

(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2004-13694

(P2004-13694A)

(43) 公開日 平成16年1月15日(2004.1.15)

(51) Int. Cl.⁷

G06F 15/00

G06F 9/46

F I

G06F 15/00 310M

G06F 15/00 310B

G06F 15/00 310R

G06F 9/46 340B

テーマコード(参考)

5B085

5B098

審査請求 未請求 請求項の数 8 O L (全 18 頁)

(21) 出願番号

特願2002-168838 (P2002-168838)

(22) 出願日

平成14年6月10日(2002.6.10)

(71) 出願人

000155469
株式会社野村総合研究所
東京都千代田区大手町二丁目2番1号

(74) 代理人

100075812
弁理士 吉武 賢次

(74) 代理人

100091982
弁理士 永井 浩之

(74) 代理人

100096895
弁理士 岡田 淳平

(74) 代理人

100105795
弁理士 名塚 聡

(74) 代理人

100106655
弁理士 森 秀行

(74) 代理人

100117787
弁理士 勝沼 宏仁

最終頁に続く

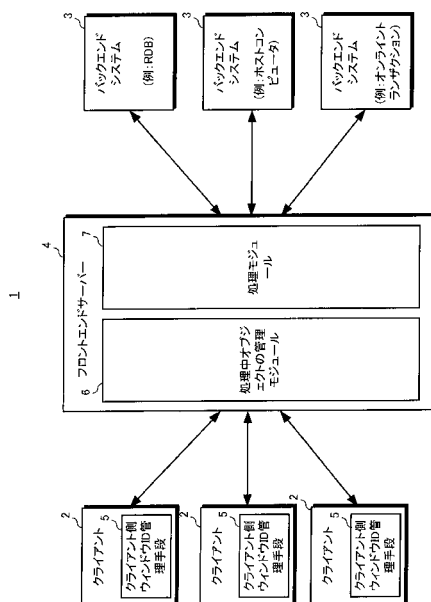
(54) 【発明の名称】 フロントエンド処理機能を有する情報処理システム

(57) 【要約】

【課題】複数のウィンドウが同時に立ち上げられ、複数の処理要求がほぼ同時に発せられた場合でも、処理結果を正しいウィンドウに表示させることができるフロントエンド処理機能を有する情報処理システムを提供する。

【解決手段】クライアント2から送信された画面データがクライアント側で新たに開いたウィンドウか否かを画面データに付されたウィンドウIDとフロントエンドサーバーに備えられたウィンドウID管理テーブルとを比較して判断し、新たに開いたウィンドウと判断した場合は新たにウィンドウIDを付与するとともに処理中オブジェクトを生成し、ウィンドウIDと処理要求IDの識別情報の組を処理要求に付して所定のバックエンドシステムに実行させ、処理結果を前記ウィンドウIDと処理要求IDの識別情報の組によって特定されるアプリケーションの所定の画面に加入しウィンドウIDを付して要求元のクライアントに送信するサーバー側ウィンドウID管理手段を有しているフロントエンドサーバー4を備えた。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

アプリケーションプログラムを実行する少なくとも一つのクライアントと業務処理ロジックを実行する少なくとも一つのバックエンドシステムの間を介し、クライアントから所定の処理要求を含むアプリケーション画面を受信し、前記処理要求を業務処理ロジックに変換し、所定のバックエンドシステムに実行させ、実行結果を要求を發したアプリケーションの所定の画面に加入して要求元のクライアントに送信するフロントエンドサーバーを有し、

前記フロントエンドサーバーは、

前記クライアントから送信された画面データがクライアント側で新たに開いたウィンドウか否かを画面データに付されたウィンドウIDとフロントエンドサーバーに備えられたウィンドウID管理テーブルとを比較して判断し、新たに開いたウィンドウと判断した場合は新たにウィンドウIDを付与するとともに処理中オブジェクトを生成し、ウィンドウIDと処理要求IDの識別情報の組を処理要求に付して所定のバックエンドシステムに実行させ、処理結果を前記処理要求IDによって特定されるアプリケーションの所定の画面に加入しウィンドウIDを付して要求元のクライアントに送信するサーバー側ウィンドウID管理手段を有していることを特徴とするフロントエンド処理機能を有する情報処理システム。

10

【請求項 2】

アプリケーションプログラムを実行する少なくとも一つのクライアントと業務処理ロジックを実行する少なくとも一つのバックエンドシステムの間を介し、クライアントから所定の処理要求を含むアプリケーション画面を受信し、前記処理要求を業務処理ロジックに変換し、所定のバックエンドシステムに実行させ、実行結果を要求を發したアプリケーションの所定の画面に加入して要求元のクライアントに送信するフロントエンドサーバーを有し、

20

前記フロントエンドサーバーは、

前記クライアントから送信された画面データがクライアント側で新たに開いたウィンドウか否かを画面データに付されたウィンドウIDとアプリケーションIDとフロントエンドサーバーに備えられたウィンドウID管理テーブルとを比較して判断し、新たに開いたウィンドウと判断した場合は新たにウィンドウIDを付与するとともに処理中オブジェクトを生成し、ウィンドウIDとアプリケーションIDと処理要求IDの識別情報の組を処理要求に付して所定のバックエンドシステムに実行させ、処理結果を前記処理要求IDによって特定されるアプリケーションの所定の画面に加入して前記ウィンドウIDとアプリケーションIDの識別情報の組を付して要求元のクライアントに送信するサーバー側ウィンドウID管理手段を有していることを特徴とするフロントエンド処理機能を有する情報処理システム。

30

【請求項 3】

前記ウィンドウID管理テーブルは、各処理中オブジェクトへのアクセス時間を記録し、前記サーバー側ウィンドウID管理手段は、前記ウィンドウID管理テーブルを所定のタイミングで参照し、クライアントから所定の時間以上アクセスがなかった処理中オブジェクトの処理を中止するための処理を行うことを特徴とする請求項 1 または 2 に記載のフロントエンド処理機能を有する情報処理システム。

40

【請求項 4】

アプリケーションプログラムを実行する少なくとも一つのクライアントと業務処理ロジックを実行する少なくとも一つのバックエンドシステムの間を、前記クライアントから所定の処理要求を含むアプリケーション画面を受信し、前記処理要求を業務処理ロジックに変換し、所定のバックエンドシステムに実行させ、実行結果を要求を發したアプリケーションの所定の画面に加入して要求元のクライアントに送信するフロントエンドサーバーを介在させたフロントエンド処理機能を有する情報処理システムにおいて、前記クライアントは、

50

ブラウザが所定のアプリケーション画面を開く時に、そのアプリケーション画面がウィンドウIDが更新されたことを示す前記フロントエンドサーバーからのリセット信号を有している時は、そのアプリケーション画面の画面データに付されているウィンドウIDをそのアプリケーション画面の識別情報として採用するとともに、処理登録済み信号をオンにし、そのアプリケーション画面が前記リセット信号を有していない時は、そのアプリケーション画面に付されているウィンドウIDをヌル信号に変換し、処理登録済み情報をオンにする、クライアント側ウィンドウID管理手段を有していることを特徴とするフロントエンド処理機能を有する情報処理システム。

【請求項5】

アプリケーションプログラムを実行する少なくとも一つのクライアントと業務処理ロジックを実行する少なくとも一つのバックエンドシステムの間、前記クライアントから所定の処理要求を含むアプリケーション画面を受信し、前記処理要求を業務処理ロジックに変換し、所定のバックエンドシステムに実行させ、実行結果を要求を発生したアプリケーションの所定の画面に加入して要求元のクライアントに送信するフロントエンドサーバーを介在させたフロントエンド処理機能を有する情報処理システムにおいて、

前記アプリケーションプログラムは、同一画面内で複数のアプリケーション処理を実行可能である場合は、前記画面のウィンドウIDに対して前記アプリケーション処理ごとにアプリケーションIDを予め決めており、

前記クライアントは、

ブラウザが所定のアプリケーション画面を開く時に、そのアプリケーション画面がウィンドウIDおよびアプリケーションIDが更新されたことを示す前記フロントエンドサーバーからのリセット信号を有している時は、そのアプリケーション画面に付されているウィンドウIDとアプリケーションIDとをそのアプリケーション画面の識別情報として採用するとともに、処理登録済み信号をオンにし、そのアプリケーション画面が前記リセット信号を有していない時は、そのアプリケーション画面に付されているウィンドウIDをヌル信号に変換し、処理登録済み情報をオンにする、クライアント側ウィンドウID管理手段を有していることを特徴とするフロントエンド処理機能を有する情報処理システム。

【請求項6】

前記フロントエンドサーバーは、

前記クライアントに表示させる画面の画面データを記憶する画面データ記憶手段と、前記クライアントから受信した画面から処理要求IDと必要な入力データを抽出し、あるいは前記クライアントに画面を表示させるために必要な処理結果を加入した画面データを生成して出力する画面制御手段と、

処理要求とその処理結果に対応する画面遷移の定義を記憶した画面遷移定義記憶手段と、を有する画面制御部を有していることを特徴とする請求項1～5のいずれかに記載のフロントエンド処理機能を有する情報処理システム。

【請求項7】

前記フロントエンドサーバーは、

前記クライアントから受信した画面の処理要求IDと必要な入力データを前記画面制御手段から取得し、前記処理要求IDに対応するトランザクション処理の要求電文のテンプレートに前記入力データを組み込んで処理要求電文を生成し、所定のバックエンドシステムに前記処理要求電文を送信し、前記バックエンドシステムから処理結果電文を受信し、前記処理結果電文から処理結果を抽出して前記画面制御部に出力する処理中継部を有していることを特徴とする請求項6に記載のフロントエンド処理機能を有する情報処理システム。

【請求項8】

前記フロントエンドサーバーは、

前記クライアントから受信した画面の処理要求IDを前記画面制御手段から取得し、処理要求IDとトランザクション処理を対応させて記憶したトランザクション定義記憶手段を参照して前記処理要求IDに対応するトランザクション処理を特定するトランザクション

10

20

30

40

50

変換手段と、

前記クライアントから受信した画面の入力データ、または、前記バックエンドシステムによる処理結果を記憶保持するデータストアと、

前記トランザクション変換手段から特定されたトランザクション処理の情報を入力し、トランザクション処理とそれに必要な処理要求電文を対応させて記憶した電文変換定義記憶手段と、各処理要求電文または処理結果電文の定義を記憶した電文定義記憶手段とを参照して必要な入力データを組み込んで処理要求電文を生成し、あるいは、前記バックエンドシステムから処理結果電文を受信し、前記処理結果電文から処理結果を抽出して前記データストアに出力する電文変換手段と、

前記電文変換手段から処理要求電文を入力し、所定のバックエンドシステムと接続して処理要求電文を送信し、前記バックエンドシステムから処理結果電文を受信して前記電文変換手段に出力するコネクタと、を有していることを特徴とする請求項6に記載のフロントエンド処理機能を有する情報処理システム。 10

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は大規模で、多様な処理を行うアプリケーションプログラムと、所定の定型的な情報処理の実行を得手とするホストコンピュータとデータベースなどのバックエンドシステムとの間に、アプリケーションプログラムからの処理要求を単純化して中継しバックエンドシステムに処理させ、バックエンドシステムからその処理結果を受信してアプリケーションプログラムがほしい形に変換して返すフロントエンド処理機能を有する情報処理システムに関する。 20

【0002】

特に本発明は、アプリケーションプログラムが一つのブラウザで複数のウィンドウを立ち上げた場合などに、バックエンドシステムからの処理結果を正確にその処理結果を反映すべきウィンドウに表示させることができる、フロントエンド処理機能を有する情報処理システムに関する。

【0003】

【従来の技術】

ユーザから直接操作を受け付けて、さまざまな処理を行うコンピュータシステムは、ユーザインターフェースに関する部分と、具体的な業務処理に関する業務処理ロジックとに分けることができる。 30

【0004】

ユーザインターフェースの設計基準は、ユーザに与える印象、操作性等さまざまな要素がある。そのため、ユーザインターフェースの設計には高い自由度が求められる。また、ユーザインターフェースは、さまざまな理由から頻繁に変更されるものである。

【0005】

これに対して業務処理ロジックは、定型的な処理が多く、単純化できれば高性能なコンピュータによって高速に処理することができる。この業務処理ロジックは、ユーザインターフェースの変更に伴って変更する必要はなく、むしろ、ユーザインターフェースを変更する度に、業務処理ロジックまで変更しなければならないとすれば、不便である。 40

【0006】

【発明が解決しようとする課題】

上記問題に対し、本願出願人は、大規模で多様なアプリケーションプログラムを実行するクライアントと、所定の定型的な情報処理を得手とするホストコンピュータなどのバックエンドシステムとの間に、クライアントとバックシステムとの間の処理要求や処理結果を中継するフロントエンドサーバーを介在させる情報処理システムを提案した。前記フロントエンドサーバーは、アプリケーションプログラムからの処理要求を受信してバックエンドシステムが処理可能な処理要求電文に変換し、バックエンドシステムの処理結果を受信してアプリケーションプログラムの所定の画面に処理結果を組み入れてクライアントに送信 50

するものであった。このフロントエンドサーバーの作用により、バックエンドシステムの業務処理ロジックとアプリケーションプログラムの画面等が分離され、バックエンドシステムから独立してアプリケーションプログラムのユーザーインターフェースを変更し得るものになった。

【0007】

上記本願出願人が提案した情報処理システムは、アプリケーションプログラムのユーザーインターフェースが要求する複雑さと多様さと柔軟性と、バックエンドシステムが要求する単純性、不変性の双方を同時に満たすものであった。

【0008】

しかし、上記本願出願人が先に提案した情報処理システムは、一つの処理要求について画面遷移が一義的に定まる処理を対象とするものであった。 10

【0009】

ところが、アプリケーションプログラムの処理内容の高度化と大規模化により、現在では同一のブラウザによって複数のウィンドウを立ち上げ、それぞれのウィンドウからほぼ同時に処理要求を発することが多くなった。あるいは、一つのフレーム内に複数の画面を配置し、それぞれの画面から処理要求を発することもしばしばある。

【0010】

クライアントが処理要求を順次発し、バックエンドシステムが順次処理を行って処理結果を変える場合は、一つの処理要求に対して一つの画面遷移のみが定義されていても問題は生じない。 20

【0011】

しかし、複数のウィンドウが同時に立ち上げられ、複数のウィンドウから同時に処理要求が発せられる場合は、一つの処理要求に対して一つの画面遷移のみが定義されていると、バックエンドシステムからの処理結果がクライアントに返されるが、クライアント側では複数同時に存在するウィンドウのいずれに処理結果を表示していいのか判断できないという問題が生じる。たとえば複数のウィンドウを立ち上げて、各ウィンドウから銀行口座A、B、Cの残高照会をする場合に、銀行口座A、B、Cの残高が異なるウィンドウに表示されることがある。

【0012】

そこで、本発明が解決しようとする課題は、複数のウィンドウが同時に立ち上げられ、複数の処理要求がほぼ同時に発せられた場合でも、処理結果を正しいウィンドウに表示させることができるフロントエンド処理機能を有する情報処理システムを提供することにある。 30

【0013】

【課題を解決するための手段】

本発明によるフロントエンド処理機能を有する情報処理システムは、アプリケーションプログラムを実行する少なくとも一つのクライアントと業務処理ロジックを実行する少なくとも一つのバックエンドシステムの間を介在し、クライアントから所定の処理要求を含むアプリケーション画面を受信し、前記処理要求を業務処理ロジックに変換し、所定のバックエンドシステムに実行させ、実行結果を要求を發したアプリケーションの所定の画面に加入して要求元のクライアントに送信するフロントエンドサーバーを有し、 40

前記フロントエンドサーバーは、

前記クライアントから送信された画面データがクライアント側で新たに開いたウィンドウか否かを画面データに付されたウィンドウIDとアプリケーションIDとフロントエンドサーバーに備えられたウィンドウID管理テーブルとを比較して判断し、新たに開いたウィンドウと判断した場合は新たにウィンドウIDを付与するとともに処理中オブジェクトを生成し、ウィンドウIDとアプリケーションIDと処理要求IDの識別情報の組を処理要求に付して所定のバックエンドシステムに実行させ、処理結果を前記ウィンドウIDとアプリケーションIDと処理要求IDの識別情報の組によって特定されるアプリケーションの所定の画面に加入して前記ウィンドウIDとアプリケーションIDと処理要求IDの 50

識別情報の組を付して要求元のクライアントに送信するサーバー側ウィンドウID管理手段を有していることを特徴とするものである。

【0014】

本発明によるフロントエンド処理機能を有する情報処理システムは、アプリケーションプログラムを実行する少なくとも一つのクライアントと業務処理ロジックを実行する少なくとも一つのバックエンドシステムの間、前記クライアントから所定の処理要求を含むアプリケーション画面を受信し、前記処理要求を業務処理ロジックに変換し、所定のバックエンドシステムに実行させ、実行結果を要求を発生したアプリケーションの所定の画面に加入して要求元のクライアントに送信するフロントエンドサーバーを介在させたフロントエンド処理機能を有する情報処理システムにおいて、

10

前記アプリケーションプログラムは、同一画面内で複数のアプリケーション処理を実行可能である場合は、前記画面のウィンドウIDに対して前記アプリケーション処理ごとにアプリケーションIDを予め決めており、

前記クライアントは、

ブラウザが所定のアプリケーション画面を開く時に、そのアプリケーション画面がウィンドウIDおよびアプリケーションIDが更新されたことを示す前記フロントエンドサーバーからのリセット信号を有している時は、そのアプリケーション画面に付されているウィンドウIDとアプリケーションIDとをそのアプリケーション画面の識別情報として採用するとともに、処理登録済み信号をオンにし、そのアプリケーション画面が前記リセット信号を有していない時は、そのアプリケーション画面に付されているウィンドウIDをヌ

20

【0015】

本発明によれば、フロントエンドサーバーが、クライアントから送信された画面データがクライアント側で新たに開いたウィンドウか否かを画面データに付されたウィンドウIDとアプリケーションIDとフロントエンドサーバーに備えられたウィンドウID管理テーブルとを比較して判断し、新たに開いたウィンドウと判断した場合は新たにウィンドウIDを付与するとともに処理中オブジェクトを生成し、ウィンドウIDとアプリケーションIDと処理要求IDの識別情報の組を処理要求に付して所定のバックエンドシステムに実行させ、処理結果を前記ウィンドウIDとアプリケーションIDと処理要求IDの識別情報

30

【0016】

一方、クライアントは、ブラウザが所定のアプリケーション画面を開く時に、そのアプリケーション画面がウィンドウIDおよびアプリケーションIDが更新されたことを示す前記フロントエンドサーバーからのリセット信号を有している時は、そのアプリケーション画面に付されているウィンドウIDとアプリケーションIDとをそのアプリケーション画面の識別情報として採用するとともに、処理登録済み信号をオンにし、そのアプリケーション画面が前記リセット信号を有していない時は、そのアプリケーション画面に付されて

40

【0017】

これにより、クライアントで新規にウィンドウを開いた時も、画面のコピーによって新規にウィンドウを開いた時も、フロントエンドサーバーによって新規のウィンドウであると認識され、新たなウィンドウIDを付与される。

【0018】

バックエンドシステムの処理結果は処理要求のウィンドウID/アプリケーションIDの情報を付されてクライアントに送信されるので、クライアント側では処理結果を正しいウィンドウに表示させることができる。

【0019】

50

【発明の実施の形態】

次に、本発明によるフロントエンド処理機能を有する情報処理システムの実施の形態について以下に説明する。

【0020】

図1は、本発明による一実施形態によるフロントエンド処理機能を有する情報処理システムの全体の構成を示したものである。

【0021】

図1に示すように、本実施形態による情報処理システム1は、複数のクライアント2と、たとえばリレーショナルデータベース(RDB)、ホストコンピュータ、オンライントランザクション処理装置等のバックエンドシステム3と、前記クライアント2とバックエンドシステム3の間に設けられたフロントエンドサーバー4とを有している。

10

【0022】

クライアント2は種々のアプリケーションプログラムを実行するコンピューター、端末である。

【0023】

バックエンドシステム3は、複数のシステムからなり、それぞれのシステムは特定の定型処理を行うように特化されている。たとえば、RDBはデータ検索等を行い、ホストコンピュータは業務処理を行い、オンライントランザクション処理装置は所定のオンライントランザクション処理を行うように特化され、大量の処理を高性能なコンピューターによって高速に処理するように構成されている。

20

【0024】

フロントエンドサーバー4は、クライアント2(実際にはクライアント2上で作動するアプリケーションプログラム)からの処理要求をバックエンドシステム3への処理要求電文に変換し、その処理要求電文をバックエンドシステム3に送信し、その処理結果電文をバックエンドシステム3から受信し、処理結果をアプリケーションプログラムが認識できる形に変換してクライアント2に送信する。

【0025】

クライアント2からの処理要求が実際には複数のバックエンドシステム3を稼働させる必要がある時は、フロントエンドサーバー4は、一つの処理要求に対して複数の処理要求電文を生成し、これらの処理要求電文を複数のバックエンドシステム3に送信し、それらのバックエンドシステム3から処理要求電文を受信し、処理結果を組み立ててクライアント2に返す。

30

【0026】

クライアント2からの処理要求はWebページによるのが好ましい。ユーザーは、クライアント2上でWebページを表示させ、当該Webページに必要なデータを入力し、その入力データを含んだWebページをフロントエンドサーバー4に送信する。フロントエンドサーバー4は、前記入力データを含んだWebページから必要な情報(入力データと処理要求の種類)を抽出し、バックエンドシステム3に送信し、バックエンドシステム3から処理結果を受信してWebページに組み入れてクライアント2に返信する。

【0027】

本発明の情報処理システム1は、クライアント2が一つのブラウザによって複数のウィンドウを立ち上げて複数の処理要求を発する場合、あるいは、一つのフレーム内に複数の画面が含まれて複数の処理要求を発することができる場合に対し、フロントエンドサーバー4からの処理結果を組み入れた画面を正しいウィンドウに表示させることができる。

40

【0028】

フロントエンドサーバー4からの処理結果を正しいウィンドウに表示させるために、情報処理システム1は以下の構成を有している。

【0029】

クライアント2は、クライアント側ウィンドウID管理手段5を有している。

【0030】

50

また、フロントエンドサーバー 4 は、処理中オブジェクトの管理モジュール 6 と処理モジュール 7 とを有している。

【0031】

クライアント側ウィンドウ ID 管理手段 5 は、クライアント 2 側でブラウザが立ち上げたウィンドウに識別情報を付して管理する手段である。また、クライアント側ウィンドウ ID 管理手段 5 は、ブラウザが開いたウィンドウがバックエンドシステム 3 から送信されたウィンドウか、すでに開かれている画面をコピーしたウィンドウかを識別し、それによって識別情報を操作する機能を有している手段である。

【0032】

クライアント側ウィンドウ ID 管理手段 5 の具体的な処理については後述する。

10

【0033】

処理中オブジェクトの管理モジュール 6 は、フロントエンドサーバー 4 が処理している途中の処理要求を管理する手段である。処理中オブジェクトの管理モジュール 6 は、各処理中の処理要求をオブジェクト（処理中オブジェクト）として管理し、クライアント 2 の各ウィンドウからの処理要求に対して一対一対応の処理中オブジェクトを生成し、終了した処理中オブジェクトを削除等する。ここで、処理中オブジェクトは、進行中の処理プロセスを表すものであって、問い合わせに対して処理状況や処理要求の種別等を返すオブジェクトである。

【0034】

処理中オブジェクトの管理モジュール 6 の具体的な処理については後述する。

20

【0035】

処理モジュール 7 は、処理中オブジェクトの管理モジュール 6 から、処理要求を受け取り、バックエンドシステム 3 に対して処理を要求し、その処理結果を受信してクライアント 2 への Web ページに組み入れて処理中オブジェクトの管理モジュール 6 に返す処理部である。

【0036】

処理モジュール 7 の具体的な処理については後述する。

【0037】

図 2 に、処理中オブジェクトの管理モジュール 6 の構成を示す。

【0038】

処理中オブジェクトの管理モジュール 6 は、サーバー側ウィンドウ ID 管理手段 8 と、処理要求定義記憶手段 9 と、ウィンドウ ID 管理テーブル 10 と、処理中オブジェクト記憶手段 11 とを有している。

30

【0039】

サーバー側ウィンドウ ID 管理手段 8 は、クライアント 2 から受信した画面データがクライアント 2 で新たに開かれたウィンドウか否かを判定し、新たに開かれたウィンドウであれば、ウィンドウ ID を付与して管理し、そのウィンドウに対応する処理中オブジェクトを生成する手段である。新たに開かれたウィンドウか否の情報は既に開かれているウィンドウのウィンドウ ID を記憶したウィンドウ ID 管理テーブル 10 を参照して判断する。

【0040】

処理中オブジェクトは、サーバー側ウィンドウ ID 管理手段 8 によって生成、削除され、処理中オブジェクト記憶手段 11 によって記憶管理される。

40

【0041】

処理要求 ID 定義記憶手段 9 は、ウィンドウ ID またはアプリケーション ID に対応する処理要求 ID の定義を記憶管理する手段である。

【0042】

ここで、ウィンドウ ID は各ウィンドウに付される識別情報であり、アプリケーション ID はウィンドウ中の処理に付される識別情報である。ウィンドウが単一の処理のみを行う画面である場合には、アプリケーション ID を省略することができる。その場合は、ウィンドウ ID と処理要求 ID とが直接対応して定義される。

50

【 0 0 4 3 】

図 3 は処理モジュール 7 の構成を示している。

【 0 0 4 4 】

処理モジュール 7 は、画面制御部 1 2 と処理中継部 1 3 とを有している。

【 0 0 4 5 】

画面制御部 1 2 は、クライアント 2 から受信した画面から必要な情報を抽出し、あるいはクライアント 2 に送信する画面を選択し、必要な情報を組み入れる処理部である。

【 0 0 4 6 】

処理中継部 1 3 は、クライアント 2 から受信した画面データが要求する処理要求を適当なバックエンドシステム 3 に行わせ、バックエンドシステム 3 から処理結果を受信して画面制御部 1 2 に出力する処理部である。

10

【 0 0 4 7 】

画面制御部 1 2 は、画面制御手段 1 4 と画面データ記憶手段 1 5 と画面遷移定義記憶手段 1 6 とを有している。

【 0 0 4 8 】

画面制御手段 1 4 は、クライアント 2 から受信した画面から処理要求 ID と必要な入力データを抽出し、あるいはクライアント 2 に画面を表示させるために必要な処理結果を加入した画面データを生成して出力する手段である。

【 0 0 4 9 】

画面データ記憶手段 1 5 は、クライアントに表示させる画面の画面データを記憶する手段である。画面データは、例えば HTML (HyperText Markup Language) 形式等であらかじめ作成されている。各画面データに基づいて表示された画面は、ユーザからの入力操作を受け付け、画面制御手段 1 4 がその操作内容に応じたフロントトランザクション処理をコールする。

20

【 0 0 5 0 】

画面遷移定義記憶手段 1 6 は、処理要求とその処理結果に対応する画面遷移の定義を記憶する手段である。

【 0 0 5 1 】

上記構成により、画面制御部 1 2 は、クライアント 2 から受信した画面データの処理要求 ID を特定し、あるいは入力されているデータ (入力データ) 等を抽出し、あるいは、クライアント 2 に送信する画面データのファイルを特定し、バックエンドシステム 3 の処理結果を組み入れて画面データを生成する。

30

【 0 0 5 2 】

処理中継部 1 3 は、データストア 1 7 と、トランザクション変換手段 1 8 と、電文変換手段 1 9 と、コネクタ 2 0 と、データストア定義記憶手段 2 1 と、電文変換定義記憶手段 2 2 と、電文定義記憶手段 2 3 と、トランザクション定義記憶手段 2 4 とを有している。

【 0 0 5 3 】

データストア 1 7 は、クライアント 2 から受信した画面データから抽出した入力データを記憶する手段である。このデータストア 1 7 は、サーバー側ウィンドウ ID 管理手段 8 によってウィンドウ ID 単位に複製される。そして該当するウィンドウに対応するウィンドウ ID に関連づけられたデータストア 1 7 の一つが処理モジュール 7 上で選択され、使用される。

40

【 0 0 5 4 】

トランザクション変換手段 1 8 は、クライアント 2 から受信した画面の処理要求 ID を画面制御手段 1 4 から取得し、処理要求 ID とトランザクション処理を対応させて記憶したトランザクション定義記憶手段 2 4 を参照して前記処理要求 ID に対応するトランザクション処理を特定する手段である。

【 0 0 5 5 】

電文変換手段 1 9 は、トランザクション変換手段 1 8 からトランザクション処理の情報を入力し、トランザクション処理とそれに必要な処理要求電文を対応させて記憶した電文変

50

換定義記憶手段22と、各処理要求電文または処理結果電文の定義を記憶した電文定義記憶手段23とを参照して必要な入力データを組み込んで処理要求電文を生成し、あるいは、バックエンドシステム3から処理結果電文を受信し、処理結果電文から処理結果を抽出してデータストア17に出力する手段である。

【0056】

データストア定義記憶手段21は、画面データに含まれるデータの入出力項目、データ型、使用画面等の定義を記憶する手段である。

【0057】

上記構成により、処理中継部13は、クライアント2から受信した画面の処理要求IDと必要な入力データを画面制御手段14から取得し、前記処理要求IDに対応するランザクション処理の要求電文のテンプレートに入力データを組み込んで処理要求電文を生成し、所定のバックエンドシステム3に前記処理要求電文を送信し、バックエンドシステム3から処理結果電文を受信し、その処理結果電文から処理結果を抽出してデータストア17を介して画面制御手段14に出力する。

10

【0058】

以下に情報処理システム1の処理の流れを説明する。

【0059】

図4は、クライアント側ウィンドウID管理手段5と、フロントエンドサーバーの処理中オブジェクトの管理モジュール6の間の処理の流れを示している。図4中、「WID」はウィンドウID、「APIID」はアプリケーションIDを示す。

20

【0060】

図4において、クライアント2が所定の画面を新規に開いて処理中オブジェクトの管理モジュール6に送信するときは、その画面データにウィンドウIDがヌル（入力されていない状態；WID=" "）とアプリケーションIDがたとえばP1（APIID=" P1 "）の情報を付して送信する（ステップS100）。なお、APIIDはアプリケーションプログラムによってウィンドウ（画面）に対応して予め定められている。

【0061】

上記画面データを受信した処理中オブジェクトの管理モジュール6では、WID=" " であるので、ウィンドウID管理テーブル10に新規のウィンドウID（図4の例ではW1）を登録し、WID=" W1 " , APIID=" P1 " にし、処理登録済み信号をオフ（OnLoadCtr=0）にするとともに、WID=" W1 " / APIID=" P1 " に対応する処理中オブジェクト（DSB）を生成する（ステップS110）。上記処理はサーバー側ウィンドウID管理手段8によって行われる。

30

【0062】

処理中オブジェクトの管理モジュール6は、要求された画面にOnLoadCtr=0 , WID=" W1 " , APIID=" P1 " の情報を付してクライアント2に送信する（ステップS120）。

【0063】

クライアント2のクライアント側ウィンドウID管理手段5は、画面が開かれた時に、画面データがリセット信号を有しているか否かを判断し、以下のような処理を行う。なお、リセット信号は、WID , APIID等がフロントエンドサーバー4によって更新されたことを示す信号であって、処理登録済み信号をオフにすることでリセット信号に代えてもよい。図4の例はこの方法によりリセットの有無を判断している。

40

【0064】

すなわち、クライアント側ウィンドウID管理手段5は、アプリケーション画面がウィンドウIDが更新されたことを示すリセット信号を有している時は、そのアプリケーション画面の画面データに付されているウィンドウIDとアプリケーションIDをそのアプリケーション画面の識別情報として採用するとともに、処理登録済み信号をオンにし、一方、そのアプリケーション画面が前記リセット信号を有していない時は、そのアプリケーション画面に付されているウィンドウIDをヌル信号に変換し、処理登録済み情報をオンにす

50

る。

【0065】

この場合、OnLoadCtr = 0 (リセット信号を有している) ので、クライアント側ウィンドウID管理手段5は、その画面データに付されているウィンドウIDとアプリケーションIDをそのアプリケーション画面の識別情報として採用するとともに(WID = "W1", APID = "P1")、処理登録済み信号をオンにし(OnLoadCtr = 1)にする(ステップS130)。

【0066】

次に、クライアント2でステップS130の画面から次の処理要求の画面データをフロントエンドサーバー4に送る時は、その画面に付されている情報WID = "W1", APID = "P1"を画面データに付して送信する(ステップS140)。 10

【0067】

上記画面データを受信した処理中オブジェクトの管理モジュール6は、WID = "W1", APID = "P1"という処理要求が既に処理中であるか否かをウィンドウID管理テーブル10を参照して判断する。この場合、既にWID = "W1", APID = "P1"がウィンドウID管理テーブル10に登録されているので(ステップS110)、処理中オブジェクトの管理モジュール6はWID = "W1" / APID = "P1"に対応する処理中オブジェクト(DSB)の処理要求IDを処理モジュール7に出力し(ステップS150)、処理モジュール7が出力した画面データに情報OnLoadCtr = 0, WID = "W1", APID = "P2"の情報を付してクライアント2に送信する(ステップ1 20 60)。

【0068】

上記処理結果を組み込んだ画面データを受信したクライアント2ではクライアント側ウィンドウID管理手段5が、OnLoadCtr = 0 (リセット信号を有している) であるので、その画面データに付されているウィンドウIDとアプリケーションIDをそのアプリケーション画面の識別情報として採用するとともに(WID = "W1", APID = "P2")、処理登録済み信号をオンにし(OnLoadCtr = 1)にする(ステップS170)。

【0069】

次に、クライアント2がステップS170の画面から次の処理要求の画面データをフロント 30
エンドサーバー4に送る時は、その画面に付されている情報WID = "W1", APID = "P2"を画面データに付して送信する(ステップS180)。

【0070】

上記画面データを受信したフロントエンドサーバー4では、処理中オブジェクトの管理モジュール6がウィンドウID管理テーブル10を参照し、同一ウィンドウ(WID = "W1")の次の処理(P2)であることを認識し、WID = "W1" / APID = "P2"に対応する処理中オブジェクト(DSB)を生成し、ステップ150と同様の処理を行う(ステップS190)。

【0071】

以上のように、一つのウィンドウを開いて、通常の処理の手順に従って処理を行い画面が 40
遷移するときは、同一のウィンドウID WID = "W1" が保持され、処理がP1、P2へと変遷する。

【0072】

この場合には、バックエンドシステム3の処理結果を組み込んだ画面のデータは、ウィンドウIDと処理要求IDの識別情報の組(WID = "W1" / APID = "P1"、WID = "W1" / APID = "P2"等)を識別情報として有しているので、クライアント2側ではクライアント側ウィンドウID管理手段5が上記画面データを正しいウィンドウに表示させることができる。

【0073】

また、クライアント2のブラウザが複数のウィンドウを立ち上げた場合には、各ウィンド 50

ウが上記ステップS100～S190のように処理され、各ウィンドウが異なるウィンドウIDを有しているため、クライアント2はフロントエンドサーバ4から受信した処理結果の画面データをいずれのウィンドウに表示させるべきかを判断でき、正しいウィンドウに処理結果を表示させることができる。

【0074】

一方、クライアント2でブラウザの画面をコピーして新たにウィンドウを開くことがある。この場合は、新しいウィンドウの画面データはフロントエンドサーバ4から送られない。

【0075】

クライアント2では、開かれている画面はいずれも処理登録済み信号(OnLoadCtrl)、ウィンドウID(WID)、アプリケーションID(APID)の情報を有しており、これらの情報はコピーによって新しく開かれたウィンドウの画面にも付される。

【0076】

クライアント側ウィンドウID管理手段5は、コピーによってか否かに関わらず新規にウィンドウが開かれると、そのウィンドウに付されている処理登録済み信号(OnLoadCtrl)、ウィンドウID(WID)、アプリケーションID(APID)を判断する。この場合、元のウィンドウのOnLoadCtrl=1の情報がそのままコピーされているので、クライアント側ウィンドウID管理手段5は、そのウィンドウに付されているウィンドウIDをヌル信号に変換し(WID=" ")、処理登録済み情報をオン(OnLoadCtrl=1)にする(ステップS210)。なお、アプリケーションIDはそのまま採用する(APID="P1")。

【0077】

次に、このコピーによって生成されたウィンドウ上からフロントエンドサーバ4に対して処理要求を送信する場合には、上記ウィンドウIDとアプリケーションID(WID=" ", APID="P1")が送信される(ステップS220)。

【0078】

上記画面データを受信したフロントエンドサーバ4では、画面データに付されたウィンドウIDがヌルになっているので(WID=" ")、処理中オブジェクトの管理モジュール6が新たにウィンドウが開かれたことを検知し、元の画面のウィンドウID(WID="W1")と異なるウィンドウID(WID="W2")を付与する(ステップS230)。

【0079】

この処理要求に対する返信の画面データには、リセット信号、ウィンドウID、アプリケーションIDが付される(ステップS240)。

【0080】

以上のように、クライアント2において、コピーによって新たにウィンドウを開く場合は、コピーされた時にウィンドウIDがヌルに変更されるので、その画面データを送信する時には、コピー元のウィンドウと混同しないように新たなウィンドウIDが付与される。

【0081】

これにより、コピーによって複数のウィンドウを生成した場合であっても、各ウィンドウが異なるウィンドウIDを有することになり、所定のウィンドウからの処理要求がフロントエンドサーバ4に送られた場合には、その処理結果が正しいウィンドウに表示されるのである。

【0082】

以上から明らかなように、本発明によれば、ブラウザがフロントエンドサーバ4にアクセスして新たにウィンドウを開く場合でも、コピーによって複数のウィンドウを開く場合でも、各ウィンドウごとに識別情報が付されることになり、これにより、バックエンドシステム3の処理結果が正しいウィンドウに表示される。

【0083】

図5は、処理中オブジェクトの管理モジュール6のオブジェクト管理の処理の流れを示し

10

20

30

40

50

ている。

【0084】

図5に示すように、サーバー側ウィンドウID管理手段8は、クライアント2から画面データを受信すると、画面データに付されたウィンドウIDとウィンドウID管理テーブル10とを比較参照する。前記ウィンドウIDがウィンドウID管理テーブル10に登録されていなければ、サーバー側ウィンドウID管理手段8は、新しいウィンドウIDを設定し、管理を開始する(ステップS300)。図5の例では、クライアント2に二つのウィンドウが開かれ、ウィンドウIDとしてW1, W2がウィンドウID管理テーブル10に登録される。

【0085】

同時に、サーバー側ウィンドウID管理手段8は、ウィンドウID管理テーブル10において各ウィンドウへのアクセスタイムの管理も開始する。

【0086】

さらに、サーバー側ウィンドウID管理手段8は、ウィンドウID/アプリケーションID/処理要求のオブジェクト25(処理中オブジェクト25)を生成する。

【0087】

次に、クライアント2からウィンドウW1の処理要求を受信すると、サーバー側ウィンドウID管理手段8は、ウィンドウW1からのアクセス時刻と前回のアクセス時刻とを比較し、所定の時間幅以内のアクセスであれば、処理要求を許容し、処理要求に応じて処理結果をクライアント2のウィンドウW1に返信する(ステップS310)。

【0088】

上記ウィンドウW1のアクセス時刻をチェックするタイミングで、サーバー側ウィンドウID管理手段8はウィンドウW2のアクセスをもチェックする。

【0089】

ウィンドウW2が一定時間以上アクセスしていなければ、サーバー側ウィンドウID管理手段8はウィンドウID管理テーブル10上のウィンドウW2についてタイムアウトの信号を書き込み、コネクタ20に通知し、さらに、ウィンドウW2の処理中オブジェクト25も削除する(ステップS310)。

【0090】

クライアント2から新たな画面データを受信した時も、サーバー側ウィンドウID管理手段8はウィンドウID管理テーブル10にアクセスし、このタイミングで各ウィンドウのアクセスタイムをチェックする(ステップS320)。

【0091】

また、図5に示すように、既にタイムアウトの状態になっているウィンドウW2からアクセスがあった時は、サーバー側ウィンドウID管理手段8は、ウィンドウID管理テーブル10を参照して当該ウィンドウが既にタイムアウトの状態になっていることを検知し、タイムアウト画面をクライアント2に送信する(ステップS330)。

【0092】

次に、処理モジュール7の処理について概略説明する。

【0093】

なお、以下の説明は図3を参照することにより明らかに理解できる。

【0094】

処理モジュール7は、処理中オブジェクトの管理モジュール6からクライアント2からの処理要求を受け付け、バックエンドシステム3に処理をさせ、バックエンドシステム3から処理結果を受け取り、処理中オブジェクトの管理モジュール6に返却する。

【0095】

画面制御手段14は、処理中オブジェクトの管理モジュール6から所定の処理を要求する画面のデータを取得する。画面制御手段14は、その画面データの処理要求を特定し、処理要求IDという形でトランザクション変換手段18に出力する。また、画面制御手段14は、たとえば検索処理を要求する画面に検索すべき対象の情報が入力される時のように

10

20

30

40

50

、ユーザーによってデータが入力されている場合は、その入力データを抽出してデータストア 17 に出力する。

【0096】

トランザクション変換手段 18 は、画面制御手段 14 から処理要求を処理要求 ID という形で受け取り、その処理要求に対応するために必要なトランザクション処理をトランザクション定義記憶手段 24 を参照して取得する。一つの処理要求に対して複数のトランザクション処理が必要な場合は複数のトランザクション処理を取得する。

【0097】

トランザクション変換手段 18 は、必要なトランザクション処理の情報を電文変換手段 19 に出力する。

10

【0098】

電文変換手段 19 は、電文変換定義記憶手段 22 と電文定義記憶手段 23 を参照して、バックエンドシステム 3 に上記トランザクション処理を処理させるための処理要求電文を生成する。なお、電文変換定義記憶手段 22 には、どのトランザクション処理がどの処理要求電文に対応するかの情報が記憶され、電文定義記憶手段 23 には、前記電文変換定義記憶手段 22 によって特定された処理要求電文のテンプレート、すなわち、処理を要求する電文のベース部分と、前記データ（たとえば検索を要求する対象のデータ）を組み入れるべき電文の構造が記憶されている。

【0099】

電文変換手段 19 は、前記処理要求電文のテンプレートを取得し、データストア 17 から入力データを取得し、これらを組み合わせて完全な処理要求電文を生成し、コネクタ 20 に出力する。

20

【0100】

コネクタ 20 は、上記処理要求電文をそれを処理すべきバックエンドシステム 3 に送信し、当該バックエンドシステム 3 とコネクションを維持する。バックエンドシステム 3 から処理結果電文を受信すると、コネクタ 20 はその処理結果電文を電文変換手段 19 に出力する。

【0101】

電文変換手段 19 は、電文変換定義記憶手段 22 と電文定義記憶手段 23 とを参照し、その処理結果電文の構造を取得することにより、その処理結果電文から処理結果のデータを抽出してデータストア 17 に出力する。

30

【0102】

画面制御手段 14 は、前記データストア 17 からバックエンドシステム 3 の処理結果のデータを取得し、画面遷移定義記憶手段 16 を参照することにより、処理結果を組み込むべき画面（遷移画面）の情報を取得する。

【0103】

画面制御手段 14 は画面遷移定義記憶手段 16 から取得した遷移画面の情報に基づいて画面データ記憶手段 15 から当該画面のデータを取得する。前記画面データを取得した画面制御手段 14 は、データストア 17 から処理結果のデータを取得し、画面データ記憶手段 15 から取得した画面データに処理結果データを組み込んでクライアント 2 に表示させる画面データを生成する。

40

【0104】

画面制御手段 14 は、前記生成した画面データを処理中オブジェクトの管理モジュール 6 に出力する。

【0105】

以上の処理モジュール 7 によって取り扱われる処理要求は、ウィンドウ ID とアプリケーション ID と処理要求 ID の識別情報の組、アプリケーション ID が不要ない場合はウィンドウ ID と処理要求 ID の識別情報の組を付され、この識別情報の組によって識別される。

【0106】

50

画面データを入力した処理中オブジェクトの管理モジュール6は、処理モジュール7から処理結果を組み込んだ画面データを受け取り、識別情報の組によって既に説明したようにして表示させるべきウィンドウの情報、すなわちウィンドウID / アプリケーションIDを画面データに付してクライアント2に送信する。

【0107】

これにより、クライアント2側ではクライアント側ウィンドウID管理手段5によってウィンドウIDが管理されているので、処理結果の画面データが正しいウィンドウに表示される。

【0108】

【発明の効果】

以上の説明から明らかなように、本発明のフロントエンド処理機能を有する情報処理システムによれば、バックエンドシステムの業務処理ロジックとアプリケーションプログラムの画面等が分離され、バックエンドシステムから独立してアプリケーションプログラムのユーザーインターフェースを変更し得る利点を有するとともに、クライアントのブラウザが複数の画面を同時に立ち上げて複数の処理を同時にフロントエンドサーバーに要求した場合でも、バックエンドシステムの処理結果をクライアント側で正しいウィンドウに表示させることができる。

【図面の簡単な説明】

【図1】本発明の一実施形態によるフロントエンド処理機能を有する情報処理システムの全体構成を示すブロック図。

【図2】処理中オブジェクトの管理モジュールの構成例を示すブロック図。

【図3】処理モジュールの構成例を示すブロック図。

【図4】クライアント側ウィンドウID管理手段によるクライアント側のウィンドウID管理の流れを示した図。

【図5】サーバー側ウィンドウID管理手段によるサーバー側のウィンドウID管理の流れを示した図。

【符号の説明】

- 1 情報処理システム
- 2 クライアント
- 3 バックエンドシステム
- 4 フロントエンドサーバー
- 5 クライアント側ウィンドウID管理手段
- 6 処理中オブジェクトの管理モジュール
- 7 処理モジュール
- 8 サーバー側ウィンドウID管理手段
- 9 処理要求ID定義記憶手段
- 10 ウィンドウID管理テーブル
- 11 処理中オブジェクト記憶手段
- 12 画面制御部
- 13 処理中継部
- 14 画面制御手段
- 15 画面データ記憶手段
- 16 画面遷移定義記憶手段
- 17 データストア
- 18 トランザクション変換手段
- 19 電文変換手段
- 20 コネクタ
- 21 データストア定義記憶手段
- 22 電文変換定義記憶手段
- 23 電文定義記憶手段

10

20

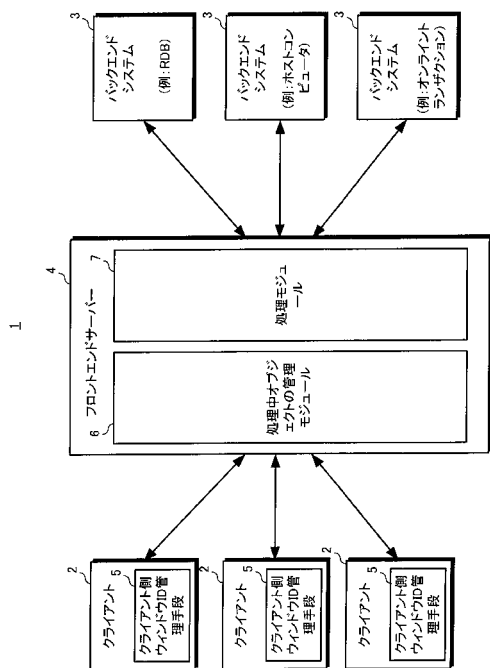
30

40

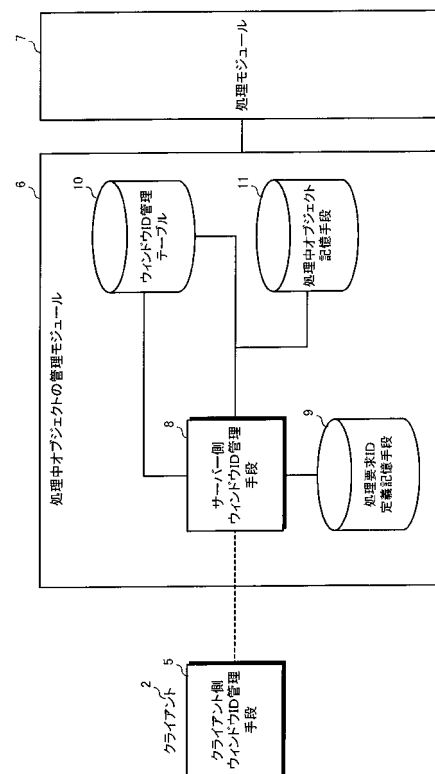
50

- 2 4 トランザクション定義記憶手段
- 2 5 処理中オブジェクト

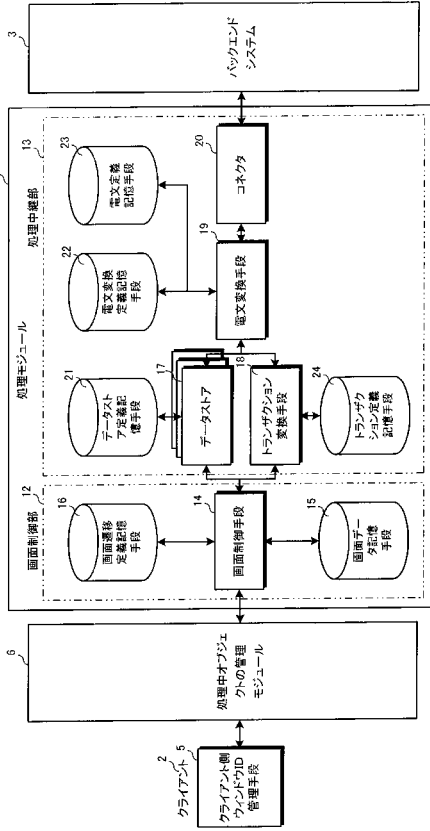
【 図 1 】



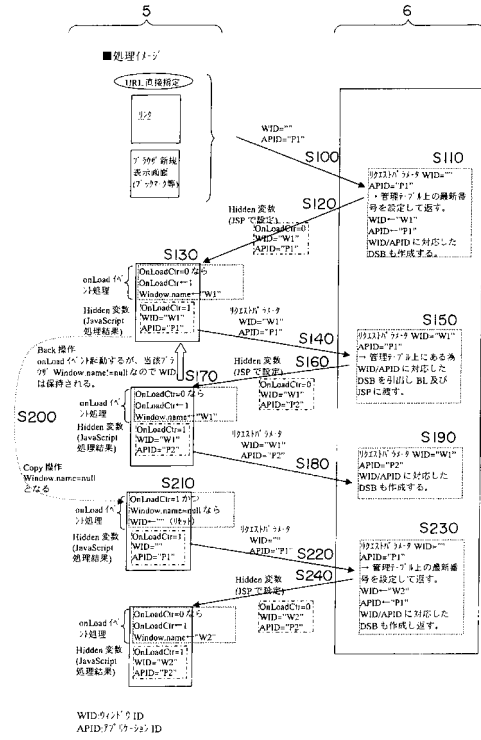
【 図 2 】



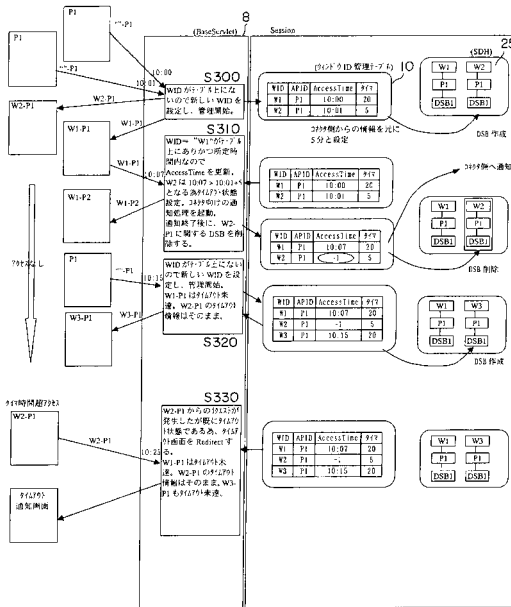
【図3】



【図4】



【図5】



フロントページの続き

- (72)発明者 瀧澤 淳一
東京都千代田区大手町二丁目2番1号 株式会社野村総合研究所内
- (72)発明者 一丁田 章
東京都千代田区大手町二丁目2番1号 株式会社野村総合研究所内
- (72)発明者 並河 英二
東京都千代田区大手町二丁目2番1号 株式会社野村総合研究所内
- (72)発明者 高 畠 勇人
東京都千代田区大手町二丁目2番1号 株式会社野村総合研究所内
- (72)発明者 渡辺 克俊
東京都千代田区大手町二丁目2番1号 株式会社野村総合研究所内
- (72)発明者 坂田 浩慈
東京都千代田区大手町二丁目2番1号 株式会社野村総合研究所内
- Fターム(参考) 5B085 AA01 BC02 BE07 CE07
5B098 AA08 GA01 GC01 GC14 GC16