(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2010-86109
(P2010-86109A)
(43)公開日　平成22年4月15日(2010.4.15)

(51)Int.Cl.		F I		テーマコード（参考）
G06F 11/28	(2006.01)	G06F 11/28	340A	5B042

審査請求 未請求　請求項の数 2　OL　（全 21 頁）

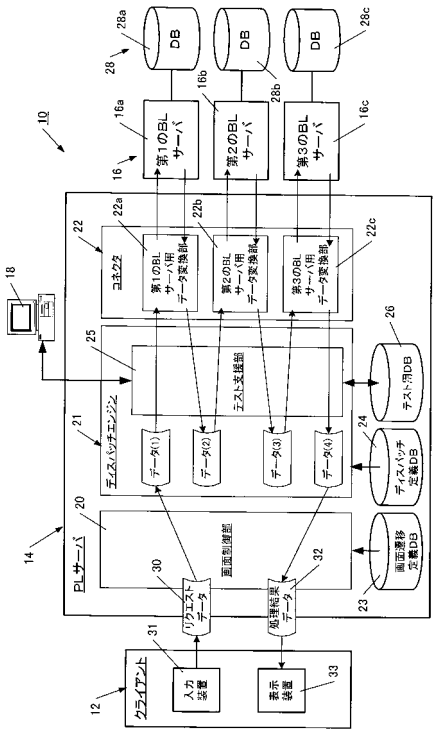
(21)出願番号　特願2008-252054 (P2008-252054)
(22)出願日　平成20年9月30日 (2008.9.30)

(71)出願人　000155469
株式会社野村総合研究所
東京都千代田区丸の内一丁目６番５号
(74)代理人　100096002
弁理士　奥田　弘之
(74)代理人　100091650
弁理士　奥田　規之
(72)発明者　高畠　勇人
東京都千代田区丸の内一丁目６番５号　株式会社野村総合研究所内
(72)発明者　溝口　拓治
東京都千代田区丸の内一丁目６番５号　株式会社野村総合研究所内

最終頁に続く

(54)【発明の名称】テスト支援システム

(57)【要約】　（修正有）
【課題】BLサーバのトランザクション単位でテストの再ランを可能とする。
【解決手段】PLサーバ14は、テストの初回ラン時に、クライアント12からのリクエストデータ30を受け付け、対応するトランザクション及び担当BLサーバ16を特定し、入力データを担当のBLサーバ16に送信し、トランザクション及びBLサーバ16の特定情報と入力データをテスト用ＤＢ26に格納する。BLサーバ16から送信された処理結果データをテスト用ＤＢ26に格納し、最終的な処理結果データをクライアント12に送信する。テストの再ラン時には、トランザクション及びBLサーバ16を特定した再ラン指示データを受け付け、対応の入力データをテスト用ＤＢ26から取り出し、担当BLサーバ16に送信する。BLサーバ16から送信された処理結果データを再ラン結果データとしてテスト用ＤＢ26に格納する。
【選択図】図１



【特許請求の範囲】
【請求項１】
クライアントと、業務処理を実行するバックエンドサーバと、上記クライアントからのリクエストデータを受け取り、担当のバックエンドサーバに必要な入力データを送信して処理を依頼し、その処理結果データをクライアントに送信するフロントエンドサーバとを備えたオンラインシステムにおいて、
上記フロントエンドサーバが、オンラインテストの初回ラン時に、
(a)クライアントからのリクエストデータを受け付ける処理と、
(b)このリクエストデータに対応する１または複数のトランザクション及び担当のバックエンドサーバを特定する処理と、
(c)上記リクエストデータに含まれた入力データを担当のバックエンドサーバに送信する処理と、
(d)この入力データを、トランザクション及び担当のバックエンドサーバを特定する情報と関連付けてテスト用データベースに格納する処理と、
(e)上記バックエンドサーバからトランザクションの処理結果データが送信された場合に、これをトランザクション及び担当のバックエンドサーバを特定する情報と関連付けて上記テスト用データベースに格納する処理と、
(f)上記バックエンドサーバからの処理結果データを入力データとして、同一または他のバックエンドサーバに次の処理を依頼する必要がある場合に、上記処理結果データに対応した入力データを担当のバックエンドサーバに送信する処理と、
(g)この入力データを、トランザクション及び担当のバックエンドサーバを特定する情報と関連付けて上記テスト用データベースに格納する処理と、
(h)上記バックエンドサーバからトランザクションの処理結果データが送信された場合に、これをトランザクション及び担当のバックエンドサーバを特定する情報と関連付けて上記テスト用データベースに格納する処理と、
(i)上記(f)～(h)の処理を必要回数繰り返し、最終的な処理結果データが得られた時点で、これをリクエストの処理結果データとして上記クライアントに送信する処理を実行し、
同フロントエンドサーバが、オンラインテストの再ラン時には、
１または複数のトランザクション及び担当のバックエンドサーバを特定した再ラン指示データを受け付ける処理と、
対応の入力データを上記テスト用データベースから取り出し、担当のバックエンドサーバに送信する処理と、
当該バックエンドサーバからトランザクションの処理結果データが送信された場合に、この再ラン結果データを、トランザクション及び担当のバックエンドサーバを特定する情報と関連付けて上記テスト用データベースに格納する処理を実行することを特徴とするテスト支援システム。
【請求項２】
上記フロントエンドサーバが、上記テスト用データベースに格納された初回ラン時の処理結果データと再ラン結果データをトランザクション単位で比較し、その結果を出力する機能を備えたことを特徴とする請求項１に記載のテスト支援システム。
【発明の詳細な説明】
【技術分野】
【０００１】
この発明はテスト支援システムに係り、特に、複数のクライアントとサーバを通信ネットワークで接続したオンラインシステムの全体動作を検証するオンラインテストを支援する技術に関する。
【背景技術】
【０００２】
複数のクライアントとサーバを通信ネットワークで接続したオンラインシステムを通じて、クライアントに対して様々なサービスを提供する機会が増えてきている。

図１８は、このようなオンラインシステムの一例を示すシステム構成図であり、オンラインシステム80は、Webブラウザプログラムを搭載したクライアント12と、PL（Presentation Logic）サーバ14と、複数のBL（Business Logic）サーバ16（第１のBLサーバ16a～第３のBLサーバ16c）を備えている。クライアント12－PLサーバ14間、及びPLサーバ14－BLサーバ16間は、通信ネットワークを介して接続されている。

【０００３】

PLサーバ14は、クライアント12からのリクエストデータ30を受け付けた後、担当のBLサーバ16に入力データを送信して処理を依頼すると共に、BLサーバ16から返された処理結果データ32をクライアントに送信するフロントエンドサーバとして機能する。

このためPLサーバ14は、画面制御部20と、ディスパッチエンジン21と、コネクタ22を備えている。

【０００４】

画面制御部20は、画面遷移定義ＤＢ23を参照し、クライアント12に対して送信する画面（Htmlファイル）を生成する機能を果たす。

またディスパッチエンジン21は、ディスパッチ定義ＤＢ24を参照し、リクエストデータ30に対応したトランザクション及び担当BLサーバを特定する機能を果たす。

またコネクタ22は、各BLサーバ16が要求する固有の形式に入力データを変換すると共に、BLサーバ16から送信された処理結果データをPLサーバ14の標準形式に変換する機能を発揮する。このため、BLサーバ16毎に対応のデータ変換部（第１のBLサーバ用データ変換部22a～第３のBLサーバ用データ変換部22c）が設けられている。

【０００５】

各BLサーバ16は、PLサーバ14から受け取った入力データに基づき、直接または図示しないDBサーバを通じて間接的に管理しているデータベース28に対する検索、更新等の業務処理を実行するバックエンドサーバとして機能する。

【０００６】

図１８では、クライアント12から送信されたリクエストデータ30に含まれるデータ(1)を第１のBLサーバ16aに送信して所定のトランザクション処理を依頼し、その処理結果データであるデータ(2)を第２のBLサーバ16bに送信して所定のトランザクション処理を依頼し、その処理結果データであるデータ(3)を第３のBLサーバ16cに送信して所定のトランザクション処理を依頼し、その処理結果データであるデータ(4)を最終的な処理結果データ32としてクライアント12に送信するパターンが例示されている。

【０００７】

このようなオンラインシステム80を構築するに際しては、各サーバに搭載された個々のアプリケーションプログラムの単体テストや結合テストが必要であることは勿論であるが、サービスのリリース前にはクライアント12と各サーバを通信ネットワークを介して接続した状態で、オンラインシステム全体の動作を検証する必要がある。

また、既存のオンラインシステムに新たな機能を追加したり、バグの修正を行った場合には、他の既存機能に悪影響（デグレード）が生じていないか等を確認する必要性が生じる。

【０００８】

この所謂オンラインテストに際しては、予め準備された複数のシナリオに従ってオペレータがクライアント12から実際の入力を行い、システムによって提供される各種機能が正常に動作するか否かが検証される。そして、何らかの不具合が生じた場合には、少し前の地点に戻って入力をやり直し、問題箇所や原因を突き止める作業が繰り返される。

【０００９】

大規模なオンラインシステムともなれば、多数のサーバの連携によって処理が実行されるため、必然的にテスト項目が多岐に亘り、また不具合が生じる可能性も増大する。この結果、全項目についての確認を完了するためには、多数のオペレータが何度も同じ入力を繰り返す必要が生じる。

このため、非特許文献１及び２に示すように、オペレータによる入力作業を軽減するた

めのツールが幾つか提案されている。
【非特許文献１】QuickTest ProfessionalインターネットURL:http://www.ashisuto.co.jp/prod/qtp/sum/index.html検索日：平成２０年６月４日
【非特許文献２】Seleniumとはインターネットurl:http://www.thinkit.co.jp/free/article/0705/2/1/検索日：平成２０年６月４日
【発明の開示】
【発明が解決しようとする課題】
【００１０】
この種の入力支援ツールは、オペレータによる入力装置31からの入力操作と、表示装置33における表示画面を入力支援用ＤＢ82に逐一記録する機能を備えているため、オペレータが再入力することなく、同じリクエストデータ30をサーバ側に何度でも再投入することができると共に、新たな表示画面と前回の表示画面を比較し、両者が一致するか否かを判定することが可能となる。
また、各BLサーバ16が管理するデータベース28のバックアップを最初に１回だけとっておけば、これをベースにして任意の地点までの入力を再現することにより、当該地点におけるデータベース28の状態を復元することが可能となる。
【００１１】
しかしながら、この従来の入力支援ツールはあくまでもクライアント12毎に導入されるアプリケーションプログラムであり、監視の対象となるデータもクライアント12から送信されたリクエストデータ30と、クライアント12が受信した処理結果データ32に限定されているという問題があった。
【００１２】
実際のところ、クライアント12から１つのリクエストデータを受け付けた場合であっても、PLサーバ14を介して複数のBLサーバ16が連携して動作し、最終的な処理結果データがもたらされることが多いにかかわらず、上記の入力支援ツールで捕捉できるのは図１８におけるリクエストデータ30と処理結果データ32に限られ、PLサーバ14内においてトランザクション単位で生起したデータ(1)～(4)のレベルでデータを保存することは出来なかった。
【００１３】
このため、従来の入力支援ツールは、テストの再実施もリクエスト単位に限定され、各BLサーバ16によるトランザクション単位でテストを再実行することもできなかった。この結果、図１９に示すように、あるデータベース28の状態がトランザクションα～γによって実際には（イ）→（ロ）→（ハ）→（ニ）と順に更新されていく場合であっても、従来の入力支援ツールで再現されるのは（イ）→（ニ）の更新のみであり、途中の（ロ）及び（ハ）の状態を復元することは不可能であった。
【００１４】
この発明は従来の上記問題を解決するために案出されたものであり、バックエンドサーバのトランザクション単位で入力データ及び処理結果データの保存が可能であり、その結果、トランザクション単位でのテストの再実行及びデータベースの復元を可能とする技術の実現を目的としている。
【課題を解決するための手段】
【００１５】
上記の目的を達成するため、請求項１に記載したテスト支援システムは、クライアントと、業務処理を実行するバックエンドサーバと、上記クライアントからのリクエストデータを受け取り、担当のバックエンドサーバに必要な入力データを送信して処理を依頼し、その処理結果データをクライアントに送信するフロントエンドサーバとを備えたオンラインシステムにおいて、上記フロントエンドサーバが、オンラインテストの初回ラン時に、(a)クライアントからのリクエストデータを受け付ける処理と、(b)このリクエストデータに対応する１または複数のトランザクション及び担当のバックエンドサーバを特定する処理と、(c)上記リクエストデータに含まれた入力データを担当のバックエンドサーバに送

信する処理と、(d)この入力データを、トランザクション及び担当のバックエンドサーバを特定する情報と関連付けてテスト用データベースに格納する処理と、(e)上記バックエンドサーバからトランザクションの処理結果データが送信された場合に、これをトランザクション及び担当のバックエンドサーバを特定する情報と関連付けて上記テスト用データベースに格納する処理と、(f)上記バックエンドサーバからの処理結果データを入力データとして、同一または他のバックエンドサーバに次の処理を依頼する必要がある場合に、上記処理結果データに対応した入力データを担当のバックエンドサーバに送信する処理と、(g)この入力データを、トランザクション及び担当のバックエンドサーバを特定する情報と関連付けて上記テスト用データベースに格納する処理と、(h)上記バックエンドサーバからトランザクションの処理結果データが送信された場合に、これをトランザクション及び担当のバックエンドサーバを特定する情報と関連付けて上記テスト用データベースに格納する処理と、(i)上記(f)～(h)の処理を必要回数繰り返し、最終的な処理結果データが得られた時点で、これをリクエストの処理結果データとして上記クライアントに送信する処理を実行し、同フロントエンドサーバが、オンラインテストの再ラン時には、１または複数のトランザクション及び担当のバックエンドサーバを特定した再ラン指示データを受け付ける処理と、対応の入力データを上記テスト用データベースから取り出し、担当のバックエンドサーバに送信する処理と、当該バックエンドサーバからトランザクションの処理結果データが送信された場合に、この再ラン結果データを、トランザクション及び担当のバックエンドサーバを特定する情報と関連付けて上記テスト用データベースに格納する処理を実行することを特徴としている。

上記の「初回ラン」とは、クライアントから入力されたリクエストデータに基づいて実施される第１回目のオンラインテストを意味している。これに対し「再ラン」とは、データベースに蓄積された初回ラン時のデータに基づいて再演される２回目以降のオンラインテストを意味している。

【００１６】

また、請求項２に記載したテスト支援システムは、請求項１のシステムであって、さらに上記フロントエンドサーバが、上記テスト用データベースに格納された初回ラン時の処理結果データと再ラン結果データをトランザクション単位で比較し、その結果を出力する機能を備えたことを特徴としている。

【発明の効果】

【００１７】

この発明に係るテスト支援システムにあっては、オンラインテストの初回ラン時に、フロントエンドサーバによって、各バックエンドサーバへの入力データ及び出力データが、担当のバックエンドサーバ及びトランザクションに関連付けて逐一テスト用データベースに保存される仕組みであるため、オンラインテストの再実施時にはトランザクション単位で特定のバックエンドサーバを動作させることが可能となり、よりきめ細かくオンラインシステムの正常／異常を検証することができる。

この結果、各バックエンドサーバが管理するデータベースの状態も、トランザクション単位で復元することが可能となる。

【発明を実施するための最良の形態】

【００１８】

図１は、この発明に係るテスト支援システムを含むオンラインシステム10を示しており、Webブラウザプログラムを搭載したクライアント12と、PLサーバ14と、複数のBLサーバ16（第１のBLサーバ16a、第２のBLサーバ16b、第３のBLサーバ16c）と、テスト端末18を備えている。クライアント12－PLサーバ14間、PLサーバ14－BLサーバ16a～16c間、PLサーバ14－テスト端末18間は、通信ネットワークを介して接続されている。

【００１９】

PLサーバ14は、クライアント12からのリクエストデータ30を受け付け、当該リクエストを実現するのに必要なトランザクション及び担当のBLサーバ16を特定した後、担当のBLサーバ16に入力データ（引数）を送信して処理を依頼すると共に、BLサーバ16から返された

最終的な処理結果データ32を含む画面をクライアント12に送信する、フロントエンドサーバとして機能する。

このためPLサーバ14は、画面制御部20と、ディスパッチエンジン21と、コネクタ22を備えている。

【００２０】

画面制御部20は、画面遷移定義ＤＢ23を参照し、クライアント12に対して送信する画面（Htmlファイル）を生成する機能を主として果たす。

【００２１】

ディスパッチエンジン21は、ディスパッチ定義ＤＢ24を参照し、クライアント12からのリクエストに対応したトランザクション及び担当のBLサーバ16を特定する機能を果たす。

また、ディスパッチエンジン21内には、テスト支援部25が生成されており、PLサーバ14のディスク内に設けられたテスト用ＤＢ26を管理している。このテスト支援部25の機能、及びその生成方法については後述する。

【００２２】

コネクタ22は、各BLサーバ16が要求する固有のフォーマットに入力データを変換すると共に、BLサーバ16から送信された処理結果データをPLサーバ14の標準形式に変換する機能を発揮する。このため、BLサーバ16a～16c毎に複数のデータ変換部（第１のBLサーバ用データ変換部22a～第３のBLサーバ用データ変換部22c）が設けられている。

例えば、第１のBLサーバ16aがＣ言語に対応したTPモニタを搭載している場合、第１のBLサーバ用データ変換部22aは、TPモニタ向けのデータ変換機能を備える。また、第２のBLサーバがEJB（Enterprise Java Beans）コンテナを搭載している場合、第２のBLサーバ用データ変換部22bは、EJBコンテナ向けのデータ変換機能を備える。また、第３のBLサーバ16cがCOBOL言語に対応したDB/BCシステムを搭載している場合、第３のBLサーバ用データ変換部22cは、ホストOS向けのデータ変換機能を備える。

【００２３】

BLサーバ16a～16cは、PLサーバ14から受け取った入力データに基づき、直接または図示しないDBサーバを通じて間接的に管理しているデータベース28に対する検索、更新等の業務処理を実行するバックエンドサーバとして機能する。

【００２４】

つぎに、図２のフローチャートに従い、このオンラインシステム10に対する初回ラン時の処理手順を説明する。

初回ラン時には、まずクライアント12からリクエストデータ30が送信される。例えば、オペレータがテストシナリオに従い、検索画面の入力欄に入力装置31を介して検索文字列を入力すると、クライアント12からPLサーバ14に対して、検索のリクエストID及び検索文字列を含んだリクエストデータ30が送信される。

【００２５】

このリクエストデータ30を受信した画面制御部20は（Ｓ10）、当該リクエストデータ30に含まれているリクエストIDを取得する（Ｓ11）。つぎに画面制御部20は、画面遷移定義ＤＢ23を参照し、当該リクエストIDに対応したPLトランザクションIDを特定する（Ｓ12）。このPLトランザクションIDは、ディスパッチエンジン21に渡される。

【００２６】

画面制御部20からPLトランザクションIDを受け取ったディスパッチエンジン21は、このPLトランザクションIDをキーにディスパッチ定義ＤＢ24を検索し、当該PLトランザクションIDに関連付けられたBEトランザクションIDを特定する（Ｓ13）。

【００２７】

BEトランザクションIDは、特定のBLサーバに対する特定の処理（トランザクション）を規定するものであり、PLトランザクションIDには１または複数のBEトランザクションIDが処理の順番通りに関連付けられている。このため、ディスパッチエンジン21はPLトランザクションIDに基づいて複数のBEトランザクションIDを特定した時点で、今回のリクエストに応えるために必要なトランザクション、それぞれの担当BLサーバ16、トランザクション

の順番等を認識することができる。

【００２８】

ここでは、クライアント12から送信された検索文字列であるデータ(1)を第１のBLサーバ16aに送信して所定の検索処理を依頼し、その処理結果データであるデータ(2)を第２のBLサーバ16bに送信して所定の検索処理を依頼し、その処理結果データであるデータ(3)を第３のBLサーバ16cに送信して所定の検索処理を依頼し、その処理結果データであるデータ(4)を最終的な処理結果データ32としてクライアント12に送信する処理手順が、ディスパッチ定義ＤＢ24に規定されているものと想定する。

【００２９】

この場合、まずディスパッチエンジン21は、データ(1)をコネクタ22の第１のBLサーバ用データ変換部22aに出力する。第１のBLサーバ用データ変換部22aは、このデータ(1)を第１のBLサーバ16a向けのデータ形式に変換した後、第１のBLサーバ16aに送信する（Ｓ14）。

【００３０】

これと並行してテスト支援部25がデータ(1)を捕捉し、検索トランザクション及び第１のBLサーバ16aを特定する情報と関連付けてテスト用ＤＢ26に格納する（Ｓ15）。具体的には、図３に示すように、BEトランザクションID「１１１」のINデータ（＝入力データ）の項目に、データ(1)が登録される。

この場合、BEトランザクションID「１１１」が、検索トランザクション及び第１のBLサーバ16aを特定する情報に該当する。

図示の通り、このテスト用ＤＢ26には、各トランザクション単位でリクエストID、PLトランザクションID、BEトランザクションIDからなる制御情報が格納されている。

【００３１】

つぎに、第１のBLサーバ16aから処理結果データが返されると（Ｓ16）、第１のBLサーバ用データ変換部22aにおいてPLサーバ14向けの標準形式に変換された後（Ｓ17）、データ(2)としてディスパッチエンジン21に出力される。

【００３２】

この際、テスト支援部25がデータ(2)を捕捉し、テスト用ＤＢ26に第１のBLサーバ16aからの検索トランザクションの処理結果データとして格納する（Ｓ18）。具体的には、BEトランザクションID「１１１」のOUTデータ（＝処理結果データ）の項目に、データ(2)が登録される。

【００３３】

つぎにディスパッチエンジン21は、上記のディスパッチ定義に従い、データ(2)をコネクタ22の第２のBLサーバ用データ変換部22bに出力する。第２のBLサーバ用データ変換部22bは、このデータ(2)を第２のBLサーバ16b向けのデータ形式に変換した後、第２のBLサーバ16bに送信する（Ｓ19）。

【００３４】

同時に、テスト支援部25がデータ(2)を捕捉し、検索トランザクション及び第２のBLサーバ16bを特定する情報と関連付けてテスト用ＤＢ26に格納する（Ｓ20）。具体的には、BEトランザクションID「２２２」のINデータの項目に、データ(2)が登録される。

この場合、BEトランザクションID「２２２」が、検索トランザクション及び第２のBLサーバ16bを特定する情報に該当する。

【００３５】

第２のBLサーバ16bから処理結果データが返されると（Ｓ21）、第２のBLサーバ用データ変換部22bにおいてPLサーバ14向けの標準形式に変換された後（Ｓ22）、データ(3)としてディスパッチエンジン21に出力される。

【００３６】

この際、テスト支援部25がデータ(3)を捕捉し、テスト用ＤＢ26に第２のBLサーバ16bからの検索トランザクションの処理結果データとして格納する（Ｓ23）。具体的には、BEトランザクションID「２２２」のOUTデータの項目に、データ(3)が登録される。

【００３７】
つぎにディスパッチエンジン21は、上記のディスパッチ定義に従い、データ(3)をコネクタ22の第３のBLサーバ用データ変換部22cに出力する。第３のBLサーバ用データ変換部22cは、このデータ(3)を第３のBLサーバ16c向けのデータ形式に変換した後、第３のBLサーバ16cに送信する（Ｓ24）。
【００３８】
同時に、テスト支援部25がデータ(3)を捕捉し、検索トランザクション及び第３のBLサーバ16cを特定する情報と関連付けてテスト用ＤＢ26に格納する（Ｓ25）。具体的には、BEトランザクションID「３３３」のINデータの項目に、データ(3)が登録される。
この場合、BEトランザクションID「３３３」が、検索トランザクション及び第３のBLサーバ16cを特定する情報に該当する。
【００３９】
第３のBLサーバ16cから処理結果データが返されると（Ｓ26）、第３のBLサーバ用データ変換部22cにおいてPLサーバ14向けの標準形式に変換された後（Ｓ27）、データ(4)としてディスパッチエンジン21に出力される。
【００４０】
この際、テスト支援部25がデータ(4)を捕捉し、テスト用ＤＢ26に第３のBLサーバ16cからの処理結果データとして格納する（Ｓ28）。具体的には、BEトランザクションID「３３３」のOUTデータの項目に、データ(4)が登録される。
【００４１】
このデータ(4)は、画面制御部20によって最終的なリクエストの処理結果データ32として処理結果表示画面中に埋め込まれ、クライアント12に送信される（Ｓ29）。
この結果、クライアント12の表示装置33上に処理結果データ32が表示される。
【００４２】
以上のように、クライアント12から一つのリクエストデータ30が送信された場合、テスト支援部25の働きにより、テスト用ＤＢ26には第１のBLサーバ16a、第２のBLサーバ16b、第３のBLサーバ16cに係る入力データ及び処理結果データが、トランザクション単位で蓄積されることとなる。
【００４３】
もちろん、複数のBLサーバ16に対して所定の処理を順番に依頼する場合に限られず、１つのBLサーバ16に対して複数の処理を連続して依頼する場合や、複数のBLサーバ16に対して同時並列的に複数の処理を依頼する場合にも、テスト支援部25によってトランザクション毎に入力データと処理結果データがテスト用ＤＢ26に記録されることとなる。
【００４４】
つぎに、初回ランによって収集したトランザクション毎の入力データ及び処理結果データに基づいて、同様のテストを再ランする場合について説明する。
図４は、再ラン時におけるオンラインシステム10のシステム構成図であり、クライアント12や画面制御部20の関与なしに再ランが実行されることを表現している。
【００４５】
以下、図５のフローチャートに従い、再ラン時の処理手順を説明する。
まず、テスト端末18からPLトランザクションID「１１１１」を特定した再ランの指示データが送信されると、これを受けたテスト支援部25は（Ｓ40）、再ランの実行範囲を特定する（Ｓ41）。ここで、PLトランザクションID「１１１１」はBEトランザクションID「１１１」～「３３３」を包含する上位概念であるため、テスト支援部25は初回ランと同じ範囲で再ランを行うべきものと判断する。
【００４６】
そこでテスト支援部25は、テスト用ＤＢ26からBEトランザクションID「１１１」の入力データであるデータ(1)を取り出し（Ｓ42）、ディスパッチエンジン21に渡す。
【００４７】
ディスパッチエンジン21は、データ(1)をコネクタ22の第１のBLサーバ用データ変換部2

2aに出力する。第１のBLサーバ用データ変換部22aは、このデータ(1)を第１のBLサーバ16a向けのデータ形式に変換した後、第１のBLサーバに送信する（Ｓ43）。

【００４８】

そして、第１のBLサーバ16aから処理結果データが返されると（Ｓ44）、第１のBLサーバ用データ変換部22aにおいてPLサーバ14向けの標準形式に変換された後（Ｓ45）、データ(2)'としてディスパッチエンジン21に出力される。

【００４９】

このデータ(2)'は、テスト支援部25によって捕捉され、テスト用ＤＢ26に格納される（Ｓ46）。具体的には、図６に示すように、BEトランザクションID「１１１」の再ラン結果データの項目に、データ(2)'が登録される。

【００５０】

つぎにテスト支援部25は、テスト用ＤＢ26からBEトランザクションID「２２２」の入力データであるデータ(2)を取り出し（Ｓ47）、ディスパッチエンジン21に渡す。

【００５１】

ディスパッチエンジン21は、データ(2)をコネクタ22の第２のBLサーバ用データ変換部22bに出力する。第２のBLサーバ用データ変換部22bは、このデータ(2)を第２のBLサーバ16b向けのデータ形式に変換した後、第２のBLサーバ16bに送信する（Ｓ48）。

【００５２】

そして、第２のBLサーバ16bから処理結果データが返されると（Ｓ49）、第２のBLサーバ用データ変換部22bにおいてPLサーバ14向けの標準形式に変換された後（Ｓ50）、データ(3)'としてディスパッチエンジン21に出力される。

【００５３】

このデータ(3)'は、テスト支援部25によって捕捉され、テスト用ＤＢ26に格納される（Ｓ51）。具体的には、BEトランザクションID「２２２」の再ラン結果データの項目に、データ(3)'が登録される。

【００５４】

つぎにテスト支援部25は、テスト用ＤＢ26からBEトランザクションID「３３３」のINデータであるデータ(3)を取り出し（Ｓ52）、ディスパッチエンジン21に渡す。

【００５５】

ディスパッチエンジン21は、データ(3)をコネクタ22の第３のBLサーバ用データ変換部22cに出力する。第２のBLサーバ用データ変換部22cは、このデータ(3)を第３のBLサーバ16c向けのデータ形式に変換した後、第３のBLサーバ16cに送信する（Ｓ53）。

【００５６】

そして、第３のBLサーバ16cから処理結果データが返されると（Ｓ54）、第３のBLサーバ用データ変換部22cにおいてPLサーバ14向けの標準形式に変換された後（Ｓ55）、データ(4)'としてディスパッチエンジン21に出力される。

【００５７】

このデータ(4)'は、テスト支援部25によって捕捉され、テスト用ＤＢ26に格納される（Ｓ56）。具体的には、BEトランザクションID「３３３」の再ラン結果データの項目に、データ(4)'が登録される。

【００５８】

つぎに、テスト支援部25はテスト用ＤＢ26に格納された初回ラン時の各BEトランザクションIDの処理結果データ（OUTデータ）と再ラン結果データとを比較し（Ｓ57）、その比較結果のリストをテスト端末18に送信する（Ｓ58）。

【００５９】

例えば、バグの修正を行う前に初回ランを実行しておき、バグの修正後に再ランを行えば、バグの修正によって悪影響が生じているか否かを確認することができる。すなわち、比較結果リストが全て「一致」であれば問題ないが、「不一致」の場合にはデグレードが生じている可能性があるため、より細かいレベルでの再ランを行う必要がある。

【００６０】

このテスト支援システムにおいては、トランザクション単位で個別に再ランを行うこともできる。

例えば、テスト端末18からBEトランザクションID「２２２」を指定した再ランの指示データが送信された場合、図７に示すように、テスト支援部25はテスト用ＤＢ26から「２２２」の入力データであるデータ(2)を取り出し、ディスパッチエンジン21に渡す。

【００６１】

ディスパッチエンジン21は、データ(2)をコネクタ22の第２のBLサーバ用データ変換部22bに出力する。第２のBLサーバ用データ変換部22bは、このデータ(2)を第２のBLサーバ16b向けのデータ形式に変換した後、第２のBLサーバ16bに送信する。

【００６２】

そして、第２のBLサーバ16bから処理結果データが返されると、第２のBLサーバ用データ変換部22bにおいてPLサーバ14向けの標準形式に変換された後、データ(3)'としてディスパッチエンジン21に出力される。

このデータ(3)'は、テスト支援部25によって捕捉され、テスト用ＤＢ26におけるBEトランザクションID「２２２」の再ラン結果データの項目に登録される。

【００６３】

このように、トランザクション単位で再ランを実施し、初回ラン時の処理結果データと再ラン結果データとを比較することにより、リクエスト単位で入力データと処理結果データを比較する従来の入力支援ツールに比べ、よりきめ細かいレベルでBLサーバ16の動作を検証することが可能となる。

【００６４】

また、このようにBLサーバ16のトランザクション単位での再ランが可能であることから、このテスト支援システムによれば、データベース28の状態もトランザクション単位で復元することが可能となる。

すなわち、各BLサーバ16が管理するデータベース28については、初回ランの直前にバックアップが取られているため、このバックアップされたデータに基づいて必要なトランザクションを順次実行することにより、各データベース28の更新具合をその都度確認することが可能となる。

【００６５】

図８は、ある住所管理データベース28に対する更新状況を例示するものであり、トランザクションαによって「TEL」が更新され、トランザクションβによって「FAX」が、またトランザクションγによって「住所」が更新される仕組みである場合、トランザクションα～γを順にワンステップずつ再ランさせることにより、データベース28が（イ）→（ロ）→（ハ）→（ニ）と更新されてきた様子を個別に確認することができる。

【００６６】

これに対し、従来の入力支援ツールを用いた場合は、リクエスト単位でしか再ランが実施できず、BLサーバ16のトランザクション単位で再ランを実施することができなかったため、図１９に示したように、データベース28の（イ）と（ハ）の状態のみを比較するしかなく、途中の（ロ）地点、（ハ）地点の状態を確認することはできなかった。

【００６７】

ところで、オンラインシステムにおいては、同一のデータベースに対して複数のクライアントが同時にアクセスし、相互に矛盾した更新処理が実行されることを防止する目的で、「楽観的排他ロック」と称する機能が一般に設けられている。

【００６８】

図９は、この楽観的排他ロックの仕組みを説明するための図であり、データベース28の各レコードには「最終更新日時」のデータ項目42が設けられている。

ここで、図９(a)に示すように、クライアント12から「社員ID＝001」を指定した検索リクエストデータ30が送信されると、PLサーバ14経由でこれを受け取ったBLサーバ16は、データベース28から検索条件にマッチしたレコードを抽出し、PLサーバ14経由でクライアント12に検索処理結果データ32を送信することとなるが、この検索処理結果データ32には「

更新日時」のデータ項目43が設けられており、データベース28の「最終更新日時」の値である「05/24/10:00」が充填されている。

【００６９】

そして、同クライアント12から「社員ID＝001」のレコードについての更新リクエストデータが送信される場合には、このデータにも上記の「更新日時」が引き継がれることとなる。

例えば、図９(b)に示すように、５月24日の14:00にクライアント12から同社員の所属を「営業部」から「総務部」に変更する内容の更新リクエストデータ30が送信される場合、「05/24/10:00」のデータが付加されている。

【００７０】

この更新リクエストデータ30を受信したBLサーバ16は、その更新日時とデータベース28に格納された対応レコードの最終更新日時を比較し、両者が一致していることを確認した上で、当該レコードへの更新を実行する。この際、当該レコードの最終更新日時は「05/24/14:00」に更新される。また、クライアント12に対しては更新処理結果データ32が送信されるが、このデータの更新日時には最新の「05/24/14:00」が記録される。

【００７１】

これに対し、更新リクエストデータ30の更新日時とデータベース28側の最終更新日時が一致しない場合には、その間に他のクライアントによって当該レコードに更新がなされたことを意味しているため、BLサーバ16からクライアント12に対してエラーメッセージが送信され、データの更新が拒絶されることとなる。

【００７２】

以上のように、楽観的排他ロック機構を備えることにより、オンラインシステムに含まれるデータベースに矛盾が生じることを有効に抑えることが可能となる。

しかしながら、この楽観的排他ロック機構を備えたオンラインシステムにこのテスト支援システムをそのまま適用すると再ラン時に問題が生じるため、これを回避するための仕組みを講じておく必要がある。以下において、この問題とその回避方法について説明する。

【００７３】

まず、図１０は初回ラン時の手順を示すものであり、図１０(a)に示すように、05/24/14:00の時点でクライアント12から(A)の更新リクエストデータ（更新日時：05/24/10:00）が送信されると、PLサーバ14経由でこれを受け取ったBLサーバ16は、当該更新リクエストデータ(A)の更新日時とデータベース28側の最終更新日時と比較し、両者が一致していることを確認した後、所属を「営業部」から「総務部」に変更する更新処理を実行する。この際、当該レコードの最終更新日時も「05/24/14:00」に更新される。

【００７４】

同時に、BLサーバ16からクライアント12に対しては、PLサーバ14経由で更新処理結果データ(B)が送信される。この更新処理結果データ(B)の更新日時にも、最終更新日時である「05/24/14:00」が記録されている。

この間、テスト用ＤＢ26には、テスト支援部25（図示省略）によって更新リクエストデータ(A)及び更新処理結果データ(B)が格納される。

【００７５】

つぎに、図１０(b)に示すように、05/24/16:00の時点でクライアント12から(C)の更新リクエストデータ（更新日時：05/24/14:00）が送信されると、BLサーバ16は当該更新リクエストデータ(C)の更新日時とデータベース28側の最終更新日時と比較し、両者が一致していることを確認した後、所属を「総務部」から「法務部」に変更する更新処理を実行する。この際、当該レコードの最終更新日時も「05/24/16:00」に更新される。

【００７６】

同時に、BLサーバ16からクライアント12に対しては、PLサーバ14経由で更新処理結果データ(D)が送信される。この更新結果データ(D)の更新日時にも、最終更新日時である「05/24/16:00」が記録されている。

この間、テスト用ＤＢ26には、テスト支援部25によって更新リクエストデータ(C)及び更新結果データ(D)が格納される。

【００７７】

つぎに、図１１は再ラン時の手順を示すものであり、図１１(a)に示すように、05/24/21:00の時点でPLサーバ14から(A)の更新リクエストデータ（更新日時：05/24/10:00）が送信されると、BLサーバ16は当該更新リクエストデータ(A)の更新日時とデータベース28側の最終更新日時と比較する。再ラン時には、初回ランの直前にバックアップされたデータが用いられるため、両日時は当然に一致することから、BLサーバ16は所属を「営業部」から「総務部」に変更する更新処理を実行する。この際、当該レコードの最終更新日時は現在時刻の「05/24/21:00」に更新される。

【００７８】

同時に、BLサーバ16からPLサーバ14に対しては、更新処理結果データ(B)'が送信される。この更新処理結果データ(B)'の更新日時にも、最終更新日時である「05/24/21:00」が記録されている。

この間、テスト用ＤＢ26には、テスト支援部25によって更新結果データ(B)'が格納される。

【００７９】

つぎに、図１１(b)に示すように、05/24/23:00の時点でPLサーバ14から(C)の更新リクエストデータ（更新日時：05/24/14:00）が送信されると、BLサーバ16は当該更新リクエストデータ(C)の更新日時とデータベース28側の最終更新日時と比較する。この場合には更新日時が不一致となるため、BLサーバ16からPLサーバ14に対してエラーメッセージが送信され、データベース28の更新処理は当然ながら拒絶されることとなる。

この間、テスト用ＤＢ26には、テスト支援部25によってエラーメッセージ(D)'が再ラン結果データとして記録される。

【００８０】

以上を要するに、楽観的排他ロック機構の下においてこのテスト支援システムをそのまま適用すると、以下の問題が生じる。

(1)初回ラン時にテスト用ＤＢ26格納された入力データの更新日時と、再ラン時にBLサーバ16によって更新されたデータベースの最終更新日時とが不一致となるため、楽観的排他ロックが発動され、再ラン結果データがエラーとなる。

(2)再ラン結果データの更新日時には再ラン時の現在日時が付与されるため、各トランザクションの処理結果データと再ラン結果データを比較する際に、更新日時の不一致を理由に「比較結果＝不一致」と判定されてしまう。

そこで、このテスト支援システムは、更新日時に関して「引継ぎ」及び「マスキング」という手法を適用することにより、上記の問題を解消している。

【００８１】

まず「引継ぎ」とは、図１２(a)に示すように、BLサーバ16によって再ラン結果データ(B)'に付与された更新日時「05/24/21:00」を用いて、後続トランザクションの入力データ(C)に対して初回ラン時に付与された「05/24/14:00」の更新日時を上書きした後、BLサーバ16に送信する手法を意味する。この更新日時の引継ぎは、テスト支援部25によって実行される。

【００８２】

この結果、更新リクエストデータ(C)の更新日時とデータベース28側の最終更新日時とが一致することとなり、社員ID＝001の所属を「総務部」から「法務部」に変更する更新処理が、BLサーバ16において正しく実行されることとなる。

また、BLサーバ16からPLサーバ14に対しては、更新結果データ(D)'が送信される。この更新結果データ(D)'の更新日時には、最終更新日時である「05/24/23:00」が記録されている。この更新結果データ(D)'は、テスト支援部25によってテスト用ＤＢ26の再ラン結果データ項目に登録される。

【００８３】

つぎに「マスキング」とは、図１３(a)に示すように、テスト用ＤＢ26に格納された処理結果データ(B)と再ラン結果データ(B)'をそのまま比較すると、更新日時が異なるために全体が不一致と判定されてしまう場合に、図１３(b)に示すように、両データの更新日時にマスク処理を施した上で比較する手法を意味する。具体的には、両データの更新日時を共通の文字列「***」等に置き換えることが該当する。

このマスク処理も、テスト支援部25によって実行される。

【００８４】

この結果、楽観的排他ロックのために便宜的に付与される更新日時項目の値の不一致によって、残りの実質的なデータ項目の値に変更がない場合であっても全体が不一致と判定されてしまう不都合を回避可能となる。

図示は省略したが、処理結果データ(D)と再ラン結果データ(D)'との比較時にも、それぞれの更新日時にはマスク処理が施される。

【００８５】

テスト支援部25は、上記の通り、ディスパッチエンジン21内に組み込まれ、ディスパッチエンジン21を通過するデータを捕捉してテスト用ＤＢ26に格納する機能、再ラン時にテスト用ＤＢ26から必要なデータを取り出してディスパッチエンジン21に渡し、所定のBLサーバ16への送信を依頼する機能、BLサーバ16から送信されたデータを捕捉してテスト用ＤＢ26に格納すると共に、初回ラン時のデータと比較する機能を発揮するが、テスト終了後の本番時においては全く不要となる。

このため、ディスパッチエンジン21用のアプリケーション内にテスト支援部25用のコードを予め組み込んでおくのは好ましくない。

また、本番時とテスト時とではマシン構成や回線容量等の環境が異なる場合が多いため、通常はテスト終了後にアプリケーションの設定を本番用に書き換えることが行われているが、この際にバグが混入する危険性があった。

【００８６】

そこでこのシステム10では、テスト対象となるアプリケーションプログラム自体には手を加えることなく、テスト時にのみテスト支援部25を外部からディスパッチエンジン21内に組み込むと共に、外部に配置されたテスト用の設定情報を反映させる仕組みを採用している。

【００８７】

図１４は、この仕組みを説明するための模式図であり、PLサーバ14のディスク領域50には、ディスパッチエンジン用実行ファイル51と、テスト支援部用ライブラリ52と、テスト設定ファイル53が格納されている。ディスパッチエンジン用実行ファイル51には、本番設定ファイル54が含まれている。

【００８８】

また、PLサーバ14のメモリ空間56には、上位クラスローダ57が設けられており、この上位クラスローダ57内に予めスイッチモジュール58が配置されている。

この上位クラスローダ57は、Java（登録商標）の動作環境であるJ2EE（Java 2 Enterprise Edition）の仕様に基づいて実現されるものであり、この上位クラスローダ57内に配置されたモジュールは、メモリ空間56に展開された全てのアプリケーションから参照できるという特徴を備えている。

このスイッチモジュール58は、スイッチモジュール用のライブラリを予めPLサーバ14の所定のディレクトリに格納しておくことで、JVM（Java Virtual Machine）の起動時に上位クラスローダ57内に自動的に配置される。

【００８９】

つぎに、図１５のフローチャートに従い、テスト支援部25の組込及びテスト用設定情報の反映手順について説明する。

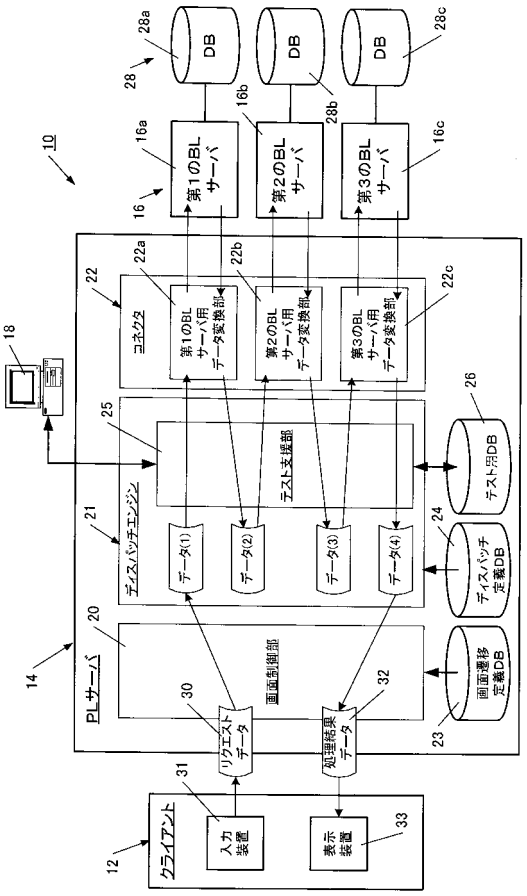
まず、ディスパッチエンジン用実行ファイル51を起動させ、メモリ空間56のWARクラスローダ59内にフレームワーク60、固有機能部61、本番設定情報62を配置させる（Ｓ51）。J2EEの環境下においては、各WebアプリケーションはWARクラスローダ単位で管理される。

【００９０】
つぎに、フレームワーク60が上位クラスローダ57内にスイッチモジュール58が存在しているか否かをチェックする（Ｓ52）。
ここで、スイッチモジュール58が存在している場合（Ｓ53／YES）、フレームワーク60はテストモードであると認識し、テスト支援部用ライブラリ52を自己のWARクラスローダ59内にロードし（Ｓ54）、テスト支援部25を生成する。
【００９１】
このテスト支援部25は、PLサーバ14のディスク領域50から対応のテスト設定ファイル53をメモリ空間56上に読み込み（Ｓ55）、WARクラスローダ59内に展開された本番設定情報をテスト設定情報で上書きする。
【００９２】
以上の結果、図１６(b)に示すように、メモリ空間のWARクラスローダ59内には、ディスパッチエンジン固有の機能を担当する固有機能部61とフレームワーク60の他に、テスト支援部25が組み込まれると共に、本番用の設定情報62がテスト用設定情報63に切り替えられることとなる。
【００９３】
そして、オンラインテストが完了した後は、スイッチモジュール用ライブラリを削除してJVMを再起動させ、上位クラスローダ57内のスイッチモジュール58を除去しておくだけで、Ｓ54～Ｓ56の処理がスキップされ、図１６(a)に示すように、WARクラスローダ59内にはフレームワーク60と固有機能部61及び本番設定情報62が配置される。
この結果、ディスパッチエンジン用実行ファイル51（Webアプリケーション）に手を加えることなく、ディスパッチエンジン21を本番仕様に切り替えることが可能となる。
【００９４】
オンラインシステムにあっては、実装言語やフレームワークを異にする複数のサーバ（ホストシステムを含む）を連携させる異機種間連携の仕組みを一般に備えているため、図１７に示すように、これまでは各BLサーバ16a～16c毎に専用のテスタ70a～70cを接続し、BLサーバ単位でIN/OUTのデータ確認を行っていた。このため、複数のテスタを準備し、それぞれの操作法をマスターせざるを得ないという問題があった。
【００９５】
これに対し、このテスト支援システムの場合には、上記のようにコネクタ22において入力データが各BLサーバ16a～16c向けの形式に変換された上で送信されると共に、各BLサーバ16a～16cからの処理結果データがコネクタ22において標準フォーマットに変換される機能を備えているため、テスト支援部25は各種BLサーバ16のテストを標準フォーマットに基づいて統一的に実行することが可能となり、オペレータもテスト支援部25の操作法のみを習得すれば済むこととなる。
【図面の簡単な説明】
【００９６】
【図１】この発明に係るテスト支援システムの初回ラン時における構成を示すシステム構成図である。
【図２】オンラインテストの初回ラン時における処理手順を示すフローチャートである。
【図３】テスト用ＤＢのデータ項目例を示す説明図である。
【図４】テスト支援システムの再ラン時における構成を示すシステム構成図である。
【図５】オンラインテストの再ラン時における処理手順を示すフローチャートである。
【図６】テスト用ＤＢのデータ項目例を示す説明図である。
【図７】テスト支援システムにおける再ランの要領を示す概念図である。
【図８】テスト支援システムの再ラン時にBLサーバのデータベースがトランザクション単位で更新される様子を示す説明図である。
【図９】楽観的排他ロックの一般的な仕組みを説明する模式図である。
【図１０】このテスト支援システムにおいて楽観的排他ロックを動作させた場合の問題点を示す模式図である。

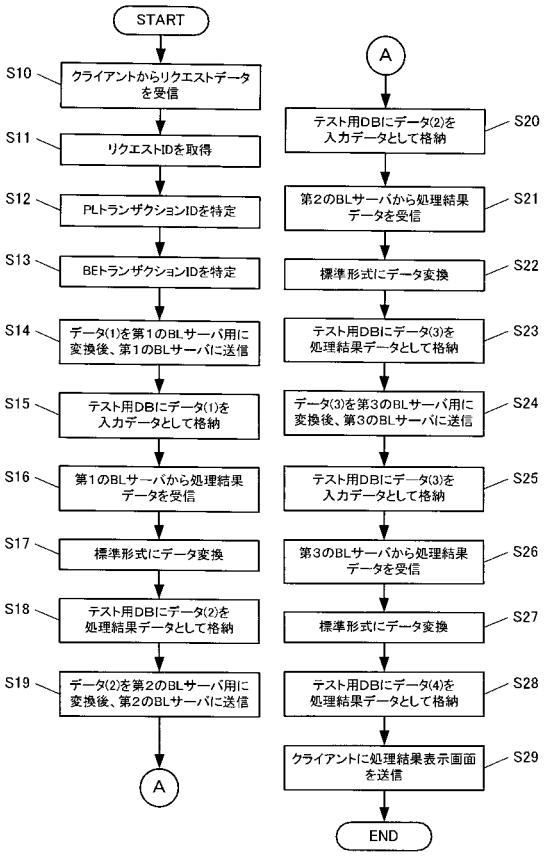
【図１１】このテスト支援システムにおいて楽観的排他ロックを動作させた場合の問題点を示す模式図である。
【図１２】テスト支援システムにおいて楽観的排他ロックを動作させた場合の問題点を回避する方法を示す模式図である。
【図１３】テスト支援システムにおいて楽観的排他ロックを動作させた場合の問題点を回避する方法を示す模式図である。
【図１４】ディスパッチエンジンにオンライン支援部を組み込みテスト用の設定を施す際の要領を示す模式図である。
【図１５】ディスパッチエンジンにオンライン支援部を組み込みテスト用の設定を施す際の手順を示すフローチャートである。
【図１６】ディスパッチエンジンをテストモードと本番モードとの間でスイッチ可能としたことを示す説明図である。
【図１７】言語使用やフレームワークの異なるBLサーバ毎に専用のテスタを用いてテストを行う従来のテスト方法を示す模式図である。
【図１８】従来のオンラインシステムの一例を示すシステム構成図である。
【図１９】従来のオンラインテストにおけるデータベースの再現性を示す説明図である。
【符号の説明】
【００９７】
10　オンラインシステム
12　クライアント
14　PLサーバ
16　BLサーバ
16a　第１のBLサーバ
16b　第２のBLサーバ
16c　第３のBLサーバ
18　テスト端末
20　画面制御部
21　ディスパッチエンジン
22　コネクタ
22a　第１のBLサーバ用データ変換部
22b　第２のBLサーバ用データ変換部
22c　第３のBLサーバ用データ変換部
25　テスト支援部
26　テスト用データベース
28　データベース
30　リクエストデータ
31　入力装置
32　処理結果データ
33　表示装置
42　「最終更新日時」のデータ項目
43　「更新日時」データ項目
50　PLサーバのディスク領域
51　ディスパッチエンジン用実行ファイル
52　テスト支援部用ライブラリ
53　テスト設定ファイル
54　本番設定ファイル
56　PLサーバのメモリ空間
57　上位クラスローダ
58　スイッチモジュール
59　WARクラスローダ

60　フレームワーク
61　固有機能部
62　本番設定情報
63　テスト用設定情報

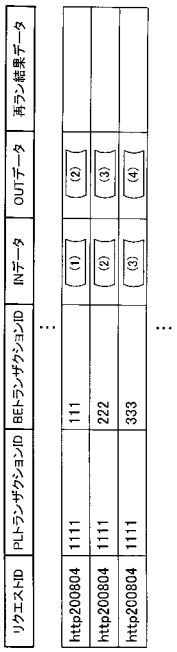
【図１】



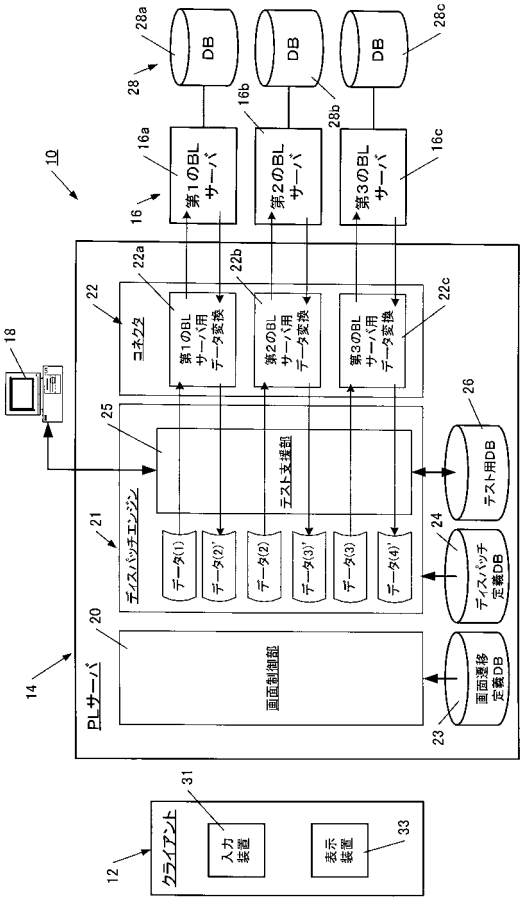
【図２】



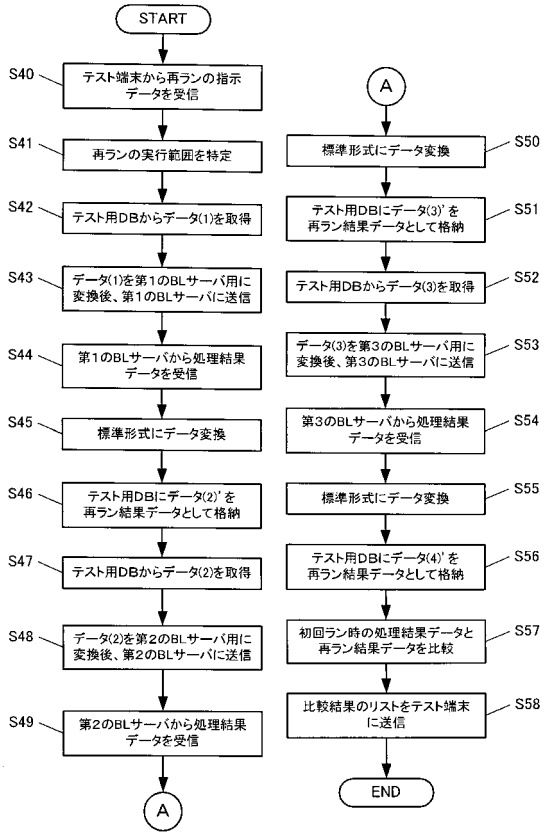
【図３】


リクエストID
PLトランザクションID
BEトランザクションID
INデータ
OUTデータ
再ラン結果データ
http200804
1111
111
(1)
(2)
(1)'
(2)'
222
(3)
(4)
(3)'
(4)'
333

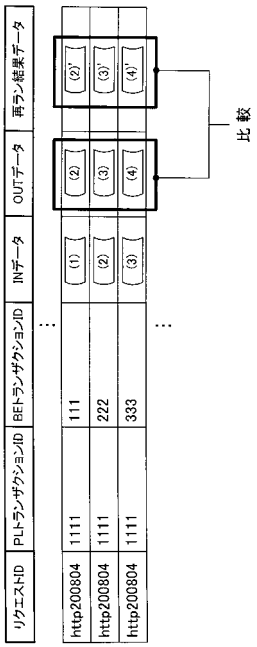
【図４】


10
12
14
16
16a
16b
16c
18
20
21
22
22a
22b
22c
23
24
25
26
28
28a
28b
28c
31
33
DB
第1のBLサーバ
第2のBLサーバ
第3のBLサーバ
コネクタ
第1のBLサーバ用データ変換
第2のBLサーバ用データ変換
第3のBLサーバ用データ変換
テスト支援部
ディスパッチエンジン
データ(1)
データ(1)'
データ(2)
データ(3)'
データ(3)
データ(4)'
テスト用DB
ディスパッチ定義DB
画面遷移定義DB
画面制御部
PLサーバ
クライアント
入力装置
表示装置

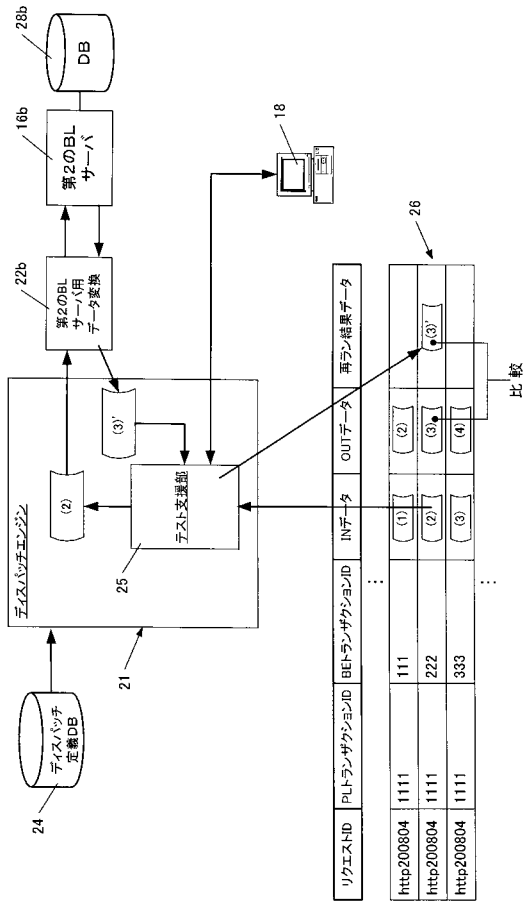
【図５】


START
S40
テスト端末から再ランの指示データを受信
S41
再ランの実行範囲を特定
S42
テスト用DBからデータ(1)を取得
S43
データ(1)を第1のBLサーバ用に変換後、第1のBLサーバに送信
S44
第1のBLサーバから処理結果データを受信
S45
標準形式にデータ変換
S46
テスト用DBにデータ(2)'を再ラン結果データとして格納
S47
テスト用DBからデータ(2)を取得
S48
データ(2)を第2のBLサーバ用に変換後、第2のBLサーバに送信
S49
第2のBLサーバから処理結果データを受信
A
A
S50
標準形式にデータ変換
S51
テスト用DBにデータ(3)'を再ラン結果データとして格納
S52
テスト用DBからデータ(3)を取得
S53
データ(3)を第3のBLサーバ用に変換後、第3のBLサーバに送信
S54
第3のBLサーバから処理結果データを受信
S55
標準形式にデータ変換
S56
テスト用DBにデータ(4)'を再ラン結果データとして格納
S57
初回ラン時の処理結果データと再ラン結果データを比較
S58
比較結果のリストをテスト端末に送信
END

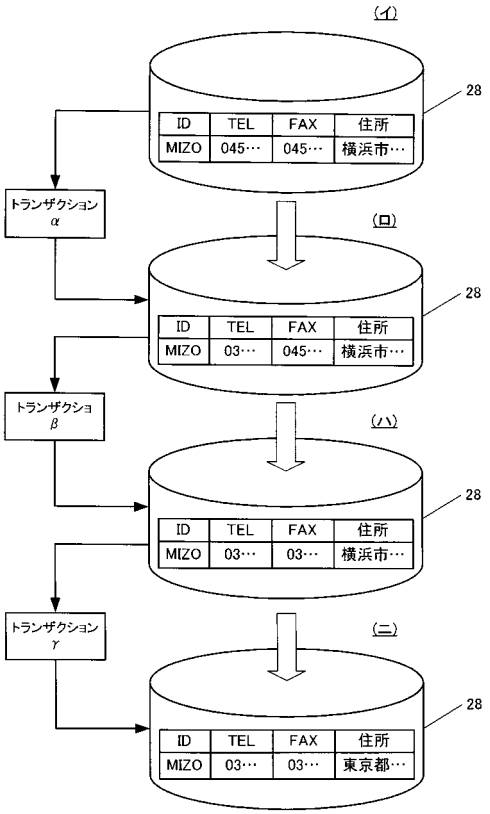
【図６】


リクエストID
PLトランザクションID
BEトランザクションID
INデータ
OUTデータ
再ラン結果データ
http200804
1111
111
(1)
(2)
(2)'
222
(3)
(4)
(3)'
(4)'
333
比較

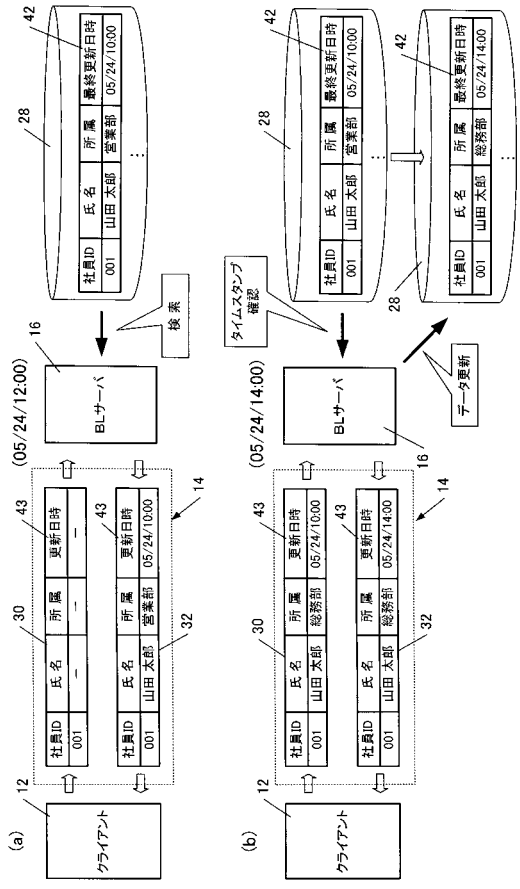
【図７】


28b
DB
16b
第2のBLサーバ
18
22b
第2のBLサーバ用データ変換
(3)'
(2)
テスト支援部
25
ディスパッチエンジン
21
24
ディスパッチ定義DB
26
比較
リクエストID
PLトランザクションID
BEトランザクションID
INデータ
OUTデータ
再ラン結果データ
http200804
1111
111
(1)
(2)
(3)'
222
(3)
(4)
333

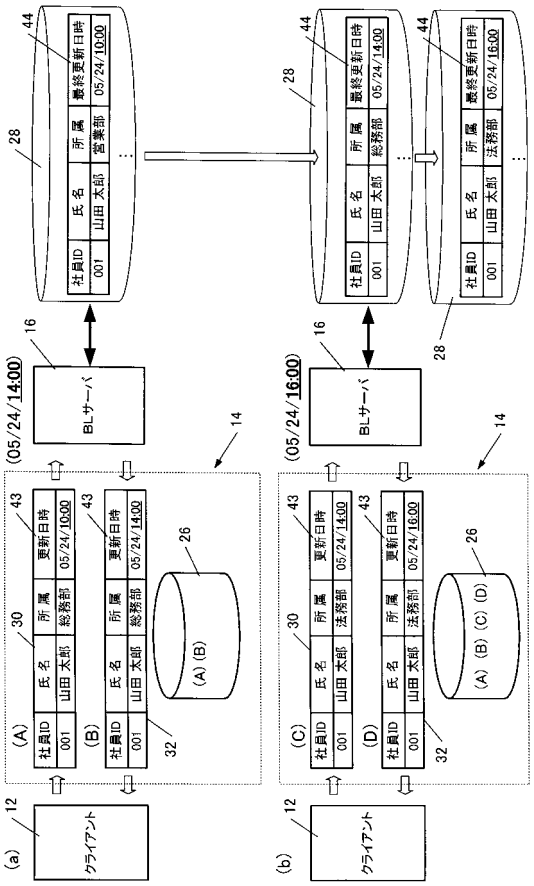
【図８】


(イ)
(ロ)
(ハ)
(ニ)
トランザクション α
トランザクション β
トランザクション γ
28
ID
TEL
FAX
住所
MIZO
045…
03…
横浜市…
東京都…

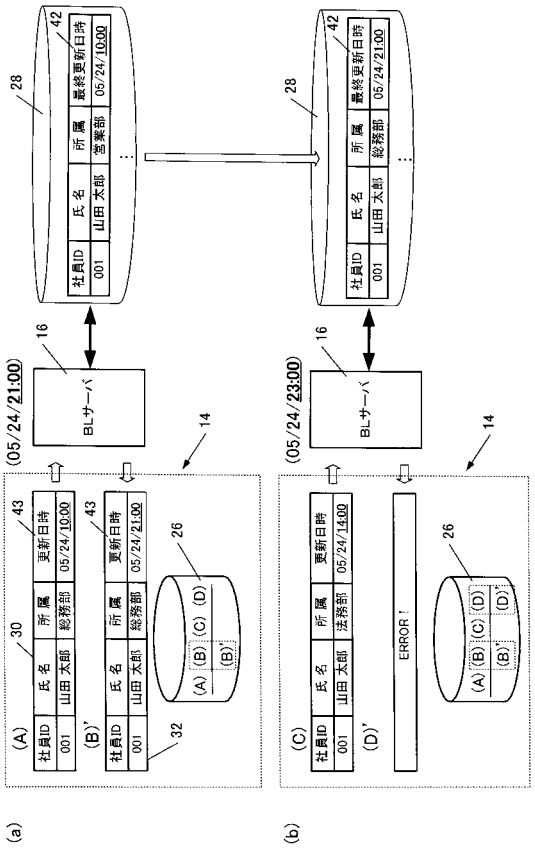
【図９】


42
28
社員ID
氏名
所属
最終更新日時
山田 太郎
営業部
総務部
05/24/10:00
05/24/14:00
検索
タイムスタンプ確認
データ更新
16
BLサーバ
(05/24/12:00)
(05/24/14:00)
14
43
更新日時
30
32
12
クライアント
(a)
(b)

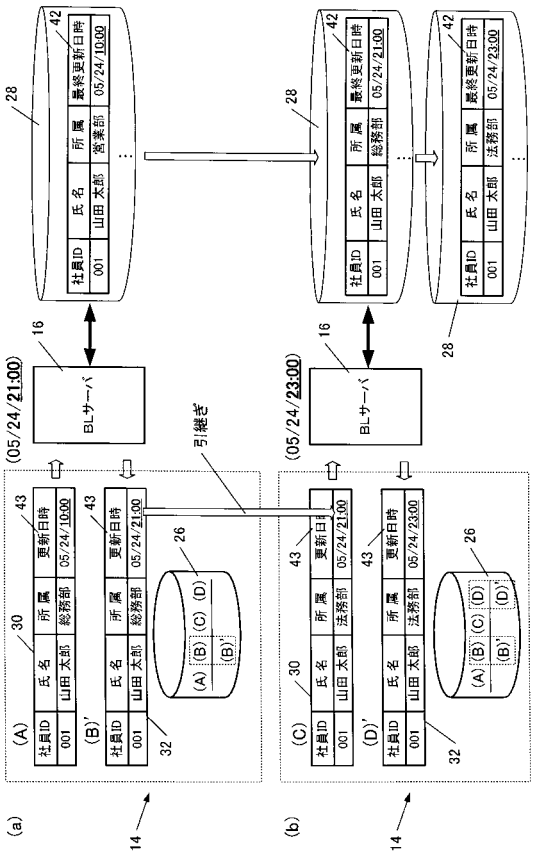
【図１０】


44
28
社員ID
氏名
所属
最終更新日時
山田 太郎
営業部
総務部
法務部
05/24/10:00
05/24/14:00
05/24/16:00
16
BLサーバ
(05/24/14:00)
(05/24/16:00)
14
43
更新日時
30
32
26
(A) (B)
(A) (B) (C) (D)
12
クライアント
(a)
(b)

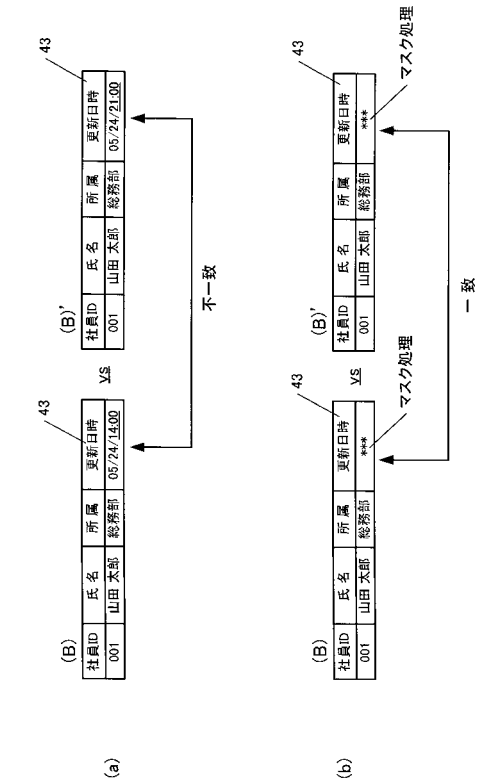
【図１１】



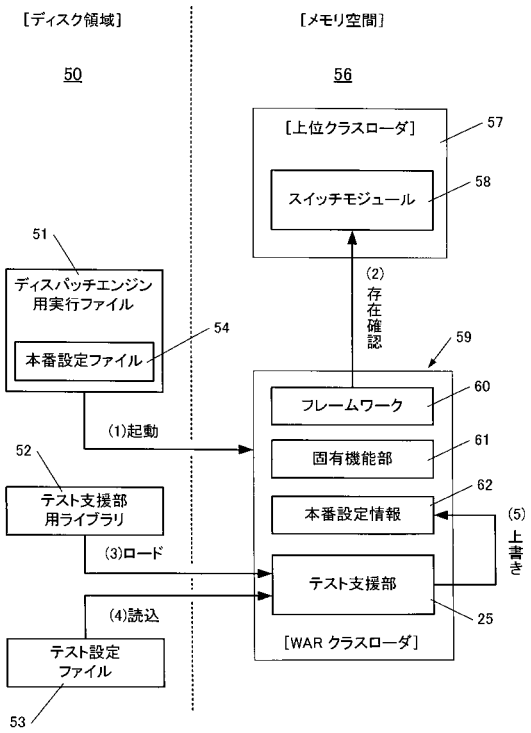
【図１２】



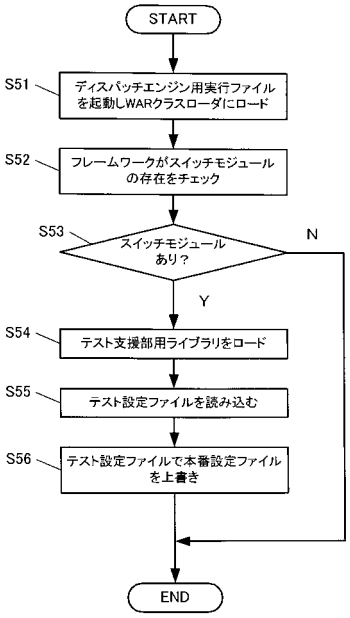
【図１３】



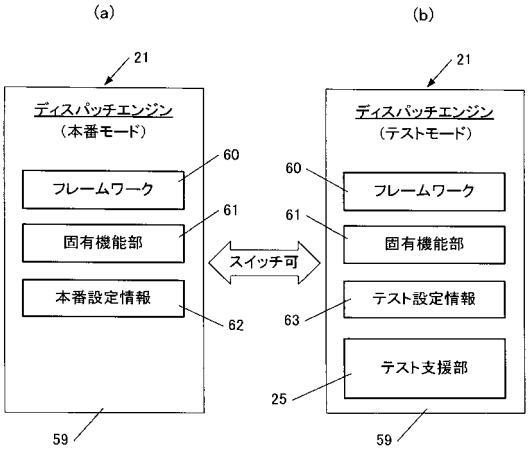
【図１４】



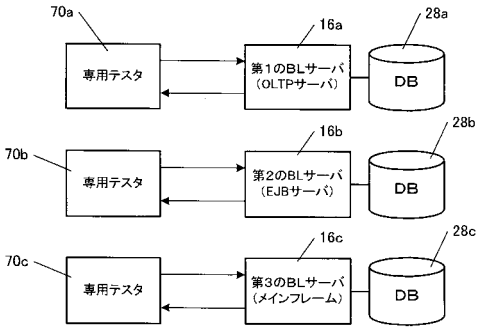
【図１５】



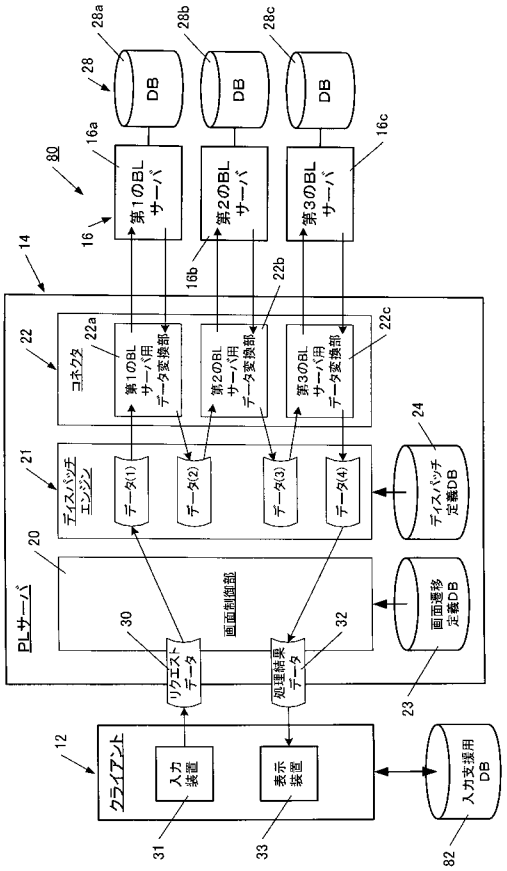
【図１６】



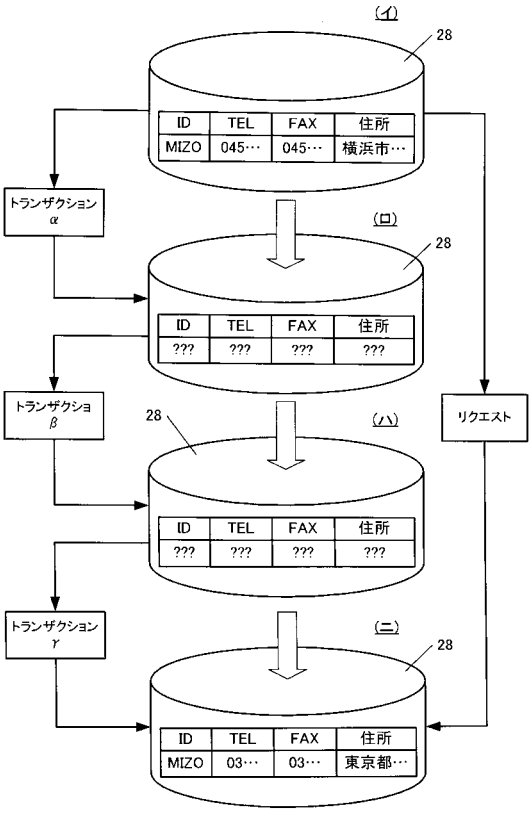
【図１７】



【図１８】



【図１９】



フロントページの続き

(72)発明者　和田　佑
東京都千代田区丸の内一丁目６番５号　株式会社野村総合研究所内
(72)発明者　渡邊　裕
東京都千代田区丸の内一丁目６番５号　株式会社野村総合研究所内
(72)発明者　國尾　慎二
東京都千代田区丸の内一丁目６番５号　株式会社野村総合研究所内

Ｆターム(参考) 5B042 GA12　GB01　HH26　MC09