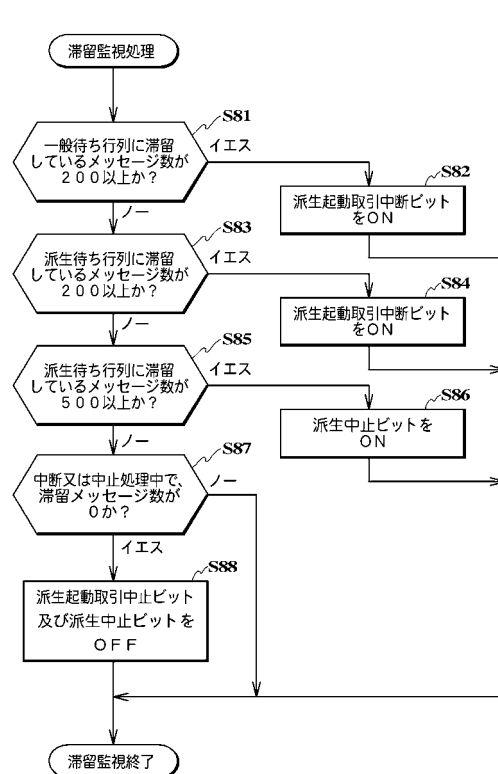
【図 17】



【図 18】



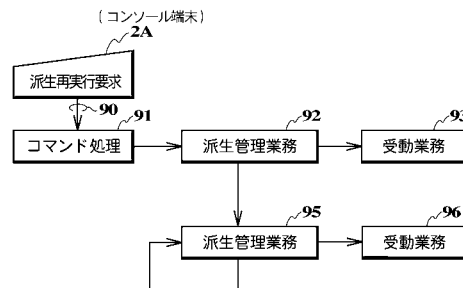
【図 19】



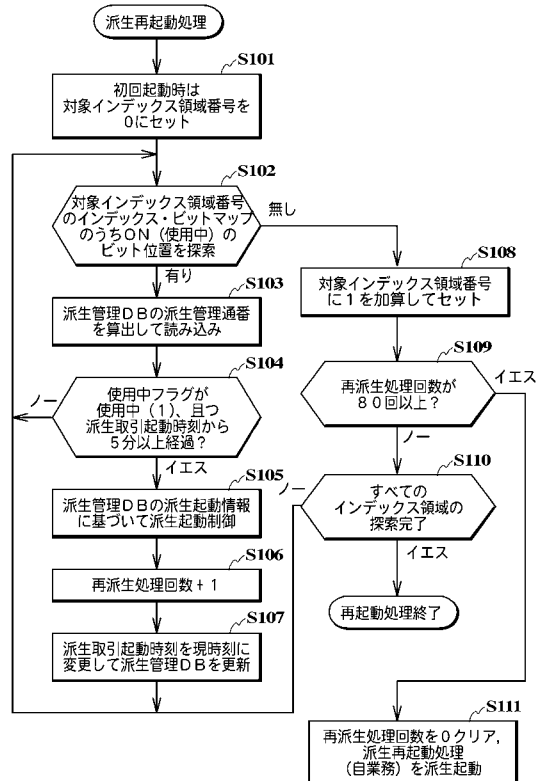
【図 20】

判定タイミング	条件	中断、中止処理
センターカット電文の入力時	派生起動取引中断ビットがON	センターカット電文の入力中断
派生起動情報を派生管理DBへ登録後	派生中止ビットがON	派生管理DBへ登録後、派生メッセージを出力せず
メッセージ受信時	派生中止ビットがON、且つ、オペコードの滞留時停止指定がYes	処理不能エラー

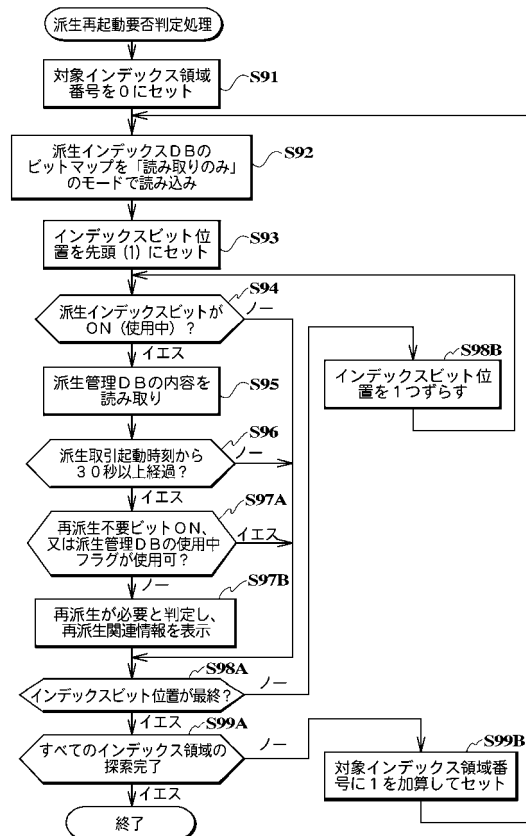
【図 21】



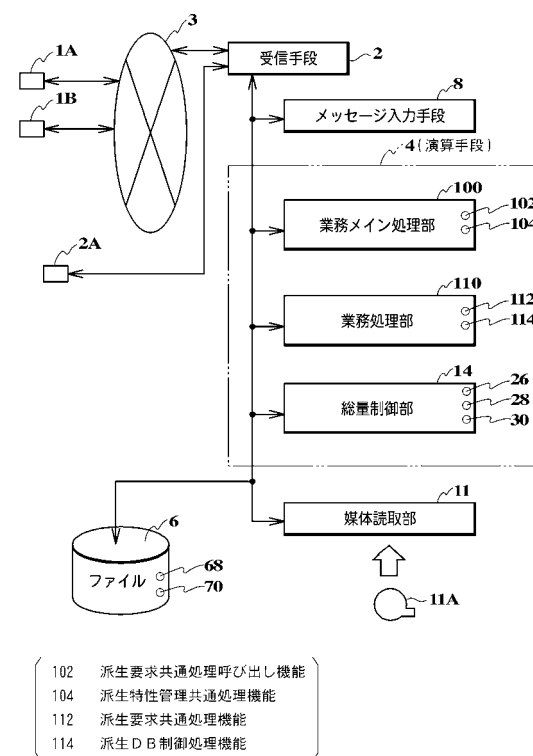
【図 23】



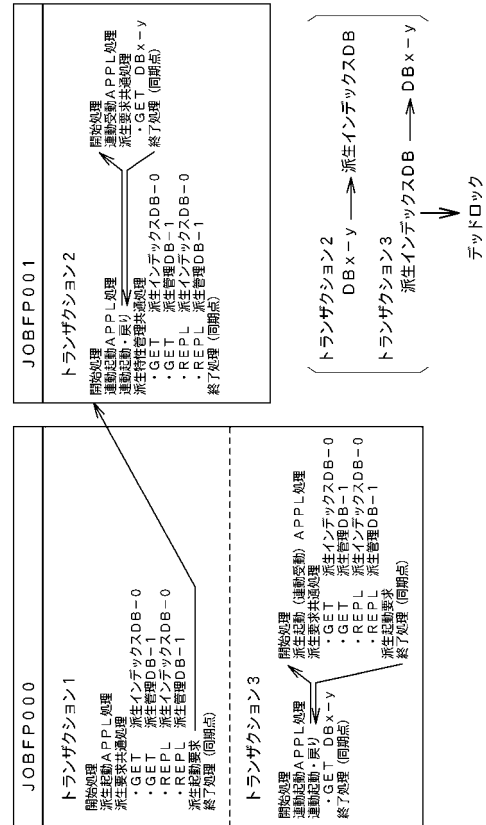
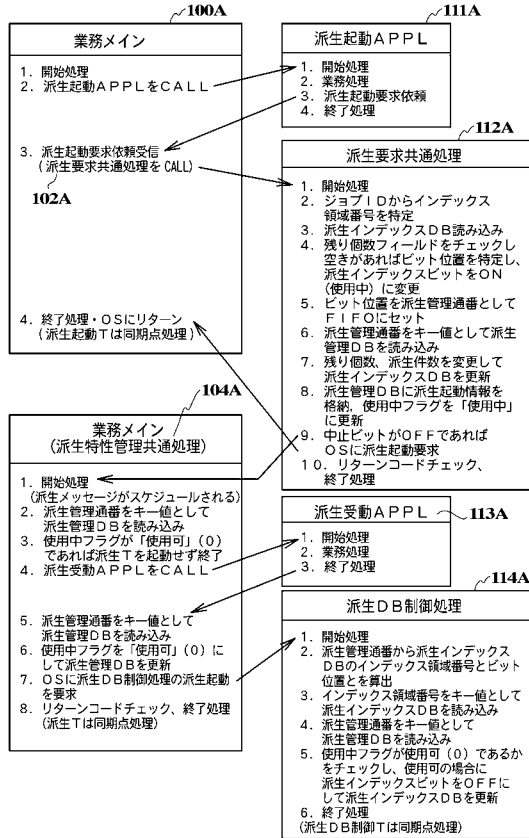
【図 22】



【図 24】



【 図 2 6 】



フロントページの続き

- (56)参考文献 特開平02-212962 (JP, A)
特表2002-536732 (JP, A)
特開平10-069418 (JP, A)
特開平02-118841 (JP, A)
特開2002-056182 (JP, A)
特開2002-279191 (JP, A)
特開平08-212163 (JP, A)
小島功, アクティブデータベースの動向と応用へのインパクトについて, 情報処理学会研究報告 (94-SE-101), 日本, 社団法人情報処理学会, 1994年11月18日, Vol. 94, No. 99, p. 73-78
ACOSソフトウェア トランザクション処理管理 TPS-EX システム利用の手引, 日本, 日本電気株式会社, 1996年12月31日, p. 20-21, 47-48
ジム・グレイ, トランザクション処理 [上], 日本, 日経BP社, 2001年10月29日, p. 405-408
Jim Gray, TRANSACTION PROCESSING, 米国, MORGAN KAUFMANN PUBLISHERS, 1993年, P.337-338

- (58)調査した分野(Int.Cl., DB名)

G06F 12/00

G06F 15/00