

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6691554号
(P6691554)

(45) 発行日 令和2年4月28日(2020.4.28)

(24) 登録日 令和2年4月14日(2020.4.14)

(51) Int.Cl. F I
G06F 8/34 (2018.01) G O 6 F 8/34
G06F 9/445 (2018.01) G O 6 F 9/445

請求項の数 10 (全 12 頁)

(21) 出願番号	特願2017-562992 (P2017-562992)	(73) 特許権者	502303739
(86) (22) 出願日	平成28年6月3日 (2016.6.3)		オラクル・インターナショナル・コーポレイション
(65) 公表番号	特表2018-516411 (P2018-516411A)		アメリカ合衆国カリフォルニア州94065
(43) 公表日	平成30年6月21日 (2018.6.21)		レッドウッド・シティー, オラクル・パークウェイ500
(86) 国際出願番号	PCT/US2016/035758	(74) 代理人	110001195
(87) 国際公開番号	W02016/196959		特許業務法人深見特許事務所
(87) 国際公開日	平成28年12月8日 (2016.12.8)	(72) 発明者	アルゲイアー, アクセル
審査請求日	平成31年1月29日 (2019.1.29)		アメリカ合衆国、94065 カリフォルニア州、レッドウッド・ショアーズ、オラクル・パークウェイ、500、エム/エス・5・オー・ピー・7、オラクル・インターナショナル・コーポレイション内
(31) 優先権主張番号	62/171,785		
(32) 優先日	平成27年6月5日 (2015.6.5)		
(33) 優先権主張国・地域又は機関	米国 (US)		
(31) 優先権主張番号	15/171,908		
(32) 優先日	平成28年6月2日 (2016.6.2)		
(33) 優先権主張国・地域又は機関	米国 (US)		

最終頁に続く

(54) 【発明の名称】 基礎をなす永続性フレームワークおよびクラウドベースの統合サービスに含まれるランタイム・エンジンからウェブ・ユーザーインターフェース・アプリケーションを隔離するためのシステム

(57) 【特許請求の範囲】

【請求項1】

統合クラウド・サービス・ランタイムに含まれるランタイム・エンジンからウェブ・インターフェース・アプリケーションを隔離するためのシステムであって、

1つ以上のマイクロプロセッサを含むコンピュータと、

前記コンピュータ上で実行中のクラウド・サービスであって、ソース・アプリケーションとターゲット・アプリケーションとの間の統合フローを作成するための前記ウェブ・インターフェース・アプリケーションと、前記統合フローを実行するための、複数のランタイム・エンジンを含むランタイムとを含むクラウド・サービスと、

統合プロジェクトを設計、デプロイ、およびモニタリングする際に使用するための複数のサービスを前記ウェブ・インターフェース・アプリケーションに対してエクスポートする抽象化アプリケーション・プログラミング・インターフェースとを備え、

前記抽象化アプリケーション・プログラミング・インターフェースは、

前記統合プロジェクトを、ランタイム・エンジン・ニュートラルフォーマットで永続化し、

デプロイメント時に、前記統合プロジェクトを、前記複数のランタイム・エンジンのうちの1つに特有のフォーマットに変換し、

前記ランタイム・エンジン上に前記変換された統合プロジェクトを実行のためにデプロイする、システム。

【請求項2】

統合クラウド・サービス・ランタイムに含まれるランタイム・エンジンからウェブ・インターフェース・アプリケーションを隔離するための方法であって、

マイクロプロセッサを含むコンピュータ上で実行中の統合クラウド・サービスであって、ソース・アプリケーションとターゲット・アプリケーションとの間の統合フローを作成するための前記ウェブ・インターフェース・アプリケーションと、前記統合フローを実行するための、複数のランタイム・エンジンを含むランタイムとを含む統合クラウド・サービスを提供するステップと、

統合プロジェクトを設計、デプロイ、およびモニタリングする際に使用するための複数のサービスを前記ウェブ・インターフェース・アプリケーションに対してエクスポートする抽象化アプリケーション・プログラミング・インターフェースを提供するステップと、

前記抽象化アプリケーション・プログラミング・インターフェースによって、前記統合プロジェクトを、ランタイム・エンジン・ニュートラルフォーマットで永続化するステップと、

デプロイメント時に、前記統合プロジェクトを、前記複数のランタイム・エンジンのうちの1つに特有のフォーマットに変換するステップと、

前記ランタイム・エンジン上に前記変換された統合プロジェクトを実行のためにデプロイするステップとを含む、方法。

【請求項3】

前記抽象化アプリケーション・プログラミング・インターフェースは、複数のプラグブル永続性フレームワークのうちの1つを使用して前記統合プロジェクトを永続化する永続性マネージャを含む、請求項2に記載の方法。

【請求項4】

前記抽象化アプリケーション・プログラミング・インターフェースは、オペレーション・マネージャを備え、当該オペレーション・マネージャは、前記ランタイム・エンジンにニュートラルな統合プロジェクトを永続ストアから読み出し、ランタイム・エンジン特有ハンドラを呼び出して前記統合プロジェクトをランタイム・エンジン特有フォーマットに変換する、請求項2または3に記載の方法。

【請求項5】

前記ランタイム・エンジンは、ユーザによってデプロイメント時に選択され、前記ランタイム・エンジン特有ハンドラを選択するために、前記抽象化アプリケーション・プログラミング・インターフェースによって使用される、請求項4に記載の方法。

【請求項6】

前記抽象化アプリケーション・プログラミング・インターフェースによって前記ウェブ・インターフェース・アプリケーションに対してエクスポートされた前記複数のサービスは、前記統合プロジェクトをコンパイル、変換、およびデプロイする際に使用するための開始サービスを含む、請求項2～5のいずれか1項に記載の方法。

【請求項7】

前記ランタイム・エンジン特有フォーマットは、`service bus`ランタイム・エンジンに特有のフォーマット、またはビジネス・プロセス実行言語(BPEL)ランタイム・エンジンに特有のフォーマットである、請求項4または5に記載の方法。

【請求項8】

前記ランタイム・エンジン・ニュートラルフォーマットは、前記ウェブ・インターフェース・アプリケーションによって定義される、請求項2～7のいずれか1項に記載の方法。

。

【請求項9】

統合クラウド・サービス・ランタイムに含まれるランタイム・エンジンからウェブ・インターフェース・アプリケーションを隔離するための一連の指示を格納したコンピュータ読み取り可能なプログラムであって、前記指示は、1つ以上のプロセッサによって実行されると、前記1つ以上のプロセッサに請求項2～8のいずれか1項に記載の方法を実行させる、コンピュータ読み取り可能なプログラム。

10

20

30

40

50

【請求項10】

統合クラウド・サービス・ランタイムに含まれるランタイム・エンジンからウェブ・インターフェース・アプリケーションを隔離するための一連の指示を格納したコンピュータ読み取り可能なプログラムであって、前記指示は、1つ以上のプロセッサによって実行されると、前記1つ以上のプロセッサに、

マイクロプロセッサを含むコンピュータ上で実行中の統合クラウド・サービスであって、ソース・アプリケーションとターゲット・アプリケーションとの間の統合フローを作成するための前記ウェブ・インターフェース・アプリケーションと、前記統合フローを実行するための、複数のランタイム・エンジンを含むランタイムとを含む統合クラウド・サービスを提供するステップと、

10

統合プロジェクトを設計、デプロイ、およびモニタリングするために使用するための複数のサービスを前記ウェブ・インターフェース・アプリケーションに対してエクスポートする抽象化アプリケーション・プログラミング・インターフェースを提供するステップと、

前記抽象化アプリケーション・プログラミング・インターフェースによって、前記統合プロジェクトを、ランタイム・エンジン・ニュートラルフォーマットで永続化するステップと、

デプロイメント時に、前記統合プロジェクトを、前記複数のランタイム・エンジンのうちの1つに特有のフォーマットに変換するステップと、

前記ランタイム・エンジン上に前記変換された統合プロジェクトを実行のためにデプロイするステップとを含むステップを実行させる、コンピュータ読み取り可能なプログラム

20

【発明の詳細な説明】**【技術分野】****【0001】****著作権表示**

本特許文献の開示の一部には、著作権保護の対象となる資料が含まれる。特許文献または特許開示は、米国特許庁の特許ファイルおよび記録に掲載されているため、著作権者は、何人によるその複製に対しても異議はないが、そうでない場合には、例外なくすべての著作権を保有する。

【0002】**発明分野**

本発明の実施形態は、全体として、クラウド・サービスに関し、特に、クラウドベース統合サービスにおいて、基礎技術からウェブ・インターフェース・アプリケーションを隔離するためのシステムおよび方法に関する。

30

【背景技術】**【0003】****背景**

オンプレミス (on-premises) ・アプリケーションから、SaaS (Software as a Service) とオンプレミス・アプリケーションとのハイブリッド・ミックスへの急速なシフトによって、エンタープライズ・アプリケーションを統合しようと試みる企業にとって、問題が生じている。サービスとしての統合プラットフォーム (iPaaS) は、これらの問題に取り組むためのクラウドベースのツール一式を提供できる。iPaaSプラットフォームは、開発者が統合プロジェクトを設計、デプロイ、およびモニタリングするための設計時間を提供できる。しかしながら、これらの統合プロジェクトは、特定のランタイム・エンジンに紐づけられている可能性があり、追加作業なしでは、異なるランタイム・エンジンにデプロイできない。

40

【発明の概要】**【課題を解決するための手段】****【0004】****概要**

50

実施形態によると、クラウドベースの統合サービスにおいて、基礎技術からウェブ・インターフェース・アプリケーションを隔離する抽象化レイヤを提供するためのシステムおよび方法が本明細書に記載されている。抽象化レイヤは、統合プロジェクトの全ライフサイクルにおいて使用するための複数のサービスをウェブ・インターフェース・アプリケーションに対してエクスポートするアプリケーション・プログラミング・インターフェースを含むことができる。ウェブ・インターフェース・アプリケーションは、統合プロジェクトを永続化する基礎フレームワーク、および統合プロジェクトを実行することになるランタイム・エンジンを問わない。アプリケーション・プログラミング・インターフェースは、統合プロジェクトを、ランタイム・エンジン・ニュートラルフォーマットで永続化でき、デプロイメント時に、統合プロジェクトをランタイム・エンジン・ニュートラルフォーマットからランタイム・エンジン特有フォーマットに変換できる。

10

【図面の簡単な説明】

【0005】

【図1】実施形態に係る統合クラウド・サービスを説明する図である。

【図2】実施形態に係る、ICSランタイムに含まれるランタイム・エンジンからUIアプリケーションを隔離するためのシステムを説明する図である。

【図3】実施形態に係る、ICSランタイムに含まれるランタイム・エンジンからウェブUIアプリケーションを隔離するためのシステムをさらに説明する図である。

【図4】実施形態に係る、ICSランタイムに含まれるランタイム・エンジンからウェブUIアプリケーションを隔離するためのシステムをさらに説明する図である。

20

【図5】実施形態に係る、ICSランタイムに含まれるランタイム・エンジンからウェブUIアプリケーションを隔離するためのシーケンス図を説明する図である。

【図6】実施形態に係る、ICSランタイムに含まれるランタイム・エンジンからウェブUIアプリケーションを隔離するための方法を説明する図である。

【発明を実施するための形態】

【0006】

詳細な説明

サービスとしての統合プラットフォーム（たとえば、Oracle統合クラウド・サービス（ICS））は、クラウドまたはオンプレミスに存在するアプリケーション同士を接続する統合フローを構築およびデプロイするためのクラウドベースのプラットフォームを提供できる。

30

【0007】

図1は、実施形態に係る統合クラウド・サービスを説明する図である。

図1に示すように、ICS107は、ICS統合フロー（flow）を設計、実行、および管理するためのクラウドベースの統合プラットフォームを提供できる。ICSは、エンタープライズ・クラウド環境（たとえば、Oracle Public Cloud）101のアプリケーション・サーバ117上で実行中のウェブ・アプリケーション109およびICSランタイム115を含むことができる。ウェブ・アプリケーションは、開発者がICS統合フローを設計、開始、管理、およびモニタリングできるように、複数のユーザ・インターフェースをエクスポートする設計時間を提供できる。開始されたICS統合フローは、ICSランタイム上でデプロイして実行できる。

40

【0008】

実施形態によると、複数のアプリケーション・アダプタ113を提供できる。複数のアプリケーション・アダプタ113は、複数のアプリケーションへの接続の基礎をなす複雑性を処理することによって、これらのアプリケーションへの接続を構成するタスクを簡素化できる。アプリケーションは、ICSベンダー（たとえば、Oracle Right Now）のエンタープライズ・クラウド・アプリケーション105と、サードパーティ・クラウド・アプリケーション（たとえば、Salesforce）103と、オンプレミス・アプリケーション119とを含むことができる。これらのアプリケーションとの通信の際に使用するために、ICSは、簡易オブジェクトアクセスプロトコル（SOAP）お

50

よび `representational state transfer (REST)` のエンドポイントをこれらのアプリケーションに対してエクスポートできる。

【0009】

実施形態によると、ICS 統合フロー（または、ICS 統合）は、ソース接続、ターゲット接続、および2つの接続の間のフィールドのマッピングを含むことができる。各接続は、アプリケーション・アダプタに基づくことができ、アプリケーションの特定インスタンスと通信するためにアプリケーション・アダプタが必要な追加情報を含めることができる。

【0010】

実施形態によると、ICS 統合フローは、ICS プロジェクトにおいて定義できる。ICS プロジェクトは、たとえば、メタデータ、Java（登録商標）コネクタアーキテクチャ（JCA）ファイル、およびウェブサービス記述言語（WSDL）ファイルなど、構造化された一連のアーティファクトを含むことができる。ICS プロジェクトは、ICS ランタイムにおいてデプロイおよび実行でき、ICS プロジェクトを使用して、クラウド・アプリケーションとオンプレミス・アプリケーションとを統合できる。このように、ICS プロジェクトは、統合プロジェクトである。本明細書全体において、用語「ICS プロジェクト」および「ICS 統合プロジェクト」は、互いに入れ替えて使用される。両方の用語は、統合プロジェクトについて言及するために使用される場合がある。

10

【0011】

抽象化レイヤ

20

上述したように、ICS は、ICS 開発者が ICS プロジェクトを設計するためのウェブ UI アプリケーションと、プロジェクトを実行するためのランタイムとを含むことができる。

【0012】

ICS プロジェクトのライフサイクルの間、開発者は、複数のインターフェースを使用する必要がある場合が多い。たとえば、開発フェーズの間、開発者は、プロジェクトに対して CRUD（生成、読み取り、更新、および削除）操作を行う場合がある。デプロイメント・フェーズの間、開発者は、プロジェクトをコンパイルしたり、デプロイしたりする場合がある。これらの操作は、互いに異なるインターフェースにおいて行われる可能性がある。加えて、ICS プロジェクトは、特定のランタイム・エンジンに特有のフォーマットで開発される可能性があり、追加作業なしでは、異なるランタイム・エンジン用に再利用できない。

30

【0013】

実施形態によると、システムは、基礎技術からウェブ・インターフェース・アプリケーションを隔離する抽象化レイヤを提供できる。抽象化レイヤは、統合プロジェクトの全ライフサイクルにおいて使用するための複数のサービスをウェブ・インターフェース・アプリケーションに対してエクスポートするアプリケーション・プログラミング・インターフェースを含む。ウェブ・インターフェース・アプリケーションは、統合プロジェクトを永続化する基礎フレームワーク、および統合プロジェクトを実行することになるランタイム・エンジンを問わない。アプリケーション・プログラミング・インターフェースは、ランタイム・エンジン・ニュートラルフォーマットで統合プロジェクトを永続化でき、デプロイメント時に、統合プロジェクトをランタイム・エンジン・ニュートラルフォーマットからランタイム・エンジン特有フォーマットに変換できる。

40

【0014】

実施形態によると、抽象化レイヤは、ICS ランタイムに含まれる1つ以上のランタイム・エンジンと、ウェブ・インターフェース・アプリケーションとの間に存在できるため、基礎統合技術から ICS ウェブ・インターフェース・レイヤを隔離できる。

【0015】

実施形態によると、複数のハンドラを提供できる。複数のハンドラは、各々、特定のランタイム・エンジン用である。ランタイム・エンジンに特有のハンドラを使用して、IC

50

Sプロジェクトをランタイム・エンジン・ニュートラルフォーマットからランタイム・エンジンに特有のフォーマットに変換できる。

【0016】

加えて、抽象化レイヤを使用して、ICSパッケージを生成したり、検証したり、ICSプロジェクトをエクスポートしたり、インポートしたり、あらかじめ定義されたICSプロジェクトおよびデータを使って永続ストアをブートしたり、ICSデータをICSプロジェクト・インスタンス間で移動したりすることができる。

【0017】

図2は、実施形態に係る、ICSランタイムに含まれるランタイム・エンジンからUIアプリケーションを隔離するためのシステムを説明する図である。

10

【0018】

図2に示すように、抽象化レイヤ209は、ICSプロジェクトのライフサイクルを管理するために使用するための複数の機能をウェブUIアプリケーションに対してエクスポートするAPIを含むことができる。たとえば、CRUD（生成、読み取り、更新、および削除）211、開始213、モニタリング215といった機能の各々は、ウェブUIアプリケーションでの1つのインターフェース上で見える。

【0019】

実施形態によると、抽象化APIは、基礎技術（たとえば、永続性フレームワークおよびランタイム・エンジン）からこれらの機能を隔離するために、複数のクラス/オブジェクトを使用する、たとえば、永続性マネージャ217およびオペレーション・マネージャ219などの複数のサービスを提供できる。これによって、柔軟性が与えられ、基礎技術が変更された場合、コードを再利用できる。

20

【0020】

実施形態によると、ICSプロジェクトの開発フェーズの間、開発者は、複数のICSオブジェクト（たとえば、接続オブジェクト、および統合フロー・オブジェクト）を生成できる。これらのオブジェクトは、読み取り、更新、または削除される必要がある。抽象化レイヤには、複数の永続性フレームワーク223をプラグインでき、オペレーション・マネージャは、ウェブUIアプリケーションの知識がなくても、複数の永続性フレームワーク223を使用してCRUD操作を行うことができる。オペレーション・マネージャは、エクスポートされた機能を実行するように構成された永続性フレームワークから当該機能を隔離することができる一連のインターフェースを提供する。

30

【0021】

さらに示すように、抽象化レイヤは、さらに、プロジェクト・ハンドラ227および複数のプロジェクト・テンプレート（たとえば、OSBテンプレート）を含むことができる。プロジェクト・ハンドラ227は、ICSプロジェクトのホストとして動作する特定のランタイムのユーザによる選択に基づいて、ランタイム・エンジン特有ハンドラ（たとえば、OSBハンドラ）を自動的に選択できる。選択されたランタイム・エンジン特有ハンドラは、オペレーション・マネージャによって永続ストアから読み出されたランタイム・エンジンにニュートラルなプロジェクトを、1つ以上のプロジェクト・テンプレート225を用いた、ランタイム・エンジン特有プロジェクトに変換できる。オペレーション・マネージャは、ランタイム・エンジン特有プロジェクトを特定のランタイム・エンジンにデプロイ233できる。ランタイム・エンジン・ニュートラルフォーマットでICSプロジェクトを開発して、デプロイメント時にそれをランタイム・エンジン特有フォーマットに変換することによって、抽象化レイヤは、ランタイム・エンジンからウェブUIアプリケーションを効果的に隔離する。

40

【0022】

実施形態によると、ICSランタイムによってエクスポートされたAPIによって集められたランタイム・メトリクス231をフォーマットして表示するために、抽象化APIは、さらに、ランタイム・メトリクス・プレゼンタ221を含む。たとえば、ランタイム・メトリクスは、所定のICS統合フローについての秒毎のトランザクションおよ

50

び失敗したトランザクションの数、成功した呼び出し毎のロギング情報、たとえば、最後の20レコード間を表示するアクティビティ・ストリームを含むことができる。また、ICSランタイムによってエクスポートされたAPIを使用して、トランザクション・ログ・ファイルをダウンロードしたり、特定の種類のランタイム・メトリックスについてのユーザ指定の要求を読み出ししたりできる。

【0023】

図3は、実施形態に係る、ICSランタイムに含まれるランタイム・エンジンからウェブUIアプリケーションを隔離するためのシステムを説明する図である。

【0024】

図3に例示するように、ウェブUIアプリケーションは、ランタイム・エンジン・ニュートラルフォーマットのプラグブル永続性フレームワークを使用して永続ストア315に永続化され得るICSプロジェクト305をICS開発者が生成できるインターフェースを提供できる。

10

【0025】

実施形態によると、ランタイム・エンジン・ニュートラルICSプロジェクトは、特定のランタイム/統合エンジンに特有ではない、ウェブUIアプリケーションによって定義され得るプロジェクト構造を表すことができる。ランタイム・エンジン・ニュートラルプロジェクトは、ソース・アプリケーション、ターゲット・アプリケーション、統合パターン、およびスキーマなど、構成を表す、XMLフォーマットで定義されたメタデータを含むことができる。

20

【0026】

実施形態によると、ICSプロジェクト・クラスは、ランタイム・エンジン・ニュートラルフォーマットのプロジェクトを表すことができ、必要なメタデータをカプセル化できる。ICSプロジェクト・クラス(ICSプロジェクト・アーティファクト314)のインスタンスは、永続ストアに格納できる。

【0027】

実施形態によると、永続性マネージャは、ICSプロジェクトの開発中に、セッション管理311およびリソースのロック/ロック解除313を行うことができる。永続性マネージャは、各リソース(たとえば、ファイルおよびメタデータ)の状態を常にモニタリングし、別の開発者がロックしているリソースを開発者がロック解除してしまうことを防ぐことができる。

30

【0028】

実施形態によると、互いに異なる永続性フレームワーク(たとえば、OSB構成フレームワークおよびOracle Enterprise Repository(OER))を、永続性マネージャにプラグインできるため、永続性フレームワークからウェブUIアプリケーションを隔離できる。

【0029】

図4は、実施形態に係る、ICSランタイムに含まれるランタイム・エンジンからウェブUIアプリケーションを隔離するためのシステムをさらに説明する図である。

【0030】

実施形態によると、オペレーション・マネージャを使用して、ICSプロジェクトを開始できる。

40

【0031】

抽象化レイヤがない場合、ICSプロジェクトを開始するには、エンタープライズ・アーカイブ(EAR)ファイルを生成するための「構築」動作をICSプロジェクトから明示的に実行し、ウェブ・インターフェースからEARファイルを選択してデプロイし、デプロイされたプロジェクトを第3のインターフェースを使用してモニタリングする必要がある。抽象化レイヤは、EARファイルの生成、ならびにICSプロジェクトのデプロイメントおよびモニタリングを、特定のインターフェースにおいて1つのステップにまとめることができる。

50

【0032】

実施形態によると、ICSプロジェクト名の横に開始ボタンを配置できる。ボタンがクリックされると、ICS開発者は、ICSプロジェクトをデプロイするための特定のランタイムを選択するように要求され得る。選択にตอบสนองして、オペレーション・マネージャは、永続性マネージャを使用して、永続ストアからランタイム・エンジン・ニュートラルプロジェクトを読み出し411、ICSプロジェクトをランタイム・エンジン特有フォーマットに変換し、EARファイルを生成し、特定のランタイム・エンジン（たとえば、Oracle Service Busランタイム・エンジン(OSB)）によってエクスポートされたAPIを呼び出して、EARファイルをその上でデプロイできる。

【0033】

実施形態によると、オペレーション・マネージャは、プロジェクト・ハンドラを呼び出すことができる。これによって、ICS開発者によって選択されたランタイムに基づいて、プロジェクトを変換するための、ランタイム・エンジン特有ハンドラを選択できる。

【0034】

図4に示すように、実施形態によると、永続性マネージャは、ランタイム・エンジン・ニュートラルプロジェクト415をプロジェクト・ハンドラに渡すことができる。プロジェクト・ハンドラは、ランタイム・エンジン特有プロジェクトをオペレーション・マネージャに返すことができる。プロジェクト・マネージャは、特定のランタイム・エンジンに変換されたプロジェクトをデプロイ419して、プロジェクトを実行できる。

【0035】

実施形態によると、抽象化レイヤによって、複数のランタイム・エンジン特有ハンドラ（たとえば、Oracle Service Bus(OSB)ハンドラ427、およびビジネス・プロセス実行言語(BPEL)ハンドラ447)を提供できる。OSBハンドラが選択されると、OSBハンドラは、たとえばデータベースからOSBテンプレートを読み出し、当該テンプレートおよび関連するメタデータを使用して、ICSプロジェクトがOSBランタイム・エンジン上で走るために必要な詳細を追加できる。読み出されたテンプレートおよびメタデータを使用するICSプロジェクトは、複数の方法を用いて変換できる。

【0036】

実施形態によると、プロジェクトがデプロイされた後、上述の開始ボタンが配置された同じユーザ・インターフェースによって、デプロイされたプロジェクトのモニタリングを行うことができる。

【0037】

実施形態によると、ICSランタイムは、所定のICS統合フローについての秒毎のトランザクションおよび失敗したトランザクションの数、成功した呼び出し毎のロギング情報、たとえば、最後の20レコードを表示するアクティビティ・ストリームなど、ランタイム・メトリックスを表示するために使用される一連のAPIをエクスポートできる。また、エクスポートされたAPIを使用して、すべてのトランザクションのログ・ファイルをダウンロードしたり、特定の種類のランタイム・メトリックスについてのユーザ指定の要求を受け付けたりできる。

【0038】

図5は、実施形態に係る、ICSランタイムに含まれるランタイム・エンジンからウェブUIアプリケーションを隔離するためのシーケンス図を説明する図である。

【0039】

図5に示すように、ICSウェブUIアプリケーション501を使用してICSプロジェクトを保存(ステップ511)できる場合、永続性マネージャ504を呼び出し、永続性フレームワークAPI505を使用してリソースを生成(ステップ512)できる。リソースの作成には、ICSプロジェクト・ファイルを永続性フレームワークAPIの基礎データベースに対して永続化することを含むことができる。

【0040】

10

20

30

40

50

さらに図5に示すように、ICSウェブUIアプリケーションから、開発者は、ICSプロジェクトを開始(ステップ515)できる。プロジェクトの開始によって、オペレーション・マネージャ506に、永続ストアから読み出された、永続化されたリソースを使用してパッケージを生成(ステップ514)させることができる。

【0041】

実施形態によると、オペレーション・マネージャは、生成されたパッケージを受け付け(ステップ516)、プロジェクト・ハンドラ508を呼び出して、変換されたプロジェクトを生成(ステップ518)できる。さらに、オペレーション・マネージャは、ランタイム・デプロイメント・ビーン(bean)509を呼び出して、変換されたプロジェクトをデプロイ(ステップ521)できる。

10

【0042】

上述したように、ICSプロジェクトを開始するためにボタンをクリックすることによって、ランタイム・エンジン・ニュートラルプロジェクトを読み出すステップと、プロジェクト・パッケージを作成する(プロジェクトをコンパイルする)ステップと、プロジェクトをランタイム・エンジン特有プロジェクトに変換するステップと、変換されたプロジェクトをデプロイするステップとを実行できる。

【0043】

実施形態によると、ICSプロジェクトがうまくデプロイされたことを示す内容をICSウェブUIアプリケーションが受け付けた(ステップ525)後、モニター画面を自動的に表示でき、ここで、デプロイされたプロジェクトのホストとして動作しているICSランタイムに、ランタイム・メトリクス要求を送る(ステップ526)ことができる。モニター用インターフェースによってランタイム・メトリクスが受け付けられ(ステップ527)、さまざまなフォーマット(たとえば、棒グラフ)で表示できる。

20

【0044】

図6は、実施形態に係る、ICSランタイムに含まれるランタイム・エンジンからウェブUIアプリケーションを隔離するための方法を説明する図である。

【0045】

図6に示すように、ステップ611において、マイクロプロセッサを含んだコンピュータ上にウェブUIアプリケーションを提供し、ここで、ウェブUIアプリケーションは、ICSを設計する時間を提供する。

30

【0046】

ステップ613において、ICSプロジェクトを設計、デプロイ、およびモニタリングする際に使用するための複数のサービスをウェブUIアプリケーションに対してエクスポートする抽象化アプリケーション・プログラミング・インターフェース(API)を提供する。

【0047】

ステップ615において、ICSプロジェクトを、ランタイム・エンジン・ニュートラルフォーマットで永続化する。

【0048】

ステップ616において、デプロイメント時に、ICSプロジェクトを、ICSランタイムに含まれるランタイム・エンジンのうちの1つに特有のフォーマットに変換する。

40

【0049】

本発明の実施形態は、本開示の教示に従ってプログラムされた1つ以上のプロセッサ、メモリ、および/またはコンピュータ読み取り可能な記憶媒体を含む、1つ以上の従来の汎用または専用のデジタル・コンピュータ、コンピューティング・デバイス、機械、またはマイクロプロセッサによって好都合に実現されてもよい。当業者であるプログラマは、ソフトウェア技術を身につけた者に明らかのように、本開示の教示に基づいて、適切なソフトウェア・コーディングを容易に用意できる。

【0050】

いくつかの実施形態において、本発明は、指示を格納した非一時的な記憶媒体またはコ

50

ンピュータ読み取り可能な記憶媒体（複数のコンピュータ読み取り可能な記憶媒体）であるコンピュータ・プログラム・プロダクトを含む。コンピュータ・プログラム・プロダクトを使用して、コンピュータを本発明のいずれの処理も実行するようにプログラムできる。たとえば、記憶媒体は、これらに限定されないが、フロッピー（登録商標）ディスク、光ディスク、DVD、CD-ROM、マイクロドライブ、および光磁気ディスクを含む任意の種類のディスク、ROM、RAM、EPROM、EEPROM、DRAM、VRAM、フラッシュメモリデバイス、磁気カードもしくは光カード、ナノシステム（分子メモリICを含む）、または指示および/またはデータを格納するのに適した任意の種類の媒体またはデバイスを含むことができる。

【0051】

10

上記の本発明の実施形態の説明は、例示および説明のために提供されている。これは、本発明を包括的であったり、開示された厳密な形態に限定したりすることを意図していない。多くの変更例および変形例は、当業者に明らかであろう。変更例および変形例は、開示の特徴の任意の適切な組み合わせを含む。実施形態は、本発明の原理およびその実用的な適用を最も適切に説明するために選ばれて記載されているため、他の当業者が、考えられる特定の使用に適したさまざまな変更例を用いて、さまざまな実施形態について本発明を理解できるようになっている。

【図1】

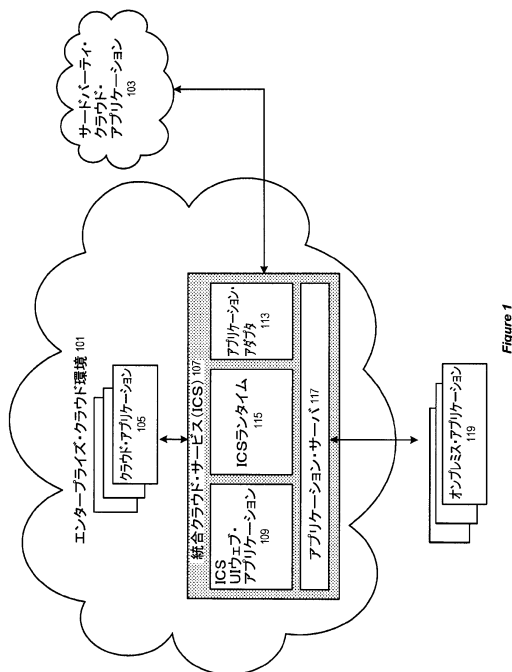


Figure 1

【図2】

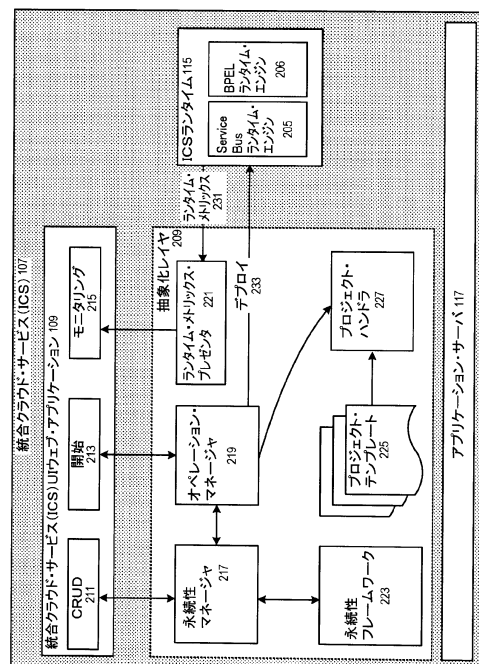


Figure 2

【 図 3 】

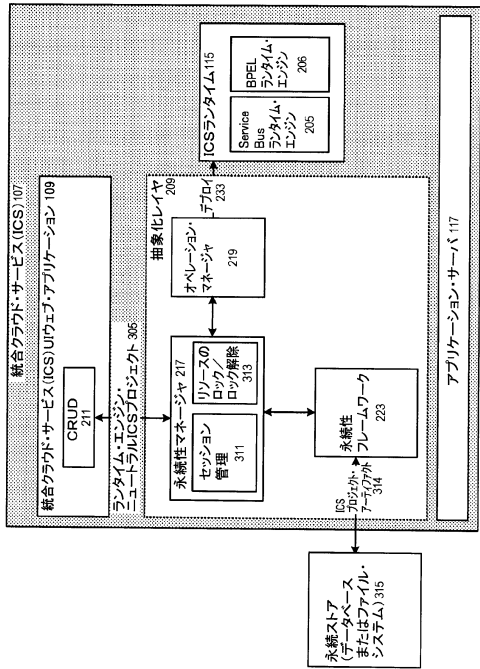


Figure 3

【 図 4 】

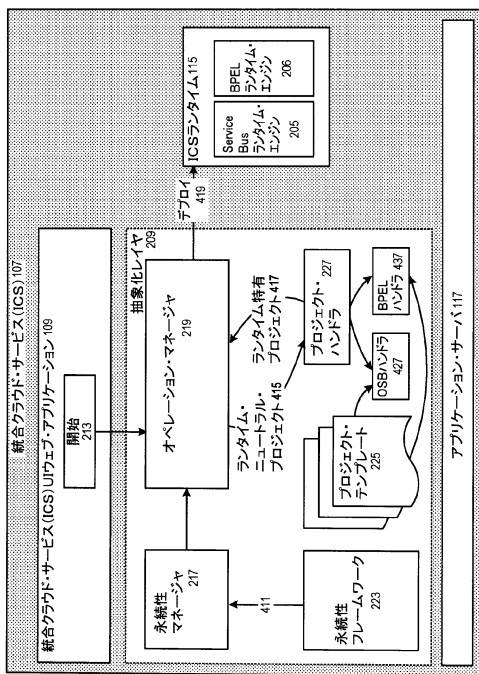


Figure 4

【 図 5 】

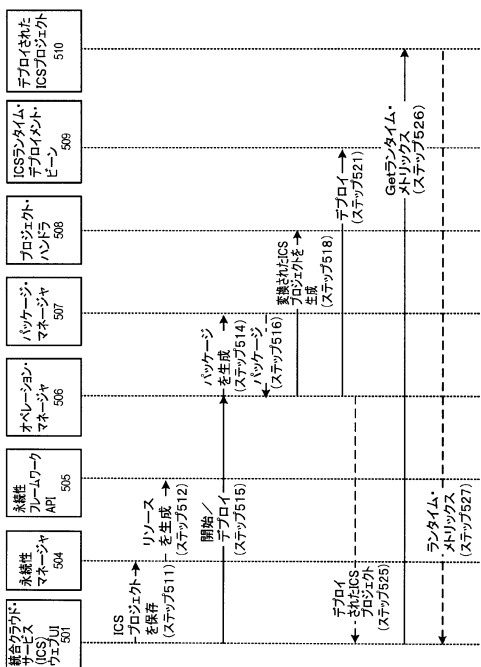


Figure 5

【 図 6 】

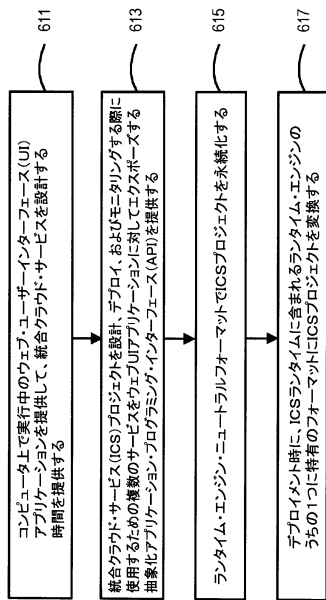


Figure 6

フロントページの続き

- (72)発明者 パラニサミー、ムスクマー
インド、560 029 バンガロール、チッカ・アダゴディ、セカンド・クロス・ロード、ナンバー・18、セント・ジョンズ・ウッズ、プレスティージ・ストリート、レキシントン・タワー、オラクル・インドア・プライベート・リミテッド内
- (72)発明者 ゴラパリ、キショア
インド、560 029 バンガロール、チッカ・アダゴディ、セカンド・クロス・ロード、ナンバー・18、セント・ジョンズ・ウッズ、プレスティージ・ストリート、レキシントン・タワー、オラクル・インドア・プライベート・リミテッド内

審査官 塚田 肇

- (56)参考文献 特表2007-508602(JP,A)
特開2010-049397(JP,A)
特開2004-280820(JP,A)
飯塚隆司ほか、ビジネスプロセスモデリングにおける時間制約条件の導入、FIT2009、日本、2009年、P225-P226
城田真琴、ビジネス単位にサービス化、企業やシステムの枠を超える、日経システムズ、日本、2010年、P92-P97

(58)調査した分野(Int.Cl., DB名)

G06F 8/34
G06F 9/445

- (54)【発明の名称】基礎をなす持続性フレームワークおよびクラウドベースの統合サービスに含まれるランタイム・エンジンからウェブ・ユーザーインターフェース・アプリケーションを隔離するためのシステムおよび方法