

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6698646号
(P6698646)

(45) 発行日 令和2年5月27日 (2020.5.27)

(24) 登録日 令和2年5月1日 (2020.5.1)

(51) Int. Cl.

F I

G 0 6 F 9/448 (2018.01)

G 0 6 F 9/448

G 0 6 F 16/17 (2019.01)

G 0 6 F 16/17 1 0 0

G 0 6 F 40/154 (2020.01)

G 0 6 F 17/22 6 7 0

請求項の数 13 (全 40 頁)

(21) 出願番号 特願2017-521556 (P2017-521556)
 (86) (22) 出願日 平成27年10月30日 (2015.10.30)
 (65) 公表番号 特表2017-535854 (P2017-535854A)
 (43) 公表日 平成29年11月30日 (2017.11.30)
 (86) 国際出願番号 PCT/US2015/058374
 (87) 国際公開番号 W02016/070081
 (87) 国際公開日 平成28年5月6日 (2016.5.6)
 審査請求日 平成30年10月25日 (2018.10.25)
 (31) 優先権主張番号 14/529,546
 (32) 優先日 平成26年10月31日 (2014.10.31)
 (33) 優先権主張国・地域又は機関
 米国 (US)

(73) 特許権者 502303739
 オラクル・インターナショナル・コーポレ
 イション
 アメリカ合衆国カリフォルニア州9406
 5レッドウッド・シティ、オラクル・パ
 ークウェイ500
 (74) 代理人 110001195
 特許業務法人深見特許事務所
 (72) 発明者 マシュー、アレックス
 インド、600077 タミル・ナードゥ
 、チェンナイ、アヤパッカム、ティルパー
 カドゥ・ロード、アンビガイ・ナガー、ブ
 イ・ジー・エヌ・プラチナ、88-89、
 エス・1

最終頁に続く

(54) 【発明の名称】 J S O Nスタイルシート言語変換

(57) 【特許請求の範囲】

【請求項 1】

J S O N データを変換するための方法であって、前記方法は、

JavaScriptオブジェクト表記法を使用して特定された第1のオブジェクトをコンピュ
 タで受信するステップを含み、前記第1のオブジェクトは一組の1つ以上の属性を有し、
 各属性は予め定められたJ S O N データタイプのもので、少なくとも1つの値を有し、前
 記方法はさらに、

JavaScriptオブジェクト表記法を使用して特定された第2のオブジェクトを前記コンピ
 ュータで受信するステップを含み、前記第2のオブジェクトは一組の1つ以上の属性を有
 し、各属性は前記第1のオブジェクトの前記一組の属性のうちの少なくとも1つの属性に
 対応しており、1つ以上の変換を定義する少なくとも1つの値を有し、前記方法はさらに
 、

前記第1のオブジェクトを、前記第2のオブジェクトに基づいて、JavaScriptオブジェ
 クト表記法を使用して特定された第3のオブジェクトに、前記コンピュータによって変換
 するステップと、

前記コンピュータに関連付けられた記憶装置に前記第3のオブジェクトを格納するステ
 ップと、

前記第2のオブジェクトが、前記第1のオブジェクトの各属性について、当該属性の対
 応する変換を定義する前記少なくとも1つの値を有するか否かを判断するステップと、

前記判断に応じて、前記第3のオブジェクトにおいて前記第1のオブジェクトの各属性

10

20

を含めるか除外するかを判断するステップを、含む、方法。

【請求項 2】

前記第 2 のオブジェクトを前記コンピュータで受信するステップは、名前が前記第 1 のオブジェクトの属性の名前に対応している属性を有する、JavaScript オブジェクト表記法を使用して特定されたオブジェクトを受信するステップを含む、請求項 1 に記載の方法。

【請求項 3】

名前が前記第 1 のオブジェクトの属性の名前に対応している属性を有する前記オブジェクトを受信するステップは、前記第 1 のオブジェクトの属性と同じ名前を有するとして属性を受信するステップを含む、請求項 2 に記載の方法。

【請求項 4】

名前が前記第 1 のオブジェクトの属性の名前に対応している属性を有する前記オブジェクトを受信するステップは、前記第 1 のオブジェクトの属性の名前を参照する値を有するとして属性を受信するステップを含む、請求項 2 に記載の方法。

【請求項 5】

前記第 2 のオブジェクトを前記コンピュータで受信するステップは、前記第 1 のオブジェクトの属性の名前または前記第 1 のオブジェクトの属性の値のいずれかに適用されるべき演算を特定する値を有する、JavaScript オブジェクト表記法を使用して特定されたオブジェクトの属性を受信するステップを含む、請求項 1 ~ 4 のいずれかに記載の方法。

【請求項 6】

前記第 2 のオブジェクトを前記コンピュータで受信するステップは、前記第 1 のオブジェクトの属性の変換に対応する前記第 3 のオブジェクトの属性のデータタイプを特定する値を有する、JavaScript オブジェクト表記法を使用して特定されたオブジェクトの属性を受信するステップを含む、請求項 1 ~ 5 のいずれかに記載の方法。

【請求項 7】

前記第 2 のオブジェクトを前記コンピュータで受信するステップは、前記第 1 のオブジェクトの 1 つ以上の属性の修正を特定する値を有する、JavaScript オブジェクト表記法を使用して特定されたオブジェクトの属性を受信するステップを含む、請求項 1 ~ 6 のいずれかに記載の方法。

【請求項 8】

前記第 2 のオブジェクトを前記コンピュータで受信するステップは、前記 1 つ以上の変換が前記第 1 のオブジェクトに適用可能かどうかを定義する 1 つ以上のテストを定義する値を有する、JavaScript オブジェクト表記法を使用して特定されたオブジェクトの属性を受信するステップを含む、請求項 1 ~ 7 のいずれかに記載の方法。

【請求項 9】

前記 1 つ以上のテストは、前記第 1 のオブジェクトの 1 つ以上の属性に対して定義される、請求項 1 ~ 8 のいずれか 1 項に記載の方法。

【請求項 10】

前記 1 つ以上のテストの肯定的評価を前記コンピュータによって判断するステップと、前記 1 つ以上のテストの肯定的評価に基づいて、前記 1 つ以上の変換を、前記コンピュータによって行なうステップとをさらに含む、請求項 1 ~ 8 のいずれか 1 項に記載の方法。

【請求項 11】

前記 1 つ以上の変換を前記コンピュータによって行なうステップは、前記第 1 のオブジェクトの属性の名前を変更するステップ、前記第 1 のオブジェクトの属性の値に対して 1 つ以上の数値演算を行なうステップ、前記第 1 のオブジェクトの属性の値に対して 1 つ以上の文字列演算を行なうステップ、オブジェクトでもある前記第 1 のオブジェクトの属性に 1 つ以上の属性を追加するステップ、または、オブジェクトでもある前記第 1 のオブジェクトの属性から 1 つ以上の属性を除去するステップを含む、請求項 10 に記載の方法。

【請求項 12】

請求項 1 ~ 11 のいずれかに記載の方法をコンピュータに実行させる、プログラム。

10

20

30

40

50

【請求項 13】

請求項 12 に記載のプログラムを格納したメモリと、
前記メモリに接続され、前記プログラムを実行するためのプロセッサとを備える、システム。

【発明の詳細な説明】

【背景技術】

【0001】

発明の背景

JSONとは、属性と値とのペアからなるデータオブジェクトを送信するために人間が読み取れるテキストを使用するオープン標準フォーマットである。それは主として、サーバとウェブアプリケーションとの間でデータを送信するために、XMLの代わりとして使用される。JSONは元々、JavaScript（登録商標）スクリプト言語から派生したものであるが、言語非依存のデータフォーマットである。JSONデータを構文解析し、生成するためのコードが、幅広い種類のプログラミング言語で容易に利用可能である。

10

【0002】

現在、企業レベルでは、JSONデータフォーマットは、他のウェブベースのデータフォーマットに比べてより柔軟で使いやすいため、多くの注目を獲得してきている。それは、モバイル世界において非常に好ましいデータフォーマットである。JSONは、データフォーマットが一般に使用される現実世界の状況において作成、読取りおよび復号のための広範なサポートを有する他のデータフォーマットの低オーバーヘッド代替物として奨励される。XMLの他に、他の例は、OGDL、YAMLおよびCSVを含んでいてもよい。

20

【発明の概要】

【発明が解決しようとする課題】

【0003】

JSONデータを変換するために、多くの変換APIが市場で入手可能である。しかしながら、これらのAPIは、ユーザが変換を定義するために別の言語を学ぶことを要求する場合があるため、ユーザフレンドリーではない。ユーザは結局、そうするために自分自身の文法を書くことになることが多い。いくつかのAPIは、JSONをXMLに変換し、変換を適用し、次にJSONに逆変換する。これは、大量のオーバーヘッド処理をもたらす。

30

【0004】

したがって、望まれるのは、JSON変換に関する問題を解決することであり、それらのうちのいくつかをここに説明する。加えて、望まれるのは、JSON変換言語に関する欠点を減らすことであり、それらのうちのいくつかをここに説明する。

【課題を解決するための手段】

【0005】

発明の簡単な概要

この開示の以下の部分は、この開示において見出された1つ以上の革新、実施形態、および/または例の簡略化された概要を、少なくとも主題の基本的理解を提供するという目的のために提示する。この概要は、任意の特定の実施形態または例の広範な概略を提供することを試みていない。加えて、この概要は、実施形態または例の主要/重要要素を識別するよう、またはこの開示の主題の範囲を定めるよう意図されてはいない。したがって、この概要の1つの目的は、この開示において見出されたいくつかの革新、実施形態、および/または例を、後に提示されるより詳細な説明の序章として、簡略化された形で提示することであり得る。

40

【0006】

JSONオブジェクトの、他のJSONオブジェクトを使用した変換を特定するためのシステムおよび方法が、提供される。JavaScriptオブジェクト表記法（JavaScript Object Notation: JSON）を使用して特定された第1のオブジェクトが受信される。第1のオブジェクトは一組の1つ以上の属性を含み、各属性は予め定められたJSONデータタ

50

イブのもので、少なくとも1つの値を有する。JavaScriptオブジェクト表記法を使用して特定された第2のオブジェクトも受信される。第2のオブジェクトは一組の1つ以上の属性を含み、各属性は第1のオブジェクトの一組の属性のうちの少なくとも1つの属性に対応しており、1つ以上の変換を定義する少なくとも1つの値を有する。第2のオブジェクトを使用して第1のオブジェクトを変換することに基づいて、JavaScriptオブジェクト表記法を使用して特定された第3のオブジェクトが生成される。

【0007】

一実施形態では、JSONデータを変換するための方法は、JavaScriptオブジェクト表記法を使用して特定された第1のオブジェクトをコンピュータで受信するステップを含む。第1のオブジェクトは一組の1つ以上の属性を有し、各属性は予め定められたJSONデータタイプのもので、少なくとも1つの値を有していてもよい。JavaScriptオブジェクト表記法を使用して特定された第2のオブジェクトが受信され、それは一組の1つ以上の属性を有し、各属性は第1のオブジェクトの一組の属性のうちの少なくとも1つの属性に対応しており、1つ以上の変換を定義する少なくとも1つの値を有する。第1のオブジェクトは、第2のオブジェクトに基づいて、JavaScriptオブジェクト表記法を使用して特定された第3のオブジェクトに変換される。

10

【0008】

ある実施形態では、第2のオブジェクトをコンピュータで受信するステップは、名前が第1のオブジェクトの属性の名前に対応している属性を有する、JavaScriptオブジェクト表記法を使用して特定されたオブジェクトを受信するステップを含む。名前が第1のオブジェクトの属性の名前に対応している属性を有するオブジェクトを受信するステップは、第1のオブジェクトの属性と同じ名前を有するとして属性を受信するステップを含み得る。名前が第1のオブジェクトの属性の名前に対応している属性を有するオブジェクトを受信するステップは、第1のオブジェクトの属性の名前を参照する値を有するとして属性を受信するステップを含み得る。

20

【0009】

さらに他の実施形態では、第2のオブジェクトをコンピュータで受信するステップは、第1のオブジェクトの属性の名前または第1のオブジェクトの属性の値のいずれかに適用されるべき演算を特定する値を有する、JavaScriptオブジェクト表記法を使用して特定されたオブジェクトの属性を受信するステップを含み得る。第2のオブジェクトをコンピュータで受信するステップは、第1のオブジェクトの属性の変換に対応する第3のオブジェクトの属性のデータタイプを特定する値を有する、JavaScriptオブジェクト表記法を使用して特定されたオブジェクトの属性を受信するステップを含み得る。第2のオブジェクトをコンピュータで受信するステップは、第1のオブジェクトの1つ以上の属性の修正を特定する値を有する、JavaScriptオブジェクト表記法を使用して特定されたオブジェクトの属性を受信するステップを含み得る。

30

【0010】

さまざまな実施形態では、第2のオブジェクトをコンピュータで受信するステップは、1つ以上の変換が第1のオブジェクトに適用可能かどうかを定義する1つ以上のテストを定義する値を有する、JavaScriptオブジェクト表記法を使用して特定されたオブジェクトの属性を受信するステップを含み得る。1つ以上のテストは、第1のオブジェクトの1つ以上の属性に対して定義され得る。1つ以上の変換を行なうために、1つ以上のテストの肯定的評価が判断され、使用されてもよい。1つ以上の変換をコンピュータによって行なうステップは、第1のオブジェクトの属性の名前を変更するステップ、第1のオブジェクトの属性の値に対して1つ以上の数値演算を行なうステップ、第1のオブジェクトの属性の値に対して1つ以上の文字列演算を行なうステップ、オブジェクトでもある第1のオブジェクトの属性に1つ以上の属性を追加するステップ、または、オブジェクトでもある第1のオブジェクトの属性から1つ以上の属性を除去するステップを含み得る。

40

【0011】

一実施形態では、非一時的なコンピュータ読取可能媒体は、JSONデータを変換する

50

ためにコンピュータシステムのプロセッサによって実行可能なコンピュータプログラム製品を格納し、または、当該コンピュータプログラム製品が組込まれている。非一時的なコンピュータ読取可能媒体は、JavaScriptオブジェクト表記法を使用して特定された第1のオブジェクトを受信するためのコードを含み得る。第1のオブジェクトは一組の1つ以上の属性を有し、各属性は予め定められたJSONデータタイプのもので、少なくとも1つの値を有する。非一時的なコンピュータ読取可能媒体はさらに、JavaScriptオブジェクト表記法を使用して特定された第2のオブジェクトを受信するためのコードを含み得る。第2のオブジェクトは一組の1つ以上の属性を有し、各属性は第1のオブジェクトの一組の属性のうちの少なくとも1つの属性に対応しており、1つ以上の変換を定義する少なくとも1つの値を有する。非一時的なコンピュータ読取可能媒体はさらに、第1のオブジェクトを、第2のオブジェクトに基づいて、JavaScriptオブジェクト表記法を使用して特定された第3のオブジェクトに変換するためのコードを含み得る。

10

【0012】

一実施形態では、JSONデータを変換するためのシステムは、ハードウェアプロセッサと、一組の命令を格納するメモリとを含み得る。一組の命令は、プロセッサによって実行されると、プロセッサに以下のステップを行なわせ、前記以下のステップは、JavaScriptオブジェクト表記法を使用して特定された第1のオブジェクトを受信するステップを含み、第1のオブジェクトは一組の1つ以上の属性を有し、各属性は予め定められたJSONデータタイプのもので、少なくとも1つの値を有し、前記以下のステップはさらに、JavaScriptオブジェクト表記法を使用して特定された第2のオブジェクトを受信するステップを含み、第2のオブジェクトは一組の1つ以上の属性を有し、各属性は第1のオブジェクトの一組の属性のうちの少なくとも1つの属性に対応しており、1つ以上の変換を定義する少なくとも1つの値を有し、前記以下のステップはさらに、第1のオブジェクトを、第2のオブジェクトに基づいて、JavaScriptオブジェクト表記法を使用して特定された第3のオブジェクトに変換するステップを含む。

20

【0013】

この開示の主題の性質および均等物（ならびに、提供されたあらゆる固有のまたは明白な利点および改良）についての一層の理解が、この開示の残りの部分、あらゆる添付図面、および請求項を参照することにより、上述の項に加えて実現されるはずである。

【0014】

図面の簡単な説明

この開示において見出されたそれらの革新、実施形態、および/または例を適正に説明し図示するために、1つ以上の添付図面を参照してもよい。1つ以上の添付図面を説明するために使用される追加の詳細または例は、請求される発明のうちのいずれか、現在説明されている実施形態および/または例のうちのいずれか、もしくは、この開示において提示されたいずれかの革新の現在理解されている最良の形態の範囲への限定として考えられるべきでない。

30

【図面の簡単な説明】

【0015】

【図1】この開示の一実施形態に従った、実施形態システムのコンポーネントによって提供されるサービスをモバイルクラウドサービスとして提供し得るシステム環境のコンポーネントの簡略ブロック図である。

40

【図2】一実施形態において、JSONデータを変換するためのワークフローの簡略ブロック図である。

【図3】一実施形態において、1つ以上の変換を定義するJSONオブジェクトを使用してJSONデータ変換を行なうための方法のフローチャートである。

【図4】一実施形態において、あるオブジェクトのJSONデータを、別のオブジェクトのJSONデータ属性言語を使用してどのように変換するかを判断するための方法のフローチャートである。

【図5】一実施形態において、JSONスタイルシート言語を使用してJSONデータを

50

どのように変換するかを判断するための方法のフローチャートである。

【図6】一実施形態において、JSONスタイルシート言語オブジェクトの属性と同じ名前を有するJSONデータ属性を変換するための方法のフローチャートである。

【図7】一実施形態において、入力JSON文字列と、JSON文字列と、JSON文字列を使用した入力JSON文字列の属性の変換から生じる出力文字列との図である。

【図8】一実施形態において、JSONスタイルシート言語オブジェクトの属性によって参照されるJSONデータ属性を変換するための方法のフローチャートである。

【図9】一実施形態において、入力JSON文字列と、JSON文字列と、JSON文字列を使用した入力JSON文字列の属性値の変換から生じる出力文字列との図である。

【図10】一実施形態において、JSONスタイルシート言語オブジェクトの属性の式を満たすJSONデータ属性を変換するための方法のフローチャートである。

【図11】一実施形態において、式が満たされた場合の、入力JSON文字列と、JSON文字列と、JSON文字列を使用した入力JSON文字列の変換から生じる出力文字列との図である。

【図12】一実施形態において、式が満たされた場合の、入力JSON文字列と、JSON文字列と、JSON文字列を使用した入力JSON文字列の変換から生じる出力文字列との図である。

【図13】一実施形態において、JSONスタイルシート言語オブジェクトを使用してJSONオブジェクト変換を行なうように構成されたハードウェアユニットのブロック図である。

【図14】実施形態のうちの1つを実現するための分散型システムの簡略図である。

【図15】この発明のさまざまな実施形態が実現され得る例示的なコンピュータシステムを示す図である。

【発明を実施するための形態】

【0016】

発明の詳細な説明

はじめに

以下の記載では、説明の目的のため、この発明の実施形態の完全な理解を提供するために特定の詳細が述べられる。しかしながら、これらの特定の詳細がなくてもさまざまな実施形態は実践され得る、ということは明らかであろう。たとえば、実施形態を必要以上に詳細に記載して不明瞭にすることがないように、回路、システム、ネットワーク、プロセス、および他のコンポーネントは、ブロック図の形のコンポーネントとして示されてもよい。他の例では、実施形態を不明瞭にしないように、周知の回路、プロセス、アルゴリズム、構造、および手法は、不要な詳細なしで示されてもよい。図および説明は、限定的であるよう意図されてはいない。むしろ、例示的な実施形態の以下の記載は、例示的な実施形態を実現するための実施可能な記載を当業者に提供するであろう。添付された請求項で述べられるようなこの発明の精神および範囲から逸脱することなく、要素の機能および構成においてさまざまな変更が行なわれてもよい、ということが理解されるべきである。

【0017】

また、個々の実施形態は、フローチャート、フロー図、データフロー図、構造図、またはブロック図として示されるプロセスとして説明され得ることに留意されたい。フローチャートは動作を順次プロセスとして説明し得るものの、動作の多くは並行してまたは同時に行なうことが可能である。加えて、動作の順序は並べ替えられてもよい。プロセスは、その動作が完了すると終了するが、図に含まれない追加のステップを有していてもよい。プロセスは、方法、機能、手順、サブルーチン、サブプログラムなどに対応していてもよい。プロセスがある機能に対応している場合、その終了は、その機能が呼出し元の機能または主機能に戻ることに対応可能である。

【0018】

「マシン読取可能媒体」または「コンピュータ読取可能媒体」という用語は、命令および/またはデータを格納し、含み、または担持することができる、携帯型または非携帯型

10

20

30

40

50

記憶装置、光学記憶装置、無線チャネル、ならびにさまざまな他の媒体を含むものの、それらに限定されない。コードセグメントまたはマシン実行可能命令が、手順、機能、サブプログラム、プログラム、ルーチン、サブルーチン、モジュール、ソフトウェアパッケージ、クラス、もしくは、命令、データ構造またはプログラム文の任意の組合せを表わしてもよい。コードセグメントは、情報、データ、引数、パラメータ、またはメモリ内容を渡し、および/または受信することによって、別のコードセグメントまたはハードウェア回路に結合されてもよい。情報、引数、パラメータ、データなどは、メモリ共有、メッセージパッシング、トークンパッシング、ネットワーク送信などを含む任意の好適な手段を介して渡され、発送され、または送信されてもよい。

【0019】

10

さらに、実施形態は、ハードウェア、ソフトウェア、ファームウェア、ミドルウェア、マイクロコード、ハードウェア記述言語、またはそれらの任意の組合せによって実現されてもよい。ソフトウェア、ファームウェア、ミドルウェアまたはマイクロコードで実現される場合、必要なタスクを行なうためのプログラムコードまたはコードセグメントが、マシン読取可能またはコンピュータ読取可能媒体に格納されてもよい。1つ以上のプロセッサが、必要なタスクを行なってもよい。

【0020】

図のうちのいくつかに示されたシステムは、さまざまな構成で提供されてもよい。いくつかの実施形態では、システムは、システムの1つ以上のコンポーネントが1つ以上のネットワークを通してクラウドコンピューティングシステムに分散された分散型システムとして構成されてもよい。さらに他の実施形態では、システムは、システムの1つ以上のコンポーネントが単一の構造またはパッケージに組込まれた単一システムとして構成されてもよい。

20

【0021】

図1は、この開示の一実施形態に従った、実施形態システムの1つ以上のコンポーネントによって提供されるサービスをクラウドサービスとして提供し得るシステム環境100の1つ以上のコンポーネントの簡略ブロック図である。図示された実施形態では、システム環境100は、1つ以上のクライアントコンピューティング装置104、106、および108にクラウドサービスを提供するクラウドインフラストラクチャシステム102を含む。クライアントコンピューティング装置104、106、および108は、クラウドインフラストラクチャシステム102とやりとりするためにユーザによって使用されてもよい。クライアントコンピューティング装置104、106、および108は、クラウドインフラストラクチャシステム102によって提供されるサービスを使用するためにクラウドインフラストラクチャシステム102とやりとりするためにクライアントコンピューティング装置のユーザによって使用され得る、ウェブブラウザ、専用クライアントアプリケーション（たとえば、オラクル・フォームズ（Oracle Forms））、または何らかの他のアプリケーションといったクライアントアプリケーションを動作させるように構成されてもよい。

30

【0022】

図1に示すクラウドインフラストラクチャシステム102は、図示されたもの以外のコンポーネントを有していてもよい、ということが理解されるべきである。また、図1に示す実施形態は、この発明の一実施形態を取入れ得るクラウドインフラストラクチャシステムの単なる一例である。いくつかの他の実施形態では、クラウドインフラストラクチャシステム102は、図1に示すものよりも多い、または少ないコンポーネントを有していてもよく、2つ以上のコンポーネントを組合せてもよく、もしくは、異なる構成または配置のコンポーネントを有していてもよい。

40

【0023】

クライアントコンピューティング装置104、106、および108は、携帯型ハンドヘルド装置（たとえば、iPhone（登録商標）、携帯電話、iPad（登録商標）、コンピューティングタブレット、携帯情報端末（PDA））、またはウェアラブル装置（

50

たとえば、グーグル・グラス（Google Glass）（登録商標）頭部装着型ディスプレイ）であってもよく、マイクロソフト・ウィンドウズ・モバイル（Microsoft Windows Mobile）（登録商標）などのソフトウェア、および／または、i O S、ウィンドウズ（登録商標）フォン、アンドロイド（登録商標）、ブラックベリー（登録商標）10、パームOSなどのさまざまなモバイルオペレーティングシステムを実行し、インターネット、電子メール、ショートメッセージサービス（short message service：SMS）、ブラックベリー（登録商標）、または他の通信プロトコルに対応している。クライアントコンピューティング装置104、106、および108は、マイクロソフト・ウィンドウズ（登録商標）、アップル・マッキントッシュ（登録商標）、および／またはLinux（登録商標）オペレーティングシステムのさまざまなバージョンを実行するパーソナルコンピュータおよび／またはラップトップコンピュータを例として含む、汎用パーソナルコンピュータであり得る。クライアントコンピューティング装置104、106、および108は、たとえばグーグル・クロームOSなどのさまざまなGNU/Linuxオペレーティングシステムを何ら限定されることなく含む、商業的に入手可能なさまざまなUNIX（登録商標）またはUNIX様オペレーティングシステムのうちのいずれかを実行するワークステーションコンピュータであり得る。それに代えて、またはそれに加えて、クライアントコンピューティング装置104、106、および108は、ネットワーク110を通して通信可能である、シンクライアントコンピュータ、インターネット対応ゲーミングシステム（たとえば、Kinect（登録商標）ジェスチャー入力装置を有する、または有さない、マイクロソフトXboxゲーミングコンソール）、および／またはパーソナルメッセージング装置といった、任意の他の電子装置であってもよい。

【0024】

例示的なシステム環境100は3つのクライアントコンピューティング装置を有して図示されているが、任意の数のクライアントコンピューティング装置がサポートされてもよい。センサを有する装置などの他の装置が、クラウドインフラストラクチャシステム102とやりとりしてもよい。

【0025】

ネットワーク110は、クライアント104、106、および108とクラウドインフラストラクチャシステム102との間でのデータの通信および交換を容易にしてもよい。ネットワーク110は、TCP/IP（transmission control protocol/Internet protocol：伝送制御プロトコル/インターネットプロトコル）、SNA（systems network architecture：システムネットワークアーキテクチャ）、IPX（Internet packet exchange：インターネットパケット交換）、アップル・トーク（Apple Talk）などを何ら限定されることなく含む、商業的に入手可能なさまざまなプロトコルのうちのいずれかを使用してデータ通信をサポートできる、当業者にはよく知られた任意のタイプのネットワークであってもよい。単なる例として、ネットワーク110は、イーサネット（登録商標）、トークンリング（Token-Ring）などに基づくものといった、ローカルエリアネットワーク（local area network：LAN）であり得る。ネットワーク110は、ワイドエリアネットワークおよびインターネットであり得る。それは、仮想プライベートネットワーク（virtual private network：VPN）を何ら限定されることなく含む仮想ネットワーク、イントラネット、エクストラネット、公衆交換電話網（public switched telephone network：PSTN）、赤外線ネットワーク、無線ネットワーク（たとえば、電気電子技術者協会（the Institute of Electrical and Electronics：IEEE）802.11プロトコルスイート、Bluetooth（登録商標）、および／または任意の他の無線プロトコルのうちのいずれかの下で動作するネットワーク）、ならびに／もしくは、これらのおよび／または他のネットワークの任意の組合せを含み得る。

【0026】

クラウドインフラストラクチャシステム102は、1つ以上のコンピュータおよび／またはサーバを含んでいてもよい。これらのコンピュータシステムまたはサーバは、1つ以上の汎用コンピュータ、専用サーバコンピュータ（PC（パーソナルコンピュータ）サー

10

20

30

40

50

バ、UNIX（登録商標）サーバ、ミッドレンジサーバ、メインフレームコンピュータ、ラックマウントサーバなどを例として含む）、サーバファーム、サーバクラス、もしくは任意の他の適切な構成および／または組合せで構成されてもよい。さまざまな実施形態では、クラウドインフラストラクチャシステム102に関連付けられた1つ以上のコンピュータシステムまたはサーバは、前述の開示で説明された1つ以上のサービスまたはソフトウェアアプリケーションを実行するように適合されてもよい。たとえば、クラウドインフラストラクチャシステム102に関連付けられた1つ以上のコンピュータシステムまたはサーバは、この開示の一実施形態に従った、ここに記載された処理を行なうためのサーバに対応していてもよい。

【0027】

クラウドインフラストラクチャシステム102に関連付けられた1つ以上のコンピュータシステムまたはサーバは、上述のもののうちのいずれかを含むオペレーティングシステム、および商業的に入手可能な任意のサーバオペレーティングシステムを実行してもよい。クラウドインフラストラクチャシステム102に関連付けられた1つ以上のコンピュータシステムまたはサーバはまた、さまざまな追加のサーバアプリケーションおよび／または中間層アプリケーションのうちのいずれかを実行してもよく、HTTP（hypertext transport protocol：ハイパーテキスト伝送プロトコル）サーバ、FTP（file transfer protocol：ファイル転送プロトコル）サーバ、CGI（common gateway interface：コモンゲートウェイインターフェイス）サーバ、JAVA（登録商標）サーバ、データベースサーバなどを含む。例示的なデータベースサーバは、オラクル、マイクロソフト、サイベ

【0028】

ある実施形態では、クラウドインフラストラクチャシステム102によって提供されるサービスは、オンラインデータストレージおよびバックアップソリューション、ウェブベースの電子メールサービス、ホスト型オフィススイートおよび文書コラボレーションサービス、データベース処理、管理された技術サポートサービスなどといった、クラウドインフラストラクチャシステム102のユーザにとってオンデマンドで利用可能にされる多数のサービスを含んでいてもよい。クラウドインフラストラクチャシステム102によって提供されるサービスは、そのユーザの必要性を満たすために動的にスケール変更され得る。クラウドインフラストラクチャシステム102によって提供されるサービスの特定のインスタンス化は、ここに「サービスインスタンス」と呼ばれる。一般に、クラウドサービスプロバイダのシステムから、インターネットなどの通信ネットワークを介してユーザに利用可能とされる任意のサービスは、「クラウドサービス」と呼ばれる。典型的には、パブリッククラウド環境では、クラウドサービスプロバイダのシステムを作り上げるサーバおよびシステムは、顧客自身の業務用サーバおよびシステムとは異なっている。たとえば、クラウドサービスプロバイダのシステムは、アプリケーションをホストしてもよく、ユーザは、インターネットなどの通信ネットワークを介してオンデマンドでアプリケーションをオーダーし、使用してもよい。

【0029】

いくつかの例では、クラウドインフラストラクチャ102によってインスタンス化されたサービスインスタンスは、クラウドベンダーによってユーザに提供されるかまたは当該技術分野において他の態様で公知であるようなストレージ、ホスト型データベース、ホスト型ウェブサーバ、ソフトウェアアプリケーション、もしくは他のサービスへの、保護されたコンピュータネットワークアクセスを含んでいてもよい。たとえば、クラウドインフラストラクチャ102によってインスタンス化されたサービスインスタンスは、インターネットを通じた、クラウド上のリモートストレージへの、パスワードで保護されたアクセスを含み得る。別の例として、クラウドインフラストラクチャ102によってインスタンス化されたサービスインスタンスは、ネットワーク化された開発者による私的使用のための、ウェブサービスベースのホスト型リレーショナルデータベースおよびスクリプト言語

ミドルウェアエンジンを含み得る。別の例として、クラウドインフラストラクチャ 102 によってインスタンス化されたサービスインスタンスは、クラウドベンダーのウェブサイト上でホストされる電子メールソフトウェアアプリケーションへのアクセスを含み得る。

【0030】

ある実施形態では、クラウドインフラストラクチャシステム 102 は、セルフサービスで、サブスクリプションベースで、弾力的にスケーラブルで、信頼でき、高可用性で、かつ安全な態様で顧客に配信される、アプリケーション、ミドルウェア、開発サービス、およびデータベースサービス提供物一式を含んでいてもよい。クラウドインフラストラクチャサービス 102 において具現化されたような、そのようなクラウドインフラストラクチャシステムの一例は、本譲受人によって提供されるオラクル・パブリック・クラウド (Oracle Public Cloud) である。

10

【0031】

クラウドインフラストラクチャシステム 102 は、異なるデプロイメントモデルを介してクラウドサービスを提供してもよい。たとえば、サービスは、クラウドインフラストラクチャシステム 102 がクラウドサービスを販売する組織によって所有され (たとえば、オラクルによって所有され)、サービスが一般大衆または異なる産業企業にとって利用可能とされる、パブリッククラウドモデルの下で提供されてもよい。別の例として、サービスは、クラウドインフラストラクチャシステム 102 が単一の組織のためにのみ動作され、その組織内の 1 つ以上のエンティティのためのサービスを提供し得る、プライベートクラウドモデルの下で提供されてもよい。クラウドサービスはまた、クラウドインフラストラクチャシステム 102、およびクラウドインフラストラクチャシステム 102 によって提供されるサービスが、関連するコミュニティにおけるいくつかの組織によって共有される、コミュニティクラウドモデルの下で提供されてもよい。クラウドサービスはまた、2 つ以上の異なるモデルの組合せであるハイブリッドクラウドモデルの下で提供されてもよい。

20

【0032】

いくつかの実施形態では、クラウドインフラストラクチャシステム 102 によって提供されるサービスは、ソフトウェア・アズ・ア・サービス (Software as a Service: SaaS) カテゴリー、プラットフォーム・アズ・ア・サービス (Platform as a Service: PaaS) カテゴリー、インフラストラクチャ・アズ・ア・サービス (Infrastructure as a Service: IaaS) カテゴリー、または、ハイブリッドサービスを含むサービスの他のカテゴリーの下で提供される、1 つ以上のサービスを含んでいてもよい。いくつかの実施形態では、クラウドインフラストラクチャシステム 102 によって提供されるサービスは、アプリケーションサービス、プラットフォームサービス、およびインフラストラクチャサービスを、何ら限定されることなく含んでいてもよい。いくつかの例では、アプリケーションサービスは、SaaS プラットフォームを介して、クラウドインフラストラクチャシステム 102 によって提供されてもよい。SaaS プラットフォームは、SaaS カテゴリーに該当するクラウドサービスを提供するように構成されてもよい。たとえば、SaaS プラットフォームは、統合された開発およびデプロイメントプラットフォーム上にオンデマンドアプリケーション一式を構築し、配信するための能力を提供してもよい。SaaS プラットフォームは、SaaS サービスを提供するための根底的なソフトウェアおよびインフラストラクチャを管理し、制御してもよい。SaaS プラットフォームによって提供されるサービスを利用することにより、顧客は、クラウドインフラストラクチャシステム上で実行されるアプリケーションを利用できる。顧客は、顧客が別々のライセンスおよびサポートを購入する必要なく、アプリケーションサービスを取得できる。さまざまな異なる SaaS サービスが提供されてもよい。例は、大型組織のための販売実績管理、企業統合、およびビジネス柔軟性についてのソリューションを提供するサービスを、何ら限定されることなく含む。

30

40

【0033】

いくつかの実施形態では、プラットフォームサービスは、PaaS プラットフォームを

50

介して、クラウドインフラストラクチャシステム 102 によって提供されてもよい。PaaS プラットフォームは、PaaS カテゴリーに該当するクラウドサービスを提供するように構成されてもよい。プラットフォームサービスの例は、(オラクルなどの)組織が共有の共通アーキテクチャ上で既存のアプリケーションを統合できるようにするサービスと、プラットフォームによって提供される共有のサービスを活用する新しいアプリケーションを構築するための能力とを、何ら限定されることなく含んでいてもよい。PaaS プラットフォームは、PaaS サービスを提供するための根底的なソフトウェアおよびインフラストラクチャを管理し、制御してもよい。顧客は、顧客が別々のライセンスおよびサポートを購入する必要なく、クラウドインフラストラクチャシステム 102 によって提供される PaaS サービスを取得できる。プラットフォームサービスの例は、オラクル Java (登録商標) クラウド・サービス (Java Cloud Service: JCS)、オラクル・データベース・クラウド・サービス (Database Cloud Service: DBCS) などを、何ら限定されることなく含む。

10

【0034】

PaaS プラットフォームによって提供されるサービスを利用することにより、顧客は、クラウドインフラストラクチャシステム 102 によってサポートされるプログラミング言語およびツールを採用するとともに、デプロイメントされたサービスを制御することができる。いくつかの実施形態では、クラウドインフラストラクチャシステム 102 によって提供されるプラットフォームサービスは、データベースクラウドサービス、ミドルウェアクラウドサービス (たとえば、オラクル・フュージョン・ミドルウェア (Oracle Fusion Middleware) サービス)、および Java クラウドサービスを含んでいてもよい。一実施形態では、データベースクラウドサービスは、組織がデータベースリソースをプールし、データベースクラウドの形をしたデータベース・アズ・ア・サービスを顧客に提供することを可能にする共有のサービスデプロイメントモデルをサポートしてもよい。ミドルウェアクラウドサービスは、顧客がさまざまなビジネスアプリケーションを開発してデプロイメントするためのプラットフォームを提供してもよく、Java クラウドサービスは、顧客がクラウドインフラストラクチャシステムにおいて Java アプリケーションをデプロイメントするためのプラットフォームを提供してもよい。

20

【0035】

クラウドインフラストラクチャシステム 102 において、さまざまな異なるインフラストラクチャサービスが、IaaS プラットフォームによって提供されてもよい。これらのインフラストラクチャサービスは、SaaS プラットフォームおよび PaaS プラットフォームによって提供されるサービスを利用する顧客のための、ストレージ、ネットワーク、ならびに他の基礎的コンピューティングリソースなどの根底的なコンピューティングリソースの管理および制御を容易にする。

30

【0036】

ある実施形態では、クラウドインフラストラクチャシステム 102 は、クラウドインフラストラクチャシステムにおけるクラウドサービス (たとえば、SaaS、PaaS、および IaaS サービス) の包括的管理を提供してもよい。一実施形態では、クラウド管理機能性は、クラウドインフラストラクチャシステム 102 によって受信された顧客のサブスクリプションをプロビジョニングし、管理し、追跡するための能力などを含んでいてもよい。さまざまな実施形態では、クラウドインフラストラクチャシステム 102 は、クラウドインフラストラクチャシステム 102 によって提供されるサービスに顧客のサブスクリプションを自動的にプロビジョニングし、管理し、追跡するように適合されてもよい。顧客は、クラウドインフラストラクチャシステム 102 によって提供される 1 つ以上のサービスを、サブスクリプションオーダーを介してオーダーしてもよい。クラウドインフラストラクチャシステム 102 は次に、顧客のサブスクリプションオーダーにおけるサービスを提供するために処理を行なう。

40

【0037】

一実施形態では、図 1 に示すように、クラウド管理機能性は、オーダー管理および監視

50

モジュール 1 1 4 などの 1 つ以上のモジュールによって提供されてもよい。これらのモジュールは、汎用コンピュータ、専用サーバコンピュータ、サーバファーム、サーバクラスター、もしくは任意の他の適切な構成および / または組合せであり得る、1 つ以上のコンピュータおよび / またはサーバを含んでいてもよく、もしくはそれらを使用して提供されてもよい。

【 0 0 3 8 】

例示的な動作では、クライアントコンピューティング装置 1 0 4、1 0 6 または 1 0 8 のうちの 1 つ以上といったクライアント装置を使用する顧客は、クラウドインフラストラクチャシステム 1 0 2 によって提供される 1 つ以上のサービスを要求することにより、クラウドインフラストラクチャシステム 1 0 2 とやりとりしてもよい。顧客は、さまざまな手段を使用して、サービス要求 1 3 4 をクラウドインフラストラクチャシステム 1 0 2 に発行してもよい。サービス要求 1 3 4 は、クラウドインフラストラクチャシステム 1 0 2 によって提供される 1 つ以上のサービスについてサブスクリプションオーダーを出すこと、クラウドインフラストラクチャシステム 1 0 2 によって提供される 1 つ以上のサービスにアクセスすること、などを含んでいてもよい。ある実施形態では、顧客は、クラウドユーザインターフェイス (User Interface : UI) であるクラウド UI 1 3 2、クラウド UI 1 3 4 および / またはクラウド UI 1 3 8 にアクセスし、これらの UI を介してサブスクリプションオーダーを出してもよい。顧客がオーダーを出したことに応答してクラウドインフラストラクチャシステム 1 0 2 が受信したオーダー情報は、顧客を識別する情報と、顧客が申し込むつもりである、クラウドインフラストラクチャシステム 1 0 2 によって提供される 1 つ以上のサービスを識別する情報とを含んでいてもよい。顧客によってオーダーが出された後で、オーダー情報がクラウド UI 1 3 2、1 3 4 および / または 1 3 8 を介して受信される。

【 0 0 3 9 】

この例では、オーダー管理および監視モジュール 1 1 4 は、顧客から受信した情報をオーダーデータベースに送信して、顧客によって出されたオーダーを記録されるよう格納させる。オーダーデータベースは、クラウドインフラストラクチャシステム 1 0 2 によって動作され、他のシステム要素とともに動作される、いくつかのデータベースのうちの 1 つであり得る。オーダー管理および監視モジュール 1 1 4 は、オーダーデータベースに格納されたオーダー情報のすべてまたは一部を含む情報をオーダー管理モジュールに発送してもよい。場合によっては、オーダー管理モジュールは、オーダーを検証し、検証後にオーダーを予約するといった、オーダーに関連する請求および課金機能を行なうように構成されてもよい。

【 0 0 4 0 】

オーダー管理および監視モジュール 1 1 4 は、顧客によって出されたオーダーのためのサービスおよびリソースのプロビジョニングをオーケストレーションするためにオーダー情報を利用するオーダーオーケストレーションモジュールに、オーダー情報のすべてまたは一部を通信してもよい。場合によっては、オーダーオーケストレーションモジュールは、オーダープロビジョニングモジュールのサービスを使用して、申し込まれたサービスをサポートするようにリソースのプロビジョニングをオーケストレーションしてもよい。

【 0 0 4 1 】

ある実施形態では、オーダーオーケストレーションモジュールは、各オーダーに関連付けられたビジネスプロセスの管理を可能にし、オーダーがプロビジョニングに進むべきか否かを判断するためにビジネスロジックを適用する。新規サブスクリプションのオーダーを受信すると、オーダーオーケストレーションモジュールは、リソースを割当ててサブスクリプションオーダーを遂行するために必要とされるリソースを構成するようにという要求を、オーダープロビジョニングモジュールに送信する。オーダープロビジョニングモジュールは、顧客によってオーダーされたサービスのためのリソースの割当てを可能にする。オーダープロビジョニングモジュールは、クラウドインフラストラクチャシステム 1 0 2 によって提供されるクラウドサービスと、要求されたサービスを提供するためのリソー

10

20

30

40

50

スをプロビジョニングするために使用される物理的実装層との間の抽象化のレベルを提供する。オーダーオーケストレーションモジュールはこのため、サービスおよびリソースが実際にオンザフライでプロビジョニングされるか否か、または予めプロビジョニングされて要求時にのみ割当てられるか否かといった実装詳細から切り離されてもよい。

【 0 0 4 2 】

ある実施形態では、オーダー管理および監視モジュール 1 1 4 は、顧客のサブスクリプションオーダーを管理し、追跡する。場合によっては、オーダー管理および監視モジュール 1 1 4 は、顧客に関連付けられた、プロビジョニングされた任意のサービスおよび/またはリソースを示す情報を受信してもよい。オーダー管理および監視モジュール 1 1 4 は、使用されるストレージの量、転送されるデータの量、ユーザの数、システムアップタイムおよびシステムダウンタイムの量といった、サブスクリプションオーダーにおけるサービスについての使用統計を収集するように構成されてもよい。

10

【 0 0 4 3 】

上述の例でサービスおよびリソースがいったんプロビジョニングされると、提供されるサービスおよび/またはリソースについて顧客に通知するサービス結果 1 3 8 が、クライアントコンピューティング装置 1 0 4、1 0 6、および/または 1 0 8 上の顧客に送信されてもよい。サービス要求 1 3 0 が、サービスにアクセスするようにという要求、またはサービスに 1 つ以上の動作を行なわせるようにという要求を含む場合、要求されたアクセスまたは任意の動作の結果、行なわれたサービス、または要求されたデータを提供するサービス結果 1 3 8 が、クライアントコンピューティング装置 1 0 4、1 0 6、および/または 1 0 8 上の顧客に送信されてもよい。

20

【 0 0 4 4 】

ある実施形態では、クラウドインフラストラクチャシステム 1 0 0 は、アイデンティティ管理モジュール 1 1 4 を含んでいてもよい。アイデンティティ管理モジュール 1 1 4 は、クラウドインフラストラクチャシステム 1 0 2 においてアクセス管理および認証サービスなどのアイデンティティサービスを提供するように構成されてもよい。いくつかの実施形態では、アイデンティティ管理モジュール 1 1 4 は、クラウドインフラストラクチャシステム 1 0 2 によって提供されるサービスを利用したい顧客についての情報を制御してもよい。そのような情報は、そのような顧客のアイデンティティを認証する情報と、さまざまなシステムリソース（たとえば、ファイル、ディレクトリ、アプリケーション、通信ポート、メモリセグメントなど）に対してそれらの顧客がどのアクションを行なうことが認可されているかを記述する情報とを含み得る。アイデンティティ管理モジュール 1 1 4 はまた、各顧客についての記述的情報と、その記述的情報が誰によってどのようにアクセスされ、修正され得るかについての記述的情報との管理を含んでいてもよい。

30

【 0 0 4 5 】

ある実施形態では、クラウドインフラストラクチャシステム 1 0 2 はまた、クラウドインフラストラクチャシステム 1 0 2 の顧客にさまざまなサービスを提供するために使用されるリソースを提供するためのインフラストラクチャリソース 1 1 6 を含んでいてもよい。一実施形態では、インフラストラクチャリソース 1 1 6 は、P a a S プラットフォームおよび S a a S プラットフォームによって提供されるサービスを実行するために、サーバ、ストレージ、およびネットワークリソースといったハードウェアの予め統合され最適化された組合せを含んでいてもよい。

40

【 0 0 4 6 】

いくつかの実施形態では、クラウドインフラストラクチャシステム 1 0 2 におけるリソースは、複数のユーザによって共有され、要望ごとに動的に再割当てされてもよい。加えて、リソースは、異なる時間帯におけるユーザに割当てられてもよい。たとえば、クラウドインフラストラクチャシステム 1 0 2 は、第 1 の時間帯における第 1 の一組のユーザが、特定数の時間、クラウドインフラストラクチャシステムのリソースを利用することを可能にし、次に、異なる時間帯に位置する別の一組のユーザへの同じリソースの再割当てを可能にして、それによりリソースの利用を最大化してもよい。

50

【 0 0 4 7 】

ある実施形態では、クラウドインフラストラクチャシステム 1 0 2 の異なるコンポーネントまたはモジュールによって、およびクラウドインフラストラクチャシステム 1 0 2 が提供するサービスによって共有される、多くの内部共有サービス 1 1 8 が提供されてもよい。これらの内部共有サービス 1 1 8 は、セキュリティおよびアイデンティティサービス、統合サービス、企業リポジトリサービス、企業マネージャサービス、ウィルススキャンングおよびホワイトリストサービス、高可用性、バックアップおよび復元サービス、クラウドサポートを可能にするためのサービス、電子メールサービス、通知サービス、ファイル転送サービスなどを、何ら限定されることなく含んでいてもよい。

【 0 0 4 8 】

ある実施形態では、クラウドインフラストラクチャシステム 1 0 2 の異なるコンポーネントまたはモジュールによって、およびクラウドインフラストラクチャシステム 1 0 2 が提供するサービスによって共有される、多くの外部共有サービス 1 2 0 が提供されてもよい。これらの外部共有サービス 1 2 0 は、セキュリティおよびアイデンティティサービス、統合サービス、企業リポジトリサービス、企業マネージャサービス、ウィルススキャンングおよびホワイトリストサービス、高可用性、バックアップおよび復元サービス、クラウドサポートを可能にするためのサービス、電子メールサービス、通知サービス、ファイル転送サービスなどを、何ら限定されることなく含んでいてもよい。

【 0 0 4 9 】

さまざまな実施形態では、外部共有サービス 1 2 0 は、アクセス、データ変換、自動化などを企業コンピュータシステム 1 2 6 に提供する 1 つ以上のコンポーネントを含んでいてもよい。企業コンピュータシステム 1 2 6 へのアクセスは、クラウドインフラストラクチャシステム 1 0 2 の異なるコンポーネントまたはモジュールによって、およびクラウドインフラストラクチャシステム 1 0 2 が提供するサービスによって、共有されてもよい。いくつかの実施形態では、企業コンピュータシステム 1 2 6 へのアクセスは、1 人以上のサブスクリバに制限される、クラウドインフラストラクチャシステム 1 0 2 が提供するサービスインスタンスによって、共有されてもよい。

【 0 0 5 0 】

さらなる実施形態では、外部共有サービス 1 2 0 は、クラウドインフラストラクチャシステム 1 0 2 の異なるコンポーネントまたはモジュールによって、およびクラウドインフラストラクチャシステム 1 0 2 が提供するサービスによって共有される、外部アプリケーションプログラミングインターフェイス (application programming interface : A P I) サービス 1 2 8 を含んでいてもよい。これらの外部 A P I サービス 1 2 8 は、他の第三者サービスまたはエンティティによって提供される A P I を、何ら限定されることなく含んでいてもよい。

【 0 0 5 1 】

クラウドインフラストラクチャシステム 1 0 2 では、モバイルクラウドサービス (mobile cloud service : M C S) 1 2 2 によって、さまざまな異なるモバイルクラウドサービスが提供されてもよい。この発明のいくつかの実施形態によれば、M C S 1 2 2 は、モバイルコンピューティング装置と企業コンピュータシステム (たとえば、企業コンピュータシステム 1 2 4 および 1 2 6) との間の通信を容易にする。M C S 1 2 2 は、企業データおよび認証情報を格納するために使用される 1 つ以上のメモリ記憶装置「 (「ローカルストレージ」) を含んでいてもよい。企業データは、企業コンピュータシステム 1 2 6 から、あるいはクライアントコンピューティング装置 1 0 4、1 0 6、または 1 0 8 から受信されてもよく、もしくは、クラウドインフラストラクチャシステム 1 0 2 によって変換された企業データを含んでいてもよく、もしくはそれらの組合せであってもよい。認証情報は、アイデンティティ管理システム 1 1 6 から受信されてもよく、および/または、クラウドインフラストラクチャシステム 1 0 2 によって生成されてもよい。いくつかの実施形態では、認証情報は、サービスについての要求に関するユーザのセキュリティ認証を示す情報を含んでいてもよい。

10

20

30

40

50

【 0 0 5 2 】

企業コンピュータシステム 1 2 6 などの企業コンピュータシステムは、クラウドインフラストラクチャシステム 1 0 2 のファイアウォールを越えて、クラウドインフラストラクチャシステム 1 0 2 とは異なる地理的場所（たとえば、リモートの地理的場所）に物理的に位置していてもよい。いくつかの実施形態では、企業コンピュータシステム 1 2 6 は、1 つ以上の異なるコンピュータまたはサーバを含んでいてもよい。いくつかの実施形態では、企業コンピュータシステム 1 2 6 は、単一のコンピュータシステムの一部であってもよい。

【 0 0 5 3 】

ある実施形態では、企業コンピュータシステム 1 2 6 は、1 つ以上の異なるプロトコルを使用して、クラウドインフラストラクチャシステム 1 0 2 と通信してもよい。企業コンピュータシステム 1 2 6 の各々は、異なる通信プロトコルを使用して、クラウドインフラストラクチャシステム 1 0 2 と通信してもよい。企業コンピュータシステム 1 2 6 は、同じまたは異なるセキュリティプロトコルをサポートしてもよい。いくつかの実施形態では、M S C 1 1 1 2 は、企業コンピュータシステム 1 2 6 との通信を扱うためのエージェントシステムを含んでいてもよい。

【 0 0 5 4 】

プロトコルは、S P D Y などの通信プロトコルを含んでいてもよい。プロトコルは、H T T P ベースのプロトコルなどのアプリケーションプロトコルを含んでいてもよい。いくつかの実施形態では、企業コンピュータシステム 1 2 6 は、R E S T または S O A P 通信プロトコルを使用して、クラウドインフラストラクチャシステム 1 0 2 と通信してもよい。たとえば、R E S T プロトコルは、U R I または U R L を含むフォーマットをサポートしてもよい。R E S T プロトコルを使用する通信のためにフォーマットされた企業データは、J S O N、コンマ区切り形式（comma-separated values：C S V）、およびリアリー・シンプル・シンジケーション（really simple syndication：R S S）などのデータフォーマットに容易に変換されてもよい。企業コンピュータシステム 1 2 6 とクラウドインフラストラクチャシステム 1 0 2 とは、リモート・プロシージャ・コール（remote procedure call：R P C）（たとえば X M L R P C）などの他のプロトコルを使用して通信してもよい。

【 0 0 5 5 】

いくつかの実施形態では、M C S 1 2 2 は、クラウドインフラストラクチャサービス 1 0 2 によって提供される 1 つ以上のサービスとの通信をサポートするように構成されたアダプタインターフェイスを含んでいてもよく、それらのサービスのうちのいくつかは、通信用の異なるプロトコルまたは手法をサポートしてもよい。いくつかの実施形態では、M C S 1 2 2 は、企業コンピュータシステム 1 2 6 との通信をサポートするように構成されたアダプタインターフェイスを含んでいてもよく、企業コンピュータシステム 1 2 6 のうちのいくつかは、通信用の異なるプロトコルまたは手法をサポートしてもよい。M C S 1 2 2 は、通信プロトコル、企業コンピュータシステムのタイプ、アプリケーションのタイプ、サービスのタイプ、またはそれらの組合せに従って通信するように各々構成され得る 1 つ以上のアダプタを含んでいてもよい。アダプタによってサポートされる通信プロトコルは、サービスに、または企業コンピュータシステム 1 2 6 のうちの 1 つ以上に特有のものであってもよい。

【 0 0 5 6 】

ある実施形態では、クライアントコンピューティング装置 1 0 4、1 0 6、および 1 0 8 は各々、M C S 1 2 2 と通信するための特定のユーザインターフェイスを提供できるアプリケーションを実装してもよい。特定の U I が、特定の通信プロトコルを使用して通信するように構成されてもよい。いくつかの実施形態では、特定の U I が、M C S 1 2 2 と通信するために呼び出され得る、呼出し可能インターフェイス、機能、ルーチン、方法、および/または動作を含んでいてもよい。特定の U I は、企業データのために、および/またはサービスを要求するために、クラウドインフラストラクチャサービス 1 0 2 によ

て提供されるサービスと通信するためのパラメータ、または企業コンピュータシステム 126 と通信するためのパラメータを、入力として受け入れてもよい。いくつかの実施形態では、MCS 122 を通じた通信は、カスタム通信プロトコルを使用する通信のために変換されてもよい。いくつかの実施形態では、特定の UI が、アプリケーションにおけるカスタムクライアントに対応していてもよい。

【0057】

MCS 122 は、1 つ以上の呼出し可能インターフェイス、たとえばアプリケーションプログラミングインターフェイス (API) を含んでいてもよい。MCS 122 に関連付けられた呼出し可能インターフェイスは、モバイルコンピューティング装置上のアプリが MCS 122 に要求を通信することを可能にしてもよい。MCS 122 に関連付けられた呼出し可能インターフェイスは共通または標準インターフェイスをサポートしてもよく、それは、パラメータを含む要求が、標準化プロトコル、アーキテクチャスタイル、および/またはフォーマット (たとえば REST プロトコル) に従って、アプリから受信されることを可能にしてもよい。MCS 122 に関連付けられた呼出し可能インターフェイスは、コンピューティング装置 104、106、または 108 のうちのいずれか 1 つのユーザによって構成可能であってもよい。MCS 122 に関連付けられた呼出し可能インターフェイスは、通信プロトコルに従って、サービスについての要求を受信してもよい。装置アプリケーション開発者らは、自身のカスタムアプリケーションのために MCS 122 に接続することができる。いくつかの実施形態では、MCS 122 に関連付けられた呼出し可能インターフェイスは、アプリを開発したのと同じ人によって構成され、その人が、MCS 122 と通信するためのカスタムアプリを実現できるようになっていてもよい。

【0058】

MCS 122 に関連付けられた呼出し可能インターフェイスはさらに、企業コンピュータシステム 126 が、標準化プロトコルまたはフォーマットに従って MCS 122 と通信することを可能にしてもよい。アプリケーション開発者らと同様に、企業コンピュータシステムを管理する人は、1 つ以上の呼出し可能インターフェイスを介して MCS 122 と通信するように構成されたコード (たとえばエージェントシステム) を実現することができる。MCS 122 に関連付けられた呼出し可能インターフェイスは、コンピューティング装置のタイプ、企業コンピュータシステムのタイプ、アプリ、エージェントシステム、サービス、プロトコル、または他の基準に基づいて実現されてもよい。いくつかの実施形態では、MCS 122 に関連付けられた呼出し可能インターフェイスは、認証、圧縮、暗号化、カーソルを用いたページネーション、クライアントベースのスロットリング、否認不可、ロギング、およびメトリクス収集を含むサービスについての要求をサポートしてもよい。いくつかの実施形態では、MCS 122 に関連付けられた呼出し可能インターフェイスは、認証、ポリシー実施、応答のキャッシング、MCS 122 への呼出しのスロットリング、非同期パターンと同期パターンとの間の変換、根底的なサービスへの呼出しのロギング、またはそれらの組合せ、といったカスタムビジネス関連サービスのために実現されてもよい。いくつかの実施形態では、MCS 122 に関連付けられた呼出し可能インターフェイスは、ユーザが、クラウドインフラストラクチャシステム 102 による実現のためにカスタムコードをロードすることを可能にしてもよい。カスタムコードは、クラウドインフラストラクチャシステム 102 のために、MCS 122 に関連付けられた 1 つ以上の呼出し可能インターフェイスを実現してもよく、それは、ユーザがカスタムサービスまたは他の企業コンピュータシステムにアクセスすることを可能にできる。

【0059】

MCS 122 に関連付けられたプロトコルトランスレータは、メッセージを処理して、メッセージ用の通信プロトコルを判断し、および/またはメッセージを宛先用の通信プロトコルに変換してもよい。MCS 122 に関連付けられたプロトコルトランスレータは、クライアントコンピューティング装置 104、106、または 108 から受信された要求を変換してもよい。要求は、クライアントコンピューティング装置 104、106、または 108 によってサポートされる通信プロトコルのフォーマットから、クラウドインフラ

ストラクチャサービス 102 または企業コンピュータシステム 126 によって提供されるサービスによってサポートされる通信プロトコルのフォーマットに変換されてもよい。MCS 122 に関連付けられたプロトコルトランスレータは、クラウドインフラストラクチャサービス 102 または企業コンピュータシステム 126 によって提供されるサービスから受信された応答を変換してもよい。応答は、クラウドインフラストラクチャサービス 102 または企業コンピュータシステム 126 によって提供されるサービスによってサポートされる通信プロトコルのフォーマットから、クライアントコンピューティング装置 104、106、または 108 によってサポートされる通信プロトコルのフォーマットに変換されてもよい。

【0060】

MCS 122 に関連付けられたセキュリティサービスは、クライアントコンピューティング装置 104、106、または 108 のうちのいずれかから受信された要求についてのセキュリティ認証を管理してもよい。MCS 122 に関連付けられたセキュリティサービスは、顧客プロセスおよび企業データの保全性を保護してもよい。システムまたはデータが損なわれないようにするために、クライアントコンピューティング装置 104、106、または 108 から要求が受信されると、セキュリティ認証が起こってもよい。セキュリティ認証は、クラウドインフラストラクチャシステム 102 による処理のために要求が發送される前に行なわれてもよい。あるユーザについて判断されたセキュリティ認証は、モバイルコンピューティング装置に関連付けられたユーザが、MCS 122 を介してサービスを要求するための認証を有することを可能にしてもよい。セキュリティ認証は、ユーザが、MCS 122 を介して要求された異なる要求および/またはサービスについて認証するための労力を減少させてもよい。MCS 122 に関連付けられたセキュリティサービスは、要求のセキュリティを認証するさまざまな動作を行なうように構成された 1 つ以上の機能ブロックまたはモジュールとして実現されてもよい。

【0061】

MCS 122 に関連付けられた認証サービスは、クライアントコンピューティング装置 104、106、または 108 から受信された要求についてのセキュリティ認証を管理してもよい。MCS 122 に関連付けられた認証サービスは、MCS 122 に要求を送信するコンピューティング装置に関連付けられたユーザについてセキュリティ認証を判断してもよい。セキュリティ認証は期間に基づいて判断されてもよく、期間は、アプリの動作（たとえば、アプリの起動）、要求、コンピューティング装置、企業コンピュータシステム、要求に関連する他の基準、またはそれらの組合せに結び付けられてもよい。セキュリティ認証は、個々の要求、1 つ以上の企業コンピュータシステム、特定のサービス、サービスのタイプ、ユーザ、コンピューティング装置、セキュリティ認証を判断するための他の基準、またはそれらの組合せ、などのうちのいずれか 1 つについて、検証され付与されてもよい。いくつかの実施形態では、クラウドインフラストラクチャシステム 102 は、企業コンピュータシステム、または企業コンピュータシステムをサポートする認証システムから受信されたユーザの認証情報を格納してもよい。クラウドインフラストラクチャシステム 102 は、要求に関連付けられたユーザのアイデンティティがそのような要求を行なう認可を有するかどうかを判断するためにルックアップ機能を行なうことによって、認証を判断してもよい。格納された認証情報は、ユーザがアクセスすることを認可され得る、要求、機能、企業コンピュータシステム、企業データなどのタイプといった情報を含んでもよい。いくつかの実施形態では、インフラストラクチャシステム 102 は、認証を判断するために、要求元のコンピューティング装置との通信を起動してもよい。

【0062】

いくつかの実施形態では、セキュリティ認証は、サービスを要求するユーザに関連付けられた役割に基づいて判断されてもよい。役割は、MCS 122 へのアクセスを要求するユーザに関連付けられてもよい。いくつかの実施形態では、ユーザは、MCS 122 によって提供されるリソースおよび/またはサービスへのアクセスを付与され得る MCS 122 のサブスクリバまたはテナントとして、サービスを要求してもよい。認証は、ユーザ

がサブスクライバとしてMCS122を介してサービスを要求することが認可され得るように、MCS122へのユーザのサブスクリプションに対応していてもよい。いくつかの実施形態では、サブスクリプションは、MCS122によって提供される特定の一組のリソースに限定されてもよい。セキュリティ認証は、MCS122のユーザがアクセス可能なリソースおよび/またはサービスに基づいていてもよい。いくつかの実施形態では、「実行時環境」と呼ばれる実行中に、要求にテンプレートがプロビジョニングされてもよい。実行時環境は、要求、ユーザ、または装置に割当てられるリソースに関連付けられてもよい。

【0063】

いくつかの実施形態では、MCS122に関連付けられた認証サービスは、アイデンティティ管理システムに、ユーザについてセキュリティ認証を判断するように要求してもよい。アイデンティティ管理システムは、クラウドインフラストラクチャシステム102によって（たとえばアイデンティティ管理114として）実現されてもよく、または、クラウドインフラストラクチャシステム102の外部の別のコンピュータシステムによって実現されてもよい。アイデンティティ管理116は、MCS122にアクセスするためのユーザの役割またはサブスクリプションに基づいて、ユーザのセキュリティ認証を判断してもよい。役割またはサブスクリプションには、企業コンピュータシステム、企業コンピュータシステムによって提供されるサービス、企業コンピュータシステムの機能または特徴、企業コンピュータシステムへのアクセスを制御するための他の基準、またはそれらの組合せに対する特権および/または権利が割当てられてもよい。

【0064】

クラウドインフラストラクチャシステム102では、アプリケーション開発フレームワーク(application development framework:ADF)124によって、さまざまな異なるアプリケーション開発フレームワーク(ADF)が提供されてもよい。ADF124は、アジャイルなSOAベースのアプリケーションを実装するためにインフラストラクチャコードを提供する。ADF124はさらに、1つ以上の開発ツール（たとえば、オラクルJDeveloper 11g開発ツール）を通して、開発への視覚的および宣言的アプローチを提供する。ADF124によって提供される1つ以上のフレームワークは、モデル・ビュー・コントローラデザインパターンを実装してもよい。そのようなフレームワークは、MVCアーキテクチャのすべての層を、オブジェクト/リレーショナルマッピング、データ持続性、再使用可能なコントローラ層、リッチなウェブユーザインターフェイスフレームワーク、UIへのデータ結合、セキュリティおよびカスタム化などのエリアに対する解決策でカバーする統合的解決策を提供する。コアウェブベースのMVCアプローチを超えて、そのようなフレームワークはまた、オラクルSOAおよびウェブセンターポータル(WebCenter Portal)フレームワークと統合されて、完全な複合アプリケーションの作成を簡略化する。

【0065】

ある実施形態では、ADF124は、クラウドインフラストラクチャシステム102によって提供されるビルトインビジネスサービスにサービスインターフェイスを結合することによって、サービスとしてデータを公開するアジャイルなアプリケーションを開発することを容易にする。ビジネスサービス実装詳細のこの分離は、ADF124においてメタデータを介して行なわれる。このメタデータ駆動型アーキテクチャの使用により、アプリケーション開発者らは、サービスがどのようにアクセスされるかの詳細ではなく、ビジネスロジックおよびユーザ体験に集中できるようになる。ある実施形態では、ADF124は、サービスの実装詳細を、モデル層におけるメタデータに格納する。これにより、開発者らは、ユーザインターフェイスを修正することなくサービスを交換できるようになり、アプリケーションが非常にアジャイルになる。加えて、ユーザインターフェイスを作成する開発者は、ビジネスサービスアクセス詳細に悩まされる必要がない。代わりに、開発者らは、アプリケーションインターフェイスおよびやりとりロジックの開発に集中することができる。ユーザ体験を作り出すことは、ビジュアルページデザイナー上に所望のビジネ

スサービスをドラッグアンドドロップし、どのタイプのコンポーネントがそのデータを表わすべきかを示すのと同じくらい単純であり得る。

【 0 0 6 6 】

さまざまな実施形態では、開発者らは A D F 1 2 4 とやりとりして、企業アプリケーションを形成するモジュールを作成する。企業アプリケーションは、クラウドインフラストラクチャシステム 1 0 2 のコンテキスト内で実行され得る。さまざまな実施形態では、開発者らは A D F 1 2 4 とやりとりして、モバイルアプリケーションを形成するモジュールを作成する。モバイルアプリケーションは、クラウドインフラストラクチャシステム 1 0 2 のコンテキスト内で実行され得る。以下に説明されるこの発明の特徴は、ここに提供される開示を読むことによって当業者には明らかであるように、プログラミング言語とアプリケーション開発フレームワークとの任意の所望の組合せを使用して実現されてもよい。

10

【 0 0 6 7 】

A D F 1 2 4 によって提供される 1 つ以上のフレームワークは、一例ではオラクル A D F として具現化され得る。したがって、A D F 1 2 4 におけるフレームワークは、モデル・ビュー・コントローラ (Model-View-Controller : M V C) デザインパターンに基づき得る。M V C アプリケーションは、1) データソースとのやりとりを扱い、ビジネスロジックを実行するモデル層と、2) アプリケーションユーザインターフェイスを扱うビュー層と、3) アプリケーションフローを管理し、モデル層とビュー層との間のインターフェイスとして作用するコントローラとに分離される。これら 3 つの層にアプリケーションを分離することは、アプリケーション間にわたるコンポーネントの保守および再使用を簡略化する。各層の他の層からの独立は、緩く結合されたサービス指向アーキテクチャ (Service Oriented Architecture : S O A) をもたらす。

20

【 0 0 6 8 】

さまざまな実施形態では、A D F 1 2 4 は、開発者らがアプリケーションを多層の形で作成することを可能にするツールおよびリソースを提供し、各層は、予め定義された仕様に従って所望のロジックを実現するコードモジュール / ファイルを含む。このため、一実施形態では、A D F 1 2 4 は、アプリケーションが、アプリケーションのユーザインターフェイスを提供するコードモジュール / ファイルを含むビュー層と、アプリケーションのフローを制御するコードモジュールを含むコントローラ層と、根底的なデータのための抽象化層を提供するデータ / コードモジュールを含むモデル層と、さまざまなソースからのデータへのアクセスを提供し、ビジネスロジックを扱うコードモジュールを含むビジネスサービス層という 4 つの層として開発されることを可能にする。

30

【 0 0 6 9 】

ある実施形態では、A D F 1 2 4 は、開発者らに、層の各々を実装する際に自分たちが使用したい技術を選択させる。E J B、ウェブサービス (Web Services)、J a v a B e a n s、J P A / エクリプスリンク (EclipseLink) / トップリンク (TopLink) オブジェクト、および他の多くがすべて、A D F 1 2 4 のためのビジネスサービスとして使用可能である。ビュー層は、J S F、デスクトップスイング (Desktop Swing) アプリケーション、および M S オフィスフロントエンドを用いて実現されるウェブベースのインターフェイスと、モバイル装置のためのインターフェイスとを含み得る。

40

【 0 0 7 0 】

一局面では、ビュー層は、開発中のアプリケーションのユーザインターフェイスを表わす。ビュー層は、デスクトップビュー、モバイルビュー、およびブラウザベースのビューを含んでいてもよく、それらの各々はユーザインターフェイスのすべてまたは一部を提供するとともに、ビュータイプに対応するさまざまな態様でアクセス可能である。たとえば、ウェブページが、対応する U R L を含むクライアント要求を受信することに応答して、アプリケーションによって送信されてもよい。ウェブページは次に、要求元クライアントシステムに関連付けられた表示部 (図示せず) 上にブラウザによって表示されてもよく、それにより、要求元クライアントシステムのユーザが企業アプリケーションとやりとりすることを可能にする。A D F 1 2 4 は、ビジネスサービスの再使用を可能にするビジネス

50

サービスへのマルチチャネルアクセスと、ウェブクライアント、クライアント・サーバ・スイングデスクトップベース・アプリケーション (client-server swing desktop-based application)、マイクロソフトエクセルスプレッドシート、スマートフォンなどのモバイル装置、などからのアクセスとをサポートする。

【 0 0 7 1 】

(ウェブページなどの) ビュー層を形成するコードファイル / モジュールは、ハイパーテキストマークアップ言語 (hypertext markup language: HTML)、Java サーバページ (Java server page: JSP)、および Java サーバフェイス (JSF) のうちの 1 つ以上を使用して実現されてもよい。それに代えて、ユーザインターフェイスは、スイング (Swing) などの Java コンポーネントおよび / または拡張マークアップ言語 (extensible markup language: XML) を使用して実現されてもよい。さらに示されるように、ユーザインターフェイスは、マイクロソフトによるワードおよびエクセルなどのデスクトップアプリケーションについてのユーザの経験および習熟を活用してもよい。

10

【 0 0 7 2 】

上述のように、ユーザが開発した関連するコード / データモジュールが、層の各々において提供される。しかしながら、各層は典型的には、ADF 124 によって提供される他の予め定義されたコード / データモジュールを含む。予め定義されたモジュールのうちのいくつかは、たとえば、ウェブページを開発するためのテンプレート、開発されたコードに所望の機能性を含めるためのテンプレートなどとして、開発中に使用されてもよい。(URL 書換モジュールなどの) 他の予め定義されたモジュールが、開発されたアプリケーションとともにデプロイメントされてもよく、企業アプリケーションの実行中に追加の機能性 (要求された URL の、内部名へのマッピング) をユーザに提供してもよい。

20

【 0 0 7 3 】

コントローラ層は、アプリケーションのフローを制御するコードモジュール / ファイルを含む。各コントローラオブジェクトは、ビュー層において情報を提示する所望の態様に従って実現されたソフトウェア命令および / またはデータを含む。所望の態様は、別のウェブページにおけるリンクがユーザによってクリック / 選択されると表示される特定のウェブページ、実行中にエラーが起こると表示される、格納 / 検索すべき特定のデータを示すページなどを含んでいてもよい。

【 0 0 7 4 】

30

一局面では、コントローラ層はアプリケーションフローを管理し、ユーザ入力を扱う。たとえば、ページ上でサーチボタンがクリックされると、コントローラは、どのアクションを行なうべきか (サーチを行なう) と、どこにナビゲートすべきか (結果ページ) とを判断する。JDeveloper では、ウェブベースのアプリケーションについて、標準の JSF コントローラ、または JSF コントローラ機能性を拡張する ADF コントローラという 2 つのコントローラオプションがある。どちらのコントローラが使用されても、アプリケーションフローは典型的には、ページおよびナビゲーションルールをダイアグラム上にレイアウトすることによってデザインされる。アプリケーションのフローは、より小さい、再使用可能なタスクフローに分割可能であり、方法コールおよび決定ポイントなどの非視覚的コンポーネントをフローに含めることができ、単一の含有ページの領域内で実行される「ページフラグメント」フローを作成することができる。

40

【 0 0 7 5 】

コントローラ層を形成するコードモジュール / ファイルはしばしば、クライアント要求を受信するとともに対応する応答として所望のウェブページを送信する Java サブレットとして実現される。コントローラオブジェクトはまた、たとえば、アパッチジャカルタストラット (Apache Jakarta Struts) コントローラとして、または JSF 規格に従って実現されてもよい。

【 0 0 7 6 】

モデル層は、図示されるように、さまざまなビジネスサービスを、それらを他の層で使用するオブジェクトに、たとえば上述のコントローラオブジェクトに、または直接デスク

50

トップアプリケーションに接続する、データ/コードモジュールを含む。モデル層の各抽象データオブジェクトは、任意のタイプのビジネスサービスにアクセスするために使用可能である対応するインターフェイスを提供し、根底的なビジネスサービス層で実行される。データオブジェクトは、クライアントからサービスのビジネスサービス実装詳細を抽象化し、および/または、データ制御方法/属性をビューコンポーネントに公開してもよく、ビュー層とデータ層との分離を提供する。

【0077】

一局面では、モデル層は、データ制御およびデータ結合という2つのコンポーネントからなり、それらは、インターフェイスを定義するためにメタデータファイルを利用する。データ制御は、クライアントからビジネスサービス実装詳細を抽象化する。データ結合は、データ制御方法および属性をUIコンポーネントに公開し、ビューとモデルとのクリーンな分離を提供する。モデル層のメタデータアーキテクチャにより、開発者らは、任意のタイプのビジネスサービス層実装をビュー層およびコントローラ層に結合する際に、同じ開発経験を得る。

【0078】

ある実施形態では、ADF 124は、開発プロセス全体にわたって宣言的プログラミングパラダイムの使用を強調して、ユーザが、実現詳細に手をつける必要なく、アプリケーション作成のロジックに集中できるようにする。高レベルでは、フュージョン(Fusion)ウェブアプリケーションのための開発プロセスは通常、アプリケーションワークスペースを作成することを伴う。ウィザードを使用して、開発者が選択した技術にとって必要なライブラリおよび構成が自動的に追加され、アプリケーションが、パッケージおよびディレクトリを有するプロジェクトへと構造化される。

【0079】

データベースオブジェクトをモデル化することにより、オンラインデータベースまたは任意のデータベースのオフラインレプリカが作成可能であり、定義が編集可能であり、スキーマが更新可能である。UMLモデラーを使用して、使用事例が次に、アプリケーションのために作成可能である。アプリケーション制御およびナビゲーションもデザイン可能である。アプリケーション制御およびナビゲーションのフローを視覚的に判断するために、ダイアグラマ(diagrammer)が使用可能である。次に、フローを記述する根底的なXMLファイルが自動的に作成可能である。開発者が、インポートされたライブラリを、アプリケーションに単純にドラッグアンドドロップすることによって閲覧および使用することを可能にするために、リソースライブラリが使用可能である。データベーステーブルから、エンティティオブジェクトが、ウィザードまたはダイアログを使用して作成可能である。それらのエンティティオブジェクトから、ビューオブジェクトが、アプリケーションにおけるページによって使用されるように作成される。検証ルールおよび他のタイプのビジネスロジックが実現可能である。

【0080】

この例では、ビジネスサービス層は、データ持続性層とのやりとりを管理する。それは、データ持続性、オブジェクト/リレーショナルマッピング、トランザクション管理、およびビジネスロジック実行などのサービスを提供する。ビジネスサービス層は、シンプルJavaクラス、EJB、ウェブサービス、JPAオブジェクト、およびオラクルADFビジネスコンポーネントといったオプションのうちのいずれかで実装可能である。加えて、データは、ファイル(XMLまたはCSV)およびRESTから直接消費され得る。このため、各ビジネスサービスは、対応するデータ持続性層とのやりとりを管理し、また、オブジェクト/リレーショナルマッピング、トランザクション管理、ビジネスロジック実行などのサービスを提供する。ビジネスサービス層は、シンプルJavaクラス、エンタープライズJavaBeans、ウェブサービスなどのうちの1つ以上を使用して実装されてもよい。

【0081】

ビジネスコンポーネントは、データベース、ウェブサービス、レガシーシステム、アプ

10

20

30

40

50

リケーションサーバなどとのやりとりを提供するために、たとえばオラクル A D F ビジネスコンポーネントを使用して実装されるビジネスサービスを表わす。一実施形態では、ビジネスサービス層のビジネスコンポーネントは、ビジネスサービス実装を提供するために協働するアプリケーションモジュール、ビュー/クエリオブジェクト、およびエンティティオブジェクトの混合を含む。アプリケーションモジュールは、アプリケーション/トランザクションデータと連携するために U I クライアントが通信するトランザクションコンポーネント/コードモジュールであり得る。アプリケーションモジュールは、更新可能なデータモデルを提供してもよく、また、ユーザトランザクションに関する手順/機能（一般にサービス方法と呼ばれる）を提供してもよい。

【 0 0 8 2 】

10

エンティティオブジェクトは、データベーステーブルにおける対応する行を表わしてもよく、対応する行に格納されたデータの操作（更新、削除など）を簡略化する。エンティティオブジェクトはしばしば、所望のビジネスルールが一貫して実施されることを確実にするために、対応する行のためのビジネスロジックをカプセル化する。エンティティオブジェクトはまた、根底的なデータベースに格納された行間に存在する関係を反映するように、他のエンティティオブジェクトに関連付けられてもよい。

【 0 0 8 3 】

J S O N スタイルシート言語変換

図 2 は、一実施形態において、J S O N データを変換するためのワークフロー 2 0 0 の簡略ブロック図である。ワークフロー 2 0 0 では、入力 J S O N データ 2 1 0 が、J S L A P I 2 2 0 で受信される。（同様に J S O N 文書である）J S L 2 3 0 が、J S L A P I 2 2 0 で受信される。J S L A P I 2 2 0 は、入力 J S O N データ 2 1 0 および J S L 2 3 0 に基づいて、変換 J S O N データ 2 4 0 を生成する。

20

【 0 0 8 4 】

この例示的な実施形態では、一般に、入力 J S O N データ 2 1 0 は、オブジェクトまたはデータの一部の論理構造を特定する J S O N 文書である。入力 J S O N データ 2 1 0 の論理構造は、1 つ以上の J S O N スキーマに従って定義されてもよい。

【 0 0 8 5 】

J S L 2 3 0 も J S O N 文書である。プロセッサ（たとえば J S L A P I 2 2 0 ）は、J S L 2 3 0 の J S L (J S O N StyleSheet Language) を理解し、入力 J S O N データ 2 1 0 に 1 つ以上の特定された変換を適用することができる。これは、X M L に変換を適用するために使用される X S L T と同様である。J S L の主な利点のうちの 1 つは、ユーザが他の言語の仕様に精通する必要がない、ということである。これは、X M L 変換を定義する X S L と同様に、J S L 自体が J S O N データフォーマットで書かれているためである。

30

【 0 0 8 6 】

市場には、J S O N データを変換する多くの変換ツールがある。しかしながら、これらのツールの多くは別の言語で書かれており、それはユーザが扱うことを困難にする。たとえば、JsonT は、X M L 界における X S L T と同様に、J S O N 状構造を変換するための言語である。この実装は、J S L ほど単純明快ではない。別の例では、json2json は、CoffeeScript で書かれており、Node.js 環境で実行されるように設計されている。この実装の欠点は、変換用のテンプレートファイルを作成するために、ユーザが Coffee 言語を学ばなければならない、ということである。しかしながら、J S L では、ユーザが変換用に特定するテンプレートは J S O N で書かれている。

40

【 0 0 8 7 】

さまざまな実施形態では、J S L 2 3 0 は、J S L 仕様に従って定義される。J S L は、入力 J S O N データ 2 1 0 の変換を実現するようにプロセッサを構成するキーワードを含み得る。一局面では、入力 J S O N データ 2 1 0 における特定されたキーについての値をどのように表わすかをプロセッサに命令するために、“value” というキーワードが、J S L ファイルにおいて提供される。“value” というキーワードは値で定義されてもよ

50

い。一実施形態では、値は、入力JSONデータ210におけるキーの語彙経路を示す語彙文字列であり得る。この経路は“root”というキーワードで始まることができ（たとえば、root.employee.id）、ここでemployeeはオブジェクトであり、idは文字列値属性である。

【0088】

別の実施形態では、値は、値と考えられる、シングルクォーテーションマーク間に定義された任意のテキストを示すテキスト文字列であってもよい。さらに別の実施形態では、値は、語彙/テキスト文字列とそれらに対して定義された算術/文字列/論理演算との組合せを示す演算文字列であってもよい。たとえば、“concat(root.employee.id, ‘abc’)”は、idフィールドの値が、変換JSONデータ240に挿入されると、文字列‘abc’と連結されることになっていることを示す。

10

【0089】

別の局面では、特定のキーについての値のデータタイプをどのように表わすかをプロセッサに命令するために、“type”というキーワードが、JSLファイルにおいて提供される。さまざまな実施形態では、“type”というキーワードは値で定義されてもよい。ある実施形態では、値は、整数、文字列、2倍、ブール、ヌルなどであり得る。JSONArray要素については、値は、objectarray、＜基本データタイプ＞アレイ、たとえばStringarrayなどとして特定され得る。

【0090】

さらに別の局面では、エンティティの親の語彙経路をどのように表わすかをプロセッサに命令するために、“source”というキーワードが、JSLファイルにおいて提供される。さまざまな実施形態では、このキーワードは、アレイ要素のためにのみ定義される。さらなる局面では、1つ以上の特定条件に対して入力JSONデータ210をどのように評価するかをプロセッサに命令するために、“testexpression”というキーワードが、JSLファイルにおいて提供される。このキーワードは、どの特定のJSONオブジェクトまたはJSONアレイが処理されるかに基づいて式を評価するために使用され得る。その式は、ブール値を返すことができる。オブジェクトレベルでは、“testexpression”結果に基づいて、プロセッサは、変換JSONデータ240においてオブジェクトが除外されるべきか含まれるべきかを判断することができる。ノードレベルでは、“testexpression”結果に基づいて、プロセッサは、変換JSONデータ240においてノードが除外されるべきか含まれるべきかを判断することができる。

20

30

【0091】

ある実施形態では、さまざまな変換演算がサポートされる。たとえば、文字列演算は、連結、比較、置換、部分文字列などを含み得る。算術演算は、加算、乗算、除算、減算などを含み得る。ブール演算は、true()、false()、not()などを含み得る。

【0092】

図3は、一実施形態において、1つ以上の変換を定義するJSONオブジェクトを使用してJSONデータ変換を行なうための方法300のフローチャートである。図3に示す方法300の実現化例または方法300における処理は、コンピュータシステムまたは情報処理装置などのロジックマシンの中央処理装置（CPUまたはプロセッサ）によって実行される場合にはソフトウェア（たとえば、命令またはコードモジュール）によって、電子装置または特定用途向け集積回路のハードウェアコンポーネントによって、もしくは、ソフトウェア要素とハードウェア要素との組合せによって行なわれてもよい。図3に示す方法300は、ステップ310で始まる。

40

【0093】

ステップ310で、第1のJSONオブジェクトが受信される。第1のJSONオブジェクトは、文字列の形をしたシリアル化オブジェクトとして受信され得る。ステップ320で、1つ以上の変換を定義する第2のJSONオブジェクトが受信される。上述のように、第2のJSONオブジェクトは、ソースオブジェクト、ソースノード、またはそれらに適用されるフィルタへの変換を示す1つ以上の定義されたキーワードを使用して、ユー

50

ザがよく知っている他のJSONデータとして書かれている。

【0094】

ステップ330で、第1のJSONオブジェクトおよび第2のJSONオブジェクトに基づいて、第3のJSONオブジェクトが生成される。したがって、中間形式または中間言語への変換は必要ない。加えて、ユーザは、変換を定義するために新しい言語を学ぶ必要がない。

【0095】

図4は、一実施形態において、あるオブジェクトのJSONデータを、別のオブジェクトのJSONデータ属性言語を使用してどのように変換するかを判断するための方法400のフローチャートである。図4に示す方法400の実現化例または方法400における処理は、コンピュータシステムまたは情報処理装置などのロジックマシンの中央処理装置（CPUまたはプロセッサ）によって実行される場合にはソフトウェア（たとえば、命令またはコードモジュール）によって、電子装置または特定用途向け集積回路のハードウェアコンポーネントによって、もしくは、ソフトウェア要素とハードウェア要素との組合せによって行なわれてもよい。図4に示す方法400は、ステップ410で始まる。

【0096】

ステップ410で、入力オブジェクトが変換オブジェクトの属性によって参照される属性を有するという判断が下される。ある実施形態では、入力オブジェクトと変換オブジェクトとの間で属性名を整合させることによって、参照が行なわれ得る。したがって、入力オブジェクトの属性は、変換オブジェクトの属性と同一の名前を有していてもよい。いくつかの実施形態では、入力オブジェクトの属性は、変換オブジェクトの属性の値によって参照され得る。たとえば、変換オブジェクトの属性の値として提供されるnameというキーワードが、入力オブジェクトの属性の名前を参照するために使用されてもよい。

【0097】

ステップ420で、変換オブジェクトの属性に基づいて入力オブジェクトの属性をどのように変換するかについて判断が下される。ある実施形態では、判断される変換は、入力オブジェクトの属性の名前の変更、入力オブジェクトの属性の値の修正、入力オブジェクトの属性を出力オブジェクトに含めるべきかどうかの判断、などを含んでいてもよい。

【0098】

ステップ430で、変換オブジェクトを使用して入力オブジェクトを変換することに基づいて、出力オブジェクトが生成される。

【0099】

図5は、一実施形態において、JSONスタイルシート言語を使用してJSONデータをどのように変換するかを判断するための方法のフローチャートである。図5に示す方法500の実現化例または方法500における処理は、コンピュータシステムまたは情報処理装置などのロジックマシンの中央処理装置（CPUまたはプロセッサ）によって実行される場合にはソフトウェア（たとえば、命令またはコードモジュール）によって、電子装置または特定用途向け集積回路のハードウェアコンポーネントによって、もしくは、ソフトウェア要素とハードウェア要素との組合せによって行なわれてもよい。図5に示す方法500は、ステップ510で始まる。

【0100】

ステップ510で、入力属性が受信される。入力属性は、キーと値とのペアによって表わされる。キーは属性の名前を提供し、値は1つ以上の基本データタイプまたはオブジェクトであり得る。ステップ520で、入力属性の対応する変換があるかどうかについて判断が下される。上述のように、入力属性と同じ名前を有する変換オブジェクトの属性、入力属性への参照を有する変換オブジェクトの属性、などによって、対応する変換が見出されてもよい。

【0101】

入力属性の対応する変換がないという判断が下された場合、ステップ530で、出力オブジェクトにおいて入力属性を含めるか除外するかについて判断が下される。あるルール

10

20

30

40

50

は、変換がないことにより、属性はそのまま出力オブジェクトに流されるようになる、ということを示してもよい。別のルールは、変換がないことにより、属性は出力オブジェクトからフィルタリングされるようになる、ということを示してもよい。

【0102】

入力属性の対応する変換があるという判断が下された場合、ステップ540で、変換オブジェクトに基づいて一組の条件が判断される。さまざまな実施形態では、変換オブジェクトの対応する属性は、1つ以上の条件を定義する式を含んでいてもよい。条件は、その条件を真と評価するために満たされるべき基準を表わし得る。その基準は、属性の名前、属性の値、それらの組合せ、または、予め定められた基準を満たすかどうかを判断するために評価されるべき他の条件、ルール、しきい値、制限などの観点から表わされ得る。

10

【0103】

ステップ550で、入力属性が定められた基準を満たすかどうかについて判断が下される。入力属性は、基準の空集合を満たすこと、または満たさないことが見出されてもよい。入力属性は、予め定められたしきい値、評価、計算、一組の演算などを満たすこと、または満たさないことが見出されてもよい。入力属性は定められた基準を満たさないという判断が下された場合、ステップ530で、上述のように、出力オブジェクトにおいて入力属性を含めるか除外するかについて判断が下される。

【0104】

入力属性は定められた基準を満たすという判断が下された場合、ステップ560で、出力オブジェクトにおける対応する属性の値を判断する変換オブジェクトに基づいて、行なうべき一組の演算が判断される。ある実施形態では、判断される一組の演算は、入力オブジェクトの属性の名前の変更、入力オブジェクトの属性の値の修正、入力オブジェクトの属性の値のすべてまたは一部を出力オブジェクトに含めるべきかどうかの判断、などを含んでいてもよい。

20

【0105】

ステップ570で、出力オブジェクトにおける対応する属性についての変換オブジェクトに基づいて、データタイプが判断される。データタイプはたとえば、変換オブジェクトによって明示的に定義されてもよく、入力オブジェクトから推論されてもよく、一組の演算を行なった結果に基づいて判断されてもよい。上述のように、判断されるデータタイプは、基本のJSONオブジェクトデータタイプを含み得る。

30

【0106】

ステップ580で、一組の演算を行なった結果の値と判断されたデータタイプとに基づいて、出力オブジェクトの対応する属性が生成される。ある実施形態では、演算の空集合によって、入力属性の値が、出力オブジェクトの対応する属性へと流されてもよい。他の実施形態では、一組の演算は、値の名前の変更、新しい値の計算、値の修正、などを行なうことができる。

【0107】

図6は、一実施形態において、JSONスタイルシート言語オブジェクトの属性と同じ名前を有するJSONデータ属性を変換するための方法600のフローチャートである。図6に示す方法600の実現化例または方法600における処理は、コンピュータシステムまたは情報処理装置などのロジックマシンの中央処理装置(CPUまたはプロセッサ)によって実行される場合にはソフトウェア(たとえば、命令またはコードモジュール)によって、電子装置または特定用途向け集積回路のハードウェアコンポーネントによって、もしくは、ソフトウェア要素とハードウェア要素との組合せによって行なわれてもよい。図6に示す方法600は、ステップ610で始まる。

40

【0108】

ステップ610で、入力オブジェクトは、変換オブジェクトの属性の名前と名前が整合している属性を有する、という判断が下される。図7は、一実施形態において、入力JSON文字列710と、JSON文字列720と、JSON文字列720を使用した入力JSON文字列710の属性の変換から生じる出力JSON文字列730との図である。図示さ

50

れているように、入力JSON文字列710およびJSL文字列720は双方とも、名前“id”による属性を有する。

【0109】

ステップ620で、同じ名前を有する変換オブジェクトの属性から、演算およびデータタイプに関して判断が下される。図7を参照して、JSL文字列720の属性“id”は、式“root.id * 2”を使用して入力JSON文字列710の属性“id”をどのようにに変換するかを特定する。言い換えれば、入力JSON文字列710の属性“id”の値に2が乗算されることになっている。JSL文字列720の属性“id”はさらに、式“root.id * 2”から生じる値は整数であるはずである、ということ特定する。

【0110】

ステップ630で、変換オブジェクトの整合する属性によって特定された、判断された演算が、入力オブジェクトの属性の値に対して行なわれる。ステップ640で、演算の結果と判断されたデータタイプとに基づいて、同じ名前を有する出力オブジェクトについての属性が作成される。図7に示すように、出力JSON文字列730は、式“root.id * 2”を使用して求められた値を有する属性“id”を含む。出力JSON文字列730の属性“id”のデータタイプは、整数である。

【0111】

図8は、一実施形態において、JSONスタイルシート言語オブジェクトの属性によって参照されるJSONデータ属性を変換するための方法800のフローチャートである。図8に示す方法800の実現化例または方法800における処理は、コンピュータシステムまたは情報処理装置などのロジックマシンの中央処理装置(CPUまたはプロセッサ)によって実行される場合にはソフトウェア(たとえば、命令またはコードモジュール)によって、電子装置または特定用途向け集積回路のハードウェアコンポーネントによって、もしくは、ソフトウェア要素とハードウェア要素との組合せによって行なわれてもよい。図8に示す方法800は、ステップ810で始まる。

【0112】

ステップ810で、入力オブジェクトは、変換オブジェクトの属性によって名前が参照される属性を有する、という判断が下される。図7に示すように、入力JSON文字列710は名前“name”による属性を有し、JSL文字列720は、入力JSON文字列710の名前“name”による属性を値である式(たとえば“value”: “root.name”)によって参照する名前“new_name”による属性を有する。

【0113】

ステップ820で、変換オブジェクトの属性から、属性名に関して判断が下される。ステップ830で、変換オブジェクトの属性から、データタイプに関して判断が下される。ステップ840で、判断された名前と判断されたデータタイプとを有する出力オブジェクトについての属性が作成される。図7に示すように、出力JSON文字列730は、入力JSON文字列710の属性“name”と同じ値(たとえば“A green door”)およびデータタイプ(たとえば文字列)を有する属性“new_name”を含む。

【0114】

図9は、一実施形態において、入力JSON文字列910と、JSL文字列920と、JSL文字列920を使用した入力JSON文字列910の属性値の変換から生じる出力JSON文字列930との図である。この例では、入力JSON文字列910とJSL文字列920とは、整合する属性“tags”を有する。JSL文字列920は、上に示すような属性名の変換ではなく、対応する属性の値の変換を特定する。文字列演算“concat(root.tags, ‘_new’)”は、“tags”アレイにおける各要素に添字“_new”を追加するために特定されたものである。したがって、出力JSON文字列930は、アレイの各要素に添字“_new”が追加された“tags”という名前の属性を含む。

【0115】

図10は、一実施形態において、JSONスタイルシート言語オブジェクトの属性の式を満たすJSONデータ属性を変換するための方法1000のフローチャートである。図

10

20

30

40

50

10に示す方法1000の実現化例または方法1000における処理は、コンピュータシステムまたは情報処理装置などのロジックマシンの中央処理装置（CPUまたはプロセッサ）によって実行される場合にはソフトウェア（たとえば、命令またはコードモジュール）によって、電子装置または特定用途向け集積回路のハードウェアコンポーネントによって、もしくは、ソフトウェア要素とハードウェア要素との組合せによって行なわれてもよい。図10に示す方法1000は、ステップ1010で始まる。

【0116】

ステップ1010で、入力オブジェクトは、変換オブジェクトの属性の式を満たす属性または属性値を有する、という判断が下される。その式は、入力属性の名前を参照する、または入力属性の値を参照する、同じ名前を有する変換オブジェクトの属性の値において特定され得る。図11は、一実施形態において、式が満たされた場合の、入力JSON文字列1110と、JSL文字列1120と、JSL文字列1120を使用した入力JSON文字列1110の変換から生じる出力JSON文字列1130との図である。図示されるように、入力JSON文字列1110は名前“emp”による属性を有し、JSL文字列1120は名前“emp”による属性を有する。JSL文字列1120の名前“emp”による属性は、empnameが“B”で始まるオブジェクトによって満たされる“textexpression”における式を含む。したがって、“emp”アレイにおけるたった1つのオブジェクトが条件を満たす（たとえば“empname”：“Bob”）。

【0117】

ステップ1020で、変換オブジェクトの属性から、演算に関して判断が下される。図11を参照して、JSL文字列1120の属性“emp”は、“textexpression”を満たす入力JSON文字列1110の属性“emp”の各オブジェクトをどのように変換するかを特定する。ステップ1030で、変換オブジェクトの属性から、データタイプに関して判断が下される。ステップ1040で、演算の結果と判断されたデータタイプとに基づいて、出力オブジェクトについての属性が作成される。図11に示すように、出力JSON文字列1130は、“empname”：“Bob”を有する単一のオブジェクトを有する“emp”属性を含む。したがって、属性または属性値をフィルタリングにより除去し、出力オブジェクト中に所望のものだけを残す条件が特定され得る。

【0118】

図12は、一実施形態において、式が満たされた場合の、入力JSON文字列1210と、JSL文字列1220と、JSL文字列1220を使用した入力JSON文字列1210の変換から生じる出力JSON文字列1230との図である。この例では、所望されない属性または属性値はフィルタリングにより除去される。したがって、CustomerTypeが“Gold”ではなく“Silver”であるため、“textexpression”はオブジェクトから“Discount”：20という属性を除去する。

【0119】

図13は、一実施形態において、JSONスタイルシート言語オブジェクトを使用してJSONオブジェクト変換を行なうように構成されたハードウェアユニット1300のブロック図である。ハードウェアユニット1300は、JSONデータ受信ユニット1310と、プリプロセッサユニット1320と、JSON変換受信ユニット1330と、変換ユニット1340と、後処理ユニット1350と、JSONデータ生成ユニット1360とを含む。各ユニットは、コンピュータプロセッサにより、メモリに格納するコンピュータ実行可能コードに従って実現されてもよい。

【0120】

JSONデータ受信ユニット1310は、JSONデータを受信するハードウェアおよび/またはソフトウェア要素を含む。JSONデータ受信ユニット1310は、ユーザ入力または他のコンピュータプロセスからJSONデータを受信することができる。プリプロセッサユニット1320は、JSONデータを前処理するハードウェアおよび/またはソフトウェア要素を含む。1つ以上の前処理ステップは、データ操作演算、データフィルタリング演算、データ増大演算などを含んでいてもよい。JSON変換受信ユニット13

10

20

30

40

50

30は、JSONデータを受信するハードウェアおよび/またはソフトウェア要素を含む。JSON変換受信ユニット1330は、ユーザ入力または他のコンピュータプロセスからJSONデータを受信することができる。この例では、JSON変換受信ユニット1330で受信されたJSONデータは、上述のJSL仕様に従ってフォーマット化される。
【0121】

変換ユニット1340は、JSONオブジェクトの形のJSONデータと、JSL変換と、適用される任意の特定された変換とを受信するハードウェアおよび/またはソフトウェア要素を含む。変換ユニット1340は、変換を実現するために、XMLなどの中間言語を必要としない。

【0122】

後処理ユニット1350は、JSONデータを処理するハードウェアおよび/またはソフトウェア要素を含む。1つ以上の処理ステップは、データ操作演算、データフィルタリング演算、データ増大演算などを含んでもよい。JSONデータ生成ユニット1360は、JSONデータを生成するハードウェアおよび/またはソフトウェア要素を含む。複数のユニットが図示されたが、すべてが必要とされるとは限らず、いくつかのユニットによって行なわれるいくつかの機能を他のユニットに組込むことができる、ということが理解されるであろう。

【0123】

結論

図14は、実施形態のうちの1つを実現するための分散型システム1400の簡略図を示す。図示された実施形態では、分散型システム1400は1つ以上のクライアントコンピューティング装置1402、1404、1406、および1408を含み、それらは、1つ以上のネットワーク1410を通して、ウェブブラウザ、専用クライアント（たとえば、オラクル・フォームズ）などのクライアントアプリケーションを実行し、動作させるように構成される。サーバ1412は、ネットワーク1410を介して、リモートのクライアントコンピューティング装置1402、1404、1406、および1408と通信可能に結合されてもよい。

【0124】

さまざまな実施形態では、サーバ1412は、システムのコンポーネントのうちの1つ以上によって提供される1つ以上のサービスまたはソフトウェアアプリケーションを実行するように適合されてもよい。いくつかの実施形態では、これらのサービスは、ウェブベースのサービスまたはクラウドサービスとして、もしくはソフトウェア・アズ・ア・サービス（SaaS）モデルまたはプラットフォーム・アズ・ア・サービス（PaaS）モデルの下で、クライアントコンピューティング装置1402、1404、1406、および/または1408のユーザに提供されてもよい。クライアントコンピューティング装置1402、1404、1406、および/または1408を動作させるユーザは次に、これらのコンポーネントによって提供されるサービスを利用するためにサーバ1412とやりとりするために、1つ以上のクライアントアプリケーションを利用してもよい。

【0125】

図14に示す構成では、システム1400のソフトウェアコンポーネント1418、1420、および1422は、サーバ1412上で実現されるとして図示されている。他の実施形態では、システム1400のコンポーネントおよび/またはこれらのコンポーネントによって提供されるサービスのうちの1つ以上も、クライアントコンピューティング装置1402、1404、1406、および/または1408のうちの1つ以上によって実現されてもよい。クライアントコンピューティング装置を動作させるユーザは次に、これらのコンポーネントによって提供されるサービスを使用するために、1つ以上のクライアントアプリケーションを利用してもよい。これらのコンポーネントは、ハードウェア、ファームウェア、ソフトウェア、またはそれらの組合せで実現されてもよい。分散型システム1400とは異なり得るさまざまな異なるシステム構成が可能であることが理解されるべきである。図に示す実施形態はこのため、実施形態システムを実現するための分散型シ

10

20

30

40

50

ステムの一例であり、限定的であるよう意図されてはいない。

【0126】

クライアントコンピューティング装置1402、1404、1406、および/または1408は、携帯型ハンドヘルド装置（たとえば、iPhone（登録商標）、携帯電話、iPad（登録商標）、コンピューティングタブレット、携帯情報端末（PDA））、またはウェアラブル装置（たとえば、グーグル・グラス（登録商標）頭部装着型ディスプレイ）であってもよく、マイクロソフト・ウィンドウズ・モバイル（登録商標）などのソフトウェア、および/または、iOS、ウィンドウズ（登録商標）フォン、アンドロイド（登録商標）、ブラックベリー（登録商標）10、パームOSなどのさまざまなモバイルオペレーティングシステムを実行し、インターネット、電子メール、ショートメッセージサービス（SMS）、ブラックベリー（登録商標）、または他の通信プロトコルに対応している。クライアントコンピューティング装置1402、1404、1406、および/または1408は、マイクロソフト・ウィンドウズ（登録商標）、アップル・マッキントッシュ（登録商標）、および/またはLinux（登録商標）オペレーティングシステムのさまざまなバージョンを実行するパーソナルコンピュータおよび/またはラップトップコンピュータを例として含む、汎用パーソナルコンピュータであり得る。クライアントコンピューティング装置1402、1404、1406、および/または1408は、たとえばグーグル・クロームOSなどのさまざまなGNU/Linuxオペレーティングシステムを何ら限定されることなく含む、商業的に入手可能なさまざまなUNIX（登録商標）またはUNIX様オペレーティングシステムのうちのいずれかを実行するワークステーションコンピュータであり得る。それに代えて、またはそれに加えて、クライアントコンピューティング装置1402、1404、1406、および1408は、ネットワーク1410を通して通信可能である、シンクライアントコンピュータ、インターネット対応ゲーミングシステム（たとえば、Kinect（登録商標）ジェスチャー入力装置を有する、または有さない、マイクロソフトXboxゲーミングコンソール）、および/またはパーソナルメッセージング装置といった、任意の他の電子装置であってもよい。

【0127】

例示的な分散型システム1400は4つのクライアントコンピューティング装置を有して図示されているが、任意の数のクライアントコンピューティング装置がサポートされてもよい。センサを有する装置などの他の装置が、サーバ1412とやりとりしてもよい。

【0128】

分散型システム1400におけるネットワーク1410は、TCP/IP（伝送制御プロトコル/インターネットプロトコル）、SNA（システムネットワークアーキテクチャ）、IPX（インターネットパケット交換）、アップル・トークなどを何ら限定されることなく含む、商業的に入手可能なさまざまなプロトコルのうちのいずれかを使用してデータ通信をサポートできる、当業者にはよく知られた任意のタイプのネットワークであってもよい。単なる例として、ネットワーク1410は、イーサネット（登録商標）、トークンリングなどに基づくものといった、ローカルエリアネットワーク（LAN）であり得る。ネットワーク1410は、ワイドエリアネットワークおよびインターネットであり得る。それは、仮想プライベートネットワーク（VPN）を何ら限定されることなく含む仮想ネットワーク、イントラネット、エクストラネット、公衆交換電話網（PSTN）、赤外線ネットワーク、無線ネットワーク（たとえば、電気電子技術者協会（IEEE）802.11プロトコルスイート、Bluetooth（登録商標）、および/または任意の他の無線プロトコルのうちのいずれかの下で動作するネットワーク）、ならびに/もしくは、これらのおよび/または他のネットワークの任意の組合せを含み得る。

【0129】

サーバ1412は、1つ以上の汎用コンピュータ、専用サーバコンピュータ（PC（パーソナルコンピュータ）サーバ、UNIX（登録商標）サーバ、ミッドレンジサーバ、メインフレームコンピュータ、ラックマウントサーバなどを例として含む）、サーバファーム、サーバクラスタ、もしくは任意の他の適切な構成および/または組合せで構成されて

もよい。さまざまな実施形態では、サーバ1412は、前述の開示で説明された1つ以上のサービスまたはソフトウェアアプリケーションを実行するように適合されてもよい。たとえば、サーバ1412は、この開示の一実施形態に従った上述の処理を行なうためのサーバに対応していてもよい。

【0130】

サーバ1412は、上述のもののうちのいずれかを含むオペレーティングシステム、および商業的に入手可能な任意のサーバオペレーティングシステムを実行してもよい。サーバ1412はまた、さまざまな追加のサーバアプリケーションおよび/または中間層アプリケーションのうちのいずれかを実行してもよく、HTTP（ハイパーテキスト伝送プロトコル）サーバ、FTP（ファイル転送プロトコル）サーバ、CGI（コモンゲートウェイインターフェイス）サーバ、JAVA（登録商標）サーバ、データベースサーバなどを含む。例示的なデータベースサーバは、オラクル、マイクロソフト、サイベース、IBM（国際ナショナル・ビジネス・マシーズ）などから商業的に入手可能なものを何ら限定されることなく含む。

【0131】

いくつかの実現化例では、サーバ1412は、クライアントコンピューティング装置1402、1404、1406、および1408のユーザから受信されたデータフィールドおよび/またはイベント更新を分析して統合するための1つ以上のアプリケーションを含んでもよい。一例として、データフィールドおよび/またはイベント更新は、センサデータアプリケーション、金融ティッカー、ネットワーク性能測定ツール（たとえば、ネットワーク監視およびトラフィック管理アプリケーション）、クリックストリーム分析ツール、自動車交通監視などに関連するリアルタイムイベントを含み得る、1つ以上の第三者情報源および連続データストリームから受信されたツイッター（登録商標）フィード、フェイスブック（登録商標）更新またはリアルタイム更新を含んでもよいが、それらに限定されない。サーバ1412はまた、クライアントコンピューティング装置1402、1404、1406、および1408の1つ以上の表示装置を介してデータフィールドおよび/またはリアルタイムイベントを表示するための1つ以上のアプリケーションを含んでもよい。

【0132】

分散型システム1400はまた、1つ以上のデータベース1414および1416を含んでもよい。データベース1414および1416は、さまざまな位置に存在していてもよい。例として、データベース1414および1416のうちの1つ以上は、サーバ1412に対してローカルな（および/または、サーバ1412内にある）非一時的記憶媒体上に存在していてもよい。それに代えて、データベース1414および1416は、サーバ1412からリモートであってもよく、ネットワークベースの接続または専用接続を介してサーバ1412と通信してもよい。一組の実施形態では、データベース1414および1416は、ストレージエリアネットワーク（storage-area network：SAN）に存在していてもよい。同様に、サーバ1412に帰する機能を行なうための任意の必要なファイルが適宜、サーバ1412上にローカルに格納されてもよく、および/またはリモートに格納されてもよい。一組の実施形態では、データベース1414および1416は、SQLフォーマットのコマンドに回答してデータを格納し、更新し、検索するように適合された、オラクルによって提供されるデータベースなどのリレーショナルデータベースを含んでもよい。

【0133】

図15は、この発明のさまざまな実施形態が実現され得る例示的なコンピュータシステム1500を示す。システム1500は、上述のコンピュータシステムのうちのいずれかを實現するために使用されてもよい。図15に示すように、コンピュータシステム1500は、バスサブシステム1502と、バスサブシステム1502を介して多くの周辺サブシステムと通信する処理部1504とを含む。これらの周辺サブシステムは、処理加速部1506と、I/Oサブシステム1508と、記憶サブシステム1518と、通信サブシ

10

20

30

40

50

ステム 1 5 2 4 とを含んでいてもよい。記憶サブシステム 1 5 1 8 は、有形のコンピュータ読取可能記憶媒体 1 5 2 2 と、システムメモリ 1 5 1 0 とを含む。

【 0 1 3 4 】

バスサブシステム 1 5 0 2 は、コンピュータシステム 1 5 0 0 のさまざまなコンポーネントおよびサブシステムを意図されるように互いに通信させるためのメカニズムを提供する。バスサブシステム 1 5 0 2 は単一のバスとして概略的に図示されているが、バスサブシステムの代替的な実施形態は複数のバスを利用してもよい。バスサブシステム 1 5 0 2 は、さまざまなバスアーキテクチャのうちのいずれかを使用するメモリバスまたはメモリコントローラ、周辺バス、およびローカルバスを含む、いくつかのタイプのバス構造のうちのいずれかであってもよい。たとえば、そのようなアーキテクチャは、IEEE P 1 3 8 6 . 1 規格で製造されるメザニンバスとして実現可能な、産業標準アーキテクチャ (Industry Standard Architecture : I S A) バス、マイクロチャネルアーキテクチャ (Micro Channel Architecture : M C A) バス、強化 I S A (E I S A) バス、ビデオエレクトロニクス標準組織 (Video Electronics Standards Association : V E S A) ローカルバス、および周辺コンポーネント相互接続 (Peripheral Component Interconnect : P C I) バスを含んでいてもよい。

【 0 1 3 5 】

1 つ以上の集積回路 (たとえば、従来のマイクロプロセッサまたはマイクロコントローラ) として実現され得る処理部 1 5 0 4 は、コンピュータシステム 1 5 0 0 の動作を制御する。処理部 1 5 0 4 には、1 つ以上のプロセッサが含まれていてもよい。これらのプロセッサは、シングルコアまたはマルチコアプロセッサを含んでいてもよい。ある実施形態では、処理部 1 5 0 4 は、各処理部にシングルまたはマルチコアプロセッサが含まれた、1 つ以上の独立した処理部 1 5 3 2 および / または 1 5 3 4 として実現されてもよい。他の実施形態では、処理部 1 5 0 4 はまた、2 つのデュアルコアプロセッサをシングルチップへと集積することによって形成されるクアッドコア処理部として実現されてもよい。

【 0 1 3 6 】

さまざまな実施形態では、処理部 1 5 0 4 は、プログラムコードに応答してさまざまなプログラムを実行でき、同時に実行される複数のプログラムまたはプロセスを維持できる。任意の所与の時間において、実行されるべきプログラムコードのうちのいくつかまたはすべては、プロセッサ 1 5 0 4 に、および / または記憶サブシステム 1 5 1 8 にあり得る。好適なプログラミングを通して、プロセッサ 1 5 0 4 は、上述のさまざまな機能性を提供できる。コンピュータシステム 1 5 0 0 は加えて処理加速部 1 5 0 6 を含んでいてもよく、それは、デジタル信号プロセッサ (digital signal processor : D S P)、専用プロセッサなどを含み得る。

【 0 1 3 7 】

I / O サブシステム 1 5 0 8 は、ユーザインターフェイス入力装置と、ユーザインターフェイス出力装置とを含んでいてもよい。ユーザインターフェイス入力装置は、キーボード、マウスまたはトラックボールなどのポインティング装置、ディスプレイに組込まれたタッチパッドまたはタッチスクリーン、スクロールホイール、クリックホイール、ダイヤル、ボタン、スイッチ、キーパッド、音声コマンド認識システム付き音声入力装置、マイクロホン、および他のタイプの入力装置を含んでいてもよい。ユーザインターフェイス入力装置は、たとえば、ジェスチャーおよび口頭コマンドを使用したナチュラルユーザインターフェイスを通して、マイクロソフト X b o x (登録商標) 3 6 0 ゲームコントローラなどの入力装置をユーザが制御し、それとやりとりすることを可能にする、マイクロソフト K i n e c t (登録商標) 運動センサなどの運動感知および / またはジェスチャー認識装置を含んでいてもよい。ユーザインターフェイス入力装置はまた、ユーザから目の活動 (たとえば、写真撮影中および / またはメニュー選択中の「まばたき」) を検出し、アイジェスチャーを入力装置 (たとえば、グーグル・グラス (登録商標)) への入力として変換する、グーグル・グラス (登録商標) まばたき検出器などのアイジェスチャー認識装置を含んでいてもよい。加えて、ユーザインターフェイス入力装置は、ユーザが音声コマン

ドを通して音声認識システム（たとえば、S i r i（登録商標）ナビゲータ）とやりとりできるようにする音声認識感知装置を含んでいてもよい。

【0138】

ユーザインターフェイス入力装置はまた、3次元（3D）マウス、ジョイスティックまたはポインティングスティック、ゲームパッドおよびグラフィックタブレット、ならびに、スピーカ、デジタルカメラ、デジタルビデオカメラ、携帯型メディアプレイヤー、ウェブカメラ、画像スキャナ、指紋スキャナ、バーコードリーダ3Dスキャナ、3Dプリンタ、レーザー測距器、および視線追跡装置などの音声／視覚装置を、何ら限定されることなく含んでいてもよい。加えて、ユーザインターフェイス入力装置は、たとえば、コンピュータ断層撮影装置、磁気共鳴撮像装置、ポジトロン放出断層撮影装置、医療用超音波検査装置などの医療用撮像入力装置を含んでいてもよい。ユーザインターフェイス入力装置はまた、たとえば、M I D Iキーボード、デジタル楽器などの音声入力装置を含んでいてもよい。

10

【0139】

ユーザインターフェイス出力装置は、表示サブシステム、表示灯、または、音声出力装置などの非視覚的ディスプレイを含んでいてもよい。表示サブシステムは、陰極線管（cathode ray tube：C R T）、液晶ディスプレイ（liquid crystal display：L C D）またはプラズマディスプレイを使用するものなどのフラットパネル装置、投影装置、タッチスクリーンなどであってもよい。一般に、「出力装置」という用語の使用は、コンピュータシステム1500からユーザまたは他のコンピュータに情報を出力するためのあらゆる可能なタイプの装置およびメカニズムを含むよう意図されている。たとえば、ユーザインターフェイス出力装置は、モニタ、プリンタ、スピーカ、ヘッドホン、自動車ナビゲーションシステム、プロッタ、音声出力装置、およびモデムといった、テキスト、グラフィックスおよび音声／映像情報を視覚的に伝えるさまざまな表示装置を、何ら限定されることなく含んでいてもよい。

20

【0140】

コンピュータシステム1500は、現在システムメモリ1510内に位置するとして図示されたソフトウェア要素を含む記憶サブシステム1518を含んでいてもよい。システムメモリ1510は、処理部1504上でロード可能および実行可能なプログラム命令と、これらのプログラムの実行中に生成されたデータとを格納してもよい。

30

【0141】

コンピュータシステム1500の構成およびタイプに依存して、システムメモリ1510は揮発性（ランダムアクセスメモリ（random access memory：R A M）など）であってもよく、および／または不揮発性（読出専用メモリ（read-only memory：R O M）、フラッシュメモリなど）であってもよい。R A Mは典型的には、処理部1504に直ちにアクセス可能であり、および／または処理部1504によって現在動作および実行中のデータおよび／またはプログラムモジュールを含む。いくつかの実現化例では、システムメモリ1510は、スタティックランダムアクセスメモリ（static random access memory：S R A M）またはダイナミックランダムアクセスメモリ（dynamic random access memory：D R A M）などの複数の異なるタイプのメモリを含んでいてもよい。いくつかの実現化例では、起動中などにコンピュータシステム1500内の要素間で情報を転送するのに役立つ基本ルーチンを含む基本入力／出力システム（basic input/output system：B I O S）が、典型的にはR O Mに格納されていてもよい。限定のためではなく例として、システムメモリ1510はまた、クライアントアプリケーション、ウェブブラウザ、中央層アプリケーション、リレーショナルデータベース管理システム（relational database management system：R D B M S）などを含み得るアプリケーションプログラム1512と、プログラムデータ1514と、オペレーティングシステム1516とを示す。例として、オペレーティングシステム1516は、マイクロソフト・ウィンドウズ（登録商標）、アップル・マッキントッシュ（登録商標）、および／またはL i n u xオペレーティングシステムのさまざまなバージョン、商業的に入手可能なさまざまなU N I X（登録商標）また

40

50

はUNIX様オペレーティングシステム（さまざまなGNU/Linuxオペレーティングシステム、グーグル・クローム（登録商標）OSなどを何ら限定されることなく含む）、および/または、iOS、ウィンドウズ（登録商標）フォン、アンドロイド（登録商標）OS、ブラックベリー（登録商標）10OS、パーム（登録商標）OSオペレーティングシステムなどのモバイルオペレーティングシステムを含んでいてもよい。

【0142】

記憶サブシステム1518はまた、いくつかの実施形態の機能性を提供する基本プログラミングおよびデータ構造を格納するための有形のコンピュータ読取可能記憶媒体を提供してもよい。プロセッサによって実行されると上述の機能性を提供するソフトウェア（プログラム、コードモジュール、命令）が、記憶サブシステム1518に格納されてもよい。これらのソフトウェアモジュールまたは命令は、処理部1504によって実行されてもよい。記憶サブシステム1518はまた、この発明に従って使用されるデータを格納するためのリポジトリを提供してもよい。

10

【0143】

記憶サブシステム1500はまた、コンピュータ読取可能記憶媒体1522にさらに接続され得るコンピュータ読取可能記憶媒体リーダ1520を含んでいてもよい。システムメモリ1510とともに、およびオプションでシステムメモリ1510と組合わされて、コンピュータ読取可能記憶媒体1522は、リモート、ローカル、固定および/またはリムーバブルの記憶装置に加えて、コンピュータ読取可能情報を一時的におよび/またはより永続的に含み、格納し、送信し、検索するための記憶媒体を包括的に表わしてもよい。

20

【0144】

コードまたはコードの一部を含むコンピュータ読取可能記憶媒体1522はまた、情報の格納および/または送信のためのあらゆる方法または技術で実現される揮発性および不揮発性でリムーバブルおよび非リムーバブルの媒体を含むがそれらに限定されない記憶媒体および通信媒体を含む、当該技術分野において公知であるかまたは使用されるあらゆる適切な媒体を含み得る。これは、RAM、ROM、電子的消去可能プログラマブルROM（electronically erasable programmable ROM：EEPROM）、フラッシュメモリまたは他のメモリ技術、CD-ROM、デジタル多用途ディスク（digital versatile disk：DVD）または他の光学ストレージ、磁気カセット、磁気テープ、磁気ディスクストレージまたは他の磁気記憶装置、もしくは他の有形のコンピュータ読取可能媒体といった、有形のコンピュータ読取可能記憶媒体を含み得る。これは、所望の情報を送信するために使用可能であり、コンピューティングシステム1500によってアクセスされ得るデータ信号、データ送信、または任意の他の媒体といった、非有形のコンピュータ読取可能媒体も含み得る。

30

【0145】

例として、コンピュータ読取可能記憶媒体1522は、非リムーバブルで不揮発性の磁気媒体から読出し、またはそれに書込むハードディスクドライブ、リムーバブルで不揮発性の磁気ディスクから読出し、またはそれに書込む磁気ディスクドライブ、ならびに、CD-ROM、DVD、およびBlu-Ray（登録商標）ディスク、または他の光学媒体といった、リムーバブルで不揮発性の光ディスクから読出し、またはそれに書込む光ディスクドライブを含んでいてもよい。コンピュータ読取可能記憶媒体1522は、Zip（登録商標）ドライブ、フラッシュメモリカード、ユニバーサルシリアルバス（universal serial bus：USB）フラッシュドライブ、セキュアデジタル（secure digital SD）カード、DVDディスク、デジタルビデオテープなどを含んでいてもよいが、それらに限定されない。コンピュータ読取可能記憶媒体1522はまた、フラッシュメモリベースのソリッドステートドライブ（solid-state drive：SSD）、企業フラッシュドライブ、ソリッドステートROMといった、不揮発性メモリに基づいたSSD、ソリッドステートRAM、ダイナミックRAM、スタティックRAM、DRAMベースのSSD、磁気抵抗RAM（MRAM）SSDといった、揮発性メモリに基づいたSSD、および、DRAMベースのSSDとフラッシュメモリベースのSSDとの組合せを使用するハイブリッドSS

40

50

Dを含んでいてもよい。ディスクドライブおよびそれらの関連付けられたコンピュータ読取可能媒体は、コンピュータ読取可能命令、データ構造、プログラムモジュールおよび他のデータの不揮発性ストレージをコンピュータシステム 1500 に提供してもよい。

【0146】

通信サブシステム 1524 は、他のコンピュータシステムおよびネットワークへのインターフェイスを提供する。通信サブシステム 1524 は、コンピュータシステム 1500 とは別のシステムからデータを受信し、別のシステムにデータを送信するためのインターフェイスとして機能する。たとえば、通信サブシステム 1524 は、コンピュータシステム 1500 がインターネットを介して 1 つ以上の装置に接続できるようにしてもよい。いくつかの実施形態では、通信サブシステム 1524 は、（たとえば、3 G、4 G、または E D G E (enhanced data rates for global evolution: エンハンスド・データレート・フォー・グローバル・エボリューション)、W i F i (I E E E 8 0 2 . 1 1 ファミリー規格)、または他の移動通信技術、またはそれらの任意の組合せといった携帯電話技術、高度なデータネットワーク技術を使用した) 無線音声および/またはデータネットワークにアクセスするための無線周波数 (radio frequency: R F) トランシーバコンポーネント、全地球測位システム (global positioning system: G P S) 受信機コンポーネント、および/または他のコンポーネントを含み得る。いくつかの実施形態では、通信サブシステム 1524 は、無線インターフェイスに加えて、またはその代わりに、有線ネットワーク接続（たとえば、イーサネット）を提供できる。

【0147】

いくつかの実施形態では、通信サブシステム 1524 はまた、コンピュータシステム 1500 を使用し得る 1 人以上のユーザのために、構造化および/または非構造化データフィード 1526、イベントストリーム 1528、イベント更新 1530 などの形をした入力通信を受信してもよい。

【0148】

例として、通信サブシステム 1524 は、ツイッター（登録商標）フィード、フェイスブック（登録商標）更新、リッチ・サイト・サマリー (Rich Site Summary: R S S) フィードなどのウェブフィード、および/または 1 つ以上の第三者情報源からのリアルタイム更新といった、ソーシャルネットワークおよび/または他の通信サービスのユーザからのデータフィード 1526 をリアルタイムで受信するように構成されてもよい。

【0149】

加えて、通信サブシステム 1524 はまた、リアルタイムイベントのイベントストリーム 1528 および/またはイベント更新 1530 を含み得る、明確な終わりがなく本質的に連続的または無限であり得る連続データストリームの形をしたデータを受信するように構成されてもよい。連続データを生成するアプリケーションの例は、たとえば、センサデータアプリケーション、金融ティッカー、ネットワーク性能測定ツール（たとえば、ネットワーク監視およびトラフィック管理アプリケーション）、クリックストリーム分析ツール、自動車交通監視などを含んでいてもよい。

【0150】

通信サブシステム 1524 はまた、構造化および/または非構造化データフィード 1526、イベントストリーム 1528、イベント更新 1530 などを、コンピュータシステム 1500 に結合された 1 つ以上のストリーミングデータソースコンピュータと通信し得る 1 つ以上のデータベースに出力するように構成されてもよい。

【0151】

コンピュータシステム 1500 は、ハンドヘルド携帯装置（たとえば、i P h o n e (登録商標) 携帯電話、i P a d (登録商標) コンピューティングタブレット、P D A)、ウェアラブル装置（たとえば、グーグル・グラス (登録商標) 頭部装着型ディスプレイ)、P C、ワークステーション、メインフレーム、キオスク、サーバラック、または任意の他のデータ処理システムを含む、さまざまなタイプのうちの 1 つであり得る。

【0152】

コンピュータおよびネットワークの絶えず変化する性質により、図に示すコンピュータシステム 1500 の説明は、単に特定の一例として意図される。図に示すシステムよりも多い、または少ないコンポーネントを有する多くの他の構成が可能である。たとえば、カスタマイズされたハードウェアも使用されてもよく、および/または、特定の要素が、ハードウェア、ファームウェア、ソフトウェア（アプレットを含む）、または組合せで実現されてもよい。また、ネットワーク入力/出力装置などの他のコンピューティング装置への接続が採用されてもよい。ここに提供される開示および教示に基づいて、当業者であれば、さまざまな実施形態を実現するための他のやり方および/または方法を理解するであろう。

【0153】

10

前述の明細書では、この発明の局面をその特定の実施形態を参照して説明したが、当業者であれば、この発明がそれに限定されないことを認識するであろう。上術の発明のさまざまな特徴および局面は、個々にまたは共同で使用されてもよい。また、実施形態は、明細書のより広範な精神および範囲から逸脱することなく、ここに記載されたものを超える任意の数の環境および用途において利用され得る。したがって、明細書および図面は、限定的でなく例示的であると見なされるべきである。

【0154】

その教示がこの開示において提示され得る 1 つ以上の発明のうちのいずれかのさまざまな実施形態が、ソフトウェア、ファームウェア、ハードウェア、またはそれらの組合せにおけるロジックの形で実現され得る。ロジックは、この開示において提示された発明のさまざまな実施形態において開示され得る一組のステップを行なうように、ロジックマシンの中央処理装置（CPU またはプロセッサ）に指示するように適合された一組の命令として、マシンアクセス可能なメモリ、マシン読取可能な物品、有形のコンピュータ読取可能媒体、コンピュータ読取可能記憶媒体、または他のコンピュータ/マシン読取可能媒体に格納されてもよい。ロジックは、コードモジュールが、この開示において提示された発明のさまざまな実施形態において方法またはプロセスを行なうために実行される際に、コンピュータシステムまたは情報処理装置のプロセッサにより動作可能になると、ソフトウェアプログラムまたはコンピュータプログラム製品の一部を形成してもよい。ここに提供されたこの開示および教示に基づいて、当業者であれば、提示された発明のうちの 1 つ以上におけるさまざまな実施形態の開示された動作または機能性のいずれかを、ソフトウェア、ファームウェア、ハードウェア、またはそれらの組合せで実現するための他のやり方、変形例、修正例、代替例、および/または方法を理解するであろう。

20

30

【0155】

その教示がこの開示において提示され得るそれらの発明のうちのいずれか 1 つの開示された例、実現化例、およびさまざまな実施形態は、この開示の教示を適正な明確性で当業者に伝えるように例示的であるに過ぎない。これらの実現化例および実施形態は例示的な図解または特定の図を参照して説明され得るため、説明された方法および/または特定の構造のさまざまな修正例または適合例が、当業者には明らかとなり得る。ここに見出されたこの開示およびこれらの教示に依存し、かつ、教示が技術を進歩させた際に経た、すべてのそのような修正例、適合例、または変形例は、その教示がこの開示において提示され得る 1 つ以上の発明の範囲内にあると考えられるべきである。よって、開示において提示された発明は、具体的に示された実施形態にまったく限定されないということが理解されるため、本説明および図面は限定的な意味で考えられるべきでない。

40

【0156】

したがって、上述の説明およびあらゆる添付図面、図解、および図は、限定的ではなく例示的であるよう意図されている。この開示において提示されたあらゆる発明の範囲はしたがって、上述の説明および図に示されたそれらの実施形態を単純に参照するのではなく、係属中の請求項をそれらの完全な範囲または均等物とともに参照して、判断されるべきである。

【図 1】

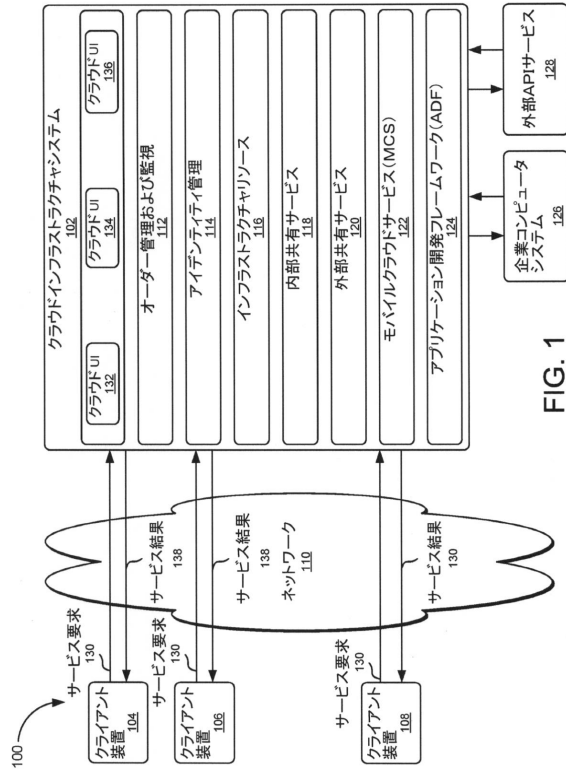


FIG. 1

【図 2】

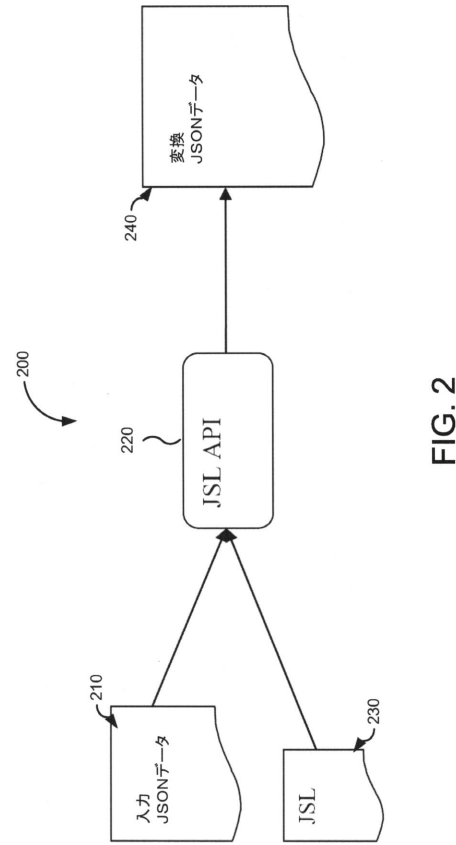


FIG. 2

【図 3】

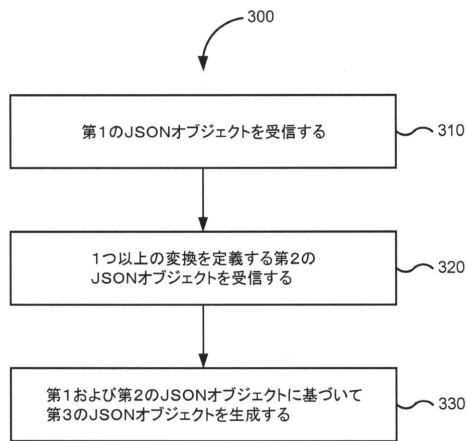


FIG. 3

【図 4】

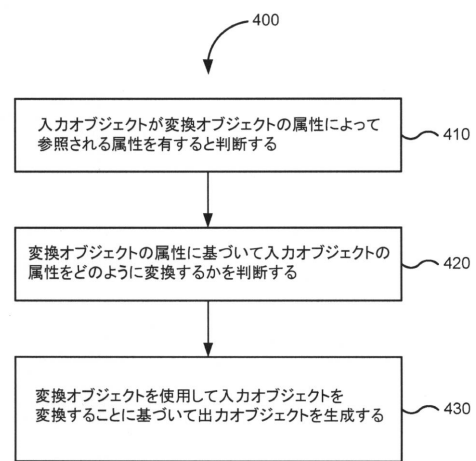


FIG. 4

【図 5】

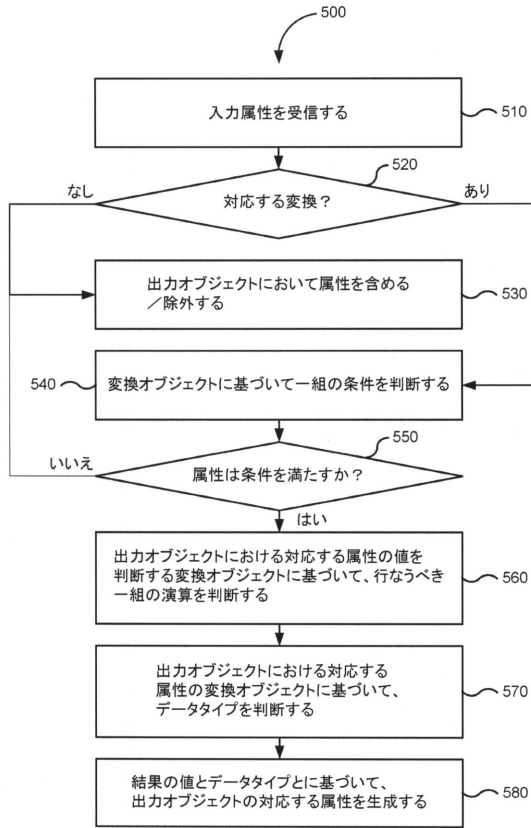


FIG. 5

【図 6】

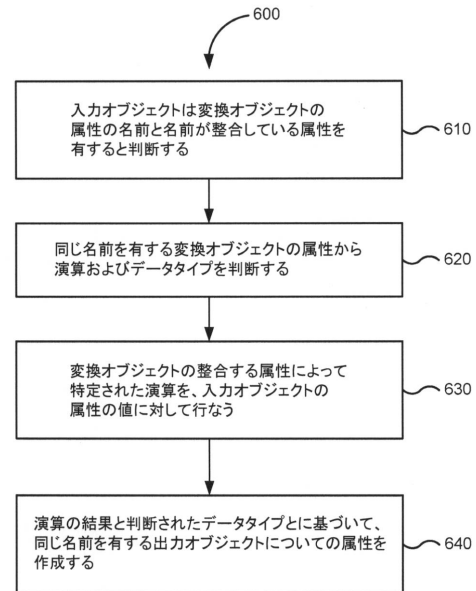


FIG. 6

【図 7】

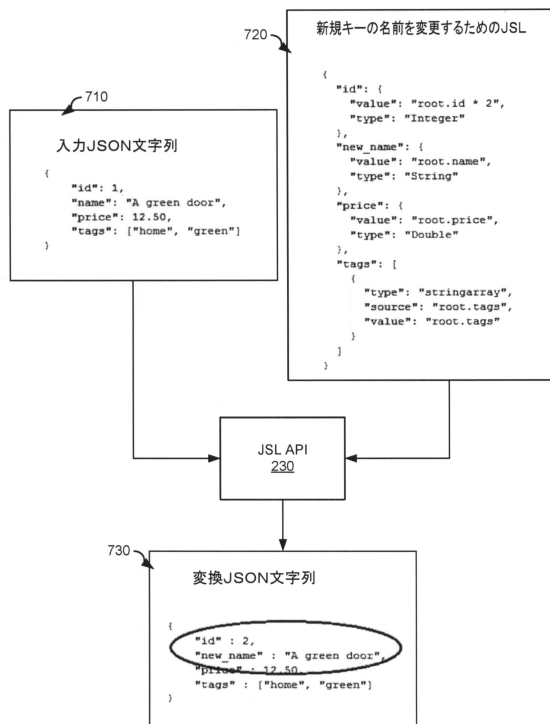


FIG. 7

【図 8】

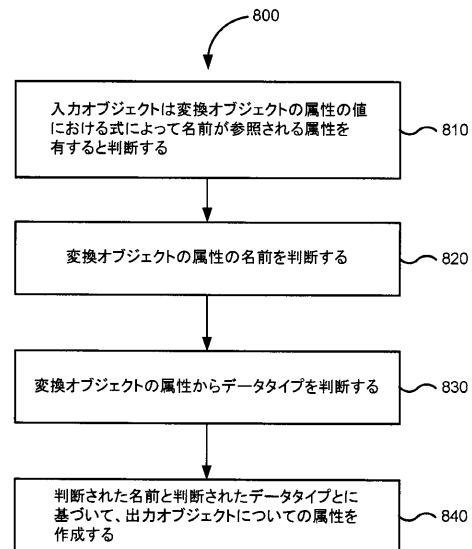


FIG. 8

【図 9】

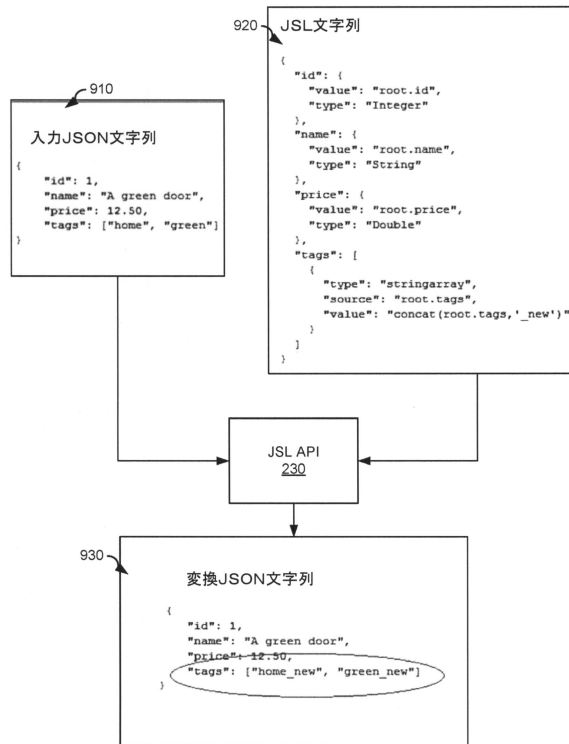


FIG. 9

【図 10】

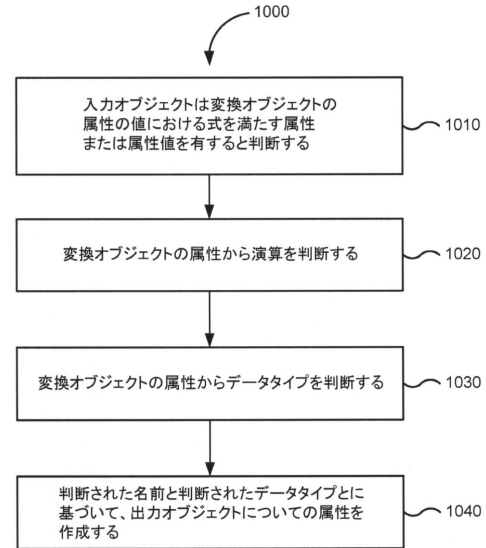


FIG. 10

【図 11】

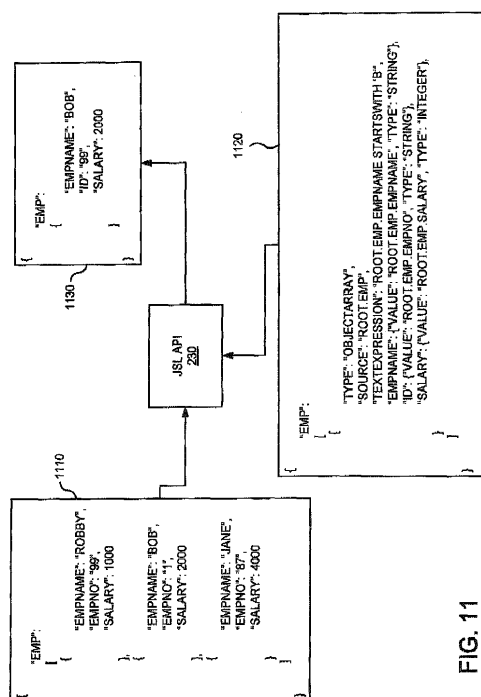


FIG. 11

【図 12】

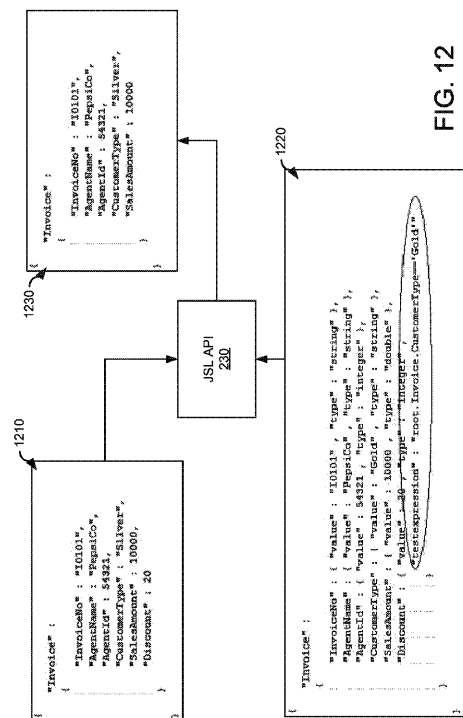


FIG. 12

【図 13】

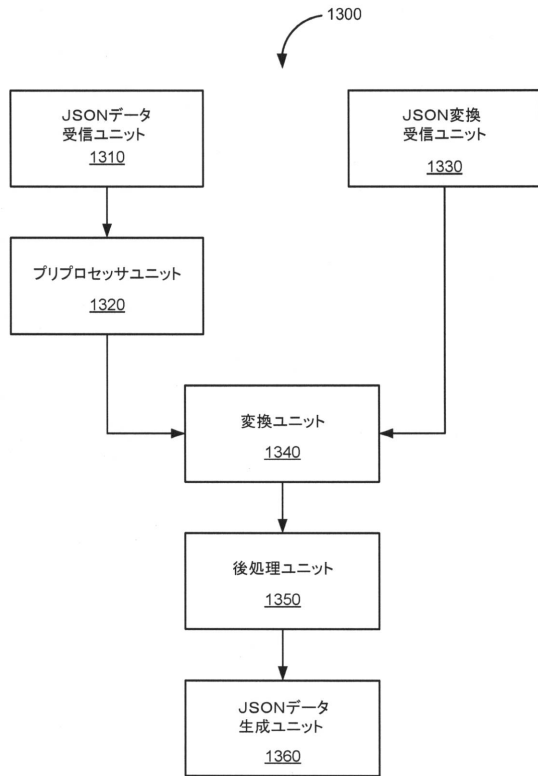


FIG. 13

【図 14】

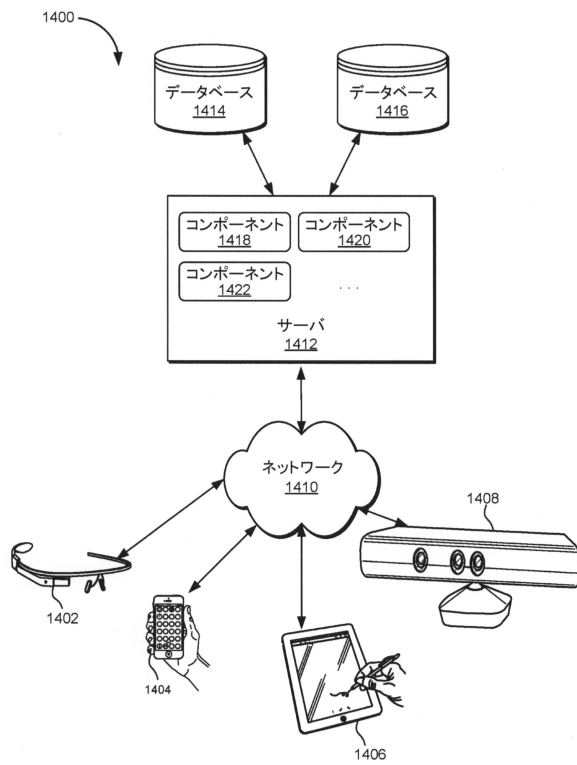


FIG. 14

【図 15】

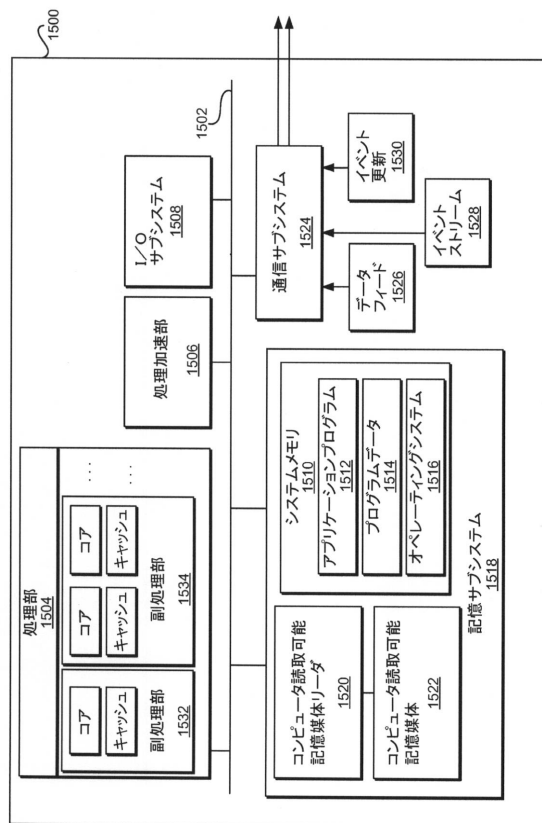


FIG. 15

フロントページの続き

(72)発明者 シンハ, ニーシャ

インド、5 6 0 0 3 7 カルナータカ、バンガロール、ムルゲシュバラヤ、チャラガッタ・ピレッジ、シュリラム・スパンダナ、エス・エフ・1 0 4

審査官 塚田 肇

(56)参考文献 特開2 0 0 5 - 1 2 8 7 1 6 (J P , A)

特開2 0 1 0 - 2 1 1 5 6 7 (J P , A)

特開2 0 1 1 - 1 8 0 8 0 9 (J P , A)

中道 義之, 今日から始めるX S L T入門, XML P R E S S , 2 0 0 0年1 2月2 5日, 第2巻, pp.196-197

(58)調査した分野(Int.Cl. , D B名)

G 0 6 F 9 / 4 4 8

G 0 6 F 1 6 / 1 7

G 0 6 F 4 0 / 1 5 4