

(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2011-159275
(P2011-159275A)

(43) 公開日 平成23年8月18日(2011.8.18)

(51) Int.Cl.	F I	テーマコード(参考)
G06F 9/46 (2006.01)	G06F 9/46 420A	5E501
G06F 3/048 (2006.01)	G06F 3/048 651A	

審査請求 未請求 請求項の数 16 O L 外国語出願 (全 20 頁)

(21) 出願番号 特願2010-235957 (P2010-235957)
 (22) 出願日 平成22年10月20日(2010.10.20)
 (31) 優先権主張番号 12/610,096
 (32) 優先日 平成21年10月30日(2009.10.30)
 (33) 優先権主張国 米国(US)

(特許庁注:以下のものは登録商標)

1. Bluetooth

(71) 出願人 510279239
 ダッソー システムズ エノビア コーポ
 レーション
 アメリカ合衆国 01851 マサチュー
 セッツ州 ローエル チェルムズフォード
 ストリート 900 タワー2 フロー
 ー5
 (74) 代理人 110001243
 特許業務法人 谷・阿部特許事務所
 (72) 発明者 デービット エドワード テュークスバリー
 アメリカ合衆国 01740 マサチュー
 セッツ州 ボルトン マウント ワチュセ
 ット レーン 9

最終頁に続く

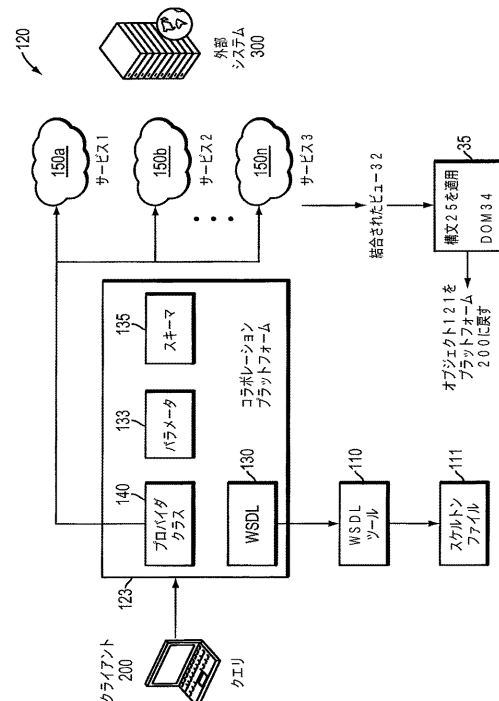
(54) 【発明の名称】 ターゲットシステムのデータへのアクセスを提供するコンピューターの方法およびシステム

(57) 【要約】

【課題】コンピューターシステムおよび方法が、ターゲットシステムのウェブ(グローバル・コンピューターネットワーク)サービスデータへのアクセスを提供する。

【解決手段】ターゲットシステムは、複数のウェブサービスを介してデータを公開する。アプリケーションインターフェイスが、ターゲットシステムと相互作用するように適合され、ターゲットシステムのためのアプリケーションの間で既存の(あらかじめ定義された)ウェブサービスを再利用する。アプリケーションインターフェイスは、公開されたデータにクエリを行う。マッピングメンバーが、公開されたデータのアプリケーションインターフェースクエリと、対象アプリケーションにおいて使用可能なオブジェクトの構文との間をマップする。マッピングメンバーは、対象アプリケーションを、クエリに応答して生成されたオブジェクトインスタンスのデータにアクセスできるようにする。

【選択図】図1C



【特許請求の範囲】**【請求項 1】**

ターゲットシステムのネットワークサービスデータへのアクセスを提供するコンピュータシステムであって、

複数のグローバルコンピューターネットワークサービスを介してデータを公開するターゲットシステムと相互作用し、かつ、公開されたデータにクエリを行うアプリケーションインターフェースであって、

前記ターゲットシステムのためのアプリケーションの間でグローバルコンピューターネットワークサービスを再利用するように、前記ターゲットシステムのために、2つ以上のアプリケーションにわたって前記複数のグローバルコンピューターネットワークサービスのうちの1つを共通して使用する、アプリケーションインターフェースと、

前記公開されたデータのクエリであって、前記アプリケーションインターフェースによって実装されるクエリ(a)と、対象アプリケーションにおいて使用可能なオブジェクトの構文(b)との間をマッピングし、前記対象アプリケーションを、前記クエリに回答して生成されたオブジェクトインスタンスのデータにアクセスできるようにするマッピングメンバーと

を含むことを特徴とするコンピュータシステム。

【請求項 2】

前記クエリの実装において、前記アプリケーションインターフェースは、前記ターゲットシステムからデータの結合されたビューをもたらす方法で、前記ターゲットシステムの前記グローバルコンピューターネットワークサービスのうちの少なくとも2つへ呼び出しを連鎖させ、

前記アプリケーションインターフェースが、前記ターゲットシステムの既存の資産を再利用するように、前記少なくとも2つのグローバルコンピューターネットワークサービスは、前記ターゲットシステムによってあらかじめ提供されていることを特徴とする請求項1に記載のコンピュータシステム。

【請求項 3】

前記マッピングメンバーは、前記結合されたビューから、前記対象アプリケーションのオブジェクトのクエリ構文をマップすることを特徴とする請求項2に記載のコンピュータシステム。

【請求項 4】

コラボレーションプラットフォームによって用いられているコンピュータシステムであって、

前記ターゲットシステムは、前記コラボレーションプラットフォームの外部システムであり、分散オブジェクトクライアントとして構成されていることを特徴とする請求項1に記載のコンピュータシステム。

【請求項 5】

前記対象アプリケーションは、XML(extensible mapping language)クライアント、RMI(remote method invocation)クライアント、EJB(enterprise JAVA Bean)クライアント、およびJNI(JAVA Native Interface)クライアントのうちのいずれかとして実装されることを特徴とする請求項1に記載のコンピュータシステム。

【請求項 6】

ターゲットシステムのネットワークサービスデータにアクセスするコンピュータ実施方法であって、

対象アプリケーションと、複数のグローバルコンピューターネットワークサービスを介してデータを公開するターゲットシステムとの間で相互作用するステップであり、公開されたデータのクエリを実装するように相互作用するステップであって、

前記ターゲットシステムのためのアプリケーションの間でグローバルコンピューターネ

10

20

30

40

50

ットワークサービスが再利用されるように、前記ターゲットシステムのために、2つ以上のアプリケーションにわたる前記複数のグローバルコンピューターネットワークサービスのうちの1つを共通して使用する、相互作用するステップと、

前記公開されたデータの前記クエリ(a)と、前記対象アプリケーションにおいて使用可能なオブジェクトの構文(b)との間をマッピングするステップであって、前記対象アプリケーションを、前記クエリに応答して生成されたオブジェクトインスタンスのデータにアクセスできるようにする、マッピングするステップと

を含むことを特徴とするコンピューター実施方法。

【請求項7】

前記クエリの実装において、前記相互作用するステップは、前記ターゲットシステムからデータの結合されたビューをもたらす方法で、前記ターゲットシステムの前記グローバルコンピューターネットワークサービスのうちの少なくとも2つへの呼び出しを連鎖させ

10

、前記相互作用するステップが、前記ターゲットシステムの既存の資産を再利用するように、前記少なくとも2つのグローバルコンピューターネットワークサービスは、前記ターゲットシステムによってあらかじめ提供されていることを特徴とする請求項6に記載のコンピューター方法。

【請求項8】

前記マッピングするステップは、前記結合されたビューから、前記対象アプリケーションのオブジェクトのクエリ構文をマッピングするステップを含むことを特徴とする請求項7に記載のコンピューター方法。

20

【請求項9】

コラボレーションプラットフォームによって利用される方法であって、

前記ターゲットシステムは、前記コラボレーションプラットフォームの外部システムであり、分散オブジェクトクライアントとして構成されていることを特徴とする請求項6に記載のコンピューター方法。

【請求項10】

前記対象アプリケーションは、XML(extensible mapping language)クライアント、RMI(remote method invocation)クライアント、EJB(enterprise JAVA Bean)クライアント、およびJNI(JAVA Native Interface)クライアントのうちのいずれかとして実装されることを特徴とする請求項6に記載のコンピューター方法。

30

【請求項11】

アプリケーションを介して少なくとも1つの外部システムによって提供される少なくとも2つのウェブサービスにアクセスし、クエリを行うためのコンピューターシステムであって、(i)各ウェブサービスは、少なくとも別のウェブサービスと統一されていないデータを保持し、(ii)前記アプリケーションは、各ウェブサービスのためのそれぞれのインターフェースを含み、

クエリを行い、統一された方法において前記ウェブサービスからクエリ結果を受信するために、前記アプリケーションのインターフェースを適合する適合ユニットと、

40

ウェブサービスからの統一されていないクエリ結果を、統一されたデータに結合するためのマッピング手段と

を含むことを特徴とするコンピューターシステム。

【請求項12】

前記適合ユニットは、前記アプリケーションのインターフェースを適合して、前記ウェブサービスのうちの2つ以上への呼び出しを連鎖させ、クエリ結果の結合したビューをもたらすことを特徴とする請求項11に記載のコンピューターシステム。

【請求項13】

前記マッピング手段は、クエリ構文を統一されたデータスキーマにマップすることを特徴とする請求項12に記載のコンピューターシステム。

50

【請求項 14】

前記インターフェースを適合する前記適合ユニットは、事実上、前記少なくとも1つの外部システムの資産の再利用であることを特徴とする請求項11に記載のコンピューターシステム。

【請求項 15】

前記少なくとも1つの外部システムは、分散オブジェクトクライアントとして構成されていることを特徴とする請求項11に記載のコンピューターシステム。

【請求項 16】

コラボレーションプラットフォームであることを特徴とする請求項11に記載のコンピューターシステム。

10

【発明の詳細な説明】

【技術分野】

【0001】

ターゲットシステムのデータへのアクセスを提供するコンピューターの方法およびシステムに関する。

【背景技術】

【0002】

インターネットなどのグローバル・コンピューターネットワークが、新たなコンピューティング・プラットフォーム、およびコンピューティング・アプリケーション（ソフトウェア・アプリケーション設計を含む）を提供している。例えば、インターネット規模の分散コンピューティングは、いわゆるプラットフォーム・コンピューティングのようなネットワークを提供する。プラットフォームとしてのネットワークは、（ウェブ対応アプリケーションとして構成された）アプリケーションについて、ブラウザを介して全面的に実行できるようにする。さらに、プラットフォームとしてのネットワークによって、ユーザは、さまざまな役割にわたり、またはその役割の中で、共同作業することが可能になる。

20

【0003】

所与のウェブ対応アプリケーションに関して、ユーザ/クライアント（分散オブジェクトアーキテクチャクライアント）は、個々のブラウザを介してアプリケーションを実行する。ブラウザは、アプリケーションサーバー（分散オブジェクト・サーバー）と通信する。さらにブラウザは、ウェブサーバーと通信する。認証サービス、トランザクションサービス、ネーミングサービス、アドミニストレーションサービス、およびセキュリティサービスなどの多くのサービスが、アプリケーションサーバーおよびウェブサーバーの両方に複製される。各ユーザ/クライアントは、一般的に、別々にプラットフォームに登録される。また、各ユーザ/クライアントは、各組のウェブサーバーサービスの実行に参与する。

30

【0004】

今までのところ、プラットフォームにおけるクライアントとしてのユーザが、複数のアプリケーションにわたって、データをコピーし、他のデータに参与する、または既存のウェブサーバーサービスを活用するためのメカニズムはない。

【0005】

プラットフォームとしてネットワークを利用する、コンピューター実施プログラムおよび協調システムの産業上の一例は、製品ライフサイクル管理（Product Lifecycle Management / PLM）システムである。PLMソリューションは、企業活動の拡大を概念にして、製品開発のために、構想の段階から製品寿命の終わりまで、企業が、製品データを共有し、共通プロセスを適用し、企業知識を活用することを支援するビジネス戦略を指す。この概念によれば、企業は、その企業の部門のみならず、ビジネスパートナー、サプライヤ、OEM（Original Equipment Manufacturers）、および顧客などの他の関係者からも構成される。これらの関係者を含めることにより、PLMは、このネットワークが単一のエンティティとして機能して、製品およびプロセスを概念化し、設計し、構築し、サポートすることを可能にする

40

50

。

【 0 0 0 6 】

PLMソリューションの中には、コンピューター支援技術があり、コンピューター支援設計 (Computer - Aided Design / CAD) が知られている。コンピューター支援設計は、製品設計を記述するためのソフトウェアソリューションに関する。同様に、CAEは、コンピューター支援エンジニアリング (Computer - Aided Engineering) の頭字語であり、例えば、将来製品の物理的な動作をシミュレートするためのソフトウェアソリューションに関する。CAMは、コンピューター支援製造 (Computer - Aided Manufacturing) を表し、一般的に、製造過程および製造作業を定義するためのソフトウェアソリューションを含む。いくつかのPLMソリューションは、例えば、デジタルモックアップ (製品の3Dグラフィカルモデル) を作成することによって、製品を設計し、開発することを可能にする。適切なアプリケーションを使用して、最初にデジタル製品を定義し、シミュレートすることができる。次いで、無駄のないデジタル製造プロセスを定義し、モデル化することができる。

10

【 0 0 0 7 】

例えば、一般に、Dassault Systemsにより (CATIA、ENOVIA、およびDELMIAの商標下において) 提供されるPLMソリューションは、製品エンジニアリング知識を構成するエンジニアリングハブ、製造エンジニアリング知識を管理するマニファクチャリングハブ、ならびにエンジニアリングハブおよびマニファクチャリングハブの両方への事業統合および事業結合を可能にするエンタープライズハブを提供する。これら全てのシステムはともに、製品、プロセス、リソースをリンクするオープンなオブジェクトモデルを実現して、動的な知識ベースの製品創造、および最適化した製品定義、製造準備、生産およびサービスを推進する意思決定支援を可能にする。

20

【 0 0 0 8 】

このPLMソリューションは、製品のリレーショナルデータベースを含む。リレーショナルデータベースは、一組のテキストデータを含み、かつデータ間の関係を含む。データは、一般的に、データの階層順に並べられた、製品に関連した技術データを含み、検索できるようにインデックス付けされている。データは、しばしばモデル化された製品およびプロセスである、モデル化されたオブジェクトを表す。

【 0 0 0 9 】

製品構成、プロセス知識、およびリソース情報を含むPLM情報は、一般的に、協調的な方法により編集されることを目的とする。例えば、協調的な方法で作業する設計者のチームは、ウェブ対応アプリケーションを介して、リモートサイトから製品に関連した共有情報にアクセスすることによって、個々に貢献して製品を設計する。各ユーザまたはリモートサイトは、システムプラットフォームに別々に登録され、それぞれが、このプロジェクトのアプリケーション用のそれぞれの組のウェブサーバーサービスの実装に関与する。ユーザは、もともと他のアプリケーション用に実装される既存のウェブサービスを有していることがあるけれども、進行中のプロジェクトに関して、これらのウェブサービスを再利用、または共有するためのメカニズムがない。プラットフォームのクライアントのユーザまたはアプリケーションが、既存のウェブサービスをコピーする、または別の方法で活用する必要はある。

30

40

【 先行技術文献 】

【 特許文献 】

【 0 0 1 0 】

【 特許文献 1 】 米国特許第 6 9 3 4 7 0 9 号明細書

【 発明の概要 】

【 0 0 1 1 】

本発明は、前述の先行技術の欠点に取り組む。詳細には、本発明は、2つ以上のアプリケーションにわたるクライアントシステムのウェブ (グローバルコンピューターネットワークサーバー) サービスの再利用、または共通使用を可能にする、コンピューター手法、

50

システム、および装置を提供する。

【0012】

好ましい実施形態では、本発明のシステムは、アプリケーションインターフェースと、マッピングメンバーを含む。アプリケーションインターフェースは、複数のウェブサービスを介して、データを公開しているターゲット（クライアント）システムと相互作用する。アプリケーションインターフェースは、公開されたデータにクエリを行う。アプリケーションインターフェースは、2つ以上のアプリケーションにわたって、複数のウェブサービスのうちの少なくとも1つを共通して使用することにより、ターゲットシステムに関するアプリケーション間で、既存の（あらかじめ提供された）ウェブサービスを再利用する。

10

【0013】

この再利用を実現するために、アプリケーションインターフェースは、ターゲットシステムからウェブサービスデータの結合されたビューをもたらす方法で、ターゲットシステムのウェブサービスのうちの少なくとも2つへ呼を連鎖させる。少なくとも2つのウェブサービスは、ターゲットシステムによってあらかじめ提供されている。

【0014】

マッピングメンバーは、グローバルコンピューターネットワークサービスを介して、公開されるデータのアプリケーションインターフェースクエリ（a）と、対象アプリケーションにおいて使用可能なオブジェクトの構文（b）の間をマップする。マッピングメンバーは、対象アプリケーションを、アプリケーションインターフェースによって実装されるクエリに回答して生成されたオブジェクトインスタンスのデータにアクセスできるようにする。

20

【0015】

いくつかの実施形態では、ターゲットシステムは、協調プラットフォームまたは類似のプラットフォームに対して外部または外的なシステムであり、分散オブジェクトアーキテクチャクライアントとして構成されている。対象アプリケーションは、任意のコンピューターシステムとして実装される。また、XML（`extensible markup language`）クライアント、RMI（`remote method invocation`）クライアント、EJB（`enterprise JAVA`（登録商標）`Bean`）クライアント、およびJNI（`JAVA`（登録商標）`Native Interface`）クライアントのうちの任意のものとして実装される。

30

【0016】

一実施形態では、本発明は、外部システム（ターゲットクライアント）と統合するコラボレーションプラットフォームの能力を強化する。具体的には、実施形態によって、プラットフォームが、クエリデータを提供するウェブサービスを供給する外部システムとデータを共有することが可能になることから、プラットフォームのストレージ層を拡張するためのインターフェースが、本発明により恩恵を受ける。本発明は、宣言的な方法により、ウェブサービス呼び出し（複数のクエリ）のシーケンスを記述する機能を提供する。本発明の実装は、ウェブサービス呼び出しのシーケンスを実行して、データの統合されたビューを生成する。これにより、外的なデータを記述する単純化した方法を促進し、この外的なデータを、PLM（製品ライフサイクル管理）データモデルに容易にマッピングする方法を可能にする。PLMデータモデルは、（PLM、CV5、その他などの）プラットフォームアプリケーションによって、固有に使われる。

40

【0017】

したがって、本発明は、関係／従属関係に基づく「統一されたビュー」を見る一意のクエリ／マッピング構文を定義し、既存のウェブサービスを利用して外部データを連合する。本発明は、いかなる追加の「コード」も書くことなく、外部システムからデータを連合する問題を解決し、それにより、時間およびコストを節約する。本発明はまた、コラボレーションプラットフォームが、データ連合を必要とするさまざまなアプリケーションをホストするための機会を広げる。その目的のために、本発明は以下の問題に取り組む。

50

1. 複数のシステムにわたって製品材料を探す無駄な時間
2. 別々の企業システムが同期を取り続けるようにする複雑な統合
3. 別々のシステムで作業するユーザ間の非効率なコラボレーション
4. データ移行の複雑さによりレガシーシステムを退却させる困難性
5. データを他のシステムに再入力することによる作業の重複
6. 手作業のユーザ介入を使用する、外部サプライヤとの非効率なデータ共有

【0018】

したがって、本発明の実施形態は、インフラストラクチャレベルのソリューションを提供し、このインフラストラクチャの使用を意図する任意のコンシューマによって定義されなければならない一意のクエリ構文を定義する。

【図面の簡単な説明】

【0019】

前述の事項は、添付の図面において図示されるように、以下の本発明の例示的な実施形態のより詳細な説明から明らかになるであろう。添付の図面は、異なる図の全体にわたって同様の参照記号が同じ部品を指す。図面は必ずしも縮尺どおりではなく、むしろ、本発明の実施形態を例示することに重点を置いている。

【図1A】本発明の実施形態が配置されたコラボレーションプラットフォームの概略ブロック図である。

【図1B】本発明の実施形態が配置されたコラボレーションプラットフォームの概略ブロック図である。

【図1C】本発明の実施形態が配置されたコラボレーションプラットフォームの概略ブロック図である。

【図2】ターゲットシステムの複数のウェブ（グローバル・コンピューターネットワーク）サービスへの呼び出しを連鎖させて、ターゲットシステムからデータの結合されたビューをもたらす、本発明の実施形態の概略図である。

【図3】本発明の原理に従って、図1のプラットフォームスキーマにマップされている図2の結合されたビューの概略図である。

【図4】図2および図3の例におけるクエリ/マッピング構文を実装する、本発明のマッピングエンジンの概略図である。

【図5】本発明の一実施形態における処理およびデータ制御の流れ図である。

【図6】本発明の実施形態が動作するコンピューターネットワーク環境の概略図である。

【図7】本発明の実施形態が動作するコンピューターネットワーク環境のブロック図である。

【発明を実施するための形態】

【0020】

以下は、本発明の例示的な実施形態の説明である。

【0021】

本発明の実施形態を用いるコラボレーションプラットフォーム200が、図1Aおよび図1Bに図示されている。コラボレーションプラットフォーム200は、分散オブジェクトアーキテクチャとして構成されている。コラボレーションプラットフォーム200の好ましい実施形態は、さまざまなウェブブラウザ228（簡略化を目的に、図1Bに1つだけを図示）、ウェブサーバー238（明確化のため1つのみを図示）、アプリケーションサーバー236、およびストレージ層220（アプリケーションサーバーデータベース、対応するデータサーバーおよびファイルサーバーを含む）を含む。ウェブブラウザ228を介して、アプリケーション202（XML（Extensible Markup Language）クライアント204'として実装される）が実行される。ウェブブラウザ228は、ウェブサーバー238'に常駐するXMLサーバー206'と（ファイアウォール208を介して）通信する。ウェブサーバー238'はまた、RMI（Remote Method Invocation）クライアント210およびXML-RMIインターフェース222を含む。

10

20

30

40

50

【0022】

同様に、このアプリケーションおよび他のアプリケーション202は、RMIクライアント、EJB(Enterprise Java(登録商標) Bean)クライアント、JNI(Java(登録商標) Native Interface)クライアントとして、および他のウェブサーバーキテクチャ上のクライアントとして実装することができる。これらのクライアントは、図1Aの204で概略的に参照される。(概略的に参照されるように)対応するサーバー206は、ウェブサーバー238'の構成と同様に(すなわちそれぞれの該当するクライアント210およびインターフェース222と共に)構成される、それぞれのウェブサーバー238(図1Aに概略的に参照される)に常駐する。

【0023】

さらに図1Bに示すように、アプリケーションサーバー236は、RMIサーバー212、EJBクライアント214、EJBサーバー216に常駐する情報サービス218(永続性、メッセージング、ライフサイクル、ワークフロー、ポリシーマネージャ、アドミニストレーション、および構造管理)、およびRMI-EJBインターフェース224を含む。ウェブサーバー238'は、それ自身に常駐するXML-RMIインターフェース222を介して、アプリケーションサーバー236と通信する。ウェブサーバー238'は、RMI-EJBインターフェース224を介して、EJBサーバー216に常駐する情報サービス218とさらに通信する。これにより、上述した構成を使用して、アプリケーション202を使用するユーザは、XMLクライアント204'からウェブサーバー238'を介して、EJBサーバー216上の情報サービス218にアクセスすることができる。CORBAベースのアーキテクチャに複製されるサービス(認証、セキュリティ、トランザクション、ネーミング、およびデータベースアドミニストレーション)は、ウェブサーバー238'によって提供され、アプリケーションサーバー236と共有される。この構成は、「ウェブセントリック(webcentric)」と呼ばれ、アプリケーション202は、(例えば、3階層CORBAベースのウェブ対応アプリケーションとは対照的に)N階層である。アプリケーション202は、ウェブ上に配置され、ウェブサーバー238'またはアプリケーションサーバー236の、サービスおよびアドミニストレーション能力を十分に活かすことができる。

【0024】

情報サービス218には、ネットワーク内の任意の地点からアクセスが可能である。例えば、ファイアウォール208内部からRMIクライアント234上のアプリケーション232を実装するユーザは、RMIサーバー212からRMI-EJBインターフェース224を介して、EJBサーバー216上の情報サービス218にアクセスすることができる。同様に、ファイアウォール208内部からEJBクライアント242上のアプリケーション240を実装するユーザは、EJBサーバー216から直接、EJBサーバー216上の情報サービス218にアクセスすることができる。アプリケーション202、232、および240は、異なるサーバー上で実装される同じアプリケーションである。

【0025】

このように、プラットフォーム200は、(CORBAベースのアーキテクチャを有するような)一般的なウェブ対応の3階層アプリケーションに限定されず、譲受人により、参照によってその全体が本明細書に組み込まれる特許文献1に記述されているシステムコンパイラによって、N階層アプリケーションを有する。ウェブサーバー238およびアプリケーションサーバー236を構築する際、任意の組合せおよび任意の数のインターフェース(N階層)を互いに積み重ねることができる。

【0026】

好ましい実施形態では、コラボレーションプラットフォーム200は、グローバルな、企業規模の配置をサポートするための柔軟なPLM(製品ライフサイクル管理)環境を提供する。好ましくは、コラボレーションプラットフォーム200は、製品構想から製品経験までの情報および知識に関連した、すべての製品のアクセスおよび管理を可能にする、さまざまな非ネイティブおよびネイティブなアプリケーション202、232、240の

10

20

30

40

50

ための単一で開かれた環境／プラットフォームである。プラットフォーム200は、すべてのエンジニアリング専門分野、およびすべてのPLM企業ビジネスプロセスとは言わないまでも、その多くにわたる。

【0027】

コラボレーションプラットフォーム200は、ビジネスプロセス、ユーザインターフェース、およびインフラストラクチャオプションを容易に構成するための柔軟性を提供して、できる限り低いトータルの維持費で、現在および将来において、それらが組織のニーズを満たすことを確実にする。その目的のために、プラットフォーム200のインフラストラクチャサービスの1つが、図1Aに示すアダプレット(Adaplet)120と呼ばれるアプリケーション適合インターフェースである。アダプレット120によって、アプリケーションサーバー236は、プラットフォーム200にネイティブではないデータソースと通信し、このデータをあたかもコラボレーションプラットフォーム200にネイティブであるかのように表すことが可能になる。

10

【0028】

本発明は、図1C～図5を参照して次に説明するように、プラットフォーム200のアダプレット120インフラストラクチャを強化する。

【0029】

アダプレットAPIは、外的な(外部の)または非ネイティブなシステム300(例えばCRM(Customer Relationship Management)、ERP(Enterprise Resource Planning)、または他の任意の企業アプリケーション)から、ビジネスエンティティ/オブジェクトを「適合させる」ことによって、コラボレーションプラットフォーム200のストレージ管理層220(図1A)の拡張を可能にする。アダプレットインターフェース/APIの具体的な実装(インスタンス)は、アダプレット120と呼ばれ、ビジネスオブジェクト/エンティティ121、およびそのリレーションシップを収集し、それらがコラボレーションプラットフォーム200(モデリングクエリ言語-MQL層、およびアプリケーション開発キット-ADK層)によって活用することができる。

20

【0030】

各アダプレット120は、ターゲット(外部)システム300に固有であり、プラットフォームネイティブアプリケーション202、232、240が活用することができる別個のヴォールト(Vault)125(図1A)として現れる。アダプレットヴォールト125に発行されたクエリは、非アダプレットヴォールト126から見分けがつかない方法で、ビジネスオブジェクト121およびリレーションシップを戻す。

30

【0031】

アダプレット120は、プラットフォーム200と他のシステム300との間でリアルタイムな相互運用を可能にし、他のシステム300の混乱を起こすことなく、ITにわずかな負荷しかかけずに、複数のソースから統合された情報を提供する。アダプレット120を用いることで、プラットフォームユーザは、他のシステム300に記憶されている情報に、あたかもそれが実際にプラットフォーム200の一部であるかのようにアクセスできる。外部から記憶された情報は、ネイティブプラットフォームデータから見分けがつかないようにされる。ネイティブプラットフォーム情報上で実行可能なオペレーションは、ターゲットシステム300から外部情報上でもまた実行可能である。

40

【0032】

アダプレット120同士は、互いに類似した性能を有する。

(a) アダプレットは、統合されたトランザクションマネジメントと双方向である。ターゲットシステム300は、二相コミットを実装してもよく、または一相データソースとしても扱われることができる。

(b) アダプレットは、ビジネスオブジェクト121を公開し、適合した/外的なシステム300のビジネスオブジェクト121同士の接続を公開する。

(c) アダプレットは、スキーマを拡張し、複合オブジェクト121を構築する機能を有

50

する。いくつかのオブジェクト 121 の属性は、コラボレーションプラットフォーム 200 に常駐し、他は適合した / 外的なシステム 300 に常駐することができる。結合されたオブジェクト 121 は、コラボレーションプラットフォーム 200 を介して、単一のオブジェクトとして現れる。

(d) アダプレットは、異なる連合体においてオブジェクト同士を接続する機能を有する。プラットフォーム 200 リレーションシップは、ネイティブビジネスオブジェクトを、適合した / 外的なヴォールト 125 のオブジェクト 121 に接続する、または、別の適合した / 外的なヴォールト 125 のオブジェクト 121 に接続するのに使用することができる。

【0033】

本発明より以前において、アダプレットインターフェースは、以下のように、2つの異なるアーキテクチャを使用して構成することができた。1つのオプションは、外部データソースへの直接データベース接続に基づいていた。アダプレットはまた、ウェブサービス API 150 を使用して実装することができたが、以下ではそれを WSA (ウェブサービスアダプレット / Web Service Adaplet) と呼ぶ。

【0034】

一実施形態では、ウェブサービスアダプレットは、ウェブサービス 150 を介して (ターゲットシステム 300 の) 外的な / 外部データベースと対話する Java (登録商標) アダプレットである。コラボレーションプラットフォーム 200 は、ウェブサービス 150 のクライアントサイドを提供する。プラットフォームクライアント 200 と、適合したシステム 300 との間の手shakeが、プラットフォーム 200 が提供する WSDL ファイル 130 によって定義される。WSDL ファイル 130 は、ウェブサービス 150 が理解するメッセージ、それらのメッセージに対する応答のフォーマット、サービス 150 がサポートするプロトコルなどを記述する。

【0035】

顧客は、適切な Java (登録商標) ツールまたは C シャープツール 110 と一緒にこの WSDL ファイル 130 を使用して、スタブおよびスケルトンファイル 111 を生成することができる。スタブおよびスケルトンファイル 111 は、ウェブサービス 150 を動かすためにランタイムで使用される。

【0036】

WSDL ツール 110 により作成されたファイル 111 のうちの 1つがスケルトンであり、スケルトンのパブリック関数は、クライアントプラットフォーム 200 によって行われた呼び出しへの返答を提供するコードで書き込まなければならない。このスケルトン 111 の関数の多くが、適合したシステム 300 からのデータを入手、または設定する。他のいくつかは、トランザクション制御に従事する。

【0037】

WSA アダプレットより戻されたオブジェクト / リレーションシップ 121 は、物理レベルおよび論理レベルの両方において、プラットフォーム 200 内で構造オブジェクト / リレーションシップに合致しなければならない。ターゲットシステム 300 からの応答データが、統合されたマッピングエンジンによって、プラットフォーム 200 のビジネスオブジェクト / リレーションシップ 121 にマップされる。その目的のために、WSA アダプレットは、適合したシステム 300 が公開するように要求する、「Types」「Attributes」「Relationships」などを記述するスキーマファイル 135 を (統合されたマッピングエンジンを介して) 使用する。

【0038】

外部システム 300 からビジネスオブジェクト / データ 121 を抽出するためには、WSA アダプレットを構成する必要があり、各ターゲットシステム 300 は、アダプレット API によって公開されたインターフェースに準拠する特定のそれぞれのウェブサービス 150 を実装 (すなわち、WSDL 130 で定義され、スケルトンファイル 111 でコード化されたオペレーション / メソッドを実装) しなければならない。実際には、これは、

10

20

30

40

50

コラボレーションプラットフォーム 200 を外的な / 外部システム 300 に橋渡しするために、統合が、事実上、中間層（ターゲットシステム 300 固有の WSA アダプレット）が作成されるのを必要とすることを意味する。これは、統合をより複雑でコストのかかるものにする。

【0039】

さらに、WSA（ウェブサービスアダプレット）は、外的なシステム 300 の既存のウェブサービス 150 を活かして、外部データを連合することができない。というのは、これらのウェブサービス 150 は、WSA アダプレットによって定義されたインターフェースに準拠していないからである。

【0040】

最後に、これらのウェブサービスアダプレットの限界として、ウェブサービスアダプレットは、データを抽出している間に連合データの関係または誘導可能性を考慮しない。

【0041】

本発明は、アダプレットインフラストラクチャを強化して、これに改良を加える。本発明により取られるアプローチは、上述されたものとは根本的に異なり、クエリ構文を定義しながら、連合データの関係およびその誘導可能性を見るものである。本発明は、以下にさらに説明するように、ウェブサービス呼び出しの連鎖を使用して、そこからオブジェクト 121 が生成される複合「ドキュメントオブジェクトモデル」を形成する。

【0042】

したがって、本発明は、先行 WSA（ウェブサービスアダプレット）の使用をより容易にする。出願人は、本発明が、WSA 技術の採用を大いに増加させると考える。また、本発明の実施形態により、コラボレーションプラットフォーム 200 は、非構造化データ（患者記録、その他）などの、従来とは異なるデータの中央レポジトリとなることができる。

【0043】

コラボレーションプラットフォーム 200 に対する本発明のいくつかの効果は以下である。

(1) 企業間、さらにビジネスパートナー間で、データを交換し、プロセスを自動化する機能を提供する。

(2) ユーザが複数のシステムからの情報によって作業できるようにすることにより、グローバルなビジネスコラボレーションを奨励する。この情報の全体論的観点、よりよい意思決定支援システム、低コストへの移行、およびより早い製品開発サイクルを可能にすることになる。

(3) 他のデータベースからのデータの段階的な結合および移動を提供する。

(4) 異なるシステムに常駐する製品情報を、それがあたかもプラットフォームデータベース / ストレージ層 220 に記憶されているかのように表示することを可能にする。

(5) 複数のデータソースをリンクする際の、開発の手間および複雑なコーディングを軽減する。

【0044】

先行ウェブサービスアダプレットと比較して、本発明の主要な効果は、外的なシステムまたはターゲットシステム 300 での既存のソフトウェア資産（既存のウェブサービス 150）の再利用である。本発明はまた、外部システム 300 からデータを連合するための、いかなる追加の「コーディング」も必要としない。

【0045】

図 1C ~ 図 4 に戻って、次に、本発明の実施形態について詳細に説明する。本発明のアダプレット 120 アーキテクチャおよび実装によって、ターゲットシステム 300 上の既存のウェブサービス 150 の利用が可能になる。この強化に加えて、本発明は、単一のルートから任意のドキュメント / サービス（応答）の任意のノードを識別することができる「一意の」クエリ / マッピング構文 25 を定義する。

【0046】

10

20

30

40

50

図2に示すように、ターゲットシステム300は、「getStandardInfo」および「getMaterialInfo」について、別々のウェブサービス150a、bを公開する。「getStandardInfo」サービス呼び出し150aからの応答は、「getMaterialInfo」ウェブサービス150bに必要な入力（「materialid」）を含む。本発明のアダプテッド120は、呼び出しを論理的に順番に追加する、または連鎖させることによって、これら2つのサービス呼び出し150a、150bの結合したビュー32を形成する。結合したビュー32は、単一の論理ドキュメントのように見ることができる。

【0047】

結合されたビュー32から、マッピングエンジン35が、プラットフォームスキーマ135へのマッピングを単純化する（図3）。マップされたビュー34が結果物となる。これは、以下のように実現される。パス言語が、単一の「ルート」から任意のドキュメント（またはサービス150）の任意のノードを識別することができる。次に、マッピングエンジン35が、図4に示すように、クエリ/マッピング構文25を実装する。

【0048】

マッピング25は、各ノードがサービス150を呼び出すことができる簡単なパスを示す。パス定義はまた、「マクロ」構文を利用して、URLに似た方法で引き数またはパラメータを渡す。マッピング25は、ルートコンテキスト21および導出されたコンテキスト23のような、オブジェクトコンテキストに基づく事前定義されたマクロを使用する。例えば、ルートコンテキスト21として、\$（名前）が、図4に示す例の中で用いられている。

【0049】

マッピング25はまた、ドキュメント/応答メッセージ内容に基づく動的なマクロを使用する。これは、図示された例の中で、導出されたコンテキスト23：\$（materialid）として示されている。

【0050】

図1Cおよび図5を参照すると、例示的な、限定しない一実施形態における本発明のアダプテッド120の実装およびオペレーションは、以下の通りである。アダプテッド120は、ターゲットシステム300の既存の（あらかじめ定義された/提供された）ウェブサービス150を介してターゲットシステムデータベースと対話するJAV A（登録商標）コンポーネントとして実装される。図1Bにおいて上で説明したように、プラットフォーム200は、JAV A（登録商標）をはじめ、他の（複数の）コンピューターキテクチャをサポートすることができる。したがって、アダプテッド120の実装は、JAV A（登録商標）に限定されない。コラボレーションプラットフォーム200は、ウェブサービス150のクライアントサイドを提供する。アダプテッド120インターフェースが、ターゲットシステム300によって公開されるウェブサービス150のために、WSDLファイル130を取得する（図5のステップ501）。WSAアダプテッドの場合において前に述べたように、クライアントプラットフォーム200と、ターゲットシステム300との間のハンドシェイクは、WSDLファイル130によって定義され、WSDLファイル130は、それぞれのウェブサービス150が理解するメッセージ、それらのメッセージに対する応答のフォーマット、サービスがサポートするプロトコルなどを記述する。

【0051】

ユーザは、適切なJAV A（登録商標）ツール110と一緒にWSDLファイル130を使用して、スタブおよびスケルトンファイル111を生成する。スタブおよびクライアントクラスは、スキーマファイル135（例えば、Schema.xml）に記述されているように複雑なビジネスオブジェクトを組み立てるために、プロバイダクラス140によってランタイムに使用される。ユーザは、先行WSAアダプテッドでのように、ターゲットシステム300によって実装されるべきメソッドを用いてスケルトンファイル111に書き込みをする必要はない。

【0052】

10

20

30

40

50

次に、アダプレットインターフェースがパラメータファイル133を記述し(図5、ステップ503)、これが構成設定を行うためのメカニズムとなる。ステップ503はまた、プロバイダクラス140、およびターゲットシステム300により公開されるウェブサービス150a、b、・・・nへのエンドポイントを定義する。アダプレット120インターフェースは、XMLファイルを、アダプレットヴォールト125の記述と一緒にデータベース123に記憶する。このXMLファイルが構文解析され、一部は「Add Vault」コマンド(後述)への入力として使用され、一部はスキーママップとしてアダプレットプロバイダのinit()エントリポイントの中に渡される。

【0053】

ステップ505で、スキーマファイル135を記述する。内容、すなわちスキーママップは、クライアントプラットフォーム200を介して、プロバイダのinit()呼び出しに回答して戻される。一実施形態では、スキーママッピングファイル135は、XMLファイルである。アダプレット120によって戻されたオブジェクト/リレーションシップ121は、コラボレーションプラットフォーム200内で構造オブジェクト/リレーションシップに合致しなければならない。マッピングは、アダプレット120実装の任務である。リレーショナルアダプレット120は、リレーショナルデータをビジネスオブジェクトに変換するマッピングエンジン35を用いて、これを実現する。

【0054】

ステップ507で、プロバイダクラス140は、インターフェース(例えば、一実施形態におけるffプロバイダインターフェース)を実装する。これは、既存の対象ウェブサービス150インターフェースを使用して、クエリインターフェースを実装するために、プロバイダクラス140によって提供される。一実施形態では、図4のクエリ/マッピング構文25が、プロバイダクラス140の中に構築される。しかしながら、他の実施形態では、クエリ/マッピング構文25は、プロバイダクラス140の外側であって、リレーションシップを誘導し、既存のウェブサービス150からの結果を結合してもよい。

【0055】

プロバイダクラス140生成インターフェースは、先行WSAアダプレットのアプローチおよび以下のユーザの要件、(i)ターゲットシステムによって実装されるメソッドを用いてスケルトンファイル111に書き込むこと、(ii)各ターゲットシステムにおいてウェブサービス/スケルトンクラスメソッドを実装すること、に取って替わることに留意されたい。

【0056】

ステップ509で、アダプレット120インターフェースは、MQLコマンド: Add Vault [vaultname] external file [filename]を実行する。ここで、[filename]が、ステップ503で記憶されたパラメータファイル133のXMLファイルから読み出される。これにより、クライアントプラットフォーム200/アダプレット120は、ターゲットシステム300の、あらかじめ存在した/提供されたウェブサービス150呼び出しの1つまたは複数を利用することによって、複雑なビジネスオブジェクトについてターゲットシステムデータに対するクエリを行う。詳細には、図2においてあらかじめ説明したように、アダプレット120は、ウェブサービス呼び出しを連鎖させる。

【0057】

次に、プロバイダクラス140は、スキーマファイル135のスキーママップに従って、対象の複雑なビジネスオブジェクト121を組み立てる(構築する)。詳細には、プロバイダクラス140は、呼び出された既存のウェブサービス150からの結果を結合し、マッピングエンジン35は、(図3および図4において上で詳述したように)ウェブサービス呼び出し構文からプラットフォーム200スキーマにマップする。

【0058】

本発明のアダプレット120は、複数のウェブサービス150応答メッセージを連鎖させ、結合されたビュー32およびマップされたビュー34(ドキュメントオブジェクトモ

10

20

30

40

50

デル)を利用して、適切かつ「一意の」マッピング25を定義するその能力においてユニークであり、それにより、外部システム300からオブジェクト121を連合する。

【0059】

可能性のある使用機会において、本発明のアダプレット120は、いかなる追加のコーディング/クラスメソッドのターゲットシステム実装なしに、既存のウェブサービス150の再利用を促進し、場合によっては、複数ドメインにわたり構文マッピング25を連結することができる。本発明のアダプレット120は、ターゲットシステム300がプラットフォーム200定義のサービスを実装することを必要とする、既存のウェブサービスアダプレット実装に改良を加える。

【0060】

本発明のアダプレット120の別の可能性のある適用は、ドキュメント/サービス150が内在する構造、従属関係を有し、誘導を必要とする場合の問題に対してである。一例は、RTF(Rich Text Format)/xmlフラグメントを、単一アセンブリに統合する要求管理(Requirements Management)においてである。別の例は、「元の場所に」残され、専用のウェブサービスを介して参照されるドキュメントである。80%の企業知識は、リレーショナルデータベースではなく、ファイルサーバーの中に放置されている。適所にあるドキュメントにアクセスすることによって、ユーザは、人の仕事の方法を変更することなく、この情報を活かすことができる。

【0061】

既存のアダプタおよび連合ツールキットが、ビジネスオブジェクトの2地点間統合を行う。それらは、関係/従属関係を有するオブジェクト/データについて、システム間のマッピングを定義する必要性には取り組まない。

【0062】

図6は、本発明を実装することができるコンピューターネットワーク、または同様のデジタル処理環境を図示する。

【0063】

クライアントコンピューター/デバイス50(例えばアプリケーション実行クライアント204)、およびサーバーコンピューター60(例えばウェブサーバー238、およびプラットフォーム200のアプリケーションサーバー236)は、アプリケーションプログラムなどを実行する処理デバイス、ストレージデバイス、および入力/出力デバイスを提供する。クライアントコンピューター/デバイス50はまた、通信ネットワーク70を介して、他のクライアントデバイス/プロセッサ50として実装されるターゲットシステム300、およびサーバーコンピューター60などの、他のコンピューティングデバイスにリンクされてよい。通信ネットワーク70は、リモートアクセスネットワーク、グローバルネットワーク(例えばインターネット)、コンピューターのワールドワイドな集合形態、ローカルエリアネットワークまたはワイドエリアネットワーク、および一般にそれぞれのプロトコル(TCP/IP、Bluetoothなど)を使用して互いに通信するゲートウェイの一部であってよい。他の電子デバイス/コンピューターネットワークアーキテクチャも適している。

【0064】

図7は、図6のコンピューターシステムにおけるコンピューター(例えばクライアントプロセッサ/デバイス50、またはサーバーコンピューター60)の内部構造の図である。各コンピューター50、60は、システムバス79を含み、ここでバスは、コンピューターまたは処理システムのコンポーネントの間でデータを転送するために使用される一組のハードウェアラインである。バス79は、基本的に、コンピューターシステムの異なる要素(例えば、プロセッサ、ディスクストレージ、メモリ、入力/出力ポート、ネットワークポート、その他)を接続する共用の電線であり、バスによって要素間で情報の転送が可能になる。システムバス79には、さまざまな入力および出力デバイス(例えば、キーボード、マウス、ディスプレイ、プリンタ、スピーカ、その他)を、コンピューター50、60に接続するためのI/Oデバイスインターフェース82が取り付けられている。ネ

10

20

30

40

50

ットワークインターフェース 86 によって、コンピュータは、ネットワーク（例えば図 6 のネットワーク 70）に取り付けられた、さまざまな他のデバイスに接続することが可能になる。メモリ 90 は、本発明の実施形態（例えば、アダプレット 120、マッピングエンジン 35、ならびにクエリ/マッピング構文 25 および詳細に上述された他のコードのサポート）を実装するのに使用されるコンピュータソフトウェア命令 92 およびデータ 94 に、揮発性ストレージを提供する。ディスクストレージ 95 は、本発明の実施形態を実施するのに使用されるコンピュータソフトウェア命令 92 およびデータ 94 に、不揮発性ストレージを提供する。中央処理装置 84 がまた、システムバス 79 に取り付けられ、コンピュータ命令の実行に備える。

【0065】

10

一実施形態では、プロセッサルーチン 92 およびデータ 94 は、コンピュータプログラム製品（総称して参照 92）であり、これは、本発明のシステムのためのソフトウェア命令の少なくとも一部を提供するコンピュータ可読媒体（例えば、1つまたは複数の DVD-ROM、CD-ROM、ディスク、テープ、その他などの着脱可能なストレージ媒体）を含む。コンピュータプログラム製品 92 は、当技術分野でよく知られているように、任意の好適なソフトウェアインストール手続きによって、インストールすることができる。別の実施形態では、ソフトウェア命令の少なくとも一部をまた、ケーブル、通信接続および/またはワイヤレス接続を通してダウンロードすることができる。他の実施形態では、本発明のプログラムは、伝播媒体（例えば、電波、赤外線波、レーザー波、音波、またはインターネットなどのグローバルネットワークもしくは他のネットワーク上で伝播される電気波）上の伝播信号で具体化されるコンピュータプログラム伝播信号製品 107 である。そのようなキャリア媒体または信号は、本発明のルーチン/プログラム 92 に、ソフトウェア命令の少なくとも一部を提供する。

20

【0066】

代替実施形態では、伝播信号は、アナログキャリア波、または伝播媒体上で搬送されるデジタル信号である。例えば、伝播信号は、グローバルネットワーク（例えばインターネット）、電気通信ネットワーク、または他のネットワーク上で伝播されるデジタル化された信号であってよい。一実施形態では、伝播信号は、所定期間にわたって伝播媒体上に伝送される信号であり、例えば、ミリ秒、秒、分、またはより長い期間にわたってネットワーク上のパケットで送信されるソフトウェア・アプリケーション用の命令である。別の実施形態では、コンピュータプログラム製品 92 のコンピュータ可読媒体は、コンピュータプログラム伝播信号製品について上述したように、例えば伝播媒体を受信し、伝播媒体において具体化された伝播信号を識別することによって、コンピュータシステム 50 が受信し、読み取ることができる伝播媒体である。

30

【0067】

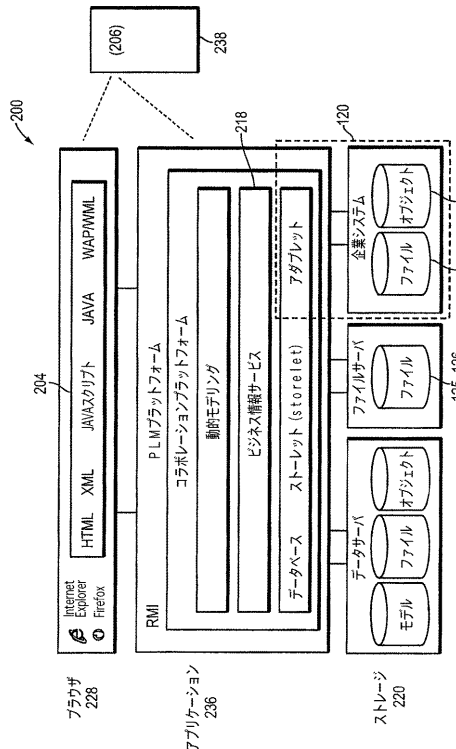
一般的に言えば、用語「キャリア媒体」すなわち遷移キャリアは、前述の遷移信号、伝播信号、伝播媒体、ストレージ媒体などを包含する。

【0068】

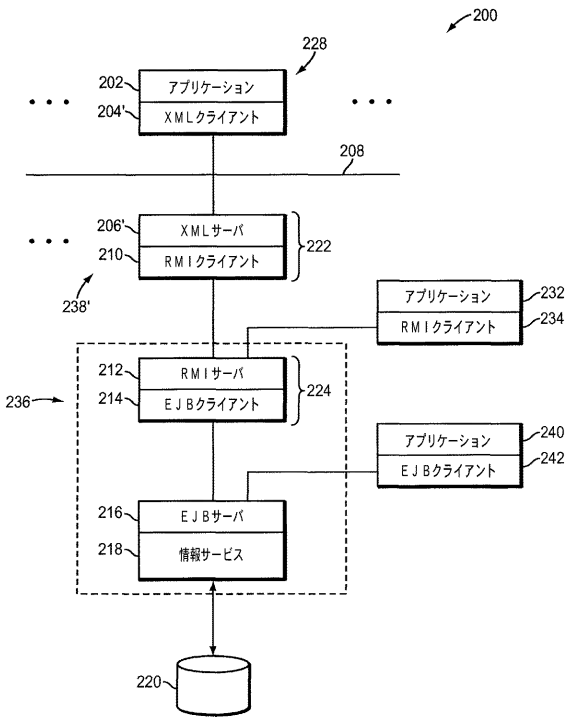
本発明を、その例示的な実施形態を参照して詳細に示し、説明してきたが、当業者であれば、添付の特許請求の範囲によって包含される本発明の範囲から逸脱することなく、本発明において、形態および詳細にさまざまな変更を行うことができることを理解するであろう。

40

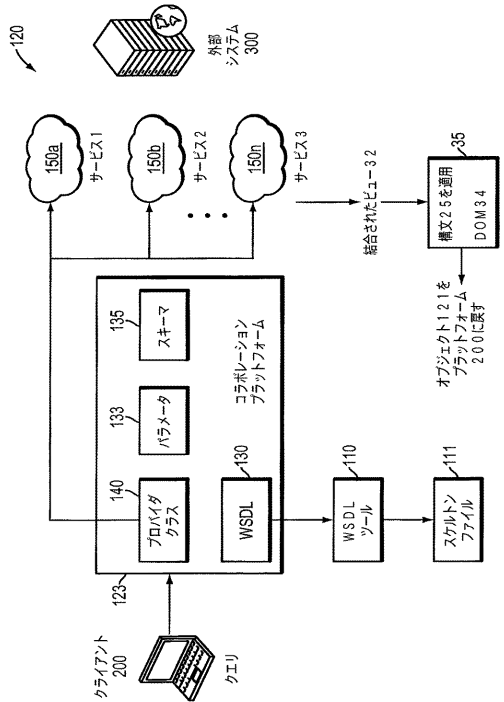
【図 1 A】



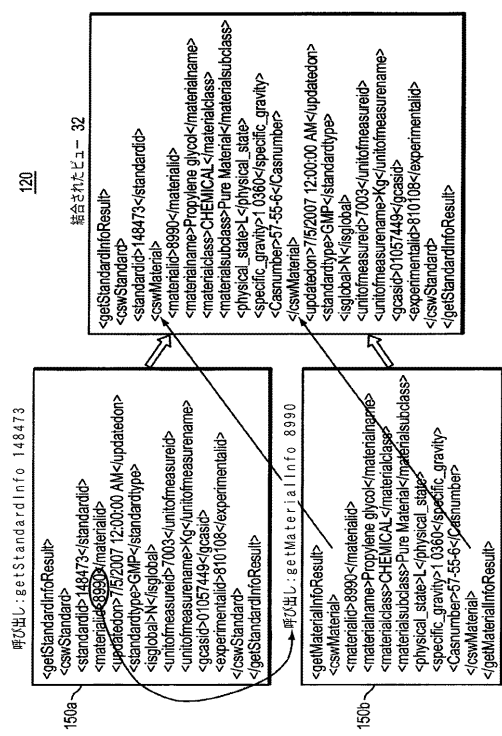
【図 1 B】



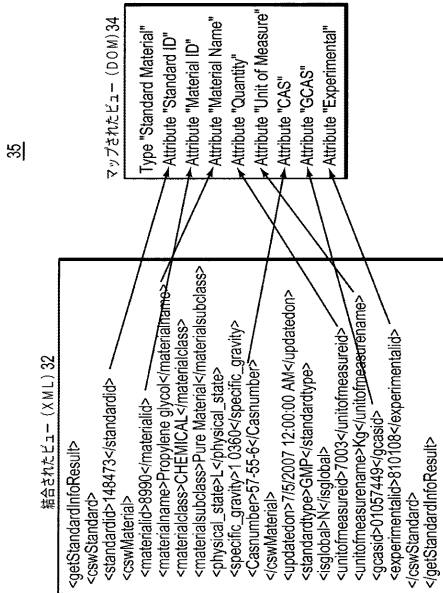
【図 1 C】



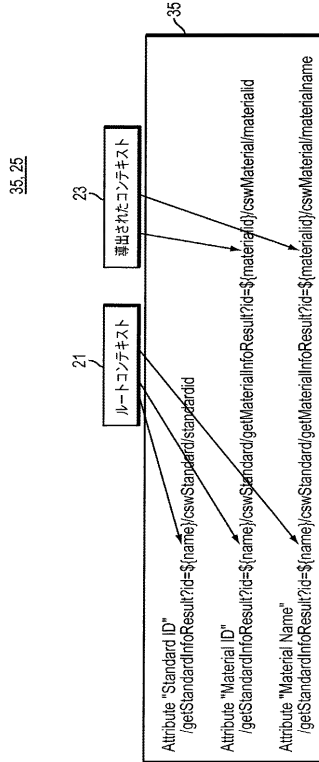
【図 2】



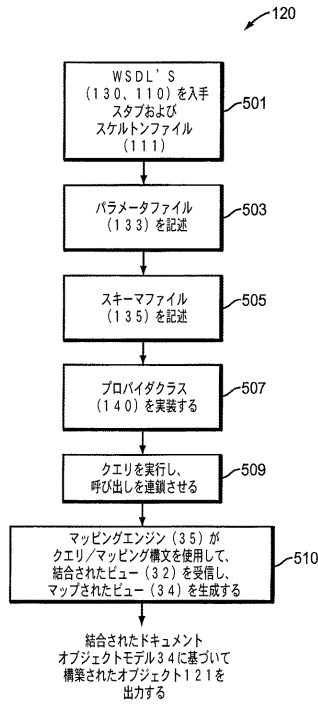
【 図 3 】



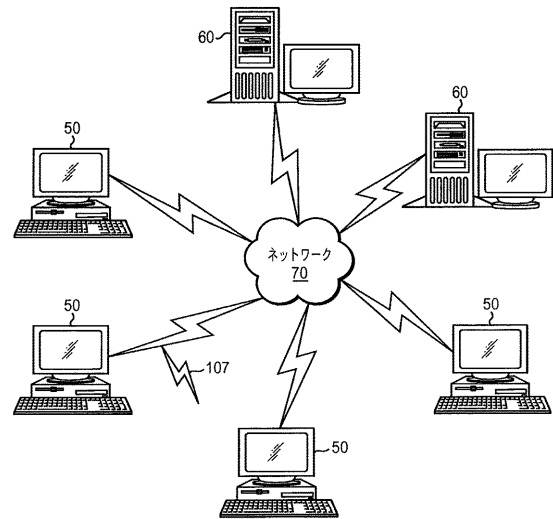
【 図 4 】



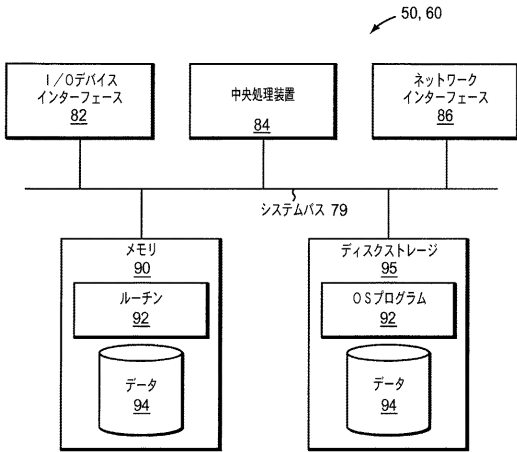
【 図 5 】



【 図 6 】



【 図 7 】



フロントページの続き

(72)発明者 アナップ チャンドラン
イギリス アールジー 3 1 5 ダブリュエフ リーディング タイルハースト ラモーナ クレセント 2

(72)発明者 ボリス タベンキン
アメリカ合衆国 0 1 8 3 3 マサチューセッツ州 ジョージタウン ロングビュー ウェイ 4
Fターム(参考) 5E501 AA01 AB15 AC23 EA12 FA13 FB25

【外国語明細書】

2011159275000001.pdf