

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6541647号
(P6541647)

(45) 発行日 令和1年7月10日(2019.7.10)

(24) 登録日 令和1年6月21日(2019.6.21)

(51) Int.Cl.

F 1

G06F 9/451 (2018.01)
G06F 8/38 (2018.01)G06F 9/451
G06F 8/38

請求項の数 16 (全 35 頁)

(21) 出願番号 特願2016-515519 (P2016-515519)
 (86) (22) 出願日 平成26年8月28日 (2014.8.28)
 (65) 公表番号 特表2016-533547 (P2016-533547A)
 (43) 公表日 平成28年10月27日 (2016.10.27)
 (86) 國際出願番号 PCT/US2014/053138
 (87) 國際公開番号 WO2015/041829
 (87) 國際公開日 平成27年3月26日 (2015.3.26)
 審査請求日 平成29年8月2日 (2017.8.2)
 (31) 優先権主張番号 61/880,767
 (32) 優先日 平成25年9月20日 (2013.9.20)
 (33) 優先権主張国 米国(US)
 (31) 優先権主張番号 61/909,205
 (32) 優先日 平成25年11月26日 (2013.11.26)
 (33) 優先権主張国 米国(US)

(73) 特許権者 502303739
 オラクル・インターナショナル・コーポレーション
 アメリカ合衆国カリフォルニア州94065レッドウッド・シティ、オラクル・パーカウェイ500
 (74) 代理人 110001195
 特許業務法人深見特許事務所
 (72) 発明者 ゴン、イーシュエン
 アメリカ合衆国、94065 カリフォルニア州、レッドウッド・シティ、シェル・パークウェイ、585、アパートメント・5211

最終頁に続く

(54) 【発明の名称】 ランタイムカスタマイゼーションインフラストラクチャ

(57) 【特許請求の範囲】

【請求項 1】

ウェブベースの複数のレイヤーからなるアプリケーションのためのユーザインターフェイスとして機能するデスクトップアプリケーションのドキュメントを作成するための方法であって、

コンピュータシステムのうちのサーバコンピュータにてドキュメントを作成することと、

前記ドキュメントについてのメタデータに関連付けられるルールのセットを前記サーバコンピュータによって決定することとを含み、前記メタデータは、前記ドキュメントに含まれる各コンポーネント、および、前記複数のレイヤーのうちの一つのレイヤーにおいて利用可能なデータにコンポーネントがどのように結合されるかを示し、前記方法はさらに、

ドキュメントの初期化の際に前記コンピュータシステムのうちのクライアントコンピュータから前記サーバコンピュータに与えられる要求に応答して、当該サーバコンピュータにおいて抽出されたメタデータに、当該ドキュメントに含まれる項目の要否に応じたカスタマイゼーションを適用することにより、前記ドキュメントのユーザのためにカスタマイズされたメタデータを前記サーバコンピュータによって生成することと、

前記ドキュメントが前記1つ以上のウェブベースのアプリケーションのためのユーザインターフェイスとして機能するように、前記サーバコンピュータから前記クライアントコンピュータに前記ドキュメントについての前記カスタマイズされたメタデータを送信する

10

20

こととを含む、方法。

【請求項 2】

前記ドキュメントについての前記メタデータに関連付けられる前記ルールのセットを前記サーバコンピュータによって決定することは、使用するべき前記ドキュメントについての前記メタデータのバージョンを規定する少なくとも1つのルールを決定することを含む、請求項1に記載の方法。

【請求項 3】

前記ドキュメントについての前記メタデータに関連付けられる前記ルールのセットを前記サーバコンピュータによって決定することは、前記ドキュメントのユーザに関連付けられる役割に基づいて、使用するべき前記ドキュメントについての前記メタデータを規定する少なくとも1つのルールを決定することを含む、請求項1に記載の方法。

10

【請求項 4】

前記ドキュメントについての前記メタデータに関連付けられる前記ルールのセットを前記サーバコンピュータによって決定することは、時間または日付情報に基づいて、使用するべき前記ドキュメントについて前記メタデータを規定する少なくとも1つのルールを決定することを含む、請求項1～3のいずれか1項に記載の方法。

【請求項 5】

前記ドキュメントについての前記メタデータに関連付けられる前記ルールのセットを前記サーバコンピュータによって決定することは、組織の部門に基づいて、使用するべき前記ドキュメントについての前記メタデータを規定する少なくとも1つのルールを決定することを含む、請求項1～4のいずれか1項に記載の方法。

20

【請求項 6】

前記ドキュメントについての前記メタデータに関連付けられる前記ルールのセットを前記サーバコンピュータによって決定することは、メタデータ管理サーバから1つ以上のルールを抽出することを含む、請求項1～5のいずれか1項に記載の方法。

【請求項 7】

前記ドキュメントについての前記メタデータに適用されるようにカスタマイゼーションを構成する前記ルールの決定されたセットに基づいて、前記ドキュメントについての前記メタデータを前記サーバコンピュータによって生成することは、前記ドキュメントについてのベースメタデータに複数のメタデータカスタマイゼーションを適用することを含む、請求項1～6のいずれか1項に記載の方法。

30

【請求項 8】

ウェブベースの複数のレイヤーからなるアプリケーションのためのユーザインターフェイスとして機能するデスクトップアプリケーションのドキュメントを作成するためのコンピュータ実行可能コードを含むコンピュータ読取可能プログラムであって、

コンピュータシステムのうちのサーバコンピュータにてドキュメントを作成するためのコードと、

前記ドキュメントについてのメタデータに関連付けられるルールのセットを前記サーバコンピュータによって決定するためのコードとを含み、前記メタデータは、前記ドキュメントに含まれる各コンポーネント、および、前記複数のレイヤーのうちの一つのレイヤーにおいて利用可能なデータにコンポーネントがどのように結合されるかを示し、さらに、

40

ドキュメントの初期化の際に前記コンピュータシステムのうちのクライアントコンピュータから前記サーバコンピュータに与えられる要求に応答して、当該サーバコンピュータにおいて抽出されたメタデータに、当該ドキュメントに含まれる項目の要否に応じたカスタマイゼーションを適用することにより、前記ドキュメントのユーザのためにカスタマイズされたメタデータを前記サーバコンピュータによって生成するためのコードと、

前記ドキュメントが前記1つ以上のウェブベースのアプリケーションのためのユーザインターフェイスとして機能するように、前記サーバコンピュータから前記クライアントコンピュータに前記ドキュメントについての前記カスタマイズされたメタデータを送信するためのコードとを含む、コンピュータ読取可能プログラム。

50

【請求項 9】

請求項 1 ~ 7 のいずれか 1 項に記載の方法をコンピュータシステムに実行させるためのコンピュータ実行可能コードを含むコンピュータ読み取り可能プログラム。

【請求項 10】

ウェブベースの複数のレイヤーからなるアプリケーションのためのユーザインターフェイスとして機能するデスクトップアプリケーションのドキュメントを作成するためのシステムであって、

プロセッサと、

指示のセットを格納するメモリとを含み、

前記指示のセットは、前記プロセッサによって実行されると、

10

ドキュメントを作成し、

前記ドキュメントについてのメタデータに関連付けられるルールのセットを決定するように前記プロセッサを構成し、前記メタデータは、前記ドキュメントに含まれる各コンポーネント、および、前記複数のレイヤーのうちの一つのレイヤーにおいて利用可能なデータにコンポーネントがどのように結合されるかを示し、さらに

ドキュメントの初期化の際にクライアントコンピュータから前記システムに与えられる要求に応答して、当該システムにおいて抽出されたメタデータに、当該ドキュメントに含まれる項目の要否に応じたカスタマイゼーションを適用することにより、前記ドキュメントのユーザのためにカスタマイズされたメタデータを生成し、

前記ドキュメントが前記 1 つ以上のウェブベースのアプリケーションのためのユーザインターフェイスとして機能するように、前記クライアントコンピュータに前記ドキュメントについての前記カスタマイズされたメタデータを送信するように前記プロセッサを構成する、システム。

20

【請求項 11】

前記ドキュメントについての前記メタデータに関連付けられる前記ルールのセットを決定するために、前記プロセッサは、使用するべき前記ドキュメントについての前記メタデータのバージョンを規定する少なくとも 1 つのルールを決定するように構成される、請求項 10 に記載のシステム。

【請求項 12】

前記ドキュメントについての前記メタデータに関連付けられる前記ルールのセットを決定するために、前記プロセッサは、前記ドキュメントのユーザに関連付けられる役割に基づいて、使用するべき前記ドキュメントについての前記メタデータを規定する少なくとも 1 つのルールを決定するように構成される、請求項 10 または 11 に記載のシステム。

30

【請求項 13】

前記ドキュメントについての前記メタデータに関連付けられる前記ルールのセットを決定するために、前記プロセッサは、時間または日付情報に基づいて、使用するべき前記ドキュメントについての前記メタデータを規定する少なくとも 1 つのルールを決定するように構成される、請求項 10 ~ 12 のいずれか 1 項に記載のシステム。

【請求項 14】

前記ドキュメントについての前記メタデータに関連付けられる前記ルールのセットを決定するために、前記プロセッサは、組織の部門に基づいて、使用するべき前記ドキュメントについての前記メタデータを規定する少なくとも 1 つのルールを決定するように構成される、請求項 10 ~ 13 のいずれか 1 項に記載のシステム。

40

【請求項 15】

前記ドキュメントについての前記メタデータに関連付けられる前記ルールのセットを決定するために、前記プロセッサは、メタデータ管理サーバから 1 つ以上のルールを抽出するように構成される、請求項 10 ~ 14 のいずれか 1 項に記載のシステム。

【請求項 16】

前記ドキュメントについての前記メタデータに適用されるようにカスタマイゼーションを構成する前記ルールの決定されたセットに基づいて前記ドキュメントについての前記メ

50

タデータを生成するために、前記プロセッサは、前記ドキュメントについてベースメタデータに複数のメタデータカスタマイゼーションを適用するように構成される、請求項10～15のいずれか1項に記載のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願への相互参照

この出願は、

2013年9月20日に出願された、「ランタイムカスタマイゼーションインフラストラクチャ (Runtime Customization Infrastructure)」という名称を有する米国仮出願第61/880,767号と、

2013年11月26日に出願された、「ワークブックコンポーザ (Workbook Composer)」という名称を有する米国仮出願第61/909,205号と、

2014年6月13日に出願された、「ランタイムカスタマイゼーションインフラストラクチャ (Runtime Customization Infrastructure)」という名称を有する米国出願第14/304,356号と、

2014年6月13日に出願された、「ワークブックコンポーザ (Workbook Composer)」という名称を有する米国出願第14/304,393号と

に対する優先権および優先権の利益を請求しており、その開示は本願明細書においてすべての目的について参照により援用される。

【背景技術】

【0002】

発明の背景

アプリケーションは、実行時において特定の所望のタスクを実行するソフトウェアプログラムを指す。一般に、いくつかのアプリケーションは、関連技術において周知のように、オペレーティングシステム、(たとえばJava(登録商標)プログラミング言語をサポートする)バーチャルマシン、デバイスドライバなどのうちの1つ以上を含むランタイム環境において実行される。

【0003】

デベロッパはしばしば、所望のアプリケーションを実装/開発するために、(それら自体がアプリケーションである)アプリケーション開発フレームワーク (ADF: Application Development Framework) を使用する。ADFは、直接的/間接的にアプリケーションの開発において使用され得るあらかじめ定義されたコード/データモジュールのセットを提供する。ADFはさらに、IDE (integrated development environment; 総合開発環境)、コードジェネレータ、デバッガなどといったツールを提供し得、これにより、デベロッパがアプリケーションの所望のロジックをコーディング/実装することをより速く/より簡素な態様で容易にする。

【0004】

一般に、ADFは、たとえば所望のタスクを実行するようコンポーネントを選択し、かつ、当該選択されたコンポーネントの外観、挙動および相互作用を定義することにより、アプリケーションデベロッパがユーザインターフェイスおよびアプリケーションロジックを定義するために使用し得る再使用可能なコンポーネントおよび総合開発環境を提供することによって、アプリケーションの開発を簡素化する。いくつかのADFは、疎結合 (loose coupling) と、より容易なアプリケーション開発およびメンテナンスとを促進するモデル・ビュー・コントローラデザインパターンに基づく。オラクルアプリケーション開発フレームワーク (Oracle Application Development Framework) はこのデザインパターンを利用するADFの一例である。

【0005】

オラクルADFは、ビルトインのHTML5およびAjax機能を有する標準ベースのJavaサーバフェイス (JSF: Java Server Face) コンポーネントのライブラリを含

10

20

30

40

50

む。これらのコンポーネントにより、ウェブで展開されるユーザインターフェイスが、以前にシッククライアントアプリケーションのために確保されたあるレベルの機能および相互作用性を有するよう開発され得る。コンポーネントは、使用が容易なコンポーネントのセットにおいて、データ相互作用、データ可視化、および、カプセル化されたブラウザ側動作を提供し、これまでよりもリッチクライアントアプリケーションの開発が容易になる。オラクル ADF はさらに、IDEにおいて単純なドラッグアンドドロップ動作を通じてビジネスサービスにUIを結合することを簡素化するデータ結合フレームワーク (data-binding framework) を提供する。これは、コンシューミングインターフェイスからのビジネスサービスの独立を保ちつつなされる。このフレームワークにより、UI デベロッパは、ビジネスサービスレイヤーの基礎をなすインプリメンテーションから隔離される。これにより、UI を構築するプロセスは、ビジネスサービスレイヤーのインプリメンテーションから真に切り離され、サービス指向アーキテクチャにおけるインプリメンテーションのためにアプリケーションがよりよく配置される。

【0006】

したがって、望まれるのは、アプリケーション開発フレームワークを使用してアプリケーションユーザインターフェイスを構築することに関連する問題を解決することであり、その問題のいくつかは本願明細書において論じられ得る。さらに、望まれるのは、アプリケーション開発フレームワークを使用してユーザインターフェイスを構築することに関連する障害を低減することであり、そのアプリケーション開発フレームワークのいくつかは本願明細書において論じられ得る。

10

【発明の概要】

【課題を解決するための手段】

【0007】

発明の概要

この開示の以下の部分は、少なくとも主題の基本的な理解を提供する目的のために、この開示において発見される1つ以上のイノベーション、実施形態および/または例の簡素化された概要を提供する。この概要是、如何なる特定の実施形態または例の広範囲な概要も提供することを試みてはいない。さらに、この概要是、実施形態または例の主な/決定的な要素を識別するようには意図されておらず、または、この開示の主題の範囲を定めるようには意図されていない。したがって、この概要の1つの目的は、この開示において発見されるいくつかのイノベーション、実施形態および/または例を簡素化された形で、後述されるさらなる詳細な説明の前置きとして提供することであり得る。

20

【0008】

さまざまな実施形態において、デベロッパが、デスクトップ統合フレームワークを使用して開発されたドキュメントのランタイムカスタマイゼーションを行うことを可能にする方法、システムおよび一時的でないコンピュータ読取可能媒体が開示される。ワークブックメタデータは、所与のワークブックが特定のウェブアプリケーションにどのように統合されるかを記述する情報のセットである。ワークブックが発行されている場合、メタデータは、発行されたワークブックにおけるローカルキャッシュおよびワークブック定義ファイルに書き込まれ得る。メタデータ管理は、発行されたワークブックにおけるローカルキャッシュおよびワークブック定義ファイルから独立して、発行されたワークブックの更新およびカスタマイゼーションを可能にするよう、メタデータサービスによって扱われ得る。

30

【0009】

さまざまな実施形態において、ウェブベースのアプリケーションのためのユーザインターフェイスとして機能するデスクトップアプリケーションのドキュメントを作成するための方法は、そのコンテンツが第2のアプリケーションに関連付けられるネイティブアプリケーションフォーマットで作成されたドキュメントであることを示す情報を第1のアプリケーションから受け取ることを含む。ドキュメントについてのメタデータに関連付けられるルールのセットが決定される。メタデータは、ドキュメントのコンテンツの部分と、第

40

50

1 のアプリケーションと通信する 1 つ以上のウェブベースのアプリケーションによって提供されるデータモデルに関連付けられるデータとの間のリンクを提供する。ドキュメントについてのメタデータに適用されるようにカスタマイゼーションを構成するルールの決定されたセットに基づいて、ドキュメントについてのメタデータが生成される。ドキュメントが 1 つ以上のウェブベースのアプリケーションのためのユーザインターフェイスとして機能するように、第 2 のアプリケーションにドキュメントについてのメタデータが通信される。

【 0 0 1 0 】

ドキュメントについてのメタデータに関連付けられるルールのセットを決定することは、使用するべきドキュメントについてのメタデータのバージョンを規定する少なくとも 1 つのルールを決定することを含み得る。当該ドキュメントについてのメタデータに関連付けられるルールのセットを決定することは、ドキュメントのユーザに関連付けられる役割に基づいて、使用するべきドキュメントについてのメタデータを規定する少なくとも 1 つのルールを決定することを含み得る。ドキュメントについてのメタデータに関連付けられるルールのセットを決定することは、時間または日付情報に基づいて、使用するべきドキュメントについてのメタデータを規定する少なくとも 1 つのルールを決定することを含み得る。ドキュメントについてのメタデータに関連付けられるルールのセットを決定することは、組織の部門に基づいて、使用するべきドキュメントについてのメタデータを規定する少なくとも 1 つのルールを決定することを含み得る。ドキュメントについてのメタデータに関連付けられるルールのセットを決定することは、メタデータ管理サーバから 1 つ以上のルールを抽出することを含み得る。

10

【 0 0 1 1 】

いくつかの実施形態において、ドキュメントについてのメタデータに適用されるようにカスタマイゼーションを構成するルールの決定されたセットに基づいて、ドキュメントについてのメタデータを生成することは、ドキュメントについてベースメタデータに複数のメタデータカスタマイゼーションを適用することを含み得る。

【 0 0 1 2 】

一実施形態において、ウェブベースのアプリケーションのためのユーザインターフェイスとして機能するデスクトップアプリケーションのドキュメントを作成するためのコンピュータ実行可能コードを格納する一時的でないコンピュータ読取可能媒体が提供される。一時的でないコンピュータ読取可能媒体は、そのコンテンツが第 2 のアプリケーションに関連付けられるネイティブアプリケーションフォーマットで作成されたドキュメントであることを示す情報を第 1 のアプリケーションから受け取るためのコードと、ドキュメントについてのメタデータに関連付けられるルールのセットを決定するためのコードとを含み、メタデータは、ドキュメントのコンテンツの部分と、第 1 のアプリケーションと通信する 1 つ以上のウェブベースのアプリケーションによって提供されるデータモデルに関連付けられるデータとの間のリンクを提供し、一時的でないコンピュータ読取可能媒体は、ドキュメントについてのメタデータに適用されるようにカスタマイゼーションを構成するルールの決定されたセットに基づいて、ドキュメントについてのメタデータを生成するためのコードと、ドキュメントが 1 つ以上のウェブベースのアプリケーションのためのユーザインターフェイスとして機能するように、第 2 のアプリケーションにドキュメントについてのメタデータを通信するためのコードとを含む。

30

【 0 0 1 3 】

一実施形態において、ウェブベースのアプリケーションのためのユーザインターフェイスとして機能するデスクトップアプリケーションのドキュメントを作成するためのシステムが提供される。システムは、プロセッサと、指示のセットを格納するメモリとを含み、指示のセットは、プロセッサによって実行されると、そのコンテンツが第 2 のアプリケーションに関連付けられるネイティブアプリケーションフォーマットで作成されたドキュメントであることを示す情報を第 1 のアプリケーションから受け取り、ドキュメントについてのメタデータに関連付けられるルールのセットを決定するように、プロセッサを構成し

40

50

、メタデータは、ドキュメントのコンテンツの部分と、第1のアプリケーションと通信する1つ以上のウェブベースのアプリケーションによって提供されるデータモデルに関連付けられるデータとの間のリンクを提供し、指示のセットはさらに、プロセッサによって実行されると、ドキュメントについてのメタデータに適用されるようにカスタマイゼーションを構成するルールの決定されたセットに基づいて、ドキュメントについてのメタデータを生成し、ドキュメントが1つ以上のウェブベースのアプリケーションのためのユーザインターフェイスとして機能するように、第2のアプリケーションにドキュメントについてのメタデータを通信するようにプロセッサを構成する。

【0014】

この開示の主題の性質および均等物（ならびに提供される如何なる固有または明白な利点および改良）のさらなる理解は、この開示の残りの部分、任意の添付の図面および請求の範囲への参照により、上記のセクションに加えて実現されるはずである。 10

【0015】

この開示において発見されるイノベーション、実施形態および／または例を適切に記載および説明するために、1つ以上の添付の図面に対して参照がなされ得る。1つ以上の添付の図面を説明するよう用いられる付加的な詳細または例は、特許請求される発明のいずれか、ここで記載される実施形態および／もしくは例のいずれか、またはこの開示において示されるいずれかのイノベーションの現在理解されている最良の形態の範囲への限定と見なされるべきではない。

【図面の簡単な説明】 20

【0016】

【図1】本発明に従った一実施形態においてアプリケーション開発フレームワーク（A D F）を示すブロック図である。

【図2】本発明に従った一実施形態において図1のA D Fのためのデスクトップ統合フレームワークを示すブロック図である。

【図3】本発明に従った一実施形態において図2のデスクトップ統合フレームワークを使用して、ドキュメントをデザインするための方法のフローチャートである。

【図4】本発明に従った一実施形態において図2のデスクトップ統合フレームワークを使用してドキュメントと相互作用するための方法のフローチャートである。

【図5】本発明に従った一実施形態において、基礎をなすモデルによってビューがドリブンされ得るドキュメントコンポーネントのスクリーンショットの図である。 30

【図6】本発明に従った一実施形態において図2のデスクトップ統合フレームワークを使用してコンポーネントのモデルドリブンの局面をデザインするための方法のフローチャートである。

【図7】本発明に従った一実施形態においてメタデータ管理を提供する図2のA D Fのためのデスクトップ統合フレームワークを示すブロック図である。

【図8】本発明に従った一実施形態においてメタデータ管理を提供するために図2のデスクトップ統合フレームワーク間の相互作用を示すブロック図である。

【図9】本発明に従った一実施形態においてメタデータ管理を有する図2のデスクトップ統合フレームワークを使用してドキュメントを実行するための方法のフローチャートである。 40

【図10】本発明に従った一実施形態においてメタデータをカスタマイズするためのユーザインターフェイスのスクリーンショットの図である。

【図11】実施形態のうちの1つを実現するための分散システムの簡略図を示す図である。

【図12】本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】

【0017】

発明の詳細な説明 50

以下の記載において本発明の実施形態が完全に理解されるように説明目的のために具体的な詳細を記載する。しかしながら、さまざまな実施形態がこれらの特定の詳細なしで実施されてもよいということは明白であろう。図および記載は、限定的であるように意図されない。

【0018】

イントロダクション

Java EE は、今日のエンタープライズアプリケーションの多くの基礎をなす標準的で、ロバストで、スケーラブルで、かつ、セキュアなプラットフォームである。Java EE は、Java 言語を使用して、多層アプリケーションを構築するための仕様のセットを提供する。過去においては、アプリケーションのロバスト性と、ロバスト性を達成するのに必要とされる複雑性との間に直接的な相関性が存在した。しかしながら、オラクル ADF (Oracle ADF) のような ADF の出現により、標準パターンおよびプラクティスに従うことによって非常にリッチな Java EE アプリケーションのインプリメンテーションが提供され得、労力が大幅に低減される。

【0019】

サービス指向アーキテクチャ (SOA : Service Oriented Architecture) 原理を利用するコンポジットアプリケーションを組織が構築する必要性の増加により、デベロッパは、非常にアジャイルなアプリケーションを作成することを強いられる。アジャイルなアプリケーションにおいてこれらのベストプラクティスを実現することは通常、かなりの量のインフラストラクチャコードを記述することを伴い、これにより、はじめて Java EE アプリケーションを構築するデベロッパにとって別の支障が加わる。ロバストで、パフォーマンスが高く (performant)、メンテナンス可能なアプリケーションの提供に加えて、オラクル ADF は、アジャイルな SOA ベースのアプリケーションを実現するようインフラストラクチャコードを提供し、これにより、組織が「自作する (rolling their own)」ことに伴う労力が除去される。

【0020】

オラクル ADF はさらに、Oracle Developer 11g 開発ツールを通じて、Java EE 開発に対して視覚的および宣言的なアプローチ (visual and declarative approach) を提供する。オラクル ADF は、モデル・ビュー・コントローラデザインパターンを実現し、かつ、オブジェクト/リレーションナルマッピング、データ持続性、再使用可能なコントローラレイヤー、リッチ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、およびカスタマイゼーションといったエリアに到るソリューションでこのアーキテクチャのすべてのレイヤーをカバーする統合ソリューションを提供する。コアウェブベース MVC アプローチを超えて、ADF はさらに、オラクル SOA と、完全なコンポジットアプリケーションの作成の簡素化する Web Center Portal フレームワークとを統合する。

【0021】

たとえば、オラクル ADF は、ADF におけ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。ビジネスサービスインプリメンテーションの詳細のこの分離は、メタデータを介してオラクル ADF において実行される。このメタデータドリブンのアーキテクチャの使用により、アプリケーションデベロッパが、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することが可能になる。

【0022】

オラクル ADF は、ADF モデルレイヤーにおいてメタデータにこれらのサービスのインプリメンテーションの詳細を格納する。これにより、デベロッパは、ユーザインターフェイスを修正することなく、サービスを交換することが可能になり、アプリケーションが非常にアジャイルになる。さらに、ユーザインターフェイスを作成するデベロッパは、ビジネスサービスアクセスの詳細に悩まされる必要がない。代わりに、デベロッパはアプリ

10

20

30

40

50

ケーションインターフェイスおよび相互作用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示すといったように単純であり得る。

【0023】

図1は、本発明に従った一実施形態において、アプリケーション開発フレームワーク(ADF)100を示すブロック図である。図1は、この開示において示される1つ以上の発明のさまざまな実施形態または実現例を組み込み得るシステムの簡略図である。図1は、本願明細書において開示される発明の実施形態または実現例を単に例示しているだけであり得、請求の範囲に記載されるような任意の発明の範囲を限定するべきでない。当業者は、この開示および本願明細書において示される教示を通じて、図において示される実施形態または実現例に対する他の変形例、修正例および/または代替例を認識し得る。

10

【0024】

ADF100は一例ではオラクルADFとして具現化され得る。したがって、ADF100はモデル・ビュー・コントローラ(MVC:Model-View-Controller)デザインパターンに基づく。MVCアプリケーションは、1)データソースとの相互作用を扱うとともにビジネスロジックを実行するモデルレイヤーと、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レイヤーと、3)アプリケーションフローを管理するとともにモデルレイヤーとビューレイヤーとの間のインターフェイスとして機能するコントローラとともに分離される。これらの3つのレイヤーにアプリケーションを分離することによって、アプリケーションを横断するコンポーネントのメンテナンスおよび再使用が簡素化される。他のレイヤーからの各レイヤーの独立によって、疎結合されたサービス指向アーキテクチャ(SOA)が得られる。

20

【0025】

この実施形態において、エンタープライズアプリケーションを形成するモジュールが、当該モジュールADF100内に存在するのが示され、ADFを使用して開発され、次いで、ADF100のコンテキストにおいて実行されることを示す。簡明性のために、JAVAプログラミング言語と、オラクル社から入手可能な開発ツールであるJDeveloper 10.1.3の一部として利用可能なオラクルADFとを使用してアプリケーションが開発されると仮定して、ADFのさまざまな内部の詳細は示されない。しかしながら、以下に記載される本発明の特徴は、関連する技術の当業者には明らかであろうように、本願明細書において提供される開示を読むことによって、プログラミング言語およびアプリケーション開発フレームワークの任意の所望の組合せを用いて実現され得る。

30

【0026】

さまざまな実施形態において、ADF100により、アプリケーションが複数のレイヤーの形態で開発され、各レイヤーは、あらかじ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。したがって、一実施形態において、ADF100によって、アプリケーションは、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レイヤー110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レイヤー120と、基礎をなすデータに対して抽象化レイヤーを提供するデータ/コードモジュールを含むモデルレイヤー130と、さまざまなソースからのデータへのアクセスを提供するとともに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レイヤー140といった4つのレイヤーとして開発され得る。

40

【0027】

オラクルADFにより、レイヤーの各々を実現する際に、デベロッパは使用することを好む技術を選択する。図1は、オラクルADFアプリケーションを構築する場合に、デベロッパに利用可能なさまざまなオプションを示す。Java EEアプリケーションのさまざまなコンポーネントを統合し開発を非常にフレキシブルにする接着剤(glue)がオラクルADFモデルレイヤーである。EJB、Web Services、Java Bea

50

ns、JPA/EclipseLink/TopLinkオブジェクトおよび他の多くのものは、すべてオラクルADFモデルのためのビジネスサービスとして使用され得る。ビューレイヤーは、JSF、デスクトップスイングアプリケーション(Desktop Swing Application)、およびMSオフィスフロントエンドを用いて実装されるウェブベースのインターフェイスと、モバイルデバイスのためのインターフェイスとを含み得る。

【0028】

そのようなレイヤーアプローチを使用するアプリケーションの開発はしばしば、さまざまなアプリケーションを横断するコンポーネント/コードモジュールのメンテナンスおよび再使用を簡素化するということが理解され得る。さらに他のレイヤーからの各レイヤーの独立は、開発されたビジネス/エンタープライズアプリケーションを複数/異なるシステム上で展開する場合に望ましくあり得る疎結合のサービス指向アーキテクチャ(SOA)に繋がる。

10

【0029】

1つの画面において、ビューレイヤー110は、開発されるアプリケーションのユーザインターフェイスを表わす。ビューレイヤー110は、各々がユーザインターフェイスのすべてまたは一部を提供し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デスクトップ、モバイルおよびブラウザベースのビューを有するのが示される。たとえば、ウェブページは、対応するURL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得る。その後、ウェブページは、要求を行ったクライアントシステムに関連付けられるディスプレイユニット(図示せず)上にブラウザによって表示され得、これにより、要求を行ったクライアントシステムのユーザがエンタープライズアプリケーションと相互作用することが可能になる。オラクルADFは、ビジネスサービスの再使用を可能にするビジネスサービスへのマルチチャンネルアクセスと、Web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、または、スマートフォンのようなモバイルデバイスなどからのアクセスとをサポートする。

20

【0030】

(ウェブページのような)ビューレイヤー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(HTML: hypertext markup language)、Javaサーバページ(JSP: Java server page)およびJavaサーバフェイス(JSF: Java Server Face)のうちの1つ以上を用いて実現され得る。代替的には、SwingのようなJavaコンポーネントおよび/または拡張マークアップ言語(XML: extensible markup language)を使用してユーザインターフェイスが実現され得る。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のような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熟知性にレバレッジをかけ得る。

30

【0031】

上に示されるように、ユーザが開発した関連するコード/データモジュールが上記レイヤーの各々において提供される。しかしながら、各レイヤーは典型的に、ADF100によって提供される他のあらかじめ定義されたコード/データモジュールを含む。たとえば、あらかじめ定義されたモジュールのうちのいくつかは、ウェブページを開発するためのテンプレート、開発したコードに所望の機能を含ませるためのテンプレートなどとして、開発の間に使用され得る。(URL書換モジュールのような)他のあらかじめ定義されたモジュールは、開発されたアプリケーションと共に展開され得、エンタープライズアプリケーションの実行の間に、付加的な機能(内部名への要求されたURLのマッピング)をユーザに提供し得る。

40

【0032】

コントローラレイヤー120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、ビューレイヤー110において情

50

報を提示する所望の態様に従って実現される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／選択される場合に特定のウェブページが表示されることと、エラーが実行の間に発生する場合に、格納／抽出されるべき特定のデータを示すページが表示されることなどを含み得る。

【0033】

1つの局面において、コントローラレイヤー120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、サーチボタンがページ上でクリックされると、コントローラは、どのアクションを実行するべきか（サーチを行う）と、どこにナビゲートするべきか（結果ページ）とを決定する。JDeveloperにおいて、ウェブベースのアプリケーションについては、JSFコントローラ機能を拡張する標準的なJSFコントローラまたはADFコントローラといった2つのコントローラオプションが存在する。どちらのコントローラが使用されるかにかかわらず、アプリケーションフローは典型的に、ページおよびナビゲーションルールをダイアグラム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、より小さく再使用可能なタスクフローへ細分され得、フローにおけるメソッドコールおよび決定ポイントのような非視覚的コンポーネントを含み、単一の包含ページの領域の内で実行される「ページフラグメント」フローを作成する。

10

【0034】

コントローラレイヤー120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ともに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サーブレットとして実現される。また、たとえば、コントローラオブジェクトは、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得る。

20

【0035】

モデルレイヤー130は、上に論じたコントローラオブジェクトのような、他のレイヤーにおいてさまざまなビジネスサービスを使用するオブジェクトに当該さまざまなビジネスサービスを接続するか、または、示されるように当該さまざまなビジネスサービスを直接的にデスク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。モデルレイヤー130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され得る対応するインターフェイスを提供し、基礎をなすビジネスサービスレイヤー140において実行する。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インプリメンテーションの詳細を抽象化し、および／または、ビューコンポーネントにデータ制御メソッド／属性を公開し、ビューレイヤーとデータレイヤーとの分離を提供する。

30

【0036】

1つの局面において、モデルレイヤー130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およびデータ結合という2つのコンポーネントからなる。データ制御は、クライアントからビジネスサービスインプリメンテーションの詳細を抽象化する。データ結合は、UIコンポーネントにデータ制御メソッドおよび属性を公開し、ビューとモデルとのクリーンな分離を提供する。モデルレイヤーのメタデータアーキテクチャにより、デベロッパは、任意のタイプのビジネスサービスレイヤーインプリメンテーションをビューレイヤーおよびコントローラレイヤーに結合する際に、同じ開発経験を得る。

40

【0037】

オラクルADFは、開発プロセスの全体にわたって宣言プログラミングパラダイムの使用を強調し、インプリメンテーションの詳細に入ることを必要とすることなくユーザがアプリケーション作成のロジックに集中することを可能にする。ハイレベルでは、Fuseonウェブアプリケーションのための開発プロセスは、アプリケーションワークスペースを作成することを通常伴う。ウィザードを使用して、デベロッパによって選択される技術に必要とされる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

50

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。

【0038】

データベースオブジェクトをモデリングすることによって、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され得、定義が編集され、スキーマが更新される。その後、UML モデラーを使用して、当該アプリケーションのためにユースケース (use case) が作成され得る。また、アプリケーション制御およびナビゲーションがデザインされ得る。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決定するために、ダイアグラム (diagrammer) が使用され得る。当該フローを記述する基礎となる XML が自動的に作成され得る。アプリケーションに単純にリソースライブラリをドロップすることによって、インポートされたライブラリをデベロッパが閲覧および使用することを可能にするよう、リソースライブラリが使用され得る。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され得る。それらのエンティティオブジェクトから、アプリケーションにおけるページによって用いられるようにビューオブジェクトが作成される。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され得る。

【0039】

この例において、ビジネスサービスレイヤー 140 は、データ持続性レイヤーとの相互作用を管理する。これにより、データ持続性、オブジェクト/リレーションナルマッピング、トランザクション管理およびビジネスロジック実行といったサービスが提供される。オラクル ADF におけるビジネスサービスレイヤーは、シンプル Java クラス、EJB、Web services、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において実現され得る。さらに、ファイル (XML または CSV) および REST からデータが直接的に消費され得る。

【0040】

したがって、各ビジネスサービスは、対応するデータ持続性レイヤーとの相互作用を管理し、オブジェクト/リレーションナルマッピング、トランザクション管理、ビジネスロジックの実行などといったサービスを提供する。ビジネスサービスレイヤーは、シンプル Java クラス、Enterprise Java Beans、ウェブサービスなどのうち 1 つ以上を使用して実現され得る。

【0041】

ビジネスコンポーネントは、たとえば、データベース、ウェブサービス、レガシーシステム、および、アプリケーションサーバなどとの相互作用を提供するようオラクル ADF ビジネスコンポーネントを使用して実現されるビジネスサービスを表わす。一実施形態において、ビジネスサービスレイヤー 140 のビジネスコンポーネントは、ビジネスサービスインプリメンテーションを提供するように協働するアプリケーションモジュール、ビューア/クエリー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ともに動作するために、UI 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、ユーザトランザクションに関する更新可能なデータモデルおよびプロシージャ/関数 (一般にサービスメソッドと称される) を提供し得る。

【0042】

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得、当該対応する行に格納されるデータの操作 (更新、削除など) を簡素化する。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保証するために、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さらに、存在するデータベースに格納される行同士の間に存在する関係を反映するよう、他のエンティティオブジェクトに関連付けられ得る。

【0043】

デスクトップ統合

10

20

30

40

50

A D F デスクトップ統合は、オラクルアプリケーション開発フレームワークをマイクロソフトエクセルのようなデスクトップアプリケーションの世界へ拡張する。アプリケーションデベロッパは、ユーザがクリティカルなビジネスデータにアクセスし編集することを可能にするよう、他のデスクトップベースのアプリケーションのスプレッドシートおよびドキュメントといった統合されたドキュメントを迅速に開発し得る。このフレームワークは、各ウェブアプリケーションのセキュリティおよびビジネスロジックインフラストラクチャにシームレスに統合する。さらに、このフレームワークは、エンドユーザがネットワークへの実際の接続なしでデータを編集することを可能にする。ひとたび再接続されると、A D F デスクトップ統合は、アプリケーションのバックエンドに対して、トランスピアレン特にすべてのユーザ変更をアップロードおよび有効化し得る。したがって、A D F デスクトップ統合は、デベロッパがウェブベースのアプリケーションによって提供される機能をデスクトップアプリケーションに拡張することを可能にする。さらに、エンドユーザはA D F デスクトップ統合を好む場合がある。なぜならば、A D F デスクトップ統合は、複雑な計算の実行または大量のデータのアップロードといった情報管理タスクを容易かつシームレスに行うために、ユーザの好むデスクトップアプリケーションにおける見慣れたユーザインターフェイスを提供するからである。

【 0 0 4 4 】

図2は、本発明に従った一実施形態において図1のA D F 1 0 0 のためのデスクトップ統合フレームワーク2 0 0 を示すブロック図である。デスクトップ統合フレームワーク2 0 0 は、この開示内に提示された1つ以上の発明のさまざまな実施形態または実現例を組み込み得る。デスクトップ統合フレームワーク2 0 0 は、本願明細書において開示される発明の実施形態または実現例を単に例示しているだけであり、請求の範囲に記載されるような任意の発明の範囲を限定するべきでない。当業者は、この開示および本願明細書において示される教示を通じて、図において示される実施形態または実現例に対する他の変形例、修正例および／または代替例を認識し得る。

【 0 0 4 5 】

この例において、デスクトップ統合フレームワーク2 0 0 は、クライアントコンピュータシステム2 1 0 およびサーバコンピュータシステム2 2 0 を含む。クライアントコンピュータシステム2 1 0 は、アクセスを提供および／またはアプリケーション2 3 0 をホスティングするように構成されるハードウェアおよび／またはソフトウェア要素を表す。クライアントコンピュータシステム2 1 0 は、パーソナルコンピュータシステム、ラップトップ、タブレット、およびモバイルデバイスなどとして具現化され得る。クライアントコンピュータシステム2 1 0 は、1つ以上のコンピュータ上で実行される1つ以上のオペレーティングシステム、アプリケーション、およびプラウザなどを含み得る。クライアントコンピュータシステム2 1 0 は、本願明細書において開示される発明の実施形態または実現例を単に例示しているだけであり、請求の範囲に記載されるような任意の発明の範囲を限定するべきでない。当業者は、この開示および本願明細書において示される教示を通じて、図において示される実施形態または実現例に対する他の変形例、修正例および／または代替例を認識し得る。

【 0 0 4 6 】

アプリケーション2 3 0 は、ユーザがドキュメントを生成、編集、または別の態様で相互作用することを可能にする1つ以上のソフトウェア要素を表す。アプリケーション2 3 0 のいくつかの例は、テキストエディタ、ワードプロセッシングアプリケーション、スプレッドシートアプリケーション、ならびに、画像編集および操作プログラムなどである。さまざまな実施形態において、デスクトップ統合フレームワーク2 0 0 は、マイクロソフトワードおよびマイクロソフトエクセルのようなマイクロソフトオフィス製品のようなデスクトップアプリケーションに特有の構成で動作する。

【 0 0 4 7 】

アプリケーション2 3 0 は、A D F - D I クライアントコンポーネント2 4 0 をさらに含むか、または、そうでなければA D F - D I クライアントコンポーネント2 4 0 と通信

10

20

30

40

50

し、かつ、ドキュメント 250 を作成する。ADF - DI クライアントコンポーネント 240 は、ウェブベースまたは他のネットワークアクセス可能なアプリケーションによって提供される機能をアプリケーション 230 に拡張する 1 つ以上のソフトウェア要素を表す。たとえば、ADF - DI クライアントコンポーネント 240 は、エンドユーザが、アプリケーション 230 に関連付けられる見慣れたユーザインターフェイスを利用して、サーバコンピュータシステム 220 にアクセスすることにより通常実行される情報管理タスクを行う。これらのタスクは、サーバコンピュータシステム 220 によってホスティングされたウェブベースまたは他のネットワークアクセス可能なアプリケーションによって実行または扱われ得る。さまざまな実施形態において、アプリケーション 230 において実行されるそのような情報管理タスクによって操作されたデータは、サーバコンピュータシステム 220 に同期される。10

【0048】

ドキュメント 250 は、電子情報の 1 つ以上のコンピュータデータファイルまたはユニットを表わす。ドキュメント 250 は、テキスト、画像、オーディオ、ビデオおよび他のマルチメディア情報を含み得る。ドキュメント 250 はさらに、アプリケーション 230 に特有のメタデータに関連付けられ得る。ドキュメント 250 (またはアプリケーション 230) は、ドキュメント 250 に関連付けられるコンテンツを作成し、相互作用し、管理するためのネイティブな機能を提供し得る。さまざまな局面において、アプリケーション 230 は、アプリケーション 230 の機能またはドキュメント 250 のコンテンツと相互作用するための 1 つ以上のインターフェイスを提供する。20

【0049】

サーバコンピュータシステム 220 は、アプリケーションサーバ 260 にアクセスを提供および / またはアプリケーションサーバ 260 をホスティングするよう構成されるハードウェアおよび / またはソフトウェア要素を表わす。サーバコンピュータシステム 220 は、ローカルサーバコンピュータシステムおよびクラウドサービスなどとして具現化され得る。サーバコンピュータシステム 220 は、1 つ以上のコンピュータ上で実行される 1 つ以上のオペレーティングシステム、サーバ、サービス、およびアプリケーションなどを含み得る。サーバコンピュータシステム 220 は、本願明細書において開示される発明の実施形態または実現例を単に例示しているだけであり、請求の範囲に記載されるような任意の発明の範囲を限定するべきでない。当業者は、この開示および本願明細書において示される教示を通じて、図において示される実施形態または実現例に対する他の変形例、修正例および / または代替例を認識し得る。30

【0050】

アプリケーションサーバ 260 は、ユーザがウェブベースのアプリケーションまたはネットワークベースのアプリケーションと相互作用することを可能にする 1 つ以上のソフトウェア要素を表わす。アプリケーションサーバ 260 のいくつかの例は、アプリケーション機能が何かであるかにかかわらず、アプリケーションサーバインプリメンテーションの作成に対する一般化されたアプローチを提供するソフトウェアフレームワークか、または、特定のインプリメンテーションインスタンスのサーバ部分のいずれかである。さまざまな実施形態において、アプリケーションサーバ 260 は、Java プラットフォーム、エンタープライズエディション、または API のコアセットおよび Java アプリケーションサーバの機能を定義する Java EE に特有の構成で動作する。アプリケーションサーバ 260 は、サーブレット、Java Server Page、および Enterprise Java Beansなどを含み得る。アプリケーションサーバ 260 は、本願明細書において開示される発明の実施形態または実現例を単に例示しているだけであり、請求の範囲に記載されるような任意の発明の範囲を限定するべきでない。当業者は、この開示および本願明細書において示される教示を通じて、図において示される実施形態または実現例に対する他の変形例、修正例および / または代替例を認識し得る。40

【0051】

ADF - DI サーバコンポーネント 270 は、アプリケーションサーバ 260 の部分の50

のような1つ以上のサーバコンポーネントを表わす。一般に、ADF-DIクライアントコンポーネント240は、ビューレイヤー110およびコントローラレイヤー120の両方として機能し、モデルレイヤー130として部分的に機能するADF-DIサーバコンポーネント270と通信して、データを同期させて、アプリケーションサーバ260によってホスティングされるアプリケーション、または、ADFモデル280を使用するアプリケーションサーバ260と通信するアプリケーションにおけるビジネスロジックを実行する。上で論じたように、モデルレイヤー130は、データ値に対して使用されるモデルレベルのビジネスルール、セキュリティおよびアプリケーションロジックとともに、アプリケーション230内のADF-DIクライアントコンポーネント240によって提示される現在のビューに関係するデータ値を表わす。この例において、ADF-DIクライアントコンポーネント240およびADF-DIサーバコンポーネント270は、エンドユーザがアプリケーション230に関連付けられる見慣れたユーザインターフェイスを利用することを可能にし、ADFモデル280にアクセスするようドキュメント250を使用してビュー/コントローラタスクを行う。10

【0052】

1つの局面において、デベロッパは、デザインモードにおいてドキュメント250を作成するためにアプリケーション230内で作業するよう、ADF-DIクライアントコンポーネント240を利用する。デベロッパは、アプリケーション230のネイティブツールを利用して所望の態様でドキュメント250を構造化およびフォーマットする。その後、デベロッパは、アプリケーションサーバ260にドキュメント250を統合するよう、ADF-DIクライアントコンポーネント240を使用してドキュメント250にコンポーネントを追加し得る。コンポーネントのいくつかの例は、入力コンポーネント（たとえばフォームコンポーネント）、出力コンポーネント、ラベル、リスト、ボタン、画像、およびテーブルなどである。20

【0053】

その後、ドキュメント250に加えられるコンポーネントは、ADF-DIサーバコンポーネント270によって提供されるまたはADF-DIサーバコンポーネント270を通じて提供される対応するモデルにマッピングされる。たとえば、モデルレイヤー130によって公開されるADF-DIサーバコンポーネント270によって提供されるまたはADF-DIサーバコンポーネント270を通じて提供される人間モデルの名前属性にドキュメント250内の入力/出力メカニズムを提供するよう、テキストボックスコンポーネントがマッピングされ得る。30

【0054】

1つの局面において、コンポーネントは、多くのアプリケーションによって使用され得る（または同じアプリケーションによって複数回使用され得る）機能を提供する再使用可能なエンティティである。コンポーネントは、ドキュメント250内に埋め込まれ得、0以上の視覚的表現を有し得る。視覚的表現を有さないコンポーネントは、表示されないが、何らかの他の機能を提供し得る。コンポーネントは一般に、プログラミングインターフェイス、データ結合インターフェイス、および視覚的インターフェイスのような1つ以上のインターフェイスを提供する。コンポーネント200は1つ以上の視覚的表現を有し得る。さらに以下に記載されるように、コンポーネントは、存在するモデルによってドリブンされる視覚的表現を有し得る。40

【0055】

1つの局面において、コンポーネントは、あるデザインタイムにて任意の数のビューを特定し得、そのいずれかがランタイムにて表示され得る。ビューアセンブリは、ランタイムにて実際に表示されるビューのセットである。アプリケーションまたはコンポーネントのためのビューアセンブリは、ある時点で表示のために選択されるビュー構成におけるビューからなる。

【0056】

ひとたびすべての所望のコンポーネントが含まれ、アプリケーションサーバ260およ50

び A D F モデル 2 8 0 にアクセス可能なデータおよび / またはモデルメタデータにマッピングされると、ドキュメント 2 5 0 は、アプリケーションサーバ 2 6 0 上に「発行 (published)」またはそうでなければ利用可能にされ得る。アプリケーションサーバ 2 6 0 は、ユーザがブラウザを介してドキュメントにアクセスすることを可能にする発行されたドキュメントへのダウンロードリンクを提供し得、サーバコンピュータシステム 2 2 0 にアクセス可能であるデータベースに格納されたデータのようなデータを閲覧、作成、および / または、操作するようアプリケーション 2 3 0 内において作動を開始し得る。さまざまな実施形態において、発行されたドキュメントは、コンポーネント、データマッピング、およびデベロッパがドキュメントと関連付けた任意のロジック、を定義するドキュメントメタデータとは別に格納される。いくつかの実施形態において、発行されたドキュメントはすべてのドキュメントメタデータを含む。

【 0 0 5 7 】

図 3 は、本発明に従った一実施形態において、図 2 のデスクトップ統合フレームワーク 2 0 0 を使用してドキュメントをデザインするための方法 3 0 0 のフロー・チャートである。図 3 に示される方法 3 0 0 の実現または処理は、コンピュータシステムまたは情報処理デバイスのようなロジックマシンの中央処理装置 (C P U またはプロセッサ) によって実行される際にソフトウェア (たとえば命令またはコードモジュール) によって実行され得るか、電子デバイス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実行され得るか、または、ソフトウェアおよびハードウェア要素の組合せによって実行され得る。図 3 に示される方法 3 0 0 は、ステップ 3 1 0 において開始する。

【 0 0 5 8 】

ステップ 3 2 0 において、ドキュメントが受け取られる。一般に、ドキュメントは、そのようなドキュメントをネイティブで作成するアプリケーションによってか、または、ネイティブフォーマットでドキュメントを作成するライブラリを使用して、作成される。図 2 に従うと、ユーザは、存在するドキュメントを開くか、または、たとえばマイクロソフトエクセルにおいて新しいスプレッドシートを作成するといったようアプリケーション 2 3 0 において新しいドキュメントを作成し得る。ドキュメントデベロッパは、ネイティブおよびノンネイティブなツールを使用して任意の所望の態様でドキュメントを編集、構造化、またはフォーマットし得る。

【 0 0 5 9 】

ステップ 3 3 0 において、ドキュメントメタデータが受け取られる。ドキュメントメタデータは、アプリケーションサーバ 2 6 0 から得られるデータに基づいてドキュメントのコンテンツを与えるよう、 A D F - D I クライアントコンポーネント 2 4 0 によって利用される情報を含む。1つの局面において、ドキュメントメタデータは、ドキュメントに含まれる各コンポーネント、および、ビジネスサービスレイヤー 1 4 0 において利用可能なデータまたはメタデータにコンポーネントがどのように結合されるかということを示す。ドキュメントメタデータはさらに、デベロッパによって提供されるアクセス情報、静的データ、および他のロジックまたはデータ操作情報を提供し得る。上で論じたように、デベロッパは、アプリケーションサーバ 2 6 0 にドキュメント 2 5 0 を統合するよう、 A D F - D I クライアントコンポーネント 2 4 0 を使用してドキュメント 2 5 0 にコンポーネントを追加し得る。

【 0 0 6 0 】

さまざまな実施形態において、 A D F - D I クライアントコンポーネント 2 4 0 は、ドキュメントに追加され得るいくつかまたはすべてのコンポーネントについて 1 つ以上の特性を含むエクスプレッションビルダーを提供する。各プロパティは、その対応するコンポーネントの挙動の局面を定義し得る。たとえば、プロパティは、コンポーネントにマッピングされるモデルもしくはオブジェクトを特定し得る、および / または、コンポーネントに対応するモデルもしくはオブジェクトの 1 つ以上の属性を特定し得る。別の例において、プロパティは、テーブルカラムヘッダーおよびワークシートリボンコマンドなどのようなドキュメント 2 5 0 の局面を特定し得る。

10

20

30

40

50

【0061】

ステップ340において、ドキュメントおよびドキュメントメタデータが発行される。上で論じたように、発行されたドキュメントは、ドキュメントメタデータとは別に格納され得る。いくつかの実施形態において、発行されたドキュメントは、ドキュメントメタデータのすべてまたは一部を含み得る。一般に、発行されたドキュメントは、ADF-DIクライアントコンポーネント240が、ドキュメントを初期化し、ADF-DIサーバコンポーネント270からの付加的な情報を要求して、ランタイムにてユーザのためにドキュメント250のコンテンツを与えることを可能にする少なくとも十分なメタデータを含む。図3はステップ350で終了する。

【0062】

ランタイムの間、ユーザは発行されたドキュメント250をダウンロードし、アプリケーション230を用いてそれを開く。一実施形態において、ADF-DIクライアントコンポーネント240は、アプリケーションプラグインまたはモジュールとしてインストールされている。その後、ADF-DIクライアントコンポーネント240は、ドキュメント250がフレームワークコンポーネントを含むよう記述されていることを検出し得、かつ、ドキュメントメタデータ、実データ、および、実行される必要がある任意のロジックを要求するようADF-DIサーバコンポーネント270にコンタクトする。たとえば、ADF-DIクライアントコンポーネント240はまず、どのコンポーネントが含まれるべきかと当該コンポーネントをどこに含むべきかとを定義するドキュメントメタデータを、ADF-DIサーバコンポーネント270からまたはADF-DIサーバコンポーネント270を通じて抽出し得る。ADF-DIクライアントコンポーネント240は、選択されたコンポーネントが使用するかまたはそうでなければ動作することになるADFモデル280からのデータを、ADF-DIサーバコンポーネント270からまたはADF-DIサーバコンポーネント270を通じて抽出し得る。ADF-DIクライアントコンポーネント240はさらに、ドキュメント250に関連付けられる任意のロジックをADF-DIサーバコンポーネント270からまたはADF-DIサーバコンポーネント270を通じて抽出し得る。その後、最終的に、ADF-DIクライアントコンポーネント240は、ドキュメントメタデータ、実データおよびロジックを利用してドキュメント250のコンテンツを与える。

【0063】

したがって、ユーザは、ドキュメントテンプレートを抽出し得るとともに、ADF-DIクライアントコンポーネント240によって実行される処理とアプリケーションサーバ260から得られたデータとに基づき自動的に更新およびフォーマットされたドキュメントコンテンツを有し得る。その後、ユーザは、アプリケーション230に関連付けられる見慣れたユーザインターフェイスを利用して、ドキュメント250を使用してタスクを実行する。

【0064】

さまざまな局面において、ユーザがドキュメント250と相互作用または操作する際に、ADF-DIクライアントコンポーネント240およびADF-DIサーバコンポーネント270は、通信したままであり得、これにより更新を送信および受信する。モデルライヤー130における対応するモデルのデータへのドキュメント250の1つ以上のコンポーネント内でなされた変更は、ADFモデル280において持続され得る。

【0065】

図4は、本発明に従った一実施形態において、図2のデスクトップ統合フレームワーク200を使用してドキュメントと相互作用するための方法400のフローチャートである。図4に示される方法400の実現または処理は、コンピュータシステムまたは情報処理デバイスのような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際にソフトウェア(たとえば命令またはコードモジュール)によって実行され得るか、電子デバイス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実行され得るか、または、ソフトウェアおよびハードウェア要素の組合せによって実行さ

10

20

30

40

50

れ得る。図4に示される方法400は、ステップ410において開始する。

【0066】

ステップ420において、ドキュメントが受け取られる。図2に従うと、ユーザは、たとえばドキュメントリンクをクリックすることによって、所望のドキュメントを抽出するようアプリケーションサーバ260と相互作用し得る。ドキュメントは、クライアントコンピュータシステム210にダウンロードまたはそうでなければ通信され得、アプリケーション230において開かれ得る。

【0067】

ステップ430において、ドキュメントメタデータが処理される。上で論じたように、ドキュメントメタデータは、アプリケーションサーバ260から得られたデータに基づいてドキュメントのコンテンツを与えるよう、ADF-DIクライアントコンポーネント240によって利用される情報を含む。したがって、ADF-DIクライアントコンポーネント240は、どのコンポーネントがドキュメントに追加されるべきかと、どこに追加されるべきかとを決定する。ADF-DIクライアントコンポーネント240はさらに、何のデータが各コンポーネントによって使用されるかと、何のデータがデベロッパによって定義される任意のロジックを適用するかとを決定する。

【0068】

ステップ440において、ドキュメントデータが処理される。上で論じたように、発行されたドキュメントは、ドキュメントメタデータとドキュメントによって使用される実データとは別に格納され得る。一般に、発行されたドキュメントは、ADF-DIクライアントコンポーネント240が、ドキュメントを初期化し、ADF-DIサーバコンポーネント270からの付加的な情報を要求して、ユーザのためにドキュメント250のコンテンツを与えることを可能にする少なくとも十分なメタデータを含む。

【0069】

ステップ450において、ドキュメントが与えられる。上で論じたように、ADF-DIクライアントコンポーネント240は、選択されたコンポーネントが使用するかまたはそうでなければ動作することになるADFモデル280からのデータを、ADF-DIサーバコンポーネント270からまたはADF-DIサーバコンポーネント270を通じて抽出し得る。ADF-DIクライアントコンポーネント240はさらに、ドキュメント250に関連付けられる任意のロジックをADF-DIサーバコンポーネント270からまたはADF-DIサーバコンポーネント270を通じて抽出し得る。その後、最終的に、ADF-DIクライアントコンポーネント240は、ドキュメントメタデータ、実データおよびロジックを利用してドキュメント250のコンテンツを与える。

【0070】

ステップ460において、ドキュメントに対する更新が存在するかどうかについて決定がなされる。ドキュメントへの更新が存在し得るさまざまな理由が存在し得る。ユーザがドキュメント250と相互作用するかまたはドキュメント250を操作する際、ADF-DIクライアントコンポーネント240およびADF-DIサーバコンポーネント270は、更新を送信および受信するよう通信したままであり得る。モデルレイヤー130における対応するモデルのデータへのドキュメント250の1つ以上のコンポーネント内でなされた変更は、ADFモデル280において持続され得る。いくつかの実施形態において、ユーザによる相互作用は新しいデータセットを必要とする場合がある。したがって、方法400のフローは、ステップ450において与えられる付加的なドキュメントデータを処理するようステップ440に戻る。図4はステップ470で終了する。

【0071】

モデルドリブンの局面

さまざまな実施形態において、デスクトップ統合フレームワーク200は、デベロッパが、ビューおよびデータが対応するモデルまたはモデル属性によってドリブンされるコンポーネントドキュメント250を含めることを可能にする。1つの局面において、コンポーネントは、ユーザのマウスがデータの部分上にある場合またはデータが選択される場合

10

20

30

40

50

に表示されるコンポーネントの与えられたコンテンツに関する有用な情報を含むツールティップ (tool tip) を提供し得る。ツールティップは、モデルまたはオブジェクトを介して定義され得、ドキュメントのコンポーネントが与えられる際に自動的に構成され得る。別の局面において、リストコンポーネントの要素には、モデルまたはオブジェクトに関連付けられる 1 つ以上の属性の以前に存在した値がポピュレートされ得る。したがって、デベロッパは、ドキュメントが与えられる場合にコンポーネントが提示するビューにおける値を特定することが必要とされない。

【 0 0 7 2 】

図 5 は、本発明に従った一実施形態において、基礎をなすモデルによってビューがドリブンされ得るドキュメントコンポーネントのスクリーンショット 500 の図である。この例において、ツリー 510 は、ドキュメント 250 において提示され得る 1 つ以上のビューのリストを提供する。この例において、EmpView520 は、「Empno」、「Ename」、「Job」、「Mgr」などとラベル付けされたテキスト、リスト、画像、および日付などといった 1 つ以上のコンポーネントを含む。

【 0 0 7 3 】

1 つの局面において、コンポーネント 530 に関連付けられる従業員オブジェクトの基礎をなすデータモデルまたは属性が「Hiredate」とラベル付けされる。当該データモデルまたは属性は、所与の従業員が組織によって雇われた日付に関連付けられる日付値を格納するように構成される。さまざまな実施形態において、コンポーネント 530 のビューまたは挙動は、デベロッパによるさらなる別の構成が存在しない日付値 (date value absent further configuration) を属性が格納するという事実によってドリブンされ得る。たとえば、コンポーネント 530 に対応するセルを有するエクセルワークブックを ADF - DI クライアントコンポーネント 240 が与える時に、ADF - DI クライアントコンポーネント 240 は、ユーザが従業員の雇用の日付を表わす値を格納するセルを選択すると、エクセルのネイティブの機能内または機能外にて日付ポップアップが提供され、これによりユーザが新しい雇用日付を選択または既存の雇用日付を修正することを可能にするように、セルを構成し得る。

【 0 0 7 4 】

別の局面において、コンポーネント 540 に関連付けられる従業員オブジェクトの基礎をなすデータモデルまたは属性は「Deptno」とラベル付けされる。当該データモデルまたは属性は、ある従業員に関連付けられる部門またはチームについての識別子を格納するように構成される。さまざまな実施形態において、コンポーネント 540 のビューまたは挙動は、属性が、デベロッパによるさらなる別の構成が存在しないデータモデル (data model absent further configuration) において特定される複数の所定の値のうちの 1 つを格納するという事実によってドリブンされ得る。たとえば、コンポーネント 540 に対応するセルを有するエクセルワークブックを ADF - DI クライアントコンポーネント 240 が与える時に、ADF - DI クライアントコンポーネント 240 は、従業員が割り当てられている部門またはチームを表わす値を格納するセルをユーザが選択すると、ドロップダウンリストがエクセルのネイティブの機能内または機能外にて提供され、これによりデータモデルから導き出される部門またはチームの所定のリストからユーザが選択することを可能にするように、セルを構成し得る。

【 0 0 7 5 】

図 6 は、本発明に従った一実施形態において、図 2 のデスクトップ統合フレームワーク 200 を使用してコンポーネントのモデルドリブンの局面をデザインするための方法 600 のフローチャートである。図 6 に示される方法 600 の実現または処理は、コンピュータシステムまたは情報処理デバイスのような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際にソフトウェア (たとえば命令またはコードモジュール) によって実行され得るか、電子デバイス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実行され得るか、または、ソフトウェアおよびハードウェア要素の組合せによって実行され得る。図 6 に示される方法 600 は、ステップ 610 におい

10

20

30

40

50

て開始する。

【0076】

ステップ620において、コンポーネント仕様が受け取られる。一般に、コンポーネント仕様は、当該コンポーネントがどのように定義されるかを特定する情報を指す。図2に従うと、アプリケーション230において、デベロッパは、存在するドキュメントを開くか、または、たとえばマイクロソフトエクセルにおいて新しいスプレッドシートを作成するといったように新しいドキュメントを作成し得る。次いで、デベロッパは、ネイティブおよびノンネイティブなツールを使用して任意の所望の態様でドキュメントを編集、構造化、またはフォーマットし得る。さらに、デベロッパは、複数の所定のコンポーネントから選択を行い、それらのコンポーネントをドキュメント250に追加し得る。

10

【0077】

ステップ630において、データ結合仕様が受け取られる。一般に、データ結合仕様は、そのソースなどのデータとコンポーネントがどのように相互作用するかを特定する情報を指す。さまざまな実施形態において、ADF-DIクライアントコンポーネント240は、この情報をを利用して、コンポーネント仕様に加えてコンポーネントをさらに構成する。1つの局面において、ADF-DIクライアントコンポーネント240は、コンポーネントに関連付けられる1つ以上のモデルまたはオブジェクトに基づいて、各コンポーネントがどのように1つ以上のビューを提示するかを示す。ADF-DIクライアントコンポーネント240は、モデル局面および値などを抽出して、コンポーネントと、任意の関連付けられるビューと、関連付けられる挙動とを構成するようADF-DIサーバコンポーネント270と相互作用し得る。さまざまな実施形態において、ADF-DIクライアントコンポーネント240は、ユーザが上に論じられたエクスプレッションビルダーを使用することに応答して、コンポーネント仕様およびデータ結合仕様を受け取り得る。

20

【0078】

ステップ640において、コンポーネントのモデルドリブンの局面が決定される。ADF-DIクライアントコンポーネント240は、ADFモデル280における属性プロパティまたはヒントとして知られるモデル局面および値などを抽出してコンポーネント、任意の関連付けられるビュー、および関連付けられる挙動を構成するようADF-DIサーバコンポーネント270と相互作用し得る。構成情報は、発行されたドキュメントに関連付けられ得るドキュメントメタデータに格納され得る。

30

【0079】

一例において、ユーザに提示されるラベルは、当該ラベルに対応するデータオブジェクトの名称としばしば異なる。たとえば、「EmpName」と称される属性が存在する場合、UIにおいて、デベロッパは、「従業員名」を表示することを望む。大抵のUIフレームワークは、デベロッパがユーザフレンドリなラベルを特定することを可能にする。しかしながら、現われる必要がある各場所にユーザフレンドリなラベルを特定することは効率的ではない。「モデルドリブン」なアプローチは、モデルレベルにてEmpName属性にユーザフレンドリなラベルを関連付けることである。その後、EmpNameを与えることを望む各UI要素（ページ、ワークシートなど）は、デザインタイムにおいてEmpNameのラベルを間接的に「参照」し、ランタイムにてそれを動的にフェッチする。このアプローチは、データオブジェクトのさまざまな異なる潜在的なプロパティに適用される。他の例は「読み取り専用」、「必須」などを含む。図6はステップ650で終了する。

40

【0080】

メタデータ管理

いくつかの実施形態において、ワークブック統合を定義するメタデータは、メタデータサービスによって管理され得る。図7は、本発明に従った一実施形態においてメタデータ管理を提供する、図2のデスクトップ統合フレームワーク200における相互作用を示すブロック図である。この例において、ドキュメントメタデータは、デザインタイムおよびランタイムの両方においてリポジトリ295内部に格納され得る。ドキュメント250がたとえばカスタマーサイトにおいて発行された後、リポジトリ295においてメタデータ

50

がカスタマイズされ得る。ADF-DIサーバコンポーネント270は、管理ルールのセットに基づいてドキュメントメタデータの1つ以上のバージョンを提供するようメタデータサーバ290と通信する。管理ルールは、たとえば時間、日付、ユーザの役割、構成などのようなドキュメントがメタデータのどのバージョンを利用するかを規定する基準および／または条件を定義し得る。これは、より多くの機能が追加される場合のように、ワークブック定義スキーマが時間にわたり変化することを可能にする。

【0081】

したがって、さまざまな実施形態において、デスクトップ統合フレームワーク200は、カスタマイズ可能なものとしてデベロッパがワークブックを指定することを可能にする。デベロッパは、良好に定義されたワークブック初期化ポイントにて、メタデータサーバ290から抽出されるワークブックメタデータを構成し得る。ワークブックメタデータは、現在のユーザのカスタマイゼーションのコンテキストについてカスタマイズされ得る。別の局面において、ランタイムワークブックは、メタデータサーバ290から抽出されたカスタマイズされたメタデータを使用して与えられ得る。

【0082】

さまざまな実施形態において、ADF-DIクライアントコンポーネント240は、ADF-DIサーバコンポーネント270へドキュメントメタデータを転送し、ADF-DIサーバコンポーネント270は、メタデータサーバ290にワークブック定義ファイル（ベースドキュメントメタデータ）およびカスタマイゼーションを格納させる。デベロッパは、ベースドキュメントおよび／またはカスタマイゼーションを管理するためのファイルベースまたはデータベースリポジトリを使用するよう選択し得る。

【0083】

図8は、本発明に従った一実施形態において、メタデータ管理を提供する、図7のデスクトップ統合フレームワーク間の相互作用を示すブロック図である。ブロック810において、クラスパス上において対応するワークブック定義ファイルを特定するよう、暗示されたメタデータパスが使用され得るような態様で、カスタマイゼーションが可能にされたワークブックが発行される。たとえば、デザインタイムワークブックを「/PROJECT_ROOT/src/apps/workbooks/myDTWorkbook.xlsx」として、それが発行される場合、当該ワークブック定義ファイルは、「/PROJECT_ROOT/src/apps/workbooks/myDTWorkbook.xlsx-workbook-definition.xml」に生成される。アプリケーションが展開されると、このワークブック定義ファイルは、パス「/apps/workbooks/myDTWorkbook.xlsx-workbook-definition.xml」を使用して、クラスパスリポジトリによって発見され得る。したがって、ランタイムワークブックは、暗示されたメタデータパスを使用してそのワークブックメタデータをロードし得るために、「/PROJECT_ROOT/public_html/oracle/apps/workbooks/」フォルダに発行されるべきである。上で論じたように、デベロッパは、ベースドキュメントおよび／またはカスタマイゼーションを管理するためのファイルベースまたはデータベースリポジトリを使用するよう選択し得る。

【0084】

ブロック820におけるワークブック初期化の際、ADF-DIクライアントコンポーネント240は、現在のワークブックのメタデータについて、サーバコンピュータシステム220への要求を発する。ブロック830において、ADF-DIサーバコンポーネント270は、MDSセッション（たとえば現在のADFコンテキスト（ADF Context）からメタデータサーバ290を介して標準的なMDSセッション）を使用してワークブックメタデータを抽出する。MDSセッションに関連付けられるカスタマイゼーションのコンテキストに基づいて、メタデータサーバ290はワークブックメタデータに全てのカスタマイゼーションを適用する。ブロック820において、カスタマイズされたワークブックメタデータは、ADF-DIクライアントコンポーネント240に返され、ワークブックのメタデータキャッシュに書き込まれる。ブロック860において、キャッシュされたカスタマイズされたワークブックメタデータは、ランタイムワークブックを初期化するよう使用される。

10

20

30

40

50

【0085】

図9は、本発明に従った一実施形態において、1つ以上のカスタマイゼーションを含む図2のデスクトップ統合フレームワーク200を使用してドキュメントを実行するための方法900のフローチャートである。図9に示される方法900の実現または処理は、コンピュータシステムまたは情報処理デバイスのような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際にソフトウェア(たとえば命令またはコードモジュール)によって実行され得るか、電子デバイス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実行され得るか、または、ソフトウェアおよびハードウェア要素の組合せによって実行され得る。図9に示される方法900は、ステップ910において開始する。

10

【0086】

ステップ920において、ドキュメントおよび初期メタデータが受け取られる。たとえば、ユーザは、発行されたドキュメント250をアプリケーションサーバ260からダウンロードすることを要求し得る。ステップ930において、メタデータカスタマイゼーションが要求される。1つの局面において、ドキュメントダウンロードは、ドキュメント250にて完了し得、その時、ドキュメント250がアプリケーション230で開かれ得る。その後、ADF-DIクライアントコンポーネント240は、ドキュメント250が与えられる必要があると検出し得る。ADF-DIクライアントコンポーネント240はADF-DIサーバコンポーネント270とコンタクトし、ドキュメントのデザインに従つて与えられるべきドキュメントメタデータおよび任意の実データを得る。

20

【0087】

さまざまな局面において、ワークブック初期化の際、ADF-DIクライアントコンポーネント240は、現在のワークブックのメタデータについてサーバコンピュータシステム220への要求を発する。ADF-DIサーバコンポーネント270は、メタデータサーバ290を介して現在のADFコンテキストから標準MDSセッションを得、このMDSセッションを使用してワークブックメタデータを抽出する。MDSセッションに関連付けられるカスタマイゼーションのコンテキストに基づいて、メタデータサーバ290は、ワークブックメタデータに全てのカスタマイゼーションを適用する。カスタマイズされたワークブックメタデータは、ADF-DIクライアントコンポーネント240に返される。カスタマイズされたワークブックメタデータは、ランタイムワークブックを初期化するために使用される。

30

【0088】

ステップ940において、ドキュメントおよびカスタマイズされたメタデータが処理される。ADF-DIクライアントコンポーネント240は、コンポーネントの配置を決定し、ステップ950においてADF-DIサーバコンポーネント270によって提供される実データを利用して当該コンポーネントをドキュメント250のコンテンツとして与える。図9はステップ960で終了する。

【0089】

ワークブックコンポーネント

さまざまな実施形態において、カスタマイゼーションがさまざまなツールおよびインターフェイスを利用してなされ得る。エンドユーザがデスクトップ統合フレームワークを使用して開発されたドキュメントのランタイムカスタマイゼーションを提供することを可能にする方法、システムおよび一時的でないコンピュータ読取可能媒体が開示される。ワークブックメタデータは、所与のワークブックが特定のウェブアプリケーションにどのように統合されるか記述する情報のセットである。ワークブックが発行されている場合、メタデータは、発行されたワークブックにおけるローカルキャッシュとワークブック定義ファイルとに書き込まれ得る。メタデータ管理は、メタデータサービスによって扱われ得、発行されたワークブックおよびワークブック定義ファイルにおけるローカルキャッシュから独立して、発行されたワークブックの更新およびカスタマイゼーションを可能にする。ワークブックカスタマイゼーションエディタは、統合されたワークブックのさまざまなカス

40

50

タマイゼーションを実行するために使用される。

【0090】

図10は、本発明に従った一実施形態においてメタデータをカスタマイズするためのユーザインターフェイス1000のスクリーンショットである。この例において、ユーザインターフェイス1000は、統合されたワークシートを所与のワークブックから選択する能力、選択されたワークシート上のコンポーネントを閲覧および選択する能力、または、選択されたコンポーネントを編集または削除する能力を提供する。いくつかの実施形態において、ユーザインターフェイス1000は、新しいワークシートを追加する能力、新しいコンポーネントを追加する能力、または、アクションセットのような既存のコンポーネントの高度なプロパティを編集する能力を提供し得る。

10

【0091】

動作の一例において、ウェブアプリケーションが送られて、A c m e社にインストールされる。A c m e社のビジネスアナリストであるラルフは、統合されたワークブックをレビューし、いくつかの調節が必要であると判断する。ラルフは、ウェブアプリケーションにログインし、ユーザインターフェイス1000にナビゲートする。ラルフは、ベースワークブックの新しいカスタマイゼーションを作成することを決定する。ラルフは、テーブルにおける標準的な65カラムから5カラムを除去する。なぜならば、ラルフは、A c m e社がそれらのカラムを必要としない／望まないことを知っているからである。ラルフはカスタマイゼーションを保存する。ここで、当該カスタマイゼーションは、ワークブックカスタマイゼーションに関連付けられるカスタマイゼーションのコンテキストと一致するユーザについて利用可能である。そのようなユーザは、60カラムを有するテーブルを見ることになる。

20

【0092】

別の例において、ユーザインターフェイス1000は、承認されたユーザが所与の統合されたワークブックのさまざまな局面を編集することを可能にするA D Fタスクフロー(ADF Task Flow)の部分を形成する。上で論じたように、ユーザインターフェイス1000は、選択されたワークブックから、統合されたワークシートのリストを提供し得る。ユーザインターフェイス1000は、ユーザがワークブックおよび／またはワークシートを選択および／または削除することを可能にする機能を含み得る。ユーザインターフェイス1000は、選択されたワークシートに現われるかまたはそうでなければ選択されたワークシートに関連付けられるコンポーネントのリストを含み得る。ユーザインターフェイス1000は、ユーザがコンポーネントを選択し、削除し、編集することを可能にする機能を含み得る。ユーザインターフェイス1000は、ユーザがテキスト入力(Input-Text)、テキスト出力(Output-Text)、ラベル(Label)、値のリスト(List-of-Values)といったコンポーネントのプロパティを編集することができる機能を含み得る。たとえば、ドキュメント250のデベロッパは、コンポーネントID、値、注釈などといったコンポーネントのどの局面が閲覧可能(閲覧のみ)であるか、または、スタイル、ツールティップ、リードオンリ、ポジションのように編集可能であるか特定し得る。デベロッパは、たとえばソース、S h o r t D e s c、ポジションなどといったイメージコンポーネントについて、また、ラベル、ポジションなどといったボタンコンポーネントについて、コンポーネントのどの局面が編集可能であるか特定し得る。別の局面において、デベロッパは、テーブルコンポーネントのどの局面が閲覧可能(閲覧のみ)および編集可能であるかを特定し得る。いくつかの例は、コンポーネントの削除のみで編集可能でない読み取り専用テーブルを含むか、または、ポジション、カラムの編集(削除、編集、再配置すべきかどうか)、セル／ヘッダスタイル、ツールティップ、ビジブル(Visible)、ヘッダラベルの編集を許可し、かつ、特別なカラムの削除を防止する。

30

【0093】

さまざまな局面において、ホストアプリケーションは、(メタデータパスおよびワークブック名を含む)編集されるべきワークブックおよびカスタマイゼーションのコンテキストのようなさまざまな情報をユーザインターフェイス1000に提供する。さらに、ホス

40

50

トアプリケーションは、ユーザインターフェイス 1000 のどのユーザがどのワークブック（存在する場合）を編集してもよいのかを制御することを担い得る。

【0094】

結論

図のうちのいくつかに示されるシステムは、さまざまな構成で提供され得る。いくつかの実施形態において、システムは、システムの1つ以上のコンポーネントがクラウドコンピューティングシステムにおいて1つ以上のネットワークにわたって分散される分散システムとして構成され得る。

【0095】

図11は、実施形態のうちの1つを実現するための分散システム1100の簡略図を示す。例示される実施形態において、分散システム1100は、1つ以上のネットワーク1110にわたって、ウェブブラウザまたはプロプラエタリクライアント（たとえばオラクルフォームズ（Oracle Forms）などといったクライアントアプリケーションを実行および動作するように構成される1つ以上のクライアントコンピューティングデバイス1102、1104、1106および1108を含む。サーバ1112は、ネットワーク1110を介してリモートクライアントコンピューティングデバイス1102、1104、1106および1108と通信可能に結合され得る。

10

【0096】

さまざまな実施形態において、サーバ1112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得る。いくつかの実施形態において、これらのサービスは、ウェブベースのサービスもしくはクラウドサービスとして、または、ソフトウェアアズアサービス（SaaS : Software as a Service）モデルで、クライアントコンピューティングデバイス1102、1104、1106および/または1108のユーザに提供され得る。クライアントコンピューティングデバイス1102、1104、1106および/または1108を操作するユーザは、1つ以上のクライアントアプリケーションを利用して、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ようサーバ1112と相互作用する。

20

【0097】

図に示される構成において、システム1100のソフトウェアコンポーネント1118、1120および1122は、サーバ1112上で実現されるのが示される。他の実施形態において、システム1100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つ以上が、クライアントコンピューティングデバイス1102、1104、1106および/または1108の1つ以上によって実現され得る。次いで、クライアントコンピューティングデバイスを操作するユーザは、1つ以上のクライアントアプリケーションを利用し、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。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の組合せで実現され得る。分散システム1100と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ということが認識されるべきである。したがって、図において示される実施形態は、実施形態のシステムを実現するための分散システムの一例であり、限定するようには意図されない。

30

【0098】

クライアントコンピューティングデバイス1102、1104、1106および/または1108は、ポータブルハンドヘルドデバイス（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））もしくはウェアラブルデバイス（たとえばGoogle Glass（登録商標）ヘッドマウントディスプレイ）であり得るか、または、Microsoft Windows Mobile（登録商標）のような実行しているソフトウェアおよび/もしくはiOS、Windows Phone、Android、BlackBerry 10およびPalm OSなどといったさまざまなモバイルのオペレーティングシステムであるか、

40

50

または、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 (S M S : short message service)、Blackberry (登録商標) または他の通信プロトコルが有効化されたものである。クライアントコンピューティングデバイスは、例として Microsoft Windows (登録商標)、Apple Macintosh (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が実行され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含む汎用の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デバイスは、たとえば Google Chrome OS のようなさまざまな GNU / Linux オペレーティングシステムを含むがそれらに限定されないさまざまな商業的に利用可能な UNIX (登録商標) または UNIX ライクなオペレーティングシステムのいずれかが実行されるワークステーションコンピュータであり得る。代替的または付加的には、クライアントコンピューティングデバイス 1102、1104、1106 および 1108 は、シンクライアントコンピュータ、インターネットが有効化されたゲームシステム (たとえば Kinect (登録商標) ジェスチャ入力デバイスを有するかまたは有さない Microsoft Xbox ゲームコンソール)、および / または、パーソナルメッセージデバイスといった、ネットワーク 1110 上で通信可能である任意の他の電子デバイスであり得る。

【0099】

例示的な分散システム 1100 が 4 つのクライアントコンピューティングデバイスを有するのが示されるが、任意数のクライアントコンピューティングデバイスがサポートされ得る。センサなどを有するデバイスのような他のデバイスがサーバ 1112 と相互作用し得る。

【0100】

分散システム 1100 におけるネットワーク 1110 は、さまざまな商業的に利用可能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し得る当業者によく知られている任意のタイプのネットワークであり得、当該プロトコルは、TCP / IP (transmission control protocol / Internet protocol ; トランスミッションコントロールプロトコル / 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エクスチェンジ)、および AppleTalkなどを含むがこれらに限定されない。単に例として、ネットワーク 1110 は、イーサネット (登録商標) および / またはトーカンリングなどに基づいたもののような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 1110 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。ネットワーク 1110 はバーチャルネットワークを含み得、当該バーチャルネットワークは、バーチャルプライベートネットワーク (VPN : virtual private network)、インターネット、エクストラネット、公衆交換電話網 (PSTN : public switched telephone network)、赤外線ネットワーク、無線ネットワーク (たとえば IEEE (Institute of Electrical and Electronics ; 電気電子技術者協会) 802.11 スイート、Blue tooth (登録商標)、および / もしくは任意の他の無線プロトコルのいずれかに基づき動作するネットワーク)、ならびに / または、これらおよび / もしくは他のネットワークの任意の組合せを含むがこれらに限定されない。

【0101】

サーバ 1112 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 (例として、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含む)、サーバファーム、サーバクラスタまたは任意の他の適切な構成および / もしくは組合せから構成され得る。さまざまな実施形態において、サーバ 1112 は、上記の開示に記載される 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適合され得る。たとえば、サーバ 1112 は、本開示の実施形態に従った、上に記載された処理を実行するためのサーバに対応し得る。

【0102】

10

20

30

40

50

サーバ1112は、上に論じられたもののいずれかを含むオペレーティングシステムと、任意の商業的に利用可能なサーバオペレーティングシステムとを実行し得る。サーバ1112はさらに、H T T P (hypertext transport protocol；ハイパーテキストransポートプロトコル) サーバと、F T P (file transfer protocol；ファイル転送プロトコル) サーバと、C G I (common gateway interface；コモンゲートウェイインターフェイス) サーバと、J A V A (登録商標) サーバと、データベースサーバなどを含む、さまざま付加的な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いずれかを実行し得る。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、S y b a s e、I B M (International Business Machines) などから商業的に利用可能なものを含むがこれらに限定されない。

10

【0103】

いくつかの実現例において、サーバ1112は、クライアントコンピューティングデバイス1102、1104、1106および1108のユーザから受け取られるデータフィードおよび／またはイベント更新を分析および統合するよう、1つ以上のアプリケーションを含み得る。例として、データフィードおよび／またはイベント更新は、T w i t t e r (登録商標) フィード、F a c e b o o k (登録商標) 更新、または、1つ以上のサードパーティ情報ソースおよび連続的なデータストリームから受け取られるリアルタイム更新を含み得るがこれらに限定されない。当該リアルタイム更新は、センサデータアプリケーション、金融相場表示器 (financial ticker)、ネットワークパフォーマンス測定ツール（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解析ツール、および自動車交通監視などに関連付けられるリアルタイムイベントを含み得る。サーバ1112はさらに、クライアントコンピューティングデバイス1102、1104、1106および1108の1つ以上のディスプレイデバイスを介してデータフィードおよび／またはリアルタイムイベントを表示するために1つ以上のアプリケーションを含み得る。

20

【0104】

分散システム1100はさらに、1つ以上のデータベース1114および1116を含み得る。データベース1114および1116はさまざまな場所に存在し得る。例として、データベース1114および1116のうちの1つ以上は、サーバ1112に対してローカルである（および／またはサーバ1112に存在する）一時的でないストレージ媒体上に存在し得る。代替的には、データベース1114および1116は、サーバ1112からリモートであり得、ネットワークベースまたは専用の接続を介してサーバ1112と通信し得る。実施形態の1つのセットにおいて、データベース1114および1116は、ストレージエリアネットワーク (S A N : storage-area network) に存在し得る。同様に、サーバ1112に起因する機能を実行するための任意の必要なファイルは、適切なよう、サーバ1112上にローカルにおよび／またはリモートに格納され得る。実施形態の1つのセットにおいて、データベース1114および1116は、オラクルによって提供されるデータベースのような、S Q L フォーマットのコマンドに応答してデータを格納、更新および抽出するように適合されるリレーションナルデータベースを含み得る。

30

【0105】

図12は、本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1200を示す。システム1200は、上に記載されたコンピュータシステムのいずれかを実現するよう使用され得る。当該図に示されるように、コンピュータシステム1200は、バスサブシステム1202を介して多くの周辺サブシステムと通信する処理ユニット1204を含む。これらの周辺サブシステムは、処理加速ユニット1206、I / Oサブシステム1208、ストレージサブシステム1218、および通信サブシステム1224を含み得る。ストレージサブシステム1218は、有形的なコンピュータ読み取り可能ストレージ媒体1222およびシステムメモリ1210を含む。

40

【0106】

バスサブシステム1202は、コンピュータシステム1200のさまざまなコンポーネ

50

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 1202 は単一のバスとして概略的に示されるが、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 1202 は、メモリバスまたはメモリコントローラと、周辺バスと、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ローカルバスと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り得る。たとえば、そのようなアーキテクチャは、産業標準アーキテクチャ (ISA : Industry Standard Architecture) バスと、マイクロチャネルアーキテクチャ (MCA : Micro Channel Architecture) バスと、エンハンスド ISA (EISA : Enhanced ISA) バスと、ビデオエレクトロニクス標準組織 (VEISA : Video Electronics Standards Association) ローカルバスと、IEEE P1386.1 規格に対して製造されるメザニーン (Mezzanine) バスとして実現され得るペリフェラルコンポーネントインターフェクト (PCI : Peripheral Component Interconnect) バスとを含み得る。

【0107】

1つ以上の集積回路として実現され得る処理ユニット 1204 (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) は、コンピュータシステム 1200 の動作を制御する。1つ以上のプロセッサが処理ユニット 1204 に含まれ得る。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み得る。ある実施形態において、処理ユニット 1204 は、各処理ユニットに含まれるシングルまたはマルチコアプロセッサを有する1つ以上の独立した処理ユニット 1232 および / または 1234 として実現され得る。他の実施形態において、処理ユニット 1204 はさらに、シングルチップに2つのデュアルコアプロセッサを統合することにより形成されるクアッドコア処理ユニットとして実現され得る。

【0108】

さまざまな実施形態において、処理ユニット 1204 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し得、複数の同時に実行されているプログラムまたはプロセスを維持し得る。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 1204 および / またはストレージサブシステム 1218 において存在し得る。好適なプログラミングを通じて、プロセッサ 1204 は上に記載されたさまざまな機能を提供し得る。コンピュータシステム 1200 はさらに、デジタル信号プロセッサ (DSP) および / または特殊目的プロセッサなどを含み得る処理加速ユニット 1206 を含み得る。

【0109】

I/Oサブシステム 1208 は、ユーザインターフェイス入力デバイスおよびユーザインターフェイス出力デバイスを含み得る。ユーザインターフェイス入力デバイスは、キーボードと、マウスまたはトラックボールのようなポインティングデバイスと、ディスプレイに組み込まれるタッチパッドまたはタッチスクリーンと、スクロールホイールと、クリックホイールと、ダイヤルと、ボタンと、スイッチと、キーパッドと、音声コマンド認識システムを有する音声入力デバイスと、マイクロフォンと、他のタイプの入力デバイスとを含み得る。ユーザインターフェイス入力デバイスは、たとえば、ジェスチャおよび音声コマンドを使用する自然なユーザインターフェイスを通じて、ユーザが Microsoft Xbox (登録商標) 360 ゲームコントローラのような入力デバイスを用いて制御および相互作用を行い得る Microsoft Kinect (登録商標) モーションセンサのようなモーション感知および / またはジェスチャ認識デバイスを含み得る。ユーザインターフェイス入力デバイスはさらに、Google Glass (登録商標) まばたき検出器のようなアイジェスチャ (eye gesture) 認識デバイスを含み得る。当該アイジェスチャ認識デバイスは、ユーザの目の活動 (たとえば写真を撮る間またはメニューの選択をする間の「まばたき」) を検出し、入力デバイス (たとえば、Google Glass (登録商標))への入力としてアイジェスチャを変換する。さらに、ユーザインターフェイス入力デバイスは、ユーザが音声コマンドを通じて音声認識システム (たとえば Siri (登録商標) ナビゲータ) と相互作用することを可能にする音声認識検出デバイス

10

20

30

40

50

を含み得る。

【 0 1 1 0 】

ユーザインターフェイス入力デバイスはさらに、3次元(3D)マウスと、ジョイスティックまたはポインティングスティックと、ゲームパッドおよびグラフィックタブレットと、スピーカ、デジタルカメラ、デジタルカムコーダ、ポータブルメディアプレイヤー、ウェブカメラ、イメージスキャナ、指紋スキャナ、バーコードリーダ3Dスキャナ、3Dプリンタ、レーザ距離測定器および視線トラッキングデバイスといったオーディオ/ビジュアルデバイスとを含み得るがこれらに限定されない。さらに、ユーザインターフェイス入力デバイスはたとえば、コンピュータ断層撮影デバイス、磁気共鳴映像デバイス、ポジトロン断層(position emission tomography)デバイス、医療超音波検査デバイスといった医療用画像入力デバイスを含み得る。ユーザインターフェイス入力デバイスはさらに、たとえばMIDIキーボードおよびデジタル楽器などのような音声入力デバイスを含み得る。

【 0 1 1 1 】

ユーザインターフェイス出力デバイスは、ディスプレイサブシステム、インジケータライト、または音声出力デバイスのような非視覚的ディスプレイを含み得る。ディスプレイサブシステムは、陰極線管(CRT)、液晶ディスプレイLCD)またはプラズマディスプレイを使用するもののようなフラットパネルデバイス、投写デバイス、およびタッチスクリーンなどであり得る。一般に、「出力デバイス」という用語の使用は、コンピュータシステム1200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すべての可能なタイプのデバイスおよびメカニズムを含むように意図される。たとえば、ユーザインターフェイス出力デバイスは、モニタ、プリンタ、スピーカ、ヘッドフォン、自動車ナビゲーションシステム、プロッタ、音声出力デバイスおよびモデムといった、テキスト、グラフィックスおよびオーディオ/ビデオ情報を視覚的に伝えるさまざまなディスプレイデバイスを含み得るがこれらに限定されない。

【 0 1 1 2 】

コンピュータシステム1200は、現在、システムメモリ1210内に位置するのが示されているソフトウェア要素を含むストレージサブシステム1218を含み得る。システムメモリ1210は、処理ユニット1204上でロード可能および実行可能であるプログラム命令と、これらのプログラムの実行の間に生成されるデータとを格納し得る。

【 0 1 1 3 】

コンピュータシステム1200の構成およびタイプに依存して、システムメモリ1210は、(ランダムアクセスメモリ(RAM)のように)揮発性および/または(リードオンリーメモリ(ROM)、フラッシュメモリなどのように)不揮発性であり得る。RAMは典型的に、処理ユニット1204に直ちにアクセス可能であるか、または、処理ユニット1204によって現在動作および実行されているデータおよび/もしくはプログラムモジュールを含む。いくつかの実現例において、システムメモリ1210は、スタティックランダムアクセスメモリ(SRAM)またはダイナミックランダムアクセスメモリ(DRAM)のような複数の異なるタイプのメモリを含み得る。いくつかの実現例において、たとえば起動の間、コンピュータシステム1200内において要素間で情報を転送するのを補助する基本ルーチンを含む基本入出力システム(BIOS)は、典型的にROMに格納され得る。限定ではなく例として、システムメモリ1210はさらに、クライアントアプリケーション、ウェブブラウザ、中間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(RDBMS: relational database management system)などを含み得るアプリケーションプログラム1212と、プログラムデータ1214と、オペレーティングシステム1216とを示す。例として、オペレーティングシステム1216は、Microsoft Windows(登録商標)、Apple Macintosh(登録商標)、および/またはLinuxオペレーティングシステム、さまざまな商業的に利用可能なUNIX(登録商標)またはUNIXライクのオペレーティングシステムのさまざまなバージョン(さまざまなGNU/LinuxオペレーティングシステムおよびGoogle)

e Chrome (登録商標) OSなどを含むがこれらに限定されない)、および/または、i OS、W i n d o w s (登録商標) P h o n e、A n d r o i d (登録商標) OS、B l a c k B e r r y (登録商標) 1 0 OSおよびP a l m (登録商標) OSオペレーティングシステムといったモバイルオペレーティングシステムを含み得る。

【 0 1 1 4 】

ストレージサブシステム 1 2 1 8 はさらに、いくつかの実施形態の機能を提供する基本的な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的なコンピュータ読取可能ストレージ媒体を提供し得る。プロセッサによって実行されると上記の機能を提供するソフトウェア(プログラム、コードモジュール、指示)がストレージサブシステム 1 2 1 8 に格納され得る。これらのソフトウェアモジュールまたは命令は、処理ユニット 1 2 0 4 によって実行され得る。ストレージサブシステム 1 2 1 8 は、さらに本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得る。

【 0 1 1 5 】

ストレージサブシステム 1 2 0 0 はさらに、コンピュータ読取可能ストレージ媒体 1 2 2 2 に接続され得るコンピュータ読取可能ストレージ媒体リーダ 1 2 2 0 を含み得る。一緒におよび隨意に、システムメモリ 1 2 1 0 と組み合わせて、コンピュータ読取可能ストレージ媒体 1 2 2 2 は、リモートストレージデバイス、ローカルストレージデバイス、固定ストレージデバイス、および/または、リムーバブルストレージデバイスに加え、コンピュータ読取可能な情報を一時的および/またはより永久的に包含、格納、送信、および抽出するためのストレージ媒体を包括的に表す。

【 0 1 1 6 】

コードまたはコードの部分を含むコンピュータ読取可能ストレージ媒体 1 2 2 2 は、ストレージ媒体および通信媒体を含む当該技術において公知または使用される任意の適切な媒体を含み得、当該媒体の例としては、情報の格納および/または送信のための任意の方法または技術において実現される揮発性媒体、不揮発性媒体、リムーバブル媒体および非リムーバブル媒体があるが、これらに限定されない。これは、R A M、R O M、電気的消却プログラム可能型R O M (E E P R O M : electronically erasable programmable ROM)、フラッシュメモリもしくは他のメモリ技術、C D - R O M、デジタルバーサタイルディスク (D V D) もしく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もしくは他の磁気ストレージデバイス、または他の有形的なコンピュータ読取可能媒体といった有形的なコンピュータ読取可能ストレージ媒体を含み得る。これはさらに、所望の情報を送信するために使用されるとともにコンピュティングシステム 1 2 0 0 によってアクセスされ得るデータ信号、データ伝送または任意の他の媒体といった非有形的なコンピュータ読取可能媒体を含み得る。

【 0 1 1 7 】

例として、コンピュータ読取可能ストレージ媒体 1 2 2 2 は、非リムーバブル不揮発性磁気媒体に対して読出または書込を行うハードディスクドライブと、リムーバブル不揮発性磁気ディスクに対して読出または書込を行う磁気ディスクドライブと、C D - R O M、D V D およびB l u - R a y (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不揮発性光学ディスクに対して読出または書込を行う光学ディスクドライブとを含み得る。コンピュータ読取可能ストレージ媒体 1 2 2 2 は、Z i p (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 (U S B) フラッシュドライブ、セキュアデジタル (S D) カード、D V D ディスク、およびデジタルビデオテープなどを含み得るがこれらに限定されない。コンピュータ読取可能ストレージメディア 1 2 2 2 はさらに、フラッシュメモリベースのS S D、エンタープライズフラッシュドライブ、およびソリッドステートR O Mなどのような不揮発性メモリに基づいたソリッドステートドライブ (S S D) と、ソリッドステートR A M、ダイナミックR A M、スタティックR A M、D R A MベースのS S D、磁気抵抗R A M (M R A M : magnetoresistive RAM) S S D、ならびに、D R A M およびフラッシュメモリベースのS S Dの組合せを使用するハイブリッドS S D といった揮発性メモリに基づくS S Dとを含み得る。ディスクドライブおよび

10

20

30

40

50

それらの関連付けられるコンピュータ読取可能媒体は、コンピュータ読取可能な指示、データ構造、プログラムモジュールおよびコンピュータシステム1200のための他のデータの不揮発性ストレージを提供し得る。

【0118】

通信サブシステム1224は、他のコンピュータシステムおよびネットワークにインターフェイスを提供する。通信サブシステム1224は、コンピュータシステム1200からデータを受け取るとともに他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1224は、コンピュータシステム1200がインターネットを介して1つ以上のデバイスに接続することを可能にし得る。いくつかの実施形態において、通信サブシステム1224は、(たとえば、携帯電話技術、3G、4GまたはEDGE (enhanced data rates for global evolution) のような高度なデータネットワーク技術、Wi-Fi (IEEE 802.11ファミリ規格、他のモバイル通信技術、またはその任意の組合せを使用する) 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 (RF) トランシーバコンポーネント、グローバルポジショニングシステム (GPS : global positioning system) 受信コンポーネント、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において、通信サブシステム1224は、無線インターフェイスに加えてまたは無線インターフェイスの代わりに、有線ネットワーク接続(たとえばイーサネット)を提供し得る。

【0119】

いくつかの実施形態において、通信サブシステム1224はさらに、コンピュータシステム1200を使用し得る1人以上のユーザに代わって、構造化されたデータおよび/または構造化されていないデータフィード1226、イベントストリーム1228およびイベント更新1230などの形態にある入力通信を受け取り得る。

【0120】

例として、通信サブシステム1224は、Twitter (登録商標) フィード、Facebook (登録商標) 更新、リッチサイトサマリ (RSS : Rich Site Summary) フィードのようなウェブフィード、および/または、1つ以上のサードパーティの情報ソース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データフィード1226をリアルタイムで受け取るように構成され得る。

【0121】

さらに、通信サブシステム1224は、リアルタイムイベントのイベントストリーム1228および/またはイベント更新1230を含み得る連続的なデータストリームの形態でデータを受け取るように構成され得る。当該連続的なデータストリームは、性質において明示的な終わりを有さない連続的または無限であり得る。連続的な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、金融相場表示器、ネットワークパフォーマンス測定ツール(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解析ツール、および自動車交通監視などを含み得る。

【0122】

通信サブシステム1224はさらに、構造化されたおよび/または構造化されていないデータフィード1226、イベントストリーム1228、およびイベント更新1230などを、コンピュータシステム1200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する1つ以上のデータベースに出力するように構成され得る。

【0123】

コンピュータシステム1200は、ハンドヘルドポータブルデバイス(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、PDA)、ウェアラブルデバイス(たとえばGoogle Glass (登録商標) ヘッドマウントディスプレイ)、PC、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

10

20

30

40

50

あり得る。

【0124】

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、図において示されるコンピュータシステム1200の記載は特定の例としてのみ意図される。当該図に示されたシステムよりも多く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得、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得る。さらに、ネットワーク入出力デバイスのような他のコンピューティングデバイスへの接続も採用され得る。本願明細書において提供される開示および教示に基づいて、当業者は、さまざまな実施形態を実現する他の態様および／または方法を認識するであろう。

10

【0125】

上記明細書において、本発明の局面はその特定の実施形態を参照して記載されるが、当業者は本発明がそれに限定されないことを認識するであろう。上記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得る。さらに、実施形態は、明細書のより広い精神および範囲から逸脱することなく本願明細書において記載された環境および適用例を越えて、任意数の環境および適用例において利用され得る。したがって、明細書および図面は、限定ではなく例示であるとみなされるべきである。

【0126】

その教示がこの開示内に示され得る1つ以上の発明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ステップのセットを実行するために、ロジックマシンの中央処理装置（CPUまたはプロセッサ）を指示するように適合される命令のセットとして、マシンアクセス可能なメモリ、マシン読取可能な物品、有形的なコンピュータ読取可能媒体、コンピュータ読取可能ストレージ媒体、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得る。ロジックは、この開示に示される発明のさまざまな実施形態における方法またはプロセスを行うよう実行される際に、コードモジュールがコンピュータシステムまたは情報処理デバイスのプロセッサにより作動状態になると、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プロダクトの一部を形成し得る。本願明細書において提供されるこの開示および教示に基づいて、示された発明の1つ以上のさまざまな実施形態の開示されたオペレーションまたは機能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の組合せで実現するための他の態様、変形例、修正例、代替例および／または方法を当業者は理解するであろう。

20

30

【0127】

その教示がこの開示に示され得るそれらの発明のいずれか1つの開示された例、実現例およびさまざまな実施形態は、当業者にこの開示の教示を妥当な明瞭さで伝えるために単に例示的である。これらの実現例および実施形態は例示的な図または特定の図を参照して記載され得る際に、記載される方法および／または特定の構造のさまざまな修正例または適合例は当業者に明らかになり得る。本願明細書において発見されるこの開示およびこれらの教示に依存し、かつ、当該教示によって技術を進歩させ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内に示され得る1つ以上の発明の範囲内に存在すると考えられるべきである。したがって、開示内に示された発明が具体的に示される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ので、この記載および図は限定的な意味で考えられるべきでない。

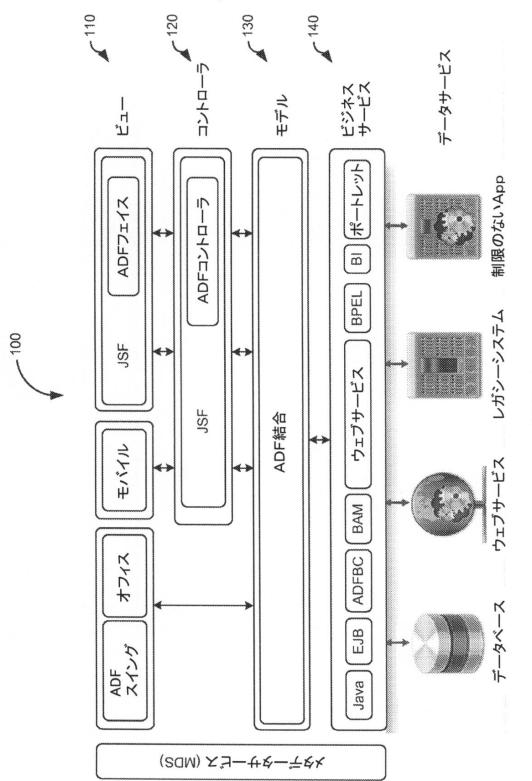
40

【0128】

したがって、上記の記載および如何なる添付の図面、説明および図は、例示的であるが限定的ではないように意図される。したがって、この開示に示される如何なる発明の範囲も、上記の記載および図に示されるそれらの実施形態を単純に参照してではなく、それらの完全な範囲または均等物とともに係属中の請求項を参照して決定されるべきである。

50

【図1】



【図2】

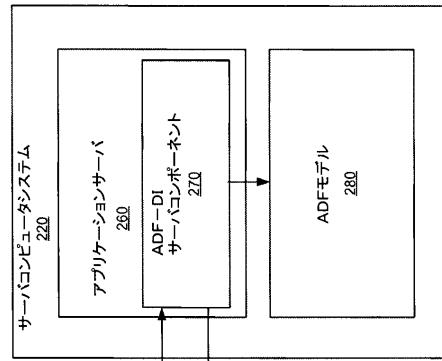
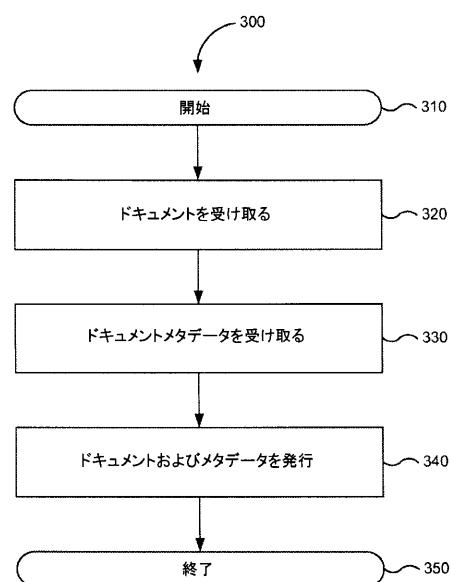


FIG. 2

【図3】



【図4】

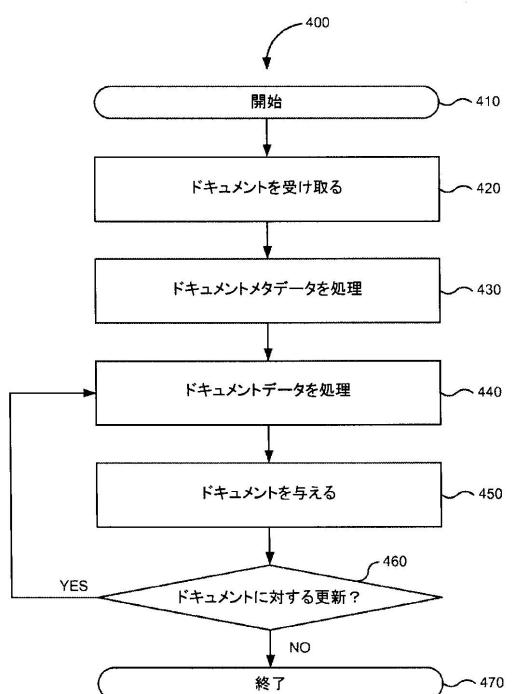
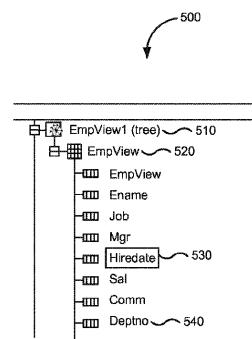


FIG. 3

FIG. 4

【図5】



【 四 6 】

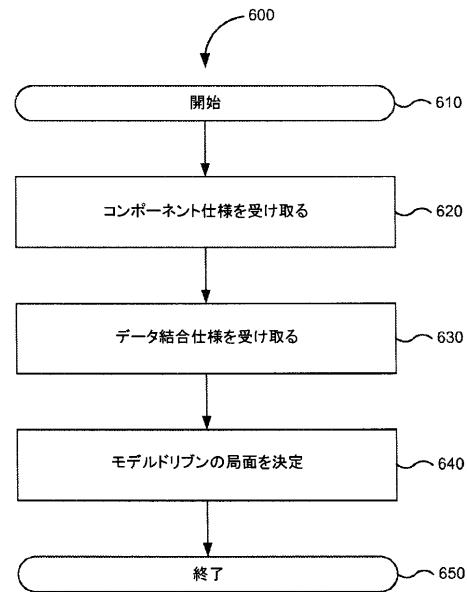


FIG. 5

FIG. 6

【図7】

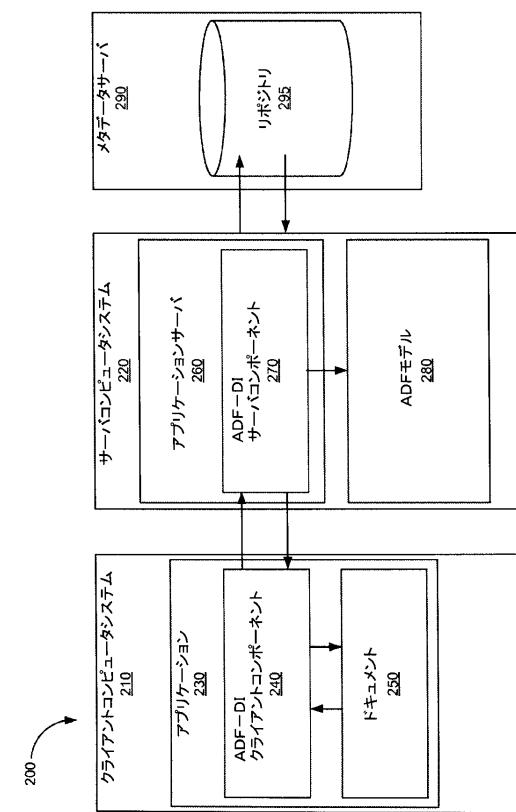
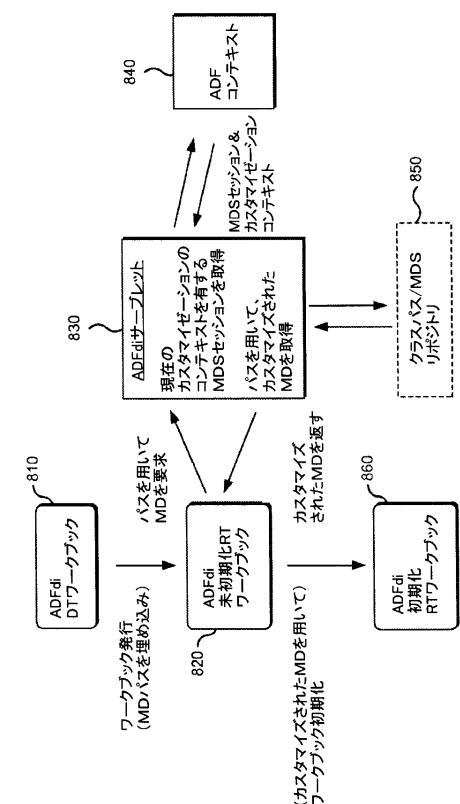


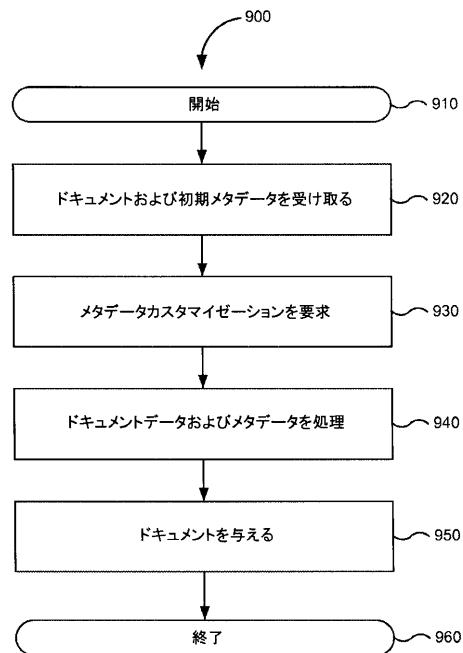
FIG. 7

【 図 8 】



8
FIG.

【図9】



【図10】

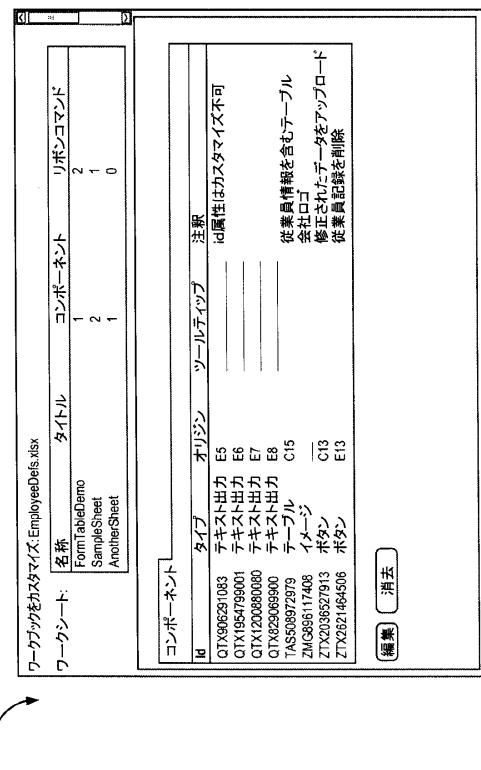
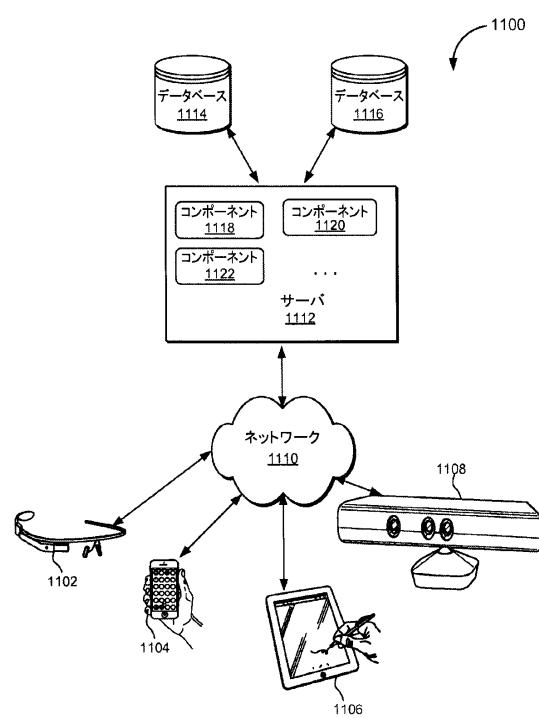


FIG. 9

【図11】



【図12】

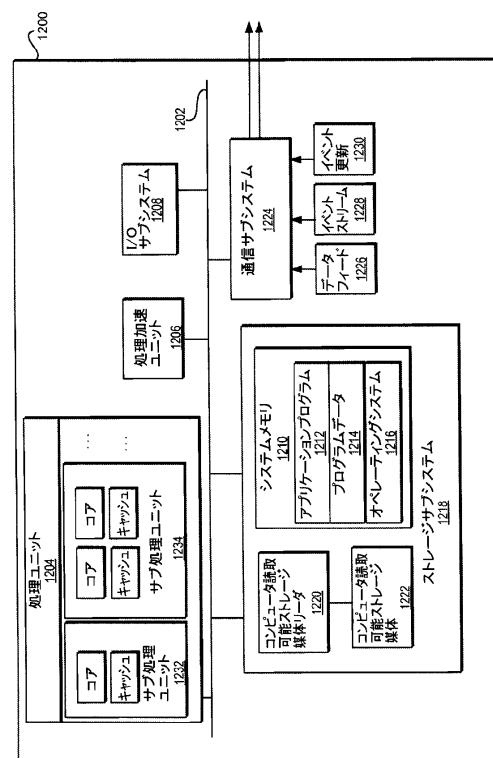


FIG. 12

FIG. 11

フロントページの続き

(31)優先権主張番号 14/304,356
(32)優先日 平成26年6月13日(2014.6.13)
(33)優先権主張国 米国(US)
(31)優先権主張番号 14/304,393
(32)優先日 平成26年6月13日(2014.6.13)
(33)優先権主張国 米国(US)

(72)発明者 デイビス , エドマンド・アレグザンダー
アメリカ合衆国、94402 カリフォルニア州、サン・マテオ、ゴールデンリッジ・コート、2
6

審査官 塚田 肇

(56)参考文献 特開2007-323234 (JP, A)
米国特許出願公開第2003/0193521 (US, A1)
特開2013-097565 (JP, A)
特開2011-203923 (JP, A)
特開2007-026348 (JP, A)
米国特許出願公開第2011/0088011 (US, A1)
米国特許出願公開第2012/0159145 (US, A1)
米国特許出願公開第2006/0253830 (US, A1)
米国特許出願公開第2011/0246444 (US, A1)
米国特許第6868413 (US, B1)

(58)調査した分野(Int.Cl. , DB名)

G 06 F 9 / 451
G 06 F 8 / 38