

(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2017-107572

(P2017-107572A)

(43) 公開日 平成29年6月15日 (2017.6.15)

(51) Int.Cl. F I テーマコード (参考)  
**G06F 13/00 (2006.01)** G06F 13/00 351N 5B089

審査請求 有 請求項の数 21 O L (全 24 頁)

<p>(21) 出願番号 特願2016-248832 (P2016-248832)</p> <p>(22) 出願日 平成28年12月22日 (2016.12.22)</p> <p>(62) 分割の表示 特願2012-533444 (P2012-533444) の分割</p> <p>原出願日 平成22年10月15日 (2010.10.15)</p> <p>(31) 優先権主張番号 61/252, 624</p> <p>(32) 優先日 平成21年10月16日 (2009.10.16)</p> <p>(33) 優先権主張国 米国 (US)</p> <p>(31) 優先権主張番号 12/905, 319</p> <p>(32) 優先日 平成22年10月15日 (2010.10.15)</p> <p>(33) 優先権主張国 米国 (US)</p> <p>(特許庁注：以下のものは登録商標)</p> <p>1. JAVASCRIPT</p> <p>2. ActiveX</p> <p>3. Modbus</p>	<p>(71) 出願人 512141699                  リアル イノベーションズ インターナショナル エルエルシー                  セントクリストファー, ネービス, チャールズタウン, メイン ストリート, ピー.オー. ボックス 556</p> <p>(71) 出願人 512095325                  トーマス, アンドリュウ                  カナダ国, オンタリオ州 エル7ジー 5 エックス7, ジョージタウン, スイート 253, 162 グェルフ ストリート</p> <p>(74) 代理人 100114775                  弁理士 高岡 亮一</p>
---	--

最終頁に続く

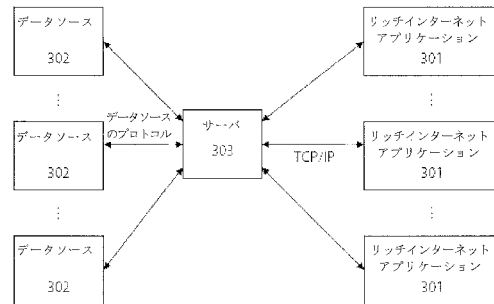
(54) 【発明の名称】 実時間データを提供するためのシステムおよび方法

(57) 【要約】 (修正有)

【課題】 リッチインターネットアプリケーション (RIA) に実時間データを提供するためのシステムおよび方法を開示する。

【解決手段】 少なくとも1つの実時間データサーバと協働するRIAフレームワークのグラフィックおよびネットワークの特徴は、低減された待ち時間とウェブブラウザにおける実時間データアプリケーションを提供することであって、データはデータソースで生成され、サーバに伝搬され、かつ、サーバで収集される。持続的な接続がRIAからサーバに対して作り出され、RIAはデータに登録される。登録されたデータは、サーバで収集された少なくとも幾つかのデータを含む。サーバは、データがサーバで収集されるように、持続的な接続を介して登録されたデータをRIAに伝搬する。

【選択図】 図3



**【特許請求の範囲】****【請求項 1】**

リッチインターネットアプリケーション（RIA）に実時間データを提供するコンピュータ実装方法であって、前記方法は、コンピュータに、データの発信元であるデータソースからデータを提供することと、前記データソースからサーバにデータを伝搬することと、前記サーバでデータを収集することであって、前記サーバで収集されたデータは、少なくとも前記データソースから伝搬されたデータを含むことと、前記サーバから前記RIAへの第1持続接続を作り出すことと、登録されたデータに前記RIAを登録することであって、前記登録されたデータは、前記サーバで収集されたデータの少なくとも一部を含み、前記サーバは、前記データソースで前記データに変更が生じる際に、前記第1持続接続を介して前記登録されたデータを前記RIAに伝搬することと、前記RIAでデータを生成することと、前記RIAから前記サーバへの第2持続接続を作り出すことと、前記第2持続接続を介して前記RIAから前記サーバに、前記RIAで生成されたデータを伝搬することと、前記サーバに命令して、前記RIAでのデータ項目の変化に対応するアクションを遂行させることと、  
を実行させることを含む、方法。

10

20

**【請求項 2】**

前記第1持続接続および前記第2持続接続は1つの接続から成る、請求項1に記載の方法。

**【請求項 3】**

前記データソースおよび前記サーバは、少なくとも1つのコンポーネントから成る、請求項1または2に記載の方法。

**【請求項 4】**

前記データは、少なくとも1つの媒介物を介して伝搬され、前記媒介物は、ソフトウェアコンポーネント、ハードウェアコンポーネント、およびネットワークを含む群から選択される、請求項1乃至3のいずれか一項に記載の方法。

30

**【請求項 5】**

前記データ項目が登録に基づいて前記RIAと前記サーバとの間で伝搬され、前記データ項目は前記データ項目の変化に応じて直ちに伝搬される、請求項1乃至4のいずれか一項に記載の方法。

**【請求項 6】**

伝搬されるデータは、数値データ、非数値データ、構成設定および実行可能コマンドを含む群から選択される、請求項1乃至5のいずれか一項に記載の方法。

**【請求項 7】**

前記RIAは、前記データに基づいてアクションを行い、前記アクションは、ユーザディスプレイの視覚表示の変更、計算、新規データの生成、既存データの変更、データの格納、可聴表示、スクリプトの実行、新規データの前記サーバへの伝搬、ユーザ可視なプログラミングの応答、およびユーザ不可視なプログラミングの応答を含む群から選択される、請求項1乃至6のいずれか一項に記載の方法。

40

**【請求項 8】**

前記RIAで生成されたデータは、前記サーバ内のデータの変更、前記サーバに接続されたデータシンクへの前記データの伝搬、スクリプトの実行、ファイルシステムへの前記データの格納、新規データの作成、前記サーバに接続されたデータシンクへの新規データの伝搬、サーバ構成の変更、サーバ動作の変更、ユーザ可視なプログラミングの応答、およびユーザ不可視なプログラミングの応答を含む群から選択されるアクションを行うよう前記サーバに指示する、請求項1乃至7のいずれか一項に記載の方法。

50

**【請求項 9】**

前記第 1 持続接続は、TCP/IP 接続および前記サーバによりサポートされた少なくとも 1 つのデータフォーマットを実装するアプリケーション・プログラミング・インターフェース、直接の TCP/IP 接続、HTTP および HTTPS を含む群から選択されるプロトコルを含む、請求項 1 乃至 8 のいずれか一項に記載の方法。

**【請求項 10】**

前記 RIA で生成されたデータは、ユーザの相互作用、タイマーイベント、前記サーバに由来するデータ変更への応答、またはユーザにより生成されない別のイベントの結果として生成される、請求項 1 乃至 9 のいずれか一項に記載の方法。

**【請求項 11】**

前記方法は、前記コンピュータに、前記データソースのデータプロトコルを、前記 RIA により処理することができるプロトコルに変換することを実行させることをさらに含む、請求項 1 乃至 10 のいずれか一項に記載の方法。

**【請求項 12】**

リッチインターネットアプリケーション (RIA) に実時間データを提供するためのコンピュータシステムであって、前記システムは、

プロセッサと、

入力手段と、

ディスプレイと、

データの発信元である少なくとも 1 つのデータソースであって、前記データソースは、前記データソースからサーバへ前記データを伝搬するように配置される、データソースと

データを収集するためのデータ収集コンポーネント、および少なくとも 1 つのデータクライアントにデータを放出するためのデータ放出コンポーネントを備える少なくとも 1 つのサーバであって、前記サーバで収集されたデータは、少なくとも前記少なくとも 1 つのデータソースから伝搬されたデータを含む、サーバと、

少なくとも 1 つの RIA であって、前記 RIA は、

前記サーバから前記 RIA への第 1 持続接続を作り出すことと、

前記サーバで収集されたデータの少なくとも一部を含む登録されたデータに前記 RIA を登録することと、

前記データソースで前記データに変更が生じる際に、前記第 1 持続接続を介して前記サーバから伝搬される登録されたデータを受信することと、

データを生成することと、

前記 RIA から前記サーバへの第 2 持続接続を作り出すことと、

前記第 2 持続接続を介して前記サーバに、前記 RIA で生成されたデータを伝搬することと、

前記サーバに命令して、前記 RIA でのデータ項目の変化に対応するアクションを遂行させることと、

を行うように構成され、

前記少なくとも 1 つの RIA は、前記少なくとも 1 つのデータクライアントのうち 1 つである、RIA と、

を備える、システム。

**【請求項 13】**

前記少なくとも 1 つのサーバが、前記データ放出コンポーネントによる放出のために前記データ収集コンポーネントにより収集されたデータのフォームを変更するためのデータ変更コンポーネントをさらに備える、請求項 12 に記載のシステム。

**【請求項 14】**

前記少なくとも 1 つのサーバが、前記データ放出コンポーネントによる放出のために前記データ収集コンポーネントにより収集されたデータのフォームを変更するためのデータ変更コンポーネントと、ユーザインターフェースと、プログラム可能なアクションを行う

10

20

30

40

50

ためのスクリプト言語コンポーネントと、HTTPリクエストを受入れるためのHTTPコンポーネントと、任意のローカルまたはネットワークでアクセス可能なコンピュータ上で前記サーバの別の実行中のインスタンスとデータを交換および同期させるための同期コンポーネントと、データが収集される速度を限定するための第1調整コンポーネントと、データが放出される速度を限定するための第2調整コンポーネントと、他のサーバへの接続性が失われたことを検出し、接続性が再び取り戻される場合に前記他のサーバに再接続するための接続性コンポーネントと、同一または類似する情報の複数の他のサーバのうち1つ以上がアクセス不能である場合に前記他のサーバのうちいずれかからのデータを収集するために、前記他のサーバと重複して接続する重複コンポーネントと、複数のデータソースにあるデータを移動させるための橋渡し用コンポーネントとから成る群から選択される少なくとも1つのコンポーネントをさらに備える、請求項12または13に記載のシステム。

10

**【請求項15】**

前記第1持続接続は、TCP/IP接続および前記サーバによりサポートされた少なくとも1つのデータフォーマットを実装するアプリケーション・プログラミング・インターフェース、直接のTCP/IP接続、HTTPおよびHTTPSを含む群から選択されるプロトコルを含む、請求項12乃至14のいずれか一項に記載のシステム。

**【請求項16】**

前記第1持続接続および前記第2持続接続は1つの接続から成る、請求項12乃至15のいずれか一項に記載のシステム。

20

**【請求項17】**

前記RIAは、前記データを表現するグラフィック描写を提示するようにさらに構成され、前記グラフィック描写は少なくとも1つのグラフィック要素を含み、前記少なくとも1つのグラフィック要素はカスタム化されたグラフィック要素であり、前記カスタム化されたグラフィック要素はプログラマによりカスタマイズ可能であり、前記カスタム化は前記RIAに実装されたアプリケーションに対する変更を伴わない、請求項12乃至16のいずれか一項に記載のシステム。

**【請求項18】**

前記RIAは、ユーザの相互作用、タイマーイベント、前記サーバに由来するデータ変更への応答、またはユーザにより生成されない別のイベントの結果としてデータを生成するように構成される、請求項12乃至17のいずれか一項に記載のシステム。

30

**【請求項19】**

前記RIAが、前記データソースのデータプロトコルを、前記RIAにより処理することができるプロトコルに変換するようにさらに構成される、請求項12乃至18のいずれか一項に記載のシステム。

**【請求項20】**

1つ以上のコンピュータで実行される場合に、リッチインターネットアプリケーション(RIA)に実時間データを提供する方法を前記コンピュータに行わせる指示を記憶するコンピュータ読み取り可能な記憶媒体であって、前記方法は、

40

データの発信元であるデータソースからデータを提供することと、

前記データソースからサーバにデータを伝搬することと、

前記サーバでデータを収集することであって、前記サーバで収集されたデータは、少なくとも前記データソースから伝搬されたデータを含むことと、

前記サーバから前記RIAへの第1持続接続を作り出すことと、

登録されたデータに前記RIAを登録することであって、前記登録されたデータは、前記サーバで収集されたデータの少なくとも一部を含み、前記サーバは、前記データソースで前記データに変更が生じる際に、前記第1持続接続を介して前記登録されたデータを前記RIAに伝搬することと、

前記RIAでデータを生成することと、

前記RIAから前記サーバへの第2持続接続を作り出すことと、

50

前記第2持続接続を介して前記R I Aから前記サーバに、前記R I Aで生成されたデータを伝搬することと、

前記サーバに命令して、前記R I Aでのデータ項目の変化に対応するアクションを遂行させることと、  
を含む、記憶媒体。

【請求項21】

前記方法が、前記データソースのデータプロトコルを、前記R I Aにより処理することができるプロトコルに変換することをさらに含む、請求項20に記載の記憶媒体。

【発明の詳細な説明】

【技術分野】

10

【0001】

著作権の告知

本特許文献の開示の一部分は、著作権保護を対象とする資料を含む。著作権者は、特許商標庁の特許出願または記録に見られる特許文献、または特許開示物による複製物に異存はないが、その他の点では、何であれ全ての著作権を保有する。

【背景技術】

【0002】

実時間データとは、データが最初に作成された後、一定時間内に処理および/または送信されるべき任意のデジタル情報またはアナログ情報を指す。データが作成された時から処理および/または送信されるまでに経過した時間は、待ち時間として知られている。任意の特定の实時間アプリケーションにとって可能な最大の待ち時間は、アプリケーションに依存する。最大待ち時間が厳しい要件であるアプリケーションは、「ハードな」実時間アプリケーションと呼ぶことができ、最大待ち時間が厳しい要件でないアプリケーションは、「ソフトな」実時間アプリケーションと呼ぶことができる。ソフトな実時間アプリケーションは、時に主観的なアプリケーション依存を満たし、一定の「十分な速さ」を満たすことだけが必要である。非実時間データは、特定の待ち時間要求に何ら応じる必要がないデータである。

20

【0003】

「データ」という用語は、ハードな実時間、ソフトな実時間、または非実時間データを指し得る。「実時間データ」は、ハードな実時間データまたはソフトな実時間データを指し得る。

30

【0004】

実時間データは、典型的に、物理的処理またはデータを処理するコンピュータシステムの外部のコンピュータプログラムに起因して作成される。例えば、実時間データは、モータの状態、流体タンクのレベル、パルプの位置、コンベヤの速度などのような産業プロセスの制御システムに由来する情報と、例えば株のような金融商品用の価格、容量と、ユーザがコンピュータディスプレイでボタンをクリックしたことを示すようなユーザインターフェースにおけるイベントと、ヒトであるオペレータによるデータ入力と、コンピュータ・オペレーティングシステムの状態変化とを含み得る。徐々に変化する視覚的な何らかの情報、実時間データとして扱うことができる。

40

【0005】

データの発信元は、「データソース」と記載され得る。例えば、データは、物理的処理として発信元となり、電氣的に計算され、デジタル表現に変換され得るか、またはデータは、デジタル表現の発信元となり得る。一般的に、データはデジタル表現としてデジタルコンピュータで利用可能に作成され、続く零以上のステップがデータをデジタル表現に変換する。データソースは、コンピュータプログラムによりアクセス可能なデジタルの状態にデータを変換するために必要な全てのコンポーネントおよび工程を含み得る。

【0006】

データソースの類似物は「データシンク」である。データシンクは、データを消費または使用する。幾つかのデータシンクの例は、処理制御システムにおけるアクチュエータ、

50

株式取引システムにおける株式取引ソフトウェア、ユーザインターフェース用アプリケーション、データベースまたは他のデータ・ストレージシステムが挙げられる。

【0007】

多くのデータソースは、データシンクでもある。従って、データソースは、データソース、データシンク、または同時に両方を含み得る。例えば、データがデータソースに送信される場合、データソースもデータシンクとして働き得る。

【0008】

コンピュータ・アプリケーションでは、データは通常「サーバ」に管理される。サーバは、データソース、データシンクのいずれか、または両方として働くことができ、「クライアント」アプリケーションは、サーバが管理するデータと相互に作用することが可能となる。

10

【0009】

一般的に、クライアント・アプリケーションは、データと相互に作用するためにサーバとの接続を開始しなければならない。この接続は、データとの一または僅かな相互作用の間のみ接続する「一時的」、またはデータと多く相互作用するために接続を持続させ、クライアント・アプリケーションの存続期間のために接続可能な「永続的」が存在し得る。永続的な接続は、「持続」接続とも呼ばれる。

【0010】

データソースは、データのデジタル表現を定義する一または複数の「データフォーマット」でデータを提供する。データフォーマットは、発行された基本的なデータソースに一致し得るか、またはデータソースに特定のであり得る。同様に、データシンクは、発行された基本フォーマットのデータまたはデータシンクに特定のフォーマットのデータを必要とし得る。

20

【0011】

データソースは、一または複数の「通信プロトコル」を介してデータへのアクセスを提供する。通信プロトコルは、データがデータソースからデータシンクに転送されるメカニズムを特定する。通信プロトコルは、基本的なデータソースに一致し得るか、またはデータソースに特異的であり得る。データソースは、サポートされていないデータフォーマットが、全てサポートされた通信プロトコルを介して送信されるようにデータフォーマットと通信プロトコルとを組み合わせ得る。一般的に「プロトコル」または「データプロトコル」とは、特定の通信プロトコルを介して送信された特定のデータフォーマットの組み合わせを指す。

30

【0012】

データシンクは、データソースにより生成されたデータを使用するために、データソースにより提供された少なくとも1つのデータプロトコルをサポートしなければならない。データプロトコルは多大に存在するため、全てのデータソースおよびデータシンクについて全てのデータプロトコルをサポートすることは実用的ではない。結果、データを使用するクライアント・アプリケーションは、通常、主要な目的のために一番必要なプロトコルだけをサポートするように作成される。同様に、データソースも一般的に主要な目的に必要なプロトコルのみをサポートする。従って、例えば、HTTPプロトコルをサポートするウェブブラウザをDDEプロトコルをサポートする表計算アプリケーションに直接接続させる方法はない。

40

【0013】

データシンクにデータソースにより提供されるデータを使用させるために、プロトコルの変換工程が行われて、データソースによってサポートされているプロトコルからデータシンクにサポートされているプロトコルにデータを変換しなければならない。この変換工程は、「ミドルウェア」アプリケーションにより行うことができる。ミドルウェア・アプリケーションの主な目的は、通常はデータソースとデータシンクとが共通してプロトコルを共有する場合は間接的に相互作用できるように、1つのプロトコルから別のプロトコルにデータを変換することによってデータソースとデータシンクとの通信を促すことであり

50

得る。

【0014】

データソースは、少なくとも2つの方法を使用してデータをデータシンクに転送し得る。

1. オンデマンド。データソースは、データソースで利用可能なデータの一部または全てを要求するデータシンクを受動的に待つ。データシンクがデータを要求した場合は、ソースが要求されたデータの現在の状態を示す結果を用いて応答する。データシンクがデータ変更を知らされる必要があるならば、データシンクはデータソースが更新されたデータを用いて応答するために、データシンクは要求を繰り返さなければならない。データシンクによる同一データに対する繰り返し要求は「ポーリング」として知られている。データシンクは、新しい要求毎にデータソースへの一時的な接続を作り出し得るか、または繰り返し要求が送信される間にわたり持続的な接続を作り出し得る。

10

2. 登録による方法。データシンクはデータソースへの持続的な接続を作り出し、データソースから利用可能な一部または全てのデータを登録する。データソースは、持続的な接続を介して変更が生じるとデータに対する何らかの変化を送信する。データソースは、データシンクが他に特定するかまたは接続が終わるまでデータに対する変化を送信し続ける。

【0015】

共有メモリ、メッセージ・キュー、およびメールボックスのようなデータ転送方法は、要求または同意方法のいずれかのバリエーションであることが理解される。データ転送、データ伝搬、またはデータ通信という用語は全て、システム内のデータの移動を指し、これらの用語は、特定のデータ通信方法に関連するように、互換可能に使用され得ることもさらに理解される。これらの方法は、通信プロトコルに基づいて独立している。

20

【0016】

実時間データを扱うコンピュータ・アプリケーションは、信頼があり、反応がよく、データソースに簡単に接続されなければならない。これは、実時間データ処理アプリケーションが、歴史的に、直接的または間接的にデータソースに接続される単独型のアプリケーションとして作成されてきたことを意味する。この単独型のアーキテクチャは、アプリケーションにコンピュータのグラフィック能力を最大限に利用することも可能とし、実時間データの高等で動的な視覚化を提供する。反対に、ウェブブラウザ技術に基づいたアプリケーションは、データ接続およびグラフィック速度の観点から不適当なことが証明されている。従って、ウェブブラウザにおける広範な実時間データ処理および表示は、利用できなかった。ウェブブラウザでActive Xコンポーネントを使用してデータ駆動型表示にアクセスする試みを行う開発者もいたが、しかしこれらのコンポーネントは、一般的に現代のブラウザではサポート性が乏しく、これらのコンポーネントが示すセキュリティーの危険性に起因して限定が課される。

30

【0017】

内蔵されたブラウザのJavaScriptエンジンを使用して、ウェブブラウザにおいて変化するデータを表示する試みが成されてきた。これは、一般的に、AJAX (Asynchronous JavaScript and XML) と呼ばれる手法を用いて実現され、ウェブブラウザは新しいデータを定期的にポーリングし、次いでこれに応じてディスプレイを更新する。このポーリングメカニズムは非常に非効率的であり、比較的少量のデータセットまたは比較的移動が遅いデータにのみ好適である。CPUまたはネットワークのバンド幅を浪費させないためにポーリングの割合を低減させることは、実時間アプリケーションを受けられないデータの待ち時間が増えることに対し効果がある。

40

【0018】

ストリーミングAJAXと呼ばれるメカニズムを介してAJAXを改善する試みは、ブラウザのページをロードするメカニズムの副作用を利用してJavaScriptコマンドを徐々にページに対して追加することによりブラウザのページを増加的に拡大させる。JavaScriptコマンドの各々は、連続的なデータの流れといった印象を与えなが

50

らコマンドが立ち上がると実行する。ウェブブラウザは、事実上、ネットワーク接続が遅い間に極めて大量のウェブページを負担しているものと思込む。この方法は、ウェブブラウザのメモリやCPUの使用が、送信されるこれまでにない大量のページに起因して続けて徐々に増えていく事実を含め、幾つかの障害を持つ。

【0019】

A J A XもストリーミングA J A Xの方法もウェブブラウザ内のディスプレイ選択肢という品質の欠損を被る。ウェブブラウザは、一般的に静止ページおよびウェブ「フォーム」のディスプレイ用に設計されており、高速または高品質なグラフィックの提示という選択肢を提供しない。グラフィカルなディスプレイという選択肢に対する改善のための試みは、ブラウザ間では両立しない虞があり、一般的に非常に実行が遅い。

10

【0020】

内蔵されたウェブブラウザの能力に基づいて全てのデータを送信する解決法は、ウェブブラウザでデータを受信することを最初に標的とする。データ通信は、一方向であり、また、サーバからデータを受信する接続は、サーバにデータを送信するために使用することもできない。サーバにデータを再送信する必要がある場合は、新たに接続を開き、HTTPリクエストを送信し、次いで再び接続を閉じなければならない。従って、ストリーミングA J A Xのような解決法は、膨大なオーバーヘッドとデータ送信の毎に新しいHTTPリクエストを発行しなければならないことにより被る待ち時間とにより、データサーバにデータを送信するのが非常に遅い。

20

【0021】

速いかのように見えるが遅く移動する（待ち時間が長い）データを提示することによりユーザ体験を改善しようというウェブベースのデータ視覚化に対する試みが成されている。これは、実際に到着するデータよりも高い頻度でウェブブラウザに補完データを表示することにより実現される。速度計を意味する環状ゲージは、5秒毎に分離された1~100の値を受信する。次に、ウェブページは、各回に4回値を変えながら1秒間にゲージダイヤルを5回取り出す。これにより、根本的なデータ伝達はそのような情報を含まないが、視聴者に滑らかに速度が変化する印象を与える。すなわち、そのような補間データの表示は、もっぱら視聴者を惑わせる場合がある。こうした種類の補間は根本的なデータの実際の動作を曖昧にし、通常、プロセス制御および株式市場の取引のような実時間アプリケーションでは許容できない。

30

【0022】

Adobe Flash（商標）およびMicrosoft Silverlight（商標）のようなリッチインターネットアプリケーション（「RIA」）フレームワークは、ウェブブラウザにおけるデータ処理およびグラフィック描写の両方について改善されたプラットフォームを提供する。これらのRIAフレームワークはまた、RIA内での直接的なTCP/IP通信をサポートする。驚くべきことに、これらの特徴を組み合わせることにより、ウェブブラウザでの実時間情報を処理および表示することが可能となる。この処理・表示能力は、RIAが主にビデオ、宣伝、およびゲームに好適であるというソフトウェア産業における認識に起因して、実時間データシステムに移転されてこなかった。

40

【発明の概要】

【0023】

本発明は、ウェブブラウザで少ない待ち時間の実時間データアプリケーションを提供するために、少なくとも1つの実時間データサーバと協働する、RIAフレームワークのグラフィック特性およびネットワーク特性を使用するためのシステムおよび方法を提供する。本発明は、データソースのデータがウェブブラウザ内で使用できないデータソースを同時に扱いながら、現在のA J A XおよびストリーミングA J A Xの限定を克服する。

【0024】

本発明は、最先端の技術により、あるポイントまでのデータ待ち時間を縮小することでウェブブラウザへの実時間データの送達を改善し、視覚化コンポーネントは、補間値ではなく実際のデータ値を使用して活発化できる。これにより、データの一時的な動作がユー

50

ザに対して一層正確に提示される。一時的なデータの動作は、通常、そのデータにより提示される実際の動的な実時間システムの理解に重要である。例えば、物理的ゲージを観察する者は、ゲージ針の動作に振動またはオーバーシュートを観察することにより重要なシステムの特性を判別することができる。本発明の一実施形態では、物理的ゲージのデジタル表現は針の原動力を捕獲し、物理的ゲージとして同じように高品質の情報を提供することができる。

**【 0 0 2 5 】**

本発明は、CPUおよびネットワークのコストを抑え、待ち時間を低減させながら、ユーザからデータサーバへのデータ送信の速度を大幅に改善する。これにより、ユーザは、システムの反応が現在の動作に重要な、一層高機能な制御シナリオに参加することが可能となる。例えば、システムは、容器に水を注ぐ間に、相互作用を維持および解除するための相互作用を必要とし得る。ユーザはボタンを押し、容器が満タンになるまで維持し、次いでボタンを離す。システムが、容器から溢れないように迅速に応答しなければならないことは明白である。この種の制御は、データ伝達の待ち時間が予測できないことに起因して、典型的なウェブベース・アプリケーションでは実行できない。驚くべきことに、本発明は、以前は遅すぎ、信頼性がなく、またはウェブブラウザを通して検討されるには原始的であったこの種の制御および実時間データ・アプリケーションを可能にする。

10

**【 0 0 2 6 】**

典型的なウェブアプリケーションは、アプリケーション設計者による特異的なフォーマットで提供されるデータを扱う。これは、エンドユーザの選択を限定する意図的な方法であり得、または単に設計に対する限定であり得る。データフォーマットが(XMLまたはJSONなどの)業界基準に従っている場合でも、データソースはアプリケーションに特異的である。本発明はまた、業界基準および所有者のソースの両方を発信元とする多種多様な実時間データを送達するための汎用的なメカニズムも提供する。有利に、本発明はさらに、種々のデータフォーマットでそのデータを提供することができる。

20

**【 0 0 2 7 】**

多くのデータソース、実時間および非実時間は、ネットワークの使用(すなわち、ネットワークを通じた送信)を意図しない。本発明によれば、Microsoft Excel(商標)(マイクロソフト株式会社)のようなデータソースからのデータをネットワークを通して任意のRIAまたはウェブベース・アプリケーションに確実にかつ迅速に送達することが可能となる。OPCに基づいたデータソースのようなデータソースは、ネットワークの使用を意図するものであったが、ウェブブラウザと通信するようには設計されていない。本発明によれば、ウェブベース・アプリケーションに確実にかつ迅速にこれらのソースを送達することが可能となる。データベース・システムのような他のデータソースは、実時間情報について全くインターフェースを提供していない。本発明によれば、データベース・アプリケーションのようなソースからの非実時間データを実時間データのように送達することが可能となり、それにより、RIAまたはウェブベース・アプリケーションのために非効率的なデータベースのポーリングの必要性を排除する。

30

**【 0 0 2 8 】**

データソースおよびデータシンクは、持続的な接続または一時的な接続によりサーバと接続し得る。サーバに接続する方法は、特定のデータソースまたはデータシンクの要求を反映させることが理解される。

40

**【 0 0 2 9 】**

本発明は、一または複数のデータソースからの実時間データがリッチインターネットアプリケーションで効率的に利用できる方法を提供する。本発明はさらに、RIAが、効率的に実時間データに変更するか、またはソースデータ(単数または複数)に再び送信することができる新規の実時間データを生成するための方法を提供する。ソースデータ(単数または複数)は、ネットワーク上の他のRIAにそのデータを再送信することができる。従って、本発明によれば、効率的に任意の数のRIAアプリケーションが実時間で別のRIAアプリケーションと通信し、一つまたは多くの実時間データソースと協働して通信す

50

ることが可能となる。抽出することにより提示可能な任意のデータが発信元のソース、提示、または転送プロトコルの有無に関わらず R I A で利用できるように、本発明は実時間データの抽出が可能である。

【図面の簡単な説明】

【 0 0 3 0 】

【図 1】本発明の一実施形態に係る、R I A とデータサーバとの間の直接接続を示す例示的なブロック図である。

【図 2】本発明の一実施形態に係る、R I A と、サーバと、別離したデータソースとの間の接続を示す例示的なブロック図である。

【図 3】本発明の一実施形態に係る、複数の R I A と、サーバと、複数の別離したデータソースとの間の接続を示す例示的なブロック図である。

【図 4】本発明の一実施形態に係る、R I A の制御フローの一方法を示すフローチャートである。

【図 5 a】本発明の一実施形態に係る、サーバ操作の一方法を示すフローチャートである。

【図 5 b】本発明の一実施形態に係る、サーバ操作の一方法を示すフローチャートである。

【図 6】本発明の一実施形態に係る、複数の R I A への同時接続を管理するデータサーバを示す例示的なブロック図である。

【図 7】本発明の一実施形態に係る、表計算アプリケーションと R I A との間のローカルネットワークまたは広域ネットワークを介したデータの実時間通信を示す例示的なブロック図である。

【図 8】本発明の一実施形態に係る、システムの実施を示す例示的なブロック図である。

【発明を実施するための形態】

【 0 0 3 1 】

以下の記載は、どの当業者も本発明を成すことができ、使用できるように提示され、本発明の特定のアプリケーションを背景として提供される。開示される実施形態に対する種々の変更が当業者には容易に明らかとなり、本明細書で定義される一般原則は、本発明の範囲から逸脱することなく他の実施形態およびアプリケーションに適用され得る。種々の実施形態および実施例に対する言及は、本発明の範囲を限定せず、本明細書に添付した請求の範囲によってのみ限定される。従って、本明細書で説明する例のどれもが限定されることを意図しておらず、主張する本発明について、可能性のある多くの実施形態のうちの幾つかを説明するにすぎない。

【 0 0 3 2 】

本発明の実施形態が実行されるプログラム環境には、例示的に、汎用コンピュータ、または携帯用コンピュータ若しくは携帯電話のような特殊目的の装置が組み込まれる。そのような装置（例えば、プロセッサ、メモリ、データストレージ、ディスプレイ、ユーザの入力手段）の詳細は、明瞭さのために省略され得る。

【 0 0 3 3 】

本発明の技術は、種々の技術を使用することにより実装され得ることも理解される。例えば、本明細書に記載される方法は、コンピュータシステムで実行されるソフトウェアに実装されるか、またはマイクロプロセッサ、若しくは他の特別に設計されたアプリケーションに特異的な集積回路、プログラム可能な論理装置のいずれかの組み合わせ若しくは種々の組み合わせを用いてハードウェアに実装され得る。詳細には、本明細書に記載される方法は、コンピュータで読み取り可能な好適な媒体に存在する、コンピュータが実行可能な一連の命令により実施され得る。コンピュータで読み取り可能な好適な媒体は、揮発性（例えば R A M）および/または不揮発性（例えば R O M、ディスク）メモリ、搬送波、並びに通信媒体（例えば、銅線、同軸ケーブル、光ファイバ媒体）を含み得る。例示的な搬送波は、ローカルネットワーク、インターネットまたは他の幾つかの通信リンクのような公的にアクセス可能なネットワークに従ってデジタルデータ・ストリームを搬送する電

10

20

30

40

50

気信号、電磁信号、または光信号の形態を取り得る。

【0034】

図面に示される例示の実施形態を参照すると、簡易化された例は明瞭さのために選択されたことが理解される。図面に見受けられる一つの構成要素における単一の例（例えば、R I A、サーバ、データソース、データシンクなど）は、複数の同じ構成要素と置換され得、それも本発明の範囲内に含まれる。

【0035】

従って、一態様において、本発明は、R I Aに実時間データを提供する方法を提供する。その方法は、データソースでデータを生成することと、サーバにデータを伝搬することと、サーバでデータを収集することと、R I Aからサーバへの持続的な接続を作り出すことと、登録されたデータに前記R I Aを登録させることとを含み、登録されたデータは、サーバで収集された少なくとも幾つかのデータを含み、サーバは、データがサーバで収集されるように持続的な接続を介して登録されたデータをR I Aに伝搬する。方法はさらに、R I Aが発信元のデータをサーバに送信することを含み得る。R I Aが発信元のデータは、持続的な接続を通じたデータに対する少なくとも1つの変化リクエストを含み得る。さらにデータは、少なくとも1つの中間コンポーネントを介して伝搬され得る。サーバは、少なくとも1つの変化リクエストを受信し得、少なくとも1つの変化リクエストをデータソースに送信し得る。少なくとも1つの変化リクエストは、中間コンポーネントを介して伝達され得る。中間コンポーネントは、中間ハードウェアコンポーネントまたは中間ソフトウェアコンポーネントであり得る。任意に、R I Aが登録されたデータに登録し得る。データソースでデータを生成し、そのデータをサーバに伝搬することは、サーバでデータを収集することともに行われ得る。R I Aは、計算またはグラフィック描写の変更など、データに基づいてアクションを遂行し得る。R I Aは、ユーザディスプレイ上にデータの視覚表示を提供し、ユーザはR I Aが発信元のデータを生成するために視覚表示とやり取りし得る。視覚表示は、R I Aフレームワーク内で実行するプログラムであり得る。R I Aが発信元のデータは、サーバを停止させるか、またはどのデータがサーバから届いたのかを変えるなど、その動作を変えるために、サーバに命令してアクションを遂行させ得る。

【0036】

例えば、R I Aが発信元のデータは、ユーザとの処理の結果、タイマーイベント、サーバに由来するデータへの応答、スクリプト、またはユーザではなく生成された別のイベントであり得る。

【0037】

別の態様では、本発明は、実時間データをR I Aに提供するコンピュータに実装された方法を提供する。前記方法は、データソースからデータを提供することと、データソースからサーバにデータソースを伝搬することと、サーバでデータを収集することと、R I Aでデータを生成することと、サーバからR I Aへの第1持続接続を作り出すことと、R I Aからサーバへの第2持続接続を作り出すことと、第2持続接続を介してR I Aからサーバにデータを伝搬することと、登録されたデータにR I Aを登録することとを含み、登録されたデータは、サーバで収集された少なくとも幾つかのデータを含み、サーバは、第1持続接続を介してR I Aに登録されたデータを伝搬する。方法はさらに、サーバからデータシンクにデータを伝搬することを含む。第1持続接続および第2持続的な接続は、単一接続から成り得る。データソース、データシンク、およびサーバは、単一のコンポーネントまたは2つのコンポーネントの任意の組み合わせから成り得る。データは、ソフトウェアコンポーネント、ハードウェア、コンポーネント、およびネットワークを含む群から選択される少なくとも1つの中間物を介して伝搬され得る。

【0038】

データ項目は、登録に基づいてR I Aとサーバとの間で伝搬され得、データ項目はデータ項目の変化に応じて直ぐに伝搬される。伝搬されたデータは、数値データ、非数値データ、構成設定、および実行コマンドを含む群から選択され得る。R I Aは、データに基づ

10

20

30

40

50

いてアクションを遂行し得、該アクションは、ユーザディスプレイの視覚表示の変更、計算、新規データの生成、既存データの変更、データの格納、可聴表示、スクリプトの実行、サーバへのデータの伝搬、ユーザ可視なプログラミングの応答、ユーザ不可視なプログラミングの応答を含む群から選択される。R I Aで生成されたデータは、サーバに命令して、サーバ内でのデータの変更、サーバに接続されたデータシンクへのデータの伝搬、スクリプトの実行、ファイルシステムへのデータの格納、新規データの作成、サーバに接続されたデータシンクへの新規データの伝搬、サーバ構成の変更、サーバ動作の変更、ユーザ可視なプログラミングの応答、およびユーザ不可視なプログラミングの応答を含む群から選択されるアクションを遂行させ得る。

【0039】

さらに別の態様では、本発明は、上述したように、一または複数のコンピュータで実行される場合に、実時間データをR I Aに提供する方法をコンピュータに行わせる命令を記憶したコンピュータ読み取り可能な記憶媒体を提供する。

【0040】

別の態様では、本発明は、R I Aに実時間を提供するためのコンピュータシステムを提供する。前記システムは、プロセッサと、入力手段と、ディスプレイと、少なくとも1つのデータソースと、少なくとも1つのデータソースからデータを収集するためのデータ収集コンポーネント、および少なくとも1つのデータクライアントにデータを放出するためのデータ放出コンポーネントを備える少なくとも1つのサーバと、少なくとも1つのR I Aと、任意に少なくとも1つのデータシンクとを備える。サーバはさらに、データ放出コンポーネントにより放出するためにデータ収集コンポーネントにより収集されたデータの形態を変更するためのデータ変更コンポーネントを備え得る。少なくとも1つのデータソースと少なくとも1つのサーバは、少なくとも1つのコンピュータプログラム（すなわち、単一のコンピュータプログラム、または2つ以上の別箇のコンピュータプログラム）に実装され得ることが理解される。

【0041】

サーバはさらに、データ変更コンポーネントと、データ生成コンポーネントと、ユーザインターフェース用コンポーネントと、コンピュータファイルシステム相互作用コンポーネントと、サーバで実行するコンピュータ上で実行する他のプログラムと相互作用するためのプログラム相互作用コンポーネントと、プログラム可能なアクションを行うためのスクリプト言語コンポーネントと、クライアントプログラムからH T T Pリクエストを受け、それらリクエストに特定される文書を用いて、「ウェブサーバ」に類似する方法で応答するためのH T T Pコンポーネントであって、前記リクエストに応じる文書を動的に構築し、サーバに在るデータの現在の値とサーバが内蔵するスクリプト言語で実行制御文の結果を文書内に含める能力を備えるH T T Pコンポーネントと、任意のローカルまたはネットワークでアクセス可能なコンピュータで実行する別の一例のサーバとデータを交換または同期し、両方のサーバがそのデータの基本的に同一の複製を維持することができ、それにより、いずれかの一例のサーバに接続されたクライアント・アプリケーションが相互に作用する 同一データセットと相互作用することができる、同期コンポーネントと、データが収集される割合を限定するための第1調整コンポーネントと、データが放出される割合を限定するための第2調整コンポーネントと、他のサーバへの接続性の損失を検出し、接続性が取り戻せる場合は他のサーバに再接続する接続コンポーネントと、他のサーバからのデータが、一または複数の他のサーバがアクセスできないように同一または類似する情報を持つ複数の他のサーバと重複して接続する重複コンポーネントと、ソース内の幾つかまたは全てのデータが別のデータと類似する値を維持するようにデータソースのデータを橋絡するか、または1つのソースにあるデータが他のソースにあるデータの数学的変換として維持されるように橋絡するブリッジングコンポーネントであって、双方向のブリッジング操作を介して順方向にも逆方向にも数学的変換を適用する能力を備えるブリッジングコンポーネントと、から選択される一または複数のコンポーネントを備え得る。この一式のサーバコンポーネントは、他のデータの収集、並びに通信メカニズム、他の処理メ

10

20

30

40

50

カニズム、および他の記憶メカニズムをサポートするために、サーバに追加機能を追加することにより拡張することができることが理解される。

【0042】

データ収集コンポーネントは、一または複数の以下の方法でデータを収集し得る：

オンデマンドであって、サーバは、別のサーバに在る幾つかまたは全てのデータのためにリクエストを送信し、他のサーバはそのリクエストに応じて、リクエストされたデータの現在の値（単数または複数）を用いて一度だけ応答する、方法と、登録による方法であって、サーバは、別のサーバにある幾つかまたは全てのデータに対し登録のためのリクエストを送信し、他のサーバは、そのデータの現在の値（単数または複数）を送信することにより応答し、次いでサーバが他のサーバとの接続を終了するか、または他のサーバが更新の送信を中止することをリクエストするまで、データの値（単数または複数）に対する逐次的変化を送信し続ける方法と、

トリガによる方法であって、クライアント、スクリプトまたはヒト（ユーザ）が、あるトリガの状況が、タイマ、時刻、データ変化、システム状態の変化、ユーザアクション、または他の幾つかの検出可能なイベントを満たす場合にのみデータを収集するようにサーバを構成する方法と、

サーバにデータを送信するために「クライアント」アプリケーションを受動的に待つ方法。

【0043】

データ放出コンポーネントは、一または複数の以下の方法でデータを放出し得る：

オンデマンドであって、「クライアント」アプリケーションは、幾つかまたは全てのデータに対してリクエストを送信し、サーバはリクエストに応じて、リクエストされたデータの現在の値（単数または複数）を用いて一度だけ応答する方法と、

登録による方法であって、クライアント・アプリケーションは、幾つかまたは全てのデータを登録するためのリクエストを送信し、サーバがデータの現在の値（単数または複数）を送信することにより応答し、次いでクライアントがサーバとの接続を終了するか、またはサーバが更新の送信を中止することをリクエストするまでデータの値（単数または複数）に対する逐次的変化を送信し続ける、方法と、

トリガによる方法であって、クライアント、スクリプト、またはヒト（ユーザ）が、あるトリガの条件がタイマ、時刻、データ変化、システム状態の変化、ユーザアクション、または他の幾つかの検出可能なイベントを満たす場合にのみデータを放出するようにサーバを構成する方法。

【0044】

データ収集コンポーネントで収集されるデータは、Dynamic Data Exchange (DDE)、OLE for Process Control (OPC)、OPCアラームおよびイベント仕様 (OPC A&E)、Unified Architecture (OPC-UA)、OPC Express インターフェース (OPC-Xi)、TCP/IP、カスタムインターフェースを介してTCP/IPを渡るSSL (セキュア・ソケット・レイヤー)、ハイパーテキスト・トランスファー・プロトコル (HTTP)、Secure HTTP (HTTPS)、オープン・データベース・接続性 (ODBC)、Microsoft Real-Time Data specification (RTD)、メッセージ・キュー、Windows Communication Foundation (WCF)、ProfibusおよびModbusなどの工業用バスのプロトコル、Windows Systemパフォーマンス・カウンタ、システムに組み込まれたTCP/IP通信、MS-Windows (登録商標) ではないシステムに由来するTCP/IP通信、Linux (登録商標) に由来するTCP/IP通信、QNXに由来するTCP/IP通信、TRONに由来するTCP/IP通信、CコンパイラおよびTCP実装を提供する任意のシステムに由来するTCP/IP通信、内蔵のスクリプト言語を使用して記載されたスクリプト、ユーザインターフェースを介してヒトが入力したデータ、ローカルディスクファイルから読み込まれたデータ、リモ

10

20

30

40

50

ートからアクセス可能なディスクファイルから読み込まれたデータ、所有者のフォーマット、ユーザが定義したフォーマット、並びにサーバへの拡張を通じて追加されたフォーマットから選択される一または複数の通信プロトコルを使用して受信され得る。所有者のフォーマットの一例は、Wonderware SuiteLink (商標) である。

【0045】

データ放出コンポーネントから放出されるデータは、Dynamic Data Exchange (DDE)、OLE for Process Control (OPC)、OPCアラームおよびイベント仕様 (OPC A&E) Unified Architecture (OPC-UA)、(OPC-UA)、OPC Express インターフェース (OPC-Xi)、TCP/IP、カスタムインターフェースを介してTCP/IPを渡るSSL (セキュア・ソケット・レイヤー)、ハイパーテキスト・トランスファー・プロトコル (HTTP)、Secure HTTP (HTTPS)、オープン・データベース・コネクティビティ (ODBC)、Microsoft Real-Time Data specification (RTD)、メッセージ・キュー、Windows Communication Foundation (WCF)、Profibus および Modbus などの工業用バスのプロトコル、Windows System パフォーマンス・カウンタ、システムに組み込まれたTCP/IP通信、MS-Windows (登録商標) ではないシステムに由来するTCP/IP通信、Linux (登録商標) に由来するTCP/IP通信、QNXに由来するTCP/IP通信、TRONに由来するTCP/IP通信、CコンパイラおよびTCP実装を提供する任意のシステムに由来するTCP/IP通信、内蔵のスクリプト言語を使用して記載されたスクリプト、ユーザインターフェースを介してヒトに提示されるデータ、ローカルディスクファイルから読み込まれたデータ、リモートからアクセス可能なディスクファイルから読み込まれたデータ、所有者のフォーマット、ユーザが定義したフォーマット、サーバへの拡張を通じて追加されたフォーマット、電子メール (Eメール)、並びにショート・メッセージ・サービス (SMS) のメッセージフォーマットから選択される一または複数の通信プロトコルを使用して送信され得る。

10

20

【0046】

さらに、データ収集コンポーネントで収集されたデータは、通信プロトコルに適したフォーマットであり得る。データ放出コンポーネントから放出されたデータは、通信プロトコルに適したフォーマットであり得る。データ収集コンポーネントで収集されたデータおよびデータ放出コンポーネントから放出されたデータもまた、括弧付き表現 (LISP など) のフォーマット、ハイパーテキスト・マークアップ言語 (HTML)、拡張マークアップ言語 (XML)、JavaScript Object Notation (JSON)、プロプライエタリ・バイナリ・フォーマット、ユーザ定義可能なテキストフォーマット、並びにサーバの拡張を通じて追加されたフォーマットから選択されるフォーマットであり得る。

30

【0047】

システムはさらに、TCP/IP接続およびサーバによりサポートされた一または複数のデータフォーマットを実装するアプリケーション・プログラミング・インターフェース (API) を備え、APIは上述した接続の確立においてプログラマを支援し得る。APIは、以下のプラットフォーム: 「C」プログラミング言語、「C++」プログラミング言語、MicrosoftのNetプログラミング環境、Microsoft SilverlightのRIAフレームワーク、Adobe FlashのRIAフレームワーク、Adobe AirのRIAフレームワーク、TCP/IP通信をサポートするプログラミング言語 (任意のスクリプト言語を含む)、並びにTCP/IP通信をサポートするRIAフレームワークのうち一または複数のプラットフォームのために実装され得る。

40

【0048】

RIAは、Microsoft Silverlight、Adobe Air、およびTCP/IP通信をサポートするRIAフレームワークから選択されるRIAフレーム

50

ワークを使用して実装され得る。R I Aフレームワークは、データを受信するためサーバへの第1の永続的T C P / I Pデータ接続を作り出し、サーバからデータを受信し、第2 T C P / I Pデータ接続を通じてサーバにデータを送信するためのサポートを備え得る。データは、オンデマンド、または登録によりサーバから受信され得る。第1 T C P / I Pデータ接続および第2 T C P / I Pデータ接続は、同じ接続であり得る。第2 T C P / I Pデータ接続永続的接続であり得る。第2 T C P / I Pデータ接続は、一時的な接続であり得る。サーバへのT C P / I Pデータ接続は、上述したA P I直接のT C P / I P接続、H T T P、およびH T T P Sから選択されるプロトコルであり得る。

**【0049】**

サーバからのデータは、括弧付き表現(L I S Pなど)のフォーマット、ハイパーテキスト・マークアップ言語(H T M L)、拡張マークアップ言語(X M L)、J a v a s c r i p t O b j e c t N o t a t i o n ( J S O N )プロプライエタリ・バイナリ・フォーマット、ユーザ定義可能なテキストフォーマット、並びにサーバへの拡張により追加されたフォーマットから選択される一または複数のフォームで受信され得るか、またはサーバへのデータは送信され得る。

10

**【0050】**

R I Aフレームワークはさらに、データを示すグラフィックディスプレイをユーザに提示するためのサポートを備え得る。グラフィックディスプレイは、テキスト表示、スライド、チャート、傾向グラフ、環状ゲージ、線形ゲージ、ボタン、チェックボックス、ラジオボタン、プログレスバー、基本のグラフィカル・オブジェクト、R I Aフレームワークにサポートされているコントロール、R I Aフレームワークを拡張するために作成されたカスタムコントロール、R I Aフレームワークを使用して実装された第三者コントロール、およびカスタム化されたグラフィック要素から選択される一または複数のグラフィック要素を備え得る。

20

**【0051】**

グラフィックディスプレイの構成情報は、サーバ上で保存されるとともにサーバからロードされ得る。グラフィック要素は、グラフィックディスプレイにおいて作成および変更され得る。グラフィック要素は、ユーザがカスタム可能なカスタム化されたグラフィック要素であり得、カスタム化はサーバ上で保存され得る。カスタム化は、R I Aフレームワークに実装されたアプリケーションに変更を必要とすることなくプログラマにより行われ得る。カスタム化されたグラフィック要素は、他のグラフィック描写でユーザが使用するために利用可能であり得る。これらのカスタム化は、ユーザインターフェース・アプリケーションにより元々サポートされているグラフィック要素に全て追加して、新たなディスプレイを作成し、既存のディスプレイを変更するためのものであり得る。グラフィック要素は、ユーザが変更可能な一または複数の特性を備え得、プログラマにより選択可能であり得る。ユーザのグラフィック要素との相互作用により、ユーザインターフェース・アプリケーションがデータに対する変更をサーバに放出させる。ユーザ専用モードが提供されると、ユーザによるグラフィックディスプレイの作成または変更を許可せず、また読み取り専用モードが提供されると、ユーザによるグラフィック要素との相互作用を許可しない。システム・アドミニストレータは、ユーザと、ユーザインターフェース・アプリケーションがユーザ専用モードおよび読み取り専用モードの1つで動作するグラフィックディスプレイを選択し得る。ユーザは、識別が必要とされる場合に自身を識別することが必要とされ、ユーザインターフェース・アプリケーションは、ユーザ専用モードおよび読み取り専用モードの少なくとも1つのモードで動作し得る。有利に、本発明の特徴によると、任意のユーザR I A端末を通してグラフィックディスプレイの変更が可能であり、保存により生じた変更は、サーバに接続された他の全てのR I A端末が直ぐに利用することができる。

30

40

**【0052】**

図1を参照すると、一実施形態において、R I A 1 0 1は、データソースおよびデータサーバ100の両方として機能するプログラムに対しデータが直接接続する。この状況は

50

、データソースがTCP/IPプロトコルを用いる生データのコレクタおよびトランスミッタの両方である場合に生じ得る。PLC内に組み込まれたOPC-UAサーバがこの例である。別の例は、データソースとして機能し、カスタム用TCP/IPインターフェースを提供するTCP/IPサーバの能力を提供する組み込み装置である。さらに別の例は、TCP/IPインターフェースを提供する株式市場のデータ供給である。

#### 【0053】

図2を参照すると、一実施形態において、別の構成は別個になったデータソース202とサーバ203とを含む。この構成は、データソース202のデータプロトコルを、RIA201が処理できるTCP/IPプロトコルに変換することにより通信モデルを拡張する。これにより、サーバ203がTCP/IPのインターフェースを直接提供しないデータソース202と相互作用することが可能になるため、データソース202の数および種類が非常に拡大される。

10

#### 【0054】

図3を参照すると、一実施形態において、サーバ303は、同時に2つ以上のデータソース302と2つ以上のRIA301との接続を維持し得る。この一層複雑な構成は、データソース302とRIA301とからのデータを、TCP/IPネットワーク上のどこからでもアクセス可能な単一のデータセットに集約する。

#### 【0055】

別の実施形態では、システムは、複数のデータソースおよび/または一若しくは複数のRIAと内部接続された複数のサーバを含み得る。

20

#### 【0056】

図4を参照すると、一実施形態におけるRIA動作の方法および制御フローが示される。RIAは、一または複数は組み込まれ得るが、明確な停止基準を必要としない。RIAは、ユーザRIAを含むウェブブラウザまたはウェブページを閉じたときに暗黙的に停止される。RIAは、2つの制御フローを同時に従い、単一のプログラムスレッドにインターリーブされるか、または別個のプログラムスレッドに実装され得る。方法は、RIAに特異的な追加処理を含み得る。

#### 【0057】

1つ目の制御フローでは、RIAはサーバへの接続を確立および維持、並びにサーバから利用可能なデータにおける変化への応答を試みる。まず、RIAは接続の確立を試みる(ステップ401)。接続が成功しない場合、RIAはただひたすらその接続を試みる。接続が成功する場合(ステップ402)、次いでRIAは、全てまたは一部分のデータセットに参加し得る(ステップ403)。代替的に、接続していることに基づいてサーバがデータにRIAを黙示的に加えることが可能であり、この場合ステップ403は省略されてよい。参加に加えて、RIAは、接続の更新の最短時間またはタイムアウトパラメータなどデータ送信動作を構成するため、サーバに他の情報も送信し得る。

30

#### 【0058】

接続を確立すると、RIAはサーバからデータにおける変化通知を待つ(ステップ404)。データ変更が生じている場合(ステップ405)、次いでRIAは何らかの方法でそのデータを処理する(ステップ407)。この処理は、RIAの内部状態を変更、グラフィック描写を変更、RIA設計者が決定する音または他の任意のプログラミングの応答を出すためであり得る。データ変化が生じない場合、RIAは何等かの理由でサーバへの接続が切れたかを特定するために調べる。接続が切れていない場合、RIAは再びデータ変化が生じるのを待つ(ステップ404)。接続が切れている場合、その時RIAは再びサーバへの接続を試みる(ステップ401)。

40

#### 【0059】

ステップ401~ステップ407と同時に、RIAはユーザ入力を受入れ、ユーザは、サーバに伝搬できるデータの変化を作成することが可能となる。RIAは、別個のプログラムスレッドまたはステップ401~ステップ407と多重化してユーザ入力を待つ(ステップ420)。

50

## 【0060】

図4は、別々にスレッド構成された方法を例示する。ユーザ入力が生じた場合（ステップ421）、次いでRIAは、結果生じたデータのサーバへの送信を試みることができる。RIAは、サーバが接続されているか（ステップ422）を確かめるためにまず調べることでこの試みを行う。接続されている場合、RIAは新しいデータをサーバに送信する（ステップ423）。接続されていない場合、RIAはさらに入力を待つ（ステップ420）。サーバ接続のチェック（ステップ422）は、データを送信するために暗黙的に試みられ、この場合、ステップ422およびステップ423は実質的に組み合わせられる。

## 【0061】

RIAはまた、ユーザ入力を受入れられないなど相互作用できない場合、ステップ420～ステップ423を省略することができる。

10

## 【0062】

図5を参照すると、一実施形態におけるデータサーバの動作方法が示される。サーバは、0以上のRIA接続にデータを出しながら、同時に0以上のデータソースからデータを収集し得る。主要な2つの制御フローを別々のスレッドか、または単一のスレッド内で2つの制御フローパスをインターリーブすることにより実装することができる。

## 【0063】

データソースと相互に作用するために、サーバはまずそのデータソースとの接続を確立しなければならない（ステップ501）。通常、サーバがデータソースへの該接続を開始する。場合によっては、データソースがサーバへのその接続を開始する。接続が成功する場合（ステップ502）、サーバはデータソースからデータを収集し始める（ステップ503）。接続が失敗する場合、サーバはデータソースへの接続を再度試みる（ステップ501）。データソースがサーバへの接続の開始側である場合、次いでステップ501およびステップ502は単一の待ち状態へと折り重なり、サーバは接続するため受動的にデータソースを待つ。データの収集（ステップ503）は、データソースに適した方法に従い、一方のデータソースと別のデータソースで異なり得る。サーバは、データをサーバに表示することができる任意のデータソースと適応するように作成することができる。新規データがデータソースから利用可能になると（ステップ504）、サーバはそのデータをサーバの内部のデータ表現に変換する。これにより、サーバは異なるデータ表現を使用して種々のデータソースを集めることが可能となる。ステップ506は、データソース、サーバ、およびRISが全て同じデータ表現を使用する簡単なケースの場合に省略することができる。次いでサーバは、データの各RIAへの送信を試みる。サーバはまず、RIAが接続されるように確立し得る（ステップ507）。一または複数のRIAが接続される場合、サーバはRIAに好適な表現へとデータを変換し（ステップ508）、そのデータを接続された各RIAに送信する（ステップ509）。RIAが接続されない場合、サーバはデータソースからデータを収集し続ける（ステップ503）。サーバは無期限にこのシーケンスを繰り返す（ステップ501～ステップ509）。データシンクがデータソースからデータを必要とするサーバに接続されていないとき、サーバはそのデータソースからデータを収集しないように選択し得る。

20

30

## 【0064】

データソースからデータを収集することと同時に、またはインターリーブして、サーバはRIAからの接続も管理する。サーバはRIAからの接続を待つ（ステップ520）。RIAがサーバとの接続を試みる時（ステップ521）、サーバは接続を受入れ（ステップ522）、他のRIAからの接続を待ち続ける。RIAとの接続を待つ間に、サーバは既存のRIA接続が切れているかを特定しなければならない（ステップ523）。RIAが切れている場合、RIA接続はサーバにおける追跡から解除され（ステップ524）、それにより、接続が切れたRIAにはその後データを送信するための試みはなされない（ステップ509）。サーバは、このシーケンスを無期限に繰り返す（ステップ520～ステップ524）。サーバは、認証失敗またはサーバがRIAインスタンスからの同時接続の最大数に限定を加えるなど、何らかの理由でサーバが接続を拒否し得るように、RI

40

50

Aが接続を試みる(ステップ522)ときに、承認基準を適用し得る。

【0065】

データソースからデータを収集すること、R I Aからの新規接続を管理することと同時に、またはインターリーブして、サーバはまた、既に接続されたR I Aからデータを受信し得る。サーバはR I Aから届くまでデータを待つ(ステップ530)。新規データが届くと(ステップ531)サーバはこのデータをサーバの内部のデータフォーマットに変換する(ステップ532)。次いでサーバは、いずれかのR I Aが現在接続されているかを特定する(ステップ533)。次いでサーバは、データR I Aが受信するのに好適なフォーマットにデータを変換し(ステップ534)、現在接続されている各R I Aにデータを送信する(ステップ535)。次いでサーバは、この情報の変更を必要とするいずれかのデータソースが現在接続されているかを判断する(ステップ536)。現在サーバに接続されている情報を必要とする各データソースのため、サーバはそのデータソースに好適なフォーマットにデータを変換し(ステップ537)、データを送信する(ステップ538)。サーバはこのシーケンスを無期限に繰り返す(ステップ530~ステップ538)

10

【0066】

ステップ501~ステップ509は、サーバが接続し得る各データソースに対し、繰り返し反復することができる。

【0067】

ステップ520~ステップ524は、サーバが接続を受信し得る各R I Aに対し、繰り返し反復することができる。

20

【0068】

ステップ530~ステップ538は、接続された各R I Aに対し反復され得るか、ステップ530は接続された全てのR I Aを同時に待つように多重に行われるか、またはこれらの選択肢を組み合わせて行われ得る。

【0069】

図4および図5で例示した方法は、R I Aおよびデータサーバの両方に対する明確な停止条件、サーバに接続するためデータソースを受動的に待つサーバの能力、R I Aに積極的に接続するサーバの能力、複数のデータソースへの接続を同時に管理するサーバの能力、複数のR I Aへの接続を同時に管理するサーバの能力、および複数のR I Aからデータを同時に受信するサーバの能力を含む追加の能力を備えるように改良され得ることが理解される。

30

【0070】

図6を参照すると、一実施形態において、複数のR I A 601への接続を同時に管理するデータサーバ603の能力によると、有利にR I A 601はサーバを介して別のR I A 601と通信することが可能となる。R I A 601からサーバ603に送信される任意の情報は、R I A 601がデータソースのように扱われ、サーバに接続されており、データに参加している他の任意のR I A 601にそのデータを伝搬する。驚くべきことに、これにより実時間で互いに更新するR I Aのネットワークを効果的に作り出す。実際に、サーバ603は、サーバがサポートするプロトコルの任意の組み合わせを使用して、任意の数のクライアント・アプリケーション間で通信することができるように使用され得る。

40

【0071】

図7を参照すると、一実施形態では、本発明の実質的な利益は、そうでなければネットワークを介してアクセスできないソースが発信元であるデータをR I A 701に提示する能力である。この実施形態では、Microsoft Excelのような表計算アプリケーション705を基にしたデータは、本発明の前は可能ではなかったローカルまたは広域ネットワークを介して送信され得る。Microsoft Excelからのデータ送信は、DDE、RTDまたは不安定なスクリプトを用いたアドホック通信に限定される。R I Aとの実時間通信を可能とするDDE、RTDまたはスクリプトを通じたアドホック通信を含む、Microsoft Excelから供給されるプロトコルはR I Aと実時間で通信できない。本発明によれば、任意のアプリケーションが任意のTCP/IPネッ

50

トワークを用いて表計算データと実時間で通信することが可能となり、表計算データ用アプリケーションの範囲が広大に拡大する。R I A 7 0 1 とのこの通信能力の組み合わせは、単一のウェブブラウザ接続を介した単一の表計算と複数の同時ユーザが相互接続する能力を提供する。この同一の機能性は、サーバ 7 0 3 がサポートする任意のプロトコルにまで拡張する。

#### 【 0 0 7 2 】

ウェブブラウザ内で R I A を実行する場合、R I A は、ウェブサーバを使用してウェブブラウザに提供されなければならない。すなわち、ユーザは、ウェブブラウザに URL を入力するか、またはその URL のためにウェブページ内のリンクをクリックして R I A を含むウェブページをウェブブラウザにロードさせる。URL は、M i c r o s o f t I I S ( 商 標 ) または A p a c h e ( 商 標 ) などのウェブサーバによりサービスが提供されている。よって、R I A をロードおよび接続するイベントのシーケンスは以下の通り。

- 1 . ユーザがウェブブラウザで URL を選択する
- 2 . ウェブブラウザがウェブサーバから R I A を含むページをロードする
- 3 . ウェブブラウザが R I A を開始する
- 4 . R I A が T C P / I P を介してデータサーバと接続する
- 5 . R I A がデータサーバのデータに参加する
- 6 . データサーバが参加に従ってデータを送信する
- 7 . R I A が接続を絶つかまたは停止するまでデータサービスが続く

#### 【 0 0 7 3 】

このシーケンスには、ウェブサーバが存在し、R I A に配信するように構成されていることが必要である。システムコンポーネントの数を縮小し、R I A のデータ機能とウェブの機能とを一層強く一体化するため、データサーバにウェブサーバの能力を組み込むことが便利であろう。

#### 【 0 0 7 4 】

組み込まれたブラウザまたは別個の非ブラウザホスト ( R I A 用ブラウザ外モードと呼ぶ場合もある ) から実行されて R I A セッションを起動し得ることが当業者には容易に明らかになるであろう。URL およびウェブブラウザは、ユーザにとって明らかではない場合がある。従って、上述したイベントの最初の 3 つのステップが変更されてこれらの代替の実施形態を反映し得る。

#### 【 0 0 7 5 】

本発明では、R I A はデータを使用または生成することができる R I A フレームワークを使用して記載された任意のアプリケーションであり得る。

#### 【 0 0 7 6 】

一実施形態では、R I ディスプレイはユーザにとって視覚的な実時間データである。視覚コンポーネントは、ゲージ、傾向グラフ、プログレスバー、ボタン、画像、及びデスクトップのアプリケーションで一般的な他の視覚表現であり得る。可能性として多種多様な表現が存在するため、特定のデータセットに最も好適な表現は、別のデータセットに最も好適な表現と異なり、R I A は、ユーザが設定可能とすべきである。これは、ユーザがデータを視覚「オブジェクト」と関連付けることが可能な視覚構成ツールとともに提示されることを意味する。これらの視覚オブジェクトの収集は共に、関連情報を表示するために共通して使用される視覚「ページ」に配置することができる。次いで、ユーザは、異なる一式の関連情報を表示するために複数のページを作成することができる。

#### 【 0 0 7 7 】

データの視覚化をカスタマイズさせる能力をユーザに提供するために、R I A は、エディタとして一層共通して参照される、一体化したカスタム機能、または別々になったカスタム機能のいずれかを提供しなければならない。このエディタは、ユーザが実時間データを表示するページの視覚的なデザインを特定する手段を提供する。また、エディタは、視覚オブジェクトの特殊なバージョンを設計する能力も提供し得る。

#### 【 0 0 7 8 】

個々の視覚オブジェクトのデザイン、視覚ページのデザインに関する情報は、ウェブサーバに格納されるべきである。これにより、ユーザは、ウェブサーバで必要なアクセス特権を用いて他の任意のユーザが閲覧できるデータの視覚化を作り出すことが可能となる。RIAは、ウェブサーバと相互作用して文書を格納し、XMLなどのフォーマットで格納された文書を回収する。この情報通信は、既存の実時間データ接続を用いるか、またはHTTP接続を使用して行われ得る。データサーバ内に内蔵されたウェブサーバは、必要ではないがこの送信の実施を簡易にする。

【0079】

本発明の一実施形態では、本発明の方法を実装するシステムは、以下のソフトウェアアプリケーションを含む。

1. データサーバとして機能するCogent DataHub (商標) (Cogent Real-Time Systems Inc.)

2. ウェブサーバとして機能するCogent DataHub (Cogent Real-Time Systems Inc.)

3. RIAフレームワークとして機能するMicrosoft Silverlight (マイクロソフト株式会社)

4. Microsoft Silverlightのためのプロトコル実装層として機能するDataHub API for .Net (Cogent Real-Time Systems Inc.)

5. ウェブブラウザに実時間データを表示するためのRIAとして機能するDataHub WebView (商標) (Cogent Real-Time Systems Inc.)

6. 視覚オブジェクトおよびページ設計のためのディスプレイ・エディタとして機能するDataHub WebView (Cogent Real-Time Systems Inc.)

【0080】

加えて、Cogent DataHubは以下を含む様々なデータソースからデータを送信または受信し得る。

1. 表計算アプリケーションとして機能するMicrosoft Excel (商標) (マイクロソフト株式会社)

2. データ通信インターフェースとして機能するOPC-DAサーバ (各社)

3. データ通信インターフェースとして機能するOPC-UAサーバ (各社)

4. データ通信インターフェースとして機能するOPC Xiサーバ (各社)

5. データベースインターフェースとして機能するODBCサーバ (各社)

【0081】

図8を参照すると、一実施形態では、特定の実装に応じて、0以上のデータソースがCogent DataHubに配置される。

【0082】

本発明の代替的な実施形態では、RIAフレームワークは、持続したネットワーク接続をサポートすることが可能な任意のRIAフレームワークであり得る。そのような代替的なRIAフレームワークの例は、Adobe Flash (商標)、およびAdobe Flex (商標)を含む。他のRIAフレームワークもまた好適であり得ることが想定される。

【0083】

RIAは、サーバのTCP/IP通信プロトコルを使用してデータを消費または生成することができるRIAフレームワークを使用して作成された任意のアプリケーションであり得る。

【0084】

RIAフレームワークは、例えば、HTML5が必要とされるTCP通信メカニズムをサポートしている場合など、ウェブブラウザに不可欠なものであり得る。

## 【 0 0 8 5 】

データサーバは、構築された R I A がアクセスできる T C P / I P 通信方法を供給できる限り、データソースからデータを収集、またはデータソース自体として機能するように設計された任意のアプリケーションであってよい。

## 【 0 0 8 6 】

データソースは、サーバにおける表示に好適なフォーマットに変換できる実時間データを生成することが可能な任意のアプリケーションまたはシステムであり得る。

## 【 0 0 8 7 】

また、データソースは、サーバにおける表示に好適なフォーマットに変換できる非実時間データを生成することが可能な任意のアプリケーションまたはシステムであり得る。サーバは、発信元のデータが実時間でない場合でも R I A にデータを提供するために、前記データを繰り返しポーリングするか、または参加により収集することができる。例えば、データベース管理システム ( D B M S ) は、一般的に実時間でなくても、データは繰り返してポーリングしてサーバ内で定期的に更新データを生成し、 D B M S における疑似的な実時間のデータ図を R I A に供給することができる。

10

## 【 0 0 8 8 】

サーバおよびデータソースは、単一のアプリケーションに組み合わされ得、その場合、T C P / I P 接続を介してそのデータへのアクセスを提供する O P C - U A サーバ、または内蔵された装置であり得る。

## 【 0 0 8 9 】

ウェブサーバは、R I A を含むウェブページを提供することができる任意のアプリケーションであり得る。

20

## 【 0 0 9 0 】

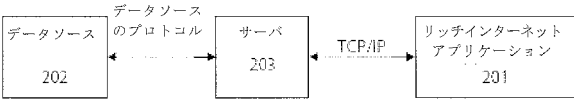
T C P / I P ソケットを開き、相互に作用することができるコンパイルまたは解釈された任意のコンピュータ言語を使用して開発されたプログラムは、ウェブブラウザ内で実行または実行しない R I A の代わりに使用され得る。同様に、本発明の方法はまた、R I A の代わりに、ブラウザ、ブラウザ以外のホストで直接またはブラウザの拡張を通じて実行可能なコードを使用して実施され得、ブラウザ、ブラウザ外のホスト、またはブラウザの拡張は開くことができ、T C P / I P ソケットと相互に作用することができ、持続したネットワーク接続することができ、任意にグラフィック能力を提供することができる。

30

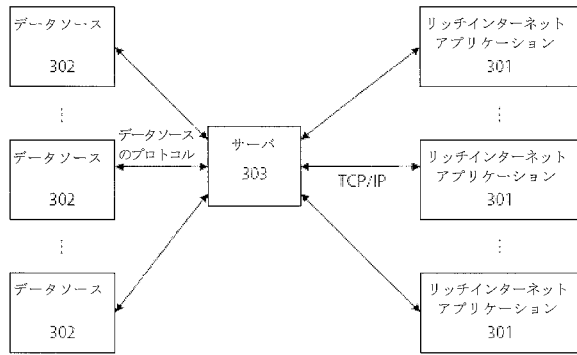
【 図 1 】



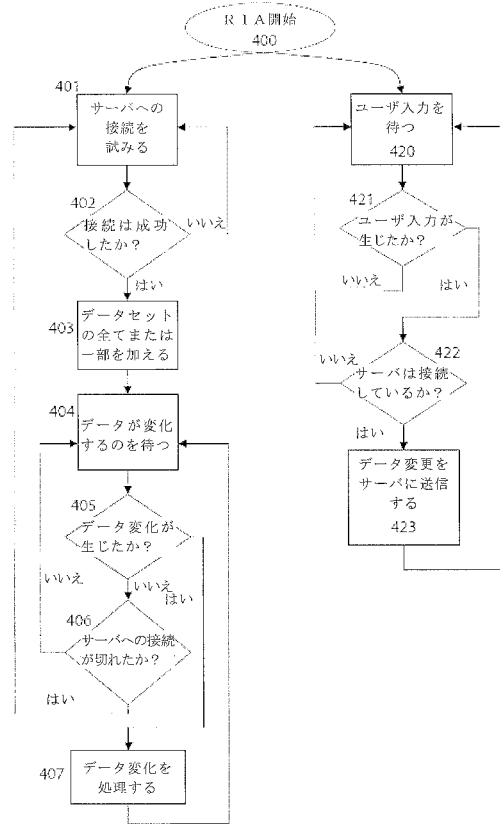
【 図 2 】



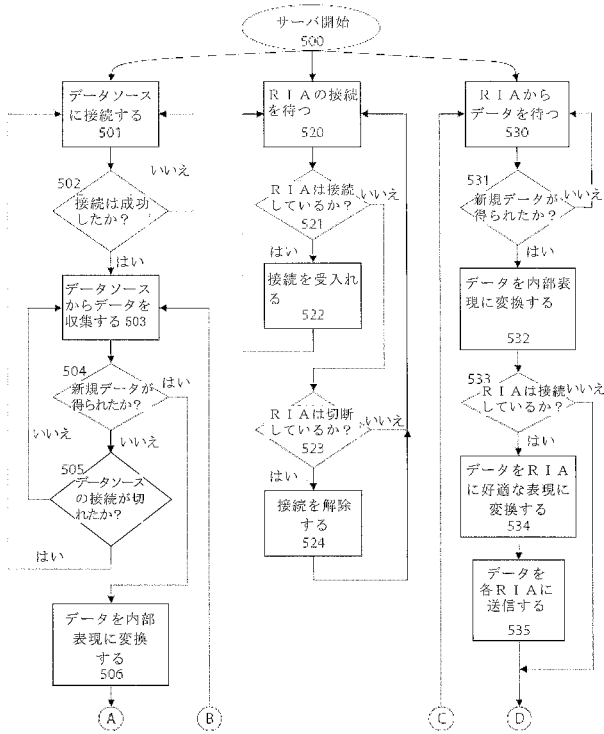
【 図 3 】



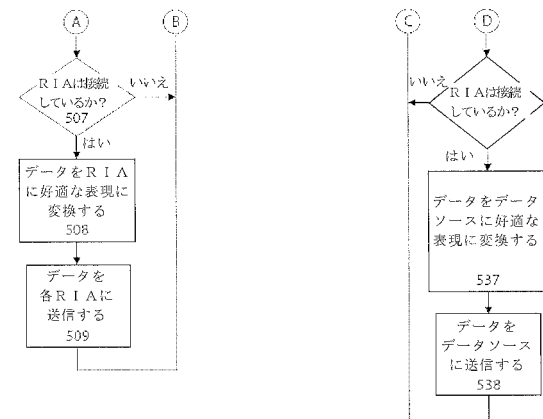
【 図 4 】



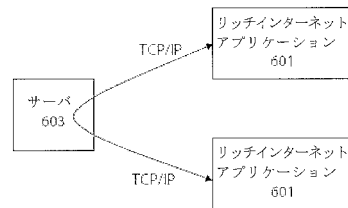
【 図 5 a 】



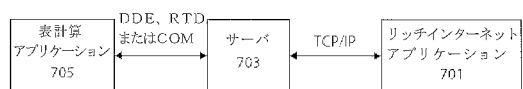
【 図 5 b 】



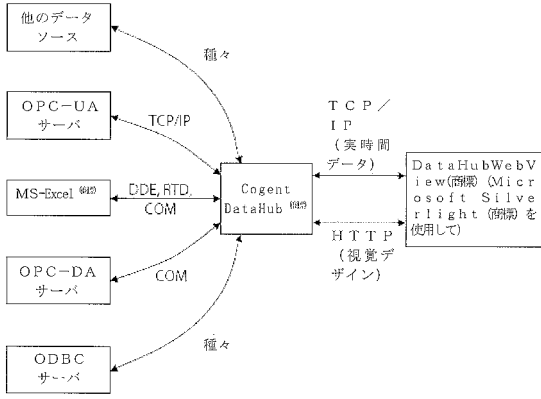
【 図 6 】



【 図 7 】



【 図 8 】



---

フロントページの続き

4 . Q N X

(72)発明者 トーマス, アンドリュー

カナダ国, オンタリオ州 エル7ジー 5エックス7, ジョージタウン, スイート 253, 16  
2 ゲルフ ストリート

Fターム(参考) 5B089 GA11 GB03 GB04 HA06 HA10 HB02 HB05 JA36 JB10 JB16  
KA05