

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5323246号  
(P5323246)

(45) 発行日 平成25年10月23日(2013.10.23)

(24) 登録日 平成25年7月26日(2013.7.26)

(51) Int.Cl. F I  
G O 6 F 12/00 (2006.01) G O 6 F 12/00 5 3 7 A

請求項の数 20 外国語出願 (全 28 頁)

(21) 出願番号	特願2012-254599 (P2012-254599)	(73) 特許権者	506332063
(22) 出願日	平成24年11月20日(2012.11.20)		セールスフォース ドット コム インコ
(62) 分割の表示	特願2008-530002 (P2008-530002) の分割		ーポレイティッド
原出願日	平成18年9月8日(2006.9.8)		アメリカ合衆国 カリフォルニア州 94
(65) 公開番号	特開2013-61972 (P2013-61972A)		105, サンフランシスコ, ザ ランドマ
(43) 公開日	平成25年4月4日(2013.4.4)		ーク アット ワン マーケット, スイー
審査請求日	平成24年12月5日(2012.12.5)	(74) 代理人	100076428
(31) 優先権主張番号	60/715,749		弁理士 大塚 康徳
(32) 優先日	平成17年9月9日(2005.9.9)	(74) 代理人	100115071
(33) 優先権主張国	米国 (US)		弁理士 大塚 康弘
早期審査対象出願		(74) 代理人	100146293
			弁理士 西川 恵雄
		(74) 代理人	100112508
			弁理士 高柳 司郎

最終頁に続く

(54) 【発明の名称】 複数の組織体のためにデータ及びオブジェクトを保存するマルチテナントデータベース環境においてアプリケーションを共有する方法

(57) 【特許請求の範囲】

【請求項1】

複数の組織体のためにデータおよびオブジェクトを保存するマルチテナントデータベースを含むマルチテナントデータベース環境において、アプリケーションを共有する方法であって、

第1の組織体に関連する1つ以上のメタデータオブジェクトのセットを参照するパッケージメタデータオブジェクトを作成する工程であって、前記第1の組織体に関連する前記パッケージメタデータオブジェクトは、アプリケーション又はアプリケーションへの参照と、前記アプリケーションで使用されるメタデータオブジェクトの参照と、を含む工程と

、  
マルチテナントデータベースシステムに、前記パッケージメタデータオブジェクトを保存する工程であって、前記マルチテナントデータベースシステムは、前記第1の組織体が、特定のテナントのアプリケーション、データベーステーブル、データエントリを単一の論理データベースの他のテナントと選択的に共有することを可能にし、かつ、仮想私的データベースの共有しないアプリケーション、データベーステーブル、データエントリのセキュリティを保つ工程と、

第2の組織体のユーザへ、前記パッケージメタデータオブジェクトへのアクセスを許可する工程と、  
を含む方法。

【請求項2】

前記パッケージメタデータオブジェクトの妥当性を確認する工程をさらに含む、請求項 1 に記載の方法。

【請求項 3】

前記妥当性を確認する工程が、前記パッケージメタデータオブジェクト中で参照される 1 つ以上のメタデータオブジェクトのセットのオブジェクト依存関係を決定する工程を含む、請求項 2 に記載の方法。

【請求項 4】

前記パッケージメタデータオブジェクトへのアクセスを許可する工程が、前記第 2 の組織体のユーザへリンクを送信する工程を含み、前記リンクの選択が、前記保存されたパッケージメタデータオブジェクトへのアクセスを提供する、請求項 1 に記載の方法。

10

【請求項 5】

前記作成する工程が、前記第 1 の組織体に関連する複数のメタデータオブジェクトからユーザが選択した 1 つ以上のメタデータオブジェクトを受取る工程を含む、請求項 1 に記載の方法。

【請求項 6】

前記 1 つ以上のメタデータオブジェクトが、前記第 1 の組織体によって使用可能な 1 つ以上のカスタムオブジェクトを含む、請求項 5 に記載の方法。

【請求項 7】

前記パッケージメタデータオブジェクトが、前記パッケージメタデータオブジェクト中で参照されるオブジェクトの主キーのセットとして前記データベースシステムに保存される、請求項 1 に記載の方法。

20

【請求項 8】

前記作成する工程が、  
前記第 1 の組織体に関連する、ユーザが選択したメタデータオブジェクトを受取る工程と、  
前記選択されたメタデータオブジェクトに依存関係を有するオブジェクトを決定する工程と、

前記選択されたオブジェクト及び前記選択されたメタデータオブジェクトに依存関係を有すると決定された任意のオブジェクトに対する参照を、前記パッケージメタデータオブジェクト中に作成する工程と、  
を含む、請求項 1 に記載の方法。

30

【請求項 9】

前記保存する工程が、前記パッケージメタデータオブジェクトへの参照を、前記第 1 の組織体に関連しない別個の組織体のためのオブジェクトとしてディレクトリ中に保存する工程を含む、請求項 1 に記載の方法。

【請求項 10】

前記パッケージメタデータオブジェクトへのアクセスを許可する工程が、前記保存されたパッケージメタデータオブジェクトへのリンクを提供する工程を含み、前記リンクの選択が前記保存されたパッケージメタデータオブジェクトへのアクセスを提供する、請求項 9 に記載の方法。

40

【請求項 11】

前記第 2 の組織体のユーザによってリンクが選択されるときに、前記第 2 の組織体に関連するデータベーススキーマ中へ前記パッケージメタデータオブジェクトがインスタンス化される、請求項 10 に記載の方法。

【請求項 12】

前記パッケージメタデータオブジェクトのデモンストレーションコピーを作成する工程をさらに含む、請求項 9 に記載の方法。

【請求項 13】

デモンストレーションパッケージがサンプルデータを含む、請求項 12 に記載の方法。

【請求項 14】

50

前記マルチテナントデータベースのためのコンテナ組織体へのアップデートを受取る工程と、

前記アップデートを、前記マルチテナントデータベース中のパッケージおよびすべての他の組織体の両方に同時に適用する工程と、  
をさらに含む、請求項 1 に記載の方法。

【請求項 15】

データベースシステムに前記パッケージメタデータオブジェクトを保存する工程は、  
コンテナ組織体が、前記マルチテナントデータベース中のパッケージおよびすべての他の組織体の両方についてアップグレードされ得ることを保証するために、前記マルチテナントデータベースシステム中の他のすべての組織体に用いられる物理的記憶機構およびスキーマを使用して、前記パッケージメタデータオブジェクトを保存する工程を含む、請求項 1 に記載の方法。

【請求項 16】

複数の組織体のためのデータおよびオブジェクトを保存するマルチテナントデータベース環境においてアプリケーションを共有するための、クライアント装置中のプロセッサによって実行可能なコードを含む、コンピュータ可読媒体であって、前記コードが、

第 1 の組織体に関連する 1 つ以上のメタデータオブジェクトのセットを参照するパッケージメタデータオブジェクトを作成する命令であって、前記第 1 の組織体に関連する前記パッケージメタデータオブジェクトは、アプリケーション又はアプリケーションへの参照と、前記アプリケーションで使用されるメタデータオブジェクトの参照と、を含む命令と

、  
前記パッケージメタデータオブジェクトをマルチテナントデータベースシステムへ保存する命令であって、前記マルチテナントデータベースシステムは、特定のテナントが、前記特定のテナントのアプリケーション、データベーステーブル、データエントリを単一の論理データベースの他のテナントと選択的に共有することを可能にし、かつ、仮想私的データベースの共有しないアプリケーション、データベーステーブル、データエントリのセキュリティを保つ、命令と、

第 2 の組織体のユーザに、前記パッケージメタデータオブジェクトへのアクセスを許可する命令と、  
を含む、コンピュータ可読媒体。

【請求項 17】

前記コードが前記パッケージメタデータオブジェクトの妥当性を確認する命令をさらに含む、請求項 16 に記載のコンピュータ可読媒体。

【請求項 18】

複数の組織体のためにデータおよびオブジェクトを保存するマルチテナントデータベースシステムであって、第 1 の組織体が、前記第 1 の組織体のアプリケーション、データベーステーブル、データエントリを単一の論理データベースの他のテナントと選択的に共有することを可能にし、かつ、仮想私的データベースの共有しないアプリケーション、データベーステーブル、データエントリのセキュリティを保つ、マルチテナントデータベースシステムと、

ネットワーク相互接続と、  
プロセッサを有するクライアント装置と、  
を備えており、

前記プロセッサは、パッケージメタデータオブジェクトを前記マルチテナントデータベースシステムに保存し、第 2 の組織体のユーザに対して、前記パッケージメタデータオブジェクトへのアクセスを許可するために、前記データベースシステムを制御可能なロジックを実行し、

前記第 1 の組織体に関連するパッケージメタデータオブジェクトは、アプリケーション又はアプリケーションへの参照と、前記アプリケーションで使用されるメタデータオブジェクトの参照を含む、コンピュータシステム。

## 【請求項 19】

マルチテナントデータベースシステムを含むマルチテナントデータベース環境中でアプリケーションを共有するためのコードを、機械を通じてアクセス可能な伝送媒体を通じて送信するための方法であって、

前記マルチテナントデータベースシステムは、特定のテナントが、前記特定のテナントのアプリケーション、データベーステーブル、データエントリを単一の論理データベースの他のテナントと選択的に共有することを可能にし、かつ、仮想私的データベースの共有しないアプリケーション、データベーステーブル、データエントリのセキュリティを保ち

、  
1つ以上のプロセッサが、第1の組織体に関連する1つ以上のメタデータオブジェクトのセットを参照するパッケージメタデータオブジェクトを作成するようにさせる、コードを送信する工程であって、前記第1の組織体に関連する前記パッケージメタデータオブジェクトは、アプリケーション又はアプリケーションへの参照と、前記アプリケーションで使用されるメタデータオブジェクトの参照と、を含む工程と、

10

1つ以上のプロセッサが前記パッケージメタデータオブジェクトをデータベースシステムへ保存するようにさせる、コードを送信する工程と、

1つ以上のプロセッサが第2の組織体のユーザに前記パッケージメタデータオブジェクトへのアクセスを許可するようにさせる、コードを送信する工程と、  
を含む方法。

## 【請求項 20】

20

複数の組織体のためのデータおよびオブジェクトを保存するマルチテナントデータベースを含むマルチテナントデータベース環境において、アプリケーションを共有する方法であって、

第1の組織体に関連するパッケージであって、アプリケーション又はアプリケーションへの参照と、前記アプリケーションで使用されるメタデータオブジェクトの参照と、を含むパッケージを作成する工程と、

マルチテナントデータベースシステムに前記パッケージを保存する工程であって、前記マルチテナントデータベースシステムは、前記第1の組織体が、前記第1の組織体のアプリケーション、データベーステーブル、データエントリを単一の論理データベースの他のテナントと選択的に共有することを可能にし、かつ、仮想私的データベースの共有しないアプリケーション、データベーステーブル、データエントリのセキュリティを保つ工程と

30

、  
第2の組織体のユーザへ前記パッケージへのアクセスを許可する工程と、  
を含む方法。

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

本出願は、2005年9月9日出願の米国特許仮出願第60/715,749号(代理人整理番号021735-001500US)について優先権を主張し、その開示全体は参照により、本明細書に組み入れられる。本発明は、一般にデータベースに、より詳細にはマルチテナントおよび/またはマルチアプリケーションデータベースシステムにおいて、カスタマイズされたアプリケーションを作成し交換するためのシステムおよび方法に関する。

40

## 【背景技術】

## 【0002】

古代人は、数および記述システムを発明してあまり経たないうちに、不注意に情報を作成してしまったことに気付いた。情報の作成と共に、人類を以後数千年間悩ますであろう：情報をどのように保存し管理するか、という問題が生じた。

## 【0003】

幸運にも、古代人によって作成された情報の量は比較的少なく、10本の手指および10本

50

の足指を用いて追跡記録することができた。より永続的な記録が望まれる場合は、情報を追跡記録するために石板または粘土板を用いた。この仕組みの初期の周知の一例は、モーゼによって用いられたものであり、モーゼは2つのそのような石板上にモーゼの十戒の本文を記録した。しかし、これらの仕組みは、書込み一回・読出し多数回実行に限られ、また容量が厳しく制限された。

#### 【0004】

グーテンベルクの印刷機の出現によって、より大量の情報を保存する能力、ならびに保存情報の大量のコピーを作成する能力が生じた。一方で、これらの機構もまた、書込み一回・読出し多数回実行に限定されたが、それらは知識の広範囲な普及を促進し、それが次には技術進歩の推進を加速した。数世紀後に、コンピュータおよびデータベースソフトウェアシステムが現われた。コンピュータデータベースは、容易に利用できるパッケージの、大容量の書込み読取り記憶装置を提供した。暫くの間は、人類の長年の情報保存および管理問題は最終的に解決されたように見えた。

#### 【発明の概要】

#### 【発明が解決しようとする課題】

#### 【0005】

コンピュータデータベースはしかし、多数の問題に悩まされている。各組織体、企業、または機構は、自分自身のデータベースのコピーをインストールする。しかしながら、ユーザは、間もなく既に提供されている標準オブジェクトおよび標準アプリケーションに加えて、自分のデータベースへ、自分自身のカスタムオブジェクトおよびアプリケーションを加えたいと望むようになった。カスタマイズの要望は、異種のスキーマ、即ちデータベースに保存されている情報の型の組織体、ならびに種々のユーザが実施しているスキーマに依存するアプリケーションをもたらす。そして異種のスキーマは、種々の組織体のユーザが相互間で情報またはアプリケーションを共有しようとするいかなる希望も絶ってしまった。

#### 【課題を解決するための手段】

#### 【0006】

いくつかの態様において、マルチテナントデータベース環境においてカスタムアプリケーションを作成し、エクスポートし、閲覧してテストし、およびインポートするための機構および方法を提供する。これらの機構および方法は、態様が、組織体の境界を越えてアプリケーションを共有するための手段を提供することを可能にする。組織体の境界を越えてアプリケーションを共有する能力は、マルチテナントデータベースシステム中のテナントが、例えばアプリケーションを、容易にまた効率的に、マルチテナント環境中の他のテナントとの間でインポートしエクスポートする、したがって共有することを可能にする。本明細書で用いる用語マルチテナントデータベースシステムとは、顧客組織体(すなわちテナント)が、一つの論理データベース中のアプリケーションおよびデータならびにデータベース資源を共有することを可能にするマルチテナントアーキテクチャを実装している、データベースシステムを指す。マルチテナントデータベース環境では、データベーステーブルそれ自体さえをも、テナントを越えて共有することができる。例えば、データモデル中の各エンティティが、各テナントに対する行を識別する組織体\_idの列を含むこともできよう。あるテナントに関するクエリおよびデータ操作は、この(インデックス付)組織体\_id列を通り抜けてフィルタされるために、適切なセキュリティおよび仮想私的データベースの外観が保証される。この戦略により、マルチテナントデータベースの態様が、アカウント、コンタクト、リード、およびオポチュニティエンティティなどの標準エンティティを、顧客へ露出することが可能になる。

#### 【0007】

いくつかの態様では、エクスポートされるテナント組織体に対して、マルチテナントデータベースシステム中の他のすべての組織体に対する同一の物理的記憶機構およびスキーマを用いることにより、将来テナント組織体がシームレスにアップグレードされることができ、これを保証する能力が提供される。

## 【0008】

いくつかの態様では、アプリケーションのエクスポートを促進するために、アプリケーションを定義するパッケージを作成する。パッケージは、アプリケーションを定義するメタデータおよび他の材料を含むであろう。受信者の組織体へパッケージがインポートされると、パッケージは組織体の寿命がなくなるまで、論理的に別々にしておかれる。このようにして、いくつかの態様ではインポートされたパッケージの独自性が保たれ、受信者組織体のユーザが個別の項目としてのパッケージをナビゲートすることが可能になる。いくつかの態様では、パッケージ中のインポートされたオブジェクトの変更ならびにパッケージの将来のアンインストールを不可能にする能力が提供される。

## 【0009】

一つの局面では例として、ある組織体においてアプリケーションを定義するパッケージ(メタデータを含む)を作成し、そのパッケージを別の組織体に供給することにより、アプリケーションの機能性を定義して、それをマルチテナントデータベース中の他の組織体と交換する能力が提供される。任意で、作成する組織体は、アプリケーションのディレクトリ中にパッケージからのメタデータを公表することにより、アプリケーションへのアクセスを提供してもよいし、または別の組織体に、例えばURLなどのパッケージへのリンクを提供してもよい。アプリケーションパッケージへのアクセスに関心を有する組織体は、そのような公表されたパッケージを閲覧してテストし、ならびにパッケージをインポートしてインストールしてもよい。

## 【0010】

別の局面では、アプリケーションのディレクトリが提供される。該ディレクトリは、ユーザが開発したアプリケーションのオンラインカタログである。ディレクトリには、公開されて利用可能なアプリケーション、およびパッケージへの正確なリンクの知識がなければアクセスできないプライベートアプリケーションが含まれる。公開ディレクトリにアクセスするビジターは、利用可能なアプリケーションについて学習し、アプリケーション制作者によって開発されたアプリケーションの説明を閲覧し、インポート(インストール)URLへアクセスすることができる。アプリケーション開発者は、公開かプライベートかにかかわらず、彼らが作成したアプリケーションの説明を提出し、維持するためにディレクトリのサイトをナビゲートする。品質を保証するために、内部ディレクトリ管理者が、アプリケーションエントリを、公開ディレクトリに加える前にレビューすることができる。一つの局面では、ディレクトリには、ユーザが提出した評価、コメント、およびインストール数の形の社会的フィードバックが含まれ、潜在的な受取人に各アプリケーションの評判および価値を測る手法を提供する。ビジターは、ビジネスソリューションカテゴリによって、または最も高い評価、開発者名、最も高い人気、および他の順序付け機構によってソートして、アプリケーションをブラウズすることができる。

## 【0011】

本発明の別の局面では、複数の組織体のためのデータおよびオブジェクトを保存するマルチテナントデータベースを含むマルチテナントデータベース環境において、アプリケーションを共有するための方法を提供する。方法には典型的には、第1の組織体に関連する一つまたは複数のメタデータオブジェクトのセットを参照するパッケージメタデータオブジェクトを作成する工程、パッケージメタデータオブジェクトをデータベースシステムに保存する工程、および第2組織体のユーザに、パッケージメタデータオブジェクトへのアクセスを許可する工程を含む。ある局面では、典型的には、一つのオブジェクトおよび該オブジェクトに依存する一つまたは複数のオブジェクトのセットを含むパッケージを作成する工程(該オブジェクトおよび依存するオブジェクトは、第1の組織体に関連している)、パッケージをデータベースシステムに保存する工程、および第2組織体のユーザへパッケージへのアクセスを許可する工程を含む方法が提供される。ある局面では、方法には、パッケージメタデータオブジェクトまたはパッケージの妥当性を確認する工程が含まれる。

## 【0012】

図面および請求の範囲を含む明細書の残りの部分の参照により、本発明の他の特徴および長所が理解されるであろう。本発明のさらなる特徴および長所、ならびに本発明の様々な態様の構造および働きを、添付した図面に関連して、以下に詳細に記述する。図面中では、同じ参照番号は、同一のまたは機能的に類似の要素を示す。

【図面の簡単な説明】

【0013】

【図1】マルチテナントデータベースシステムを用いるであろう環境を示す。

【図2】図1の要素および要素間の様々な相互接続を示す。

【図3】ある態様におけるアプリケーションのディレクトリのデータモデルを示す。

【図4 - 1】ある態様におけるディレクトリエントリオブジェクトおよび他のオブジェクトの定義を示す。 10

【図4 - 2】ある態様におけるディレクトリエントリオブジェクトおよび他のオブジェクトの定義を示す。

【図5】ある態様におけるソリューションズカテゴリの階層構造を示す。

【図6】ある態様における、ユーザがパッケージを作成することを可能にするGUI画面の例を示す。

【図7】ある態様における、作成されたパッケージの情報(それに含まれるアイテムを含む)を示しているGUIのスクリーンショットの例を示す。

【図8】ある態様における、プロジェクトデータモデル定義およびプロジェクトメンバデータモデル定義の例を示す。 20

【発明を実施するための形態】

【0014】

態様では本発明は、マルチテナントデータベースシステム中で、カスタマイズされたアプリケーションを作成し交換するためのシステムおよび方法を提供する。

【0015】

顧客は、既に提供されている標準オブジェクトおよび標準アプリケーションに加えて、自分のデータベースシステムへ自分自身のカスタムオブジェクトおよびアプリケーションを加えたいであろう。顧客がそれ自身の物理的データベースを持つ従来のクライアント/サーバアプリケーションでは、カスタムオブジェクトの付加は、典型的には、そのデータベースに対して、DDL (データ定義言語)によって新しい物理スキーマ - テーブルおよび列を作成することにより行われる。オンラインマルチテナントデータベースシステムでは、このアプローチは、様々な理由で受け入れられない可能性がある。例えば、多数のテナント(例えば約1,000または10,000またはそれ以上のテナント)を有するデータベースシステムについては、すべての所望のスキーマを合わせると、基礎データディクショナリカタログ(例えばOracleディクショナリ)の手に負えなくなるであろう。さらに、これらの全スキーマオブジェクトの保守は、DBA (データベース管理者)にとって不可能に近い負担になるであろう。さらに、最近のリレーショナルデータベースは、組織体が論理的に独立したままであることができるほど十分には、オンラインDDL(極めて同時並行的な業務処理システムでは)をサポートしていない。具体的には、一つの組織体によるスキーマの生成は、他のすべての顧客に対してアプリケーションをロックし、承諾しがたい遅れを引き起こす可能性がある。 30 40

【0016】

多くのアプリケーションプラットフォームは、インストールすることができる種々のアプリケーションのコンセプトを有する。例えばWindowsは、Linuxおよび他のオペレーティングシステムもそうであるように、インストールすることができるアプリケーションを有している。いくつかのウェブサイトは、ダウンロードおよびインストール用のアプリケーションをブラウズし選択することを可能にする。例えばCNETは、PCベースのアプリケーションをダウンロードすることができるdownload.comのサイトを有する。他の企業ソフトウェアツールキットは、メタデータのパッケージを指定し、一つの環境からそれをエクスポートし、別の環境にインポートする能力を有する。例えばPeoplesoftは、odbc接続またはフ 50

ラットファイルによるインポート/エクスポートのいずれかを用いて、これを行う能力を有する。フラットファイルおよびodbcに基づく方法によると、旧バージョンを用いて定義されたかまたはエクスポートされたパッケージをインポートすることができないという危険性が存在する。Microsoftはオフィスディレクトリを有し、それにより中心となるウェブサイトから有用なスプレッドシートまたはワード文書テンプレートをダウンロードすることが可能になる。しかし、これらのシステムでは、マルチテナント環境において、ユーザがアプリケーションを容易にかつ効率的にインポートし、エクスポートすることが可能にはならない。これらのシステムではまた、インポートされたアプリケーションの独自性を保持せず；別個のアイテムとしてそれにナビゲートすることができない。他のシステムでもまた、将来、インポートしたアプリケーション中のインポートしたオブジェクトの変更を出来なくすることおよびアプリケーションをアンインストールすることは、できない。

10

**【 0 0 1 7 】**

図1は、マルチテナントデータベースシステムが用いられるであろう環境を示す。図1に示すように、任意のユーザシステム12は、ネットワーク14を介してマルチテナントデータベースシステム(MTS) 16と対話することができよう。それらのユーザシステム12のユーザ達は、種々の容量のユーザであろうし、また特定のユーザシステム12の容量は、現在のユーザの権限(権限レベル)によって、完全に決定されるであろう。例えば、販売員が特定のユーザシステム12を用いてMTS 16と対話している場合には、そのユーザシステムはその販売員に割り当てられる容量を有する。しかしながら、管理者がそのユーザシステムを用いてMTS 16と対話している間は、そのユーザシステムはその管理者に割り当てられる容量を有する。このように、異なるユーザは、ユーザの権限レベルに依存して、タブおよびタブセット定義ならびにプロファイル情報を含むアプリケーションおよびデータベース情報への、アクセスおよび改変に関して異なる資格を有するであろう。

20

**【 0 0 1 8 】**

ネットワーク14は、LAN(ローカルエリアネットワーク)、WAN(広域ネットワーク)、無線ネットワーク、2地点間ネットワーク、星状ネットワーク、トークンリングネットワーク、ハブネットワーク、または他の構成であり得る。現在使用されているネットワークの最も普通の型は、しばしば大文字「I」で示され「Internet」を指す、ネットワークのグローバルインターネットネットワークなどの、TCP/IP(転送制御プロトコルおよびインターネットプロトコル)ネットワークであり、それが本明細書の多くの実施例中で用いられることになる。しかし、TCP/IPは現在好ましいプロトコルではあるが、本発明が用いるであろうネットワークはそれに限定されないことは当然である。

30

**【 0 0 1 9 】**

ユーザシステム12は、TCP/IPを用いてMTS 16と通信し、またより高いネットワークレベルで、HTTP、FTP、AFS、WAP等の、他の一般的な通信用インターネットプロトコルを用いてもよい。一例として、HTTPを用いる場合は、ユーザシステム12は、HTTPメッセージをMTS 16のHTTPサーバに送受信するための、一般に「ブラウザ」と呼ばれるHTTPクライアントを含むであろう。そのようなHTTPサーバが、MTS 16とネットワーク14の間の唯一のネットワークインターフェースとして実装されるかもしれないが、しかし他の技術もまた、あるいは代わりに、用いられるかもしれない。いくつかの実装例では、MTS 16とネットワーク14の間のインターフェースには、負荷をバランスさせかつ入力してくるHTTPリクエストを複数のサーバに平等に分配するラウンドロビンHTTPリクエスト分配器などの、負荷分割機能が含まれる。複数のサーバの各々は、少なくともそのサーバにアクセスしているユーザに関するMTSのデータへアクセスできる。

40

**【 0 0 2 0 】**

一つの局面では、図1に示すシステムはウェブに基づく顧客関係管理(CRM)システムを実装する。例えば一つの局面では、MTS 16は、CRMソフトウェアアプリケーションを実装しおよび実行し、ならびにユーザシステム12との間で関係するデータ、コード、フォーム、ウェブページおよび他の情報の授受を提供し、またデータベースシステムに関係するデー

50



た、オブジェクトおよびウェブページの内容をデータベースシステムに保存しかつ取出すように構成された、アプリケーションサーバを含むことができる。マルチテナントシステムでは好ましくは、一つのテナントのデータが論理上他のテナントのデータと分離しているように保たれるように、テナントデータを配置して、一つのテナントが別のテナントのデータに(そのデータが明白に共有されていない限り)アクセスできないようにする。いくつかの局面では、システム16は、CRMアプリケーションでない、またはそれに加えて、アプリケーションを実施する。例えばシステム16は、CRMアプリケーションを含む、複数のホストされた(標準およびカスタムの)アプリケーションへのテナントアクセスを提供することができる。

**【0021】**

図1にMTS 16の要素の一つの配置を示す。前記要素には、ネットワークインターフェース20、テナントデータ用記憶装置22、MTS 16および恐らく複数のテナントにアクセス可能なシステムデータ用記憶装置24、MTS 16の各種機能を実行するためのプログラムコード26、ならびにMTSシステムプロセスおよびテナント特異的プロセスを実行する(アプリケーションホスティングサービスの一部としてアプリケーションを実行するなど)ためのプロセス空間28、が含まれる。

**【0022】**

図1に示すシステムのいくつかの要素には、ここで詳細に説明する必要のない従来から周知の要素が含まれる。例えば、各ユーザシステム12は、デスクトップパソコン、ワークステーション、ラップトップ、PDA、携帯電話または任意のWAP対応デバイス、あるいはインターネットまたは他のネットワーク接続に直接もしくは間接的に接続することができる任意の他の計算装置を含むことができる。ユーザシステム12は、典型的にはHTTPクライアント(例えば、Microsoftのインターネットエクスプローラブラウザ、Netscapeのナビゲーターブラウザ、Operaのブラウザ、あるいは携帯電話、PDAまたは他の無線装置の場合のWAP対応ブラウザ、などのブラウザプログラム)を実行して、ユーザシステム12のユーザ(例えばマルチテナントデータベースシステムの加入者)が、ネットワーク14を通じてMTS 16から利用可能な、情報、ページおよびアプリケーションにアクセスし、処理しおよび閲覧することを可能にする。各ユーザシステム12はまた典型的には、MTS 16または他のシステムやサーバによって提供されるページ、フォーム、アプリケーションおよび他の情報に関連して、ディスプレイ(例えばモニター画面、LCDディスプレイ、その他)上にブラウザによって提供されるグラフィカルユーザインターフェース(GUI)と対話するために、キーボード、マウス、タッチスクリーン、ペン等の、一つまたは複数のユーザインターフェース装置を含む。例えばユーザインターフェース装置を用いることにより、タブおよびタブセットを選択し、アプリケーションを作成し修飾し、あるいはユーザが、例えば米国特許出願第11/075,546号(参照によりその全体が本明細書に組み入れられる)に述べられているような様々なGUIページと対話することが可能になる。

**【0023】**

上に議論したように、本発明はインターネット(ネットワークの特異的グローバルインターネットネットワークを指す)と一緒にの使用に適している。しかし、インターネットの代わりに、イントラネット、エクストラネット、仮想プライベートネットワーク(VPN)、非TCP/IPに基づくネットワーク、任意のLANまたはWAN等の、他のネットワークを用いることができることは当然である。

**【0024】**

一つの態様によれば、各ユーザシステム12およびその構成要素のすべてが、Intel Pentiumプロセッサなどの中央処理装置を用いて実行されるコンピュータコードを含む、ブラウザなどのアプリケーションを用いて、オペレータにより設定可能である。同様に、MTS 16(および一つより多いMTSが存在する場合は、MTSの追加のインスタンス)ならびにそれらの構成要素の全ては、Intel(登録商標) Pentiumプロセッサなどの中央処理装置または複数の処理装置を用いて実行されるコンピュータコードを含むアプリケーションを用いて、オペレータにより設定可能であろう。MTS 16を操作しかつ構成して、ウェブページ、ア

10

20

30

40

50

アプリケーションならびに本明細書に記述される他のデータおよびメディアコンテンツと相互通信し、処理するためのコンピュータコードを、好ましくはダウンロードし、ハードディスクに保存する。しかし、プログラムコードの全体またはその一部を、ROMまたはRAMなどの周知の任意の他の揮発性もしくは不揮発性メモリ媒体またはデバイスに保存しても、あるいはプログラムコードを保存することができるコンパクトディスク(CD)媒体、デジタル多用途ディスク(DVD)媒体、フロッピーディスク等の、他の情報記憶媒体に与えてもよい。さらに、プログラムコード全体またはその一部が、ソフトウェアソースから例えばインターネットを介してまたは周知の任意の別のサーバから送信されダウンロードされてもよく、あるいは任意の通信媒体および周知のプロトコル(例えばTCP/IP、HTTP、HTTPS、Ethernet、その他)を用いて、周知の任意の他の従来型ネットワーク接続(例えばエクストラネット、VPN、LAN、その他)を介して送信されてもよい。本発明のいくつかの局面を実施するためのコンピュータコードを、サーバまたはサーバシステム上で実行することができる任意のプログラミング言語：例えば、C、C++、HTML、任意の他のマークアップ言語、Java (商標)、JavaScript、VBScriptなどの任意の他のスクリプト言語、および周知の他の多くのプログラミング言語などで、実行することができることを、さらに認識することになる(Java (商標)は、Sun Microsystems, Inc. の商標である)。

10

## 【 0 0 2 5 】

一つの態様によれば、MTS 16のテナントとしてのユーザシステム12によるアクセスをサポートするために、各MTS 16を、ユーザシステム12にウェブページ、フォーム、アプリケーション、データおよびメディアコンテンツを提供するように構成する。このように、MTS 16は、データが共有されない限り、各テナントのデータを分離したままにしておくための、セキュリティ機構を提供する。一つより多いMTSを用いる場合、それらは、互いにすぐ近く(例えば、単一のビルまたはキャンパス内のサーバファーム中)に位置してもよいし、またはそれらは、互いに遠く離れた位置(例えば都市Aにある一つまたは複数のサーバ、および都市Bにある一つまたは複数のサーバ)に分配されてもよい。本明細書で用いるように、各MTSは、局所的に、または一つまたは複数の地理的な位置にわたって分散された、論理的および/または物理的に接続された一つまたは複数のサーバを含むことができよう。さらに、用語「サーバ」とは、当技術分野において周知のように、処理用ハードウェアおよびプロセス空間を含むコンピュータシステム、ならびに関連する保存システムおよびデータベースアプリケーション(例えば、OODBMSまたはRDBMS)を含むことを意味する。さらに「サーバシステム」と「サーバ」は、本明細書でしばしば交換可能に用いられる、と理解するべきである。同様に、本明細書に記述されるデータベースは、単一のデータベース、分散型データベース、分散型データベースの集合、冗長オンラインまたはオフラインバックアップまたは他の冗長度を有するデータベース、その他として実装することができる、また分散型データベースまたは記憶装置ネットワーク、および関連する処理用知能を含んでもよいであろう。

20

30

## 【 0 0 2 6 】

図2は、ある態様におけるMTS 16の要素およびこれらの要素間の様々な相互接続を示す。この例では、ネットワークインターフェースは、一つまたは複数のHTTPアプリケーションサーバ100として実装される。さらに、個々のテナントプロセス空間104、システムデータベース106、テナントデータベース108およびテナント管理プロセス空間110を含む、システムプロセス空間102が示されている。テナントデータベース108は個々のテナント記憶領域112に分割されてもよく、それは物理的配備または論理的配備のいずれでもよい。各テナント記憶領域112内では、ユーザ記憶部114を同様に各ユーザに割り当てることができよう。

40

## 【 0 0 2 7 】

各アプリケーションサーバ100を、データベースシステム(例えばシステムデータベース106およびテナントデータベース108)に、異なるネットワーク接続を介して、通信可能に接続することができることもまた当然である。例えば、一つのサーバ1001は、インターネット14を介して接続され、別のサーバ100N-1は、直接のネットワークリンクを介して接続

50

され、また別のサーバ100Nは、さらに異なるネットワーク接続を介して接続されるかもしれない。転送制御プロトコルおよびインターネットプロトコル(TCP/IP)は、サーバ100とデータベースシステム間で通信するための好ましいプロトコルであるが、しかし用いるネットワーク相互接続に依存して、他のトランスポートプロトコルを用いてシステムを最適化することができることは、当業者には明白であろう。

**【0028】**

局面において、各アプリケーションサーバ100を、任意のユーザ/組織体のリクエストを扱うように構成する。サーバプールからアプリケーションサーバを、任意の時期に任意の理由で、加えたり除去したりすることができることが望ましいので、ユーザおよび/または組織体が、特定のアプリケーションサーバ100に対してサーバの好みを持たないことが好ましい。したがって、一つの態様では、サーバ100へのリクエストを分配するために、サーバ100とユーザシステム12の間に、負荷バランス機能を実施するインターフェースシステム(例えばF5 Big-IPロードバランサ)(図示せず)を通信可能に連結する。一つの局面では、負荷バランサは、サーバ100へのユーザリクエストの経路を決めるために最小接続アルゴリズムを用いる。ラウンドロビンおよび監視応答時間などの、負荷バランスアルゴリズムの他の例もまた、用いることができる。例えば、ある局面では、同じユーザからの三つの連続リクエストが三つの異なるサーバに届くこともあり得ようし、また異なるユーザからの三つのリクエストが同じサーバに届くこともあり得よう。このようにMTS 16は、いろいろなユーザおよび組織体に亘って種々のオブジェクト、データおよびアプリケーションを保存し、それらにアクセスする、マルチテナントである。

**【0029】**

記憶装置の例として、一つのテナントは、各販売員が自分の販売プロセスを管理するためにMTS 16を用いる販売部隊を運用する会社であるかもしれない。この場合、ユーザは、コンタクトデータ、リードデータ、顧客フォローアップデータ、売上データ、目標および経過データ、その他(それらはすべて、そのユーザの個人的販売プロセスに適用可能である)を保持する(例えばテナントデータベース108中に)かもしれない。好ましいMTS配備では、このデータおよびアクセス、閲覧、改変、報告、送信、計算、その他をするアプリケーションのすべてが、ネットワークアクセスだけをするユーザシステムによって維持されアクセスされることができるので、ユーザは、任意の多くのいろいろなユーザシステムから、自分の販売努力および販売サイクルを管理することができる。例えば、販売員が顧客をナビゲート中で、顧客が彼らのロビーにインターネットアクセスを持っていれば、販売員は、顧客がロビーに到着するのを待つ間に、その顧客に関する決定的な最新情報を得ることができる。

**【0030】**

各ユーザのデータは、各ユーザの雇用者に関わらず、他のユーザのデータと分離しているかもしれないが、あるデータは、テナントである所与の組織体の、複数のユーザによってまたはすべてのユーザによって、共有されまたはアクセス可能な組織体全体に亘るデータかもしれない。このように、テナントレベルに割当てられた、MTS 16によって管理されるいくつかのデータ構造が存在するかもしれないし、一方で他のデータ構造は、ユーザレベルで管理されるかもしれない。MTSは、競争者である可能性のあるテナントを含む複数のテナントをサポートするかもしれないので、データ、アプリケーション、およびアプリケーションの使用を分離しておくセキュリティプロトコルを有するべきである。さらに、多くのテナントは、自分自身のシステムを維持するよりも、MTSへのアクセスを選ぶ傾向があるので、冗長度、稼働時間およびバックアップが、追加の重要な機能であり、MTS中に実装する必要がある。

**【0031】**

ユーザ固有データおよびテナント固有データに加えて、MTS 16は、さらに複数のテナントによって使用可能なシステムレベルのデータまたは他のデータを保持する可能性がある。そのようなシステムレベルのデータには、テナント間で共用可能な業界レポート、ニュース、掲示等が挙げられるであろう。

## 【 0 0 3 2 】

ある局面では、クライアントシステム12は、アプリケーションサーバ100と通信して、MTS 16からのシステムレベルおよびテナントレベルのデータをリクエストしアップデートする(それはデータベースシステム106および/またはデータベースシステム108への一つまたは複数のクエリを必要とする)。例えば、一つの局面では、MTS 16(例えば、MTS 16中のアプリケーションサーバ100が)が、所望の情報にアクセスするように設計された一つまたは複数のSQLステートメントを含むSQLクエリを自動的に生成する。

## 【 0 0 3 3 】

各データベースを一般に、所定のカテゴリに入れられたデータを含む論理テーブルのセットなどのオブジェクトの集合として閲覧することができる。「テーブル」とは、データオブジェクトの一つの表現であり、本発明によるオブジェクトおよびカスタムオブジェクトの概念記述を単純化するために、本明細書で用いられる。「テーブル」および「オブジェクト」は、本明細書で交換可能に用いられ得ることを理解しなければならない。各テーブルは一般に、例えば、閲覧することができるスキーマ中の列またはフィールドとして、論理的に配列された一つまたは複数のデータカテゴリを含む。テーブルの各行またはレコードは、フィールドによって定義された各カテゴリのデータの事例を含む。例えば、CRMデータベースは、名前、住所、電話番号、ファックス番号、その他の基本的なコンタクト情報に関するフィールドによって顧客を記述するテーブルを含んでもよい。別のテーブルは、顧客、製品、販売価格、日付、その他の情報に関するフィールドを含む購入注文を記述する可能性もあろう。いくつかのマルチテナントデータベースシステムでは、すべてのテナントが使用するために、標準エンティテーブルが提供されるであろう。CRMデータベースアプリケーションについては、そのような標準エンティティは、アカウント、コンタクト、リード、およびオポチュニティのデータに関するテーブル(各々が所定のフィールドを含む)を含むであろう。

## 【 0 0 3 4 】

一つの局面によれば、ユーザは、カスタムオブジェクト、カスタムタブ、カスタムフィールドおよびカスタムページレイアウトを含む、自分自身のカスタムアプリケーションを設計することができる。2004年4月2日出願の米国特許出願番号第10/817,161号、表題「Custom Entities and Fields in a Multi-Tenant Database System」(その全体は参照により本明細書に組み入れられる)は、エンティティおよびフィールドなどのオブジェクトを作成しカスタマイズするための、システムおよび方法を開示する。そこに提示されたシステムおよび方法は、固定された物理的スキーマ中に、可変スキーマデータを記憶させる柔軟なアプローチを提示する。タブおよびタブセットを作成しカスタマイズして、カスタムオブジェクトとフィールド、標準オブジェクトとフィールド、およびアプリケーションと関連データの追跡、の間の関係を定義することができる。2005年3月8日出願の米国特許出願番号第11/075,546号、表題「Systems and Methods for Implementing Multi-Application Tabs and Tab Sets」(その全体は参照により本明細書に組み入れられる)は、マルチテナント環境において、タブおよびタブセットを作成しカスタマイズするためのシステムおよび方法を開示する。そこに記述されている、タブおよびタブセット作成および機能の簡潔な概要を、以下に述べる。

## 【 0 0 3 5 】

## カスタムタブおよびタブセット

いくつかの態様では、タブは、アプリケーションの要素への、またはデータベースオブジェクトへのユーザインターフェースとなる。タブの選択によって、そのタブによって表わされるアプリケーションのオブジェクトまたは要素へのユーザアクセスが提供される。タブセットは、一単位として働いて、アプリケーションの機能性を提供する関連するタブのグループである。新しいタブおよびタブセットを定義しタブセットビューをカスタマイズして、エンドユーザが、定義されたタブおよびタブセットによって表わされる様々なオブジェクトおよびアプリケーション要素の間を、容易にかつ便利に切り替えることができるようにする。一つの局面では、タブとタブセットを、例えば、オンデマンドウェブに基

10

20

30

40

50

くホストされたアプリケーション環境などの複数アプリケーション環境中で、アプリケーションの間を切り替える手段として、用いてもよい。

【0036】

タブセットには典型的に、名前、ロゴおよびタブの順序付きリストが含まれる。タブセットを典型的には、グラフィカルユーザーインターフェース(GUI)環境において、例えばユーザのコンピュータシステム上で作動するブラウザアプリケーションを用いて閲覧する。標準タブセット定義が、ホストシステム(例えばMTS 16)によって提供されてもよい。標準タブセットは、例えばソースからインポートされた、タブのあらかじめ定義されたセットであり、該ソースは、どのタブ、タブセット、およびデータが、テナントまたはユーザに最初に供給されるかを決定する能力(例えばテンプレートとなる能力)を提供する。標準タブセットの一例が、salesforce.comのウェブサイトにより、その加入CRMサービスを通じて提供される。これらの標準タブセットを用いて、ユーザに、アカウント、コンタクト、リード、およびオポチュニティエンティティなどの、標準テーブルまたはエンティティへのアクセスが提供される。別の例として、salesforce.comのサービスでは、ユーザは、標準エンティティのためのカスタムフィールドだけでなく、カスタムエンティティも作成することができ、ユーザは、カスタムエンティティおよびカスタムフィールドを表わすタブを含む、タブセットを作成することができる。

10

【0037】

ユーザは、カスタムタブセットおよびカスタムタブを作成してもよい。好ましくは、管理者レベルのユーザのみが、それらの保存された権限に基づいて、タブセット生成機能を提供される。さらに、ユーザは、表示されるタブの順序およびタブセットの中のどのタブを表示するかを含む、自分のタブセットのビューをカスタマイズしてもよい。ユーザが便利にタブを組織体することを可能にするために、もし所望であれば、各タブが任意のおよび全てのタブセットに現われてもよい。好ましくは、任意のユーザが、タブの組合せおよび順序を編集することができるが、しかしロゴを改名したりまたは交換することはできない；タブセットの命名およびロゴの選択は、好ましくは管理者レベルのみの機能である。例えば、管理者が新しいタブセットを作成し、既存のタブセットをカスタマイズしてもよい。すべてのタブセットに対して、管理者は、どのタブが含まれるかを、およびタブを表示する順序を指定することができる。組織体特異的なタブセットに対して、管理者はまた、名前を指定し、任意でロゴを提供することができる。例えばSalesforceおよびSupportforceのタブセットなどの、salesforce.comによって提供されるタブセットのような、ホストシステムによって提供される標準タブセットに対しては、管理者は名前またはロゴを変更することが禁止され、また管理者が標準タブセットを削除することもできない。好ましくは、任意のユーザは、見る権限を有するすべてのタブセットの自分のビューを完全にカスタマイズすることができる。ユーザが閲覧することができる(また使用できる)タブは、そのユーザの権限レベルに基づく。各タブセットのプロファイルは、管理者レベルのユーザが、タブおよびタブセットのプロファイルレベル閲覧可能性を設定できるようになっており、例えば、その結果、ある権限レベルのユーザのグループは、あるタブまたはタブセットを閲覧する(また使用する)ことを制限され、したがって、制限されたタブまたはタブセットが参照するあるオブジェクトおよびアプリケーションにアクセスすること、または

20

30

40

【0038】

このように、一つの局面では、タブセットは、既存のプロファイル - レベルのタブ可視性定義を覆うフィルタと考えることができる。管理者は、各タブセットフィルタに含まれるデフォルトタブを設定するが、しかし各ユーザは、好きなように無視することができ - 唯一彼らに変更することができないものは好ましくはタブセット名およびロゴである。正味の結果は、タブセットは極めて軽量であり、柔軟であるということである。タブセットに特別な意味は強調されない；各ユーザは一般に、自分が望むようにタブセットを用いることができる。

【0039】

50

#### アプリケーションの作成および交換

一つの態様では、ユーザは、アプリケーションを作成し、掲載し、および交換する能力を有する。一つの局面では、本明細書で用いるようにアプリケーションは、マルチテナントデータベースのデータモデル、ユーザインターフェースおよびビジネスロジックを定義する、マルチテナントデータベースのセットアップデータ(例えばメタデータ)のグループ即ちパッケージである。例えば、ユーザは、アプリケーションを作り上げるメタデータのグループを論理的に定義するタブセットを定義してもよい。本明細書で用いるように「パッケージ」とは、組織体の中でのアプリケーションで用いられるメタデータオブジェクトのセットを参照するメタデータオブジェクトである。本明細書で用いるように「組織体」とは、マルチテナントシステム中の単一テナントおよび/またはマルチテナントシステム中の単一テナントのためのメタデータのセット(アプリケーションデータおよびメタデータの両方)を意味することができる。

10

#### 【0040】

一つの態様では、本発明は、アプリケーションを作り上げる全てのセットアップデータ(例えば、カスタムオブジェクト定義、ページレイアウト定義、ワークフロールール、その他)を含むメタデータ「パッケージ」を、ユーザが正確に定義できるようにして、タブセットの概念を洗練させる。パッケージは、0、1、またはそれ以上のタブセットを含んでもよい。ユーザは次に、一つの「ソース」組織体から、このパッケージを、データベースシステム中のいかなるテナントにも関連していない「コンテナ」組織体へエクスポートすることができる。エクスポートされたパッケージは、アプリケーションディレクトリに登録されるかまたはリストされる。パッケージを作成するかおよび/またはエクスポートするユーザを、本明細書では、ソースユーザまたはエクスポートユーザと呼ぶ。ソースユーザは、アプリケーションディレクトリの公開部分にパッケージ名およびアプリケーションをリストすることを選択できる。別のユーザは、別個の「標的」組織体へパッケージをインポートして、この組織体がアプリケーションを、作成する組織体と無関係に用いることができるようにすることができる。パッケージを閲覧したり、および/またはインポートするユーザを、本明細書では、閲覧するユーザまたはインポートするユーザと呼ぶことにする。

20

#### 【0041】

インポート用パッケージを選択するときには、サーバシステム上のコードがコンテナ組織体からメタデータを取り出し、適切な受信者組織体を選び、コンテナ組織体からメタデータを読み出し、そのメタデータを受信者組織体に書き込む。インポートするユーザは、任意の衝突するメタデータ(例えばオブジェクトまたはフィールド名)を、例えば受信者組織体のオブジェクト、フィールド、その他を停止することによってまたは改名することによって、解消することができる。インポートプロセスが、好ましくは任意の既存の受信者組織体のフィールドに上書きしないことに注意するべきである。さらに、インポートユーザは、インポートされたアプリケーションをアンインストールすることができる。

30

#### 【0042】

パッケージ作成プロセスならびにエクスポートおよびインポートプロセスの態様についてここで記述する。

40

#### 【0043】

##### パッケージ作成

ソース組織体では、ソースユーザが適切なセットアップデータ(メタデータ)を選択することによりパッケージ定義を作成する。これが、アプリケーションを作り上げるセットアップデータのグループを定義する。例えば、ソースユーザは、高レベルタブセット(タブのグループ)を選んでよい。システムは次に、パッケージに含まれるメタデータのオブジェクト依存関係を自動的に決定する。例えば、一つの局面では、システムは、依存関係発見プロセス(例えば、オブジェクトスキーマ中を「スパイダーして(リンクを張って)」オブジェクト依存関係を探索するプロセス)を自動的に実行して、タブセットによって提供される機能を用いるために要求される全ての関連オブジェクトを決定する。一つの局面

50

では、2パスでこれを行う：最初にトップダウン、次にボトムアップで、システム中のオブジェクトとタブセットが同定したオブジェクトとの間のすべての逆相関を決定する。追加してまたは代わりに、ソースユーザは、パッケージに含めるセットアップデータの集合を明示的に指定してもよい。

#### 【 0 0 4 4 】

一つの局面では、パッケージ中に、次のアイテム(数個のメタデータ)の一つまたは複数が含まれることができる。

1. タブセット(これは、タブセット定義によって参照されるすべてのものをコピーするであろう)
2. カスタムオブジェクト 10
  - a. カスタムフィールド、
  - b. 関連性(マスター-ディテールおよびルックアップ)
  - c. ピックリスト値、
  - d. ページレイアウト、
  - e. サーチレイアウト、
  - f. 関連リストレイアウト、
  - g. 公開リストビュー、
  - h. カスタムリンク
  - i. カスタムオブジェクトに関連する任意の他のアイテム
3. カスタムタブ定義 20
4. S-コントロール(それは一つの局面において、カスタムユーザインターフェースおよびビジネス論理処理を遂行するJavaScriptプログラムである)。パッケージ作成者は、タブまたはページの一部で作動するs-コントロールを指定して、ユーザがタブまたはページをナビゲートした場合に、アプリケーションサーバが、実行用のJavaScriptをブラウザ上にダウンロードするようにできる。必要な場合には、サービスヘデータバックアップを保存するためにAPIを用いる。
5. カスタムレポート(新しいアプリケーションそれぞれのために一つのレポートフォルダが作成されるであろう)
6. ダッシュボード
7. Eメールテンプレート 30
8. ドキュメント
9. プロファイルパッケージ(カスタムオブジェクト用のFLSを含む)(プロファイルに関連する権限データの束(後で定義される))
10. 依存ピックリスト
11. ワークフロールール
12. レコード型

#### 【 0 0 4 5 】

コピーされる追加のメタデータアイテムには以下のものが含まれてもよい。

1. 標準オブジェクト用のカスタムフィールド
2. メールマージテンプレート 40
3. ビジネスプロセス
4. 割り当てルール(ワークフロー形式)
5. 自動応答ルール(ワークフロー形式)
6. エスカレーションルール(ワークフロー形式)
7. 重大事警告 / オポチュニティリマインダー
8. セルフサービスポータル設定
9. VLO特性(一つの局面では、これには会社または組織体内の種々の「部門」の定義が含まれる。VLO特性により、アプリケーションが、データを部門別に分けることが可能になり、したがって、レポートおよびリストブラウズの範囲を制限することができる)。
10. 委任管理設定 50

## 11. ホームページの構成要素

### 【0046】

図6は、ユーザが、パッケージを作成し(例えば、ポインティングデバイスで「New」を選択することによる)、ならびに、既存のパッケージを削除し、編集し、インストール済みパッケージの歴史を閲覧し、およびパッケージをインストールするディレクトリにアクセスすることを可能にする、GUI画面の例を示す。図7は、作成されたパッケージの情報(含まれるアイテムを含む)を示すGUIスクリーンショットの例を示す。ユーザは、そのような画面を用いて、パッケージを編集し、削除し、公表することができる。

### 【0047】

ユーザは、パッケージを編集する、例えばアイテムを削除するかまたはアイテムを加える、ことを選択することができる。例えば、ユーザが、カスタムオブジェクト(またはパッケージに含まれる他のメタデータ)を削除することを選択すれば、システムはこれを検出してユーザに警告する。もしユーザが続けることを望めば、そこでメタデータはパッケージから除かれる。次回にユーザがパッケージアイテムを閲覧すれば、削除したアイテムは消失しているであろう。もしパッケージがエクスポートされてしまっても、エクスポートされたパッケージ中のメタデータは影響を受けない。この特徴は、メタデータのバックアップは与えるが、作成されたかもしれないいかなるレコードのバックアップも与えることはない(しかしこれらのレコードを、データローダまたはExcelのplug-inを用いて保存することができる)。ユーザがパッケージに新しいアイテムを追加することを決定すれば、包含のために利用可能なアイテムのピクリストが提供されるであろう。任意のアイテムが、複数のパッケージに含まれてもよい。

### 【0048】

アプリケーション(例えば、タブセットおよび/または他のメタデータアイテム)をパッケージへ加える場合は、システムは、このアプリケーションと関係する全てのカスタムアイテムを加えるべきである。これには、カスタムタブ、タブの背後のオブジェクト、およびこれらと関係する任意のカスタムオブジェクトが含まれる。理想的には、依存関係ファインダプロセス(例えばスパイダープロセス)が、追加されるアプリケーションを作り上げる任意の標準タブと関係する任意のカスタムオブジェクトもまた検出するであろう。パッケージにこれらの標準オブジェクトが含まれていなくても、接続オブジェクトおよび他の関連オブジェクトは、アプリケーション中で具体化されている機能の一部である。一定のオブジェクト依存関係を有するものは、常に含まれているべきである。例えば、ある局面では、カスタムオブジェクト上の全てのカスタムフィールドはカスタムオブジェクトと共に含まれ、全てのページレイアウトはカスタムオブジェクトと共に含まれる。ある局面では、カスタムオブジェクトに関係したリストレイアウト(それは標準オブジェクトに現われる)もまた自動的に含まれる。

### 【0049】

一旦ユーザがパッケージを定義し終えたら、パッケージの妥当性を確認する。これは、例えば、ユーザが、パッケージが完成したことを表示したとき、システムがこれを自動的に行うことができるし、あるいは例えば、ユーザが、GUI上の「妥当性を確認する(validate)」または同様のボタンを選択するなどの、妥当性確認のユーザリクエストに回答して、それが行なわれてもよい。これが、パッケージ定義に全ての必要なセットアップデータが含まれることを確実にするスパイダープロセスを呼び出す。パッケージに最初にメタデータアイテムを加えた後で、ユーザがパッケージ中のメタデータを変更した(例えば、ある関係、別のタブもしくはオブジェクト、その他を加えた)場合には、これが有用である。

### 【0050】

一つの態様では、パッケージが、パッケージに含まれるオブジェクトの主キーのセットとして、データベースに保存される。図8は、ある態様による、プロジェクトデータモデルの定義、およびプロジェクトメンバデータモデルの定義の例を示す。

### 【0051】

10

20

30

40

50



### パッケージのエクスポート

パッケージを定義した後に、ソースユーザは、パッケージをエクスポートすることを選択できる。パッケージをエクスポートすると、他の組織体がインポートすることが可能になる。一つの局面では、パッケージをエクスポートする場合、ソースユーザは、アプリケーションが他の組織体へインポートされた後に、そのカスタマイズを許可するかどうかを指定することができる。例えば、ある局面では、パッケージ定義中にフラグがセットされて(ソースユーザにより)、エクスポートされたパッケージのカスタマイズがアップグレードに従うべきかどうかを表示する。ある局面では、フラグがセットされて(ソースユーザにより)、パッケージ中の特定の構成要素/オブジェクトをとにかく変更することができるかどうかを表示する。

10

#### 【0052】

エクスポートは、一つの局面では、システムが自動的に新しい組織体定義を作成することによって、例えば「コンテナ」型の組織体(それは、エクスポートユーザによってパッケージ定義に明示的には含まれない依存関係オブジェクトメタデータを含む、パッケージ中のすべてのアイテム(メタデータ)のコピーを含む)によって、実行される。一つの局面では、このコンテナ組織体は、他の全ての組織体と同じ物理的データベース(例えばOracleデータベース)スキーマを共有する。しかし、コンテナ組織体は、異なる離れたデータベース中に存在することができる。さらに、標準の終了および課金プロセスは、適用できる場合は、好ましくはこの型の組織体(org)を無視する。ある局面では、複数のデータベースインスタンスがデータベースシステム環境の中に存在する場合に、エクスポートされたパッケージは、データベースインスタンスの一つにコンテナ組織体として保存される。コンテナ組織体は、一つまたは複数のあるいは全ての残存データベースインスタンスに複製される。複数のデータベースインスタンスは、例えば異なる地理的領域に関係していてもよい。例えば、一つのデータベースインスタンスはヨーロッパ(EP)に関係しており、また一つは北アメリカ(NA)に関係しているかもしれない。このようにして、データベースインスタンスのジグザグ状で独立したアップグレードが促進される。例えば、データベースシステムにとっての新版/アップグレード版が異なるインスタンスで実行されたときに、パッケージのインストールは、続いて行われてもよい。例えば、NAインスタンスの前(例えば1週間か2週間前)にEPインスタンスをアップグレードする場合、NAへのパッケージXのインストールがNAインスタンスから起こり、EPへのパッケージXのインストールは、パッケージXのコピーを複製されたEPインスタンスから起こるのである。これが、類似のバージョンからのインストールが起こることを保証する。

20

30

#### 【0053】

エクスポートプロセスが完了すると、ソースユーザはコンテナ組織体の独自のキーを含むURLを受け取る。URLを知っている人は誰でも、同定された「コンテナ」組織体中のパッケージにアクセスし、自分の組織体の中へインポートすることができ、例えば、そのパッケージに関連したメタデータがその組織体に関連したスキーマにコピーされるかまたはインスタンス化される。ソースユーザは、別の会社の別のユーザへURLを送ってもよく、またはソースユーザは、より詳細に後述するように、ディレクトリへURLを掲載してもよい。

40

#### 【0054】

一旦パッケージがエクスポートされれば、パッケージはソース組織体ではオープンのままであり、ソースユーザはそれを変更し続けることができる。しかし、コンテナ組織体中のパッケージのコピーは、さらに変更することを好ましくはロックされる。

#### 【0055】

ソースユーザは、同じパッケージを複数回エクスポートして、複数のコンテナ組織体を作成することができる。これによって、ソースユーザはパッケージが時間とともにどのように変化するかというバージョンの履歴を作成することが可能になる。

#### 【0056】

ソースユーザがパッケージを作成して、コンテナ組織体へエクスポートした後に、彼ら

50

は任意で「デモンストレーション」組織体型の第2の新しい組織体オブジェクトを作成することができる。デモンストレーション組織体は、それ自身のユーザIDおよびパスワードを有する。IDおよびパスワードを知っているユーザは誰でも、デモンストレーション組織体にログインしてエクスポートされたパッケージを閲覧することができる。一旦ログインすれば、閲覧ユーザは、必要なオブジェクトが存在すること、およびアプリケーションが予想通りに作動することを、手動で妥当性を確認することができる。エクスポートユーザがエクスポートパッケージにサンプルデータを含めるように指示したならば、閲覧ユーザは、デモンストレーション組織体にログイン時に、サンプルデータを見ることができるか、または彼らはデモンストレーション組織体へ自分自身のサンプルデータを直接加えることができる。一つの局面では、ソースユーザは、パッケージに関するデモンストレーション組織体を、彼らが公開ディレクトリにパッケージを公表する前に、作成しなければならない。これが、ディレクトリをブラウズするユーザが、彼らがパッケージをダウンロードするかまたはインポートすることを選ぶ前に、アプリケーションを「試用する」場所を有することを保証する。例えば、パッケージはウェブ統合リンク(WIL)を含む可能性があり、WILによって使用されるサービスが信頼できることを確かめるために、デモンストレーション組織体中でWILを検査することは非常に有用である。

#### 【0057】

一旦パッケージに対してデモンストレーション組織体が作成されたならば、ソースユーザは、中央化された公開ディレクトリへパッケージを「公表」することができる。公表に際して、システムは、ディレクトリサービスにメッセージを送ることにより、中央ディレクトリにパッケージを含めるよう通知する。このメッセージは、ある局面ではユーザが公開ディレクトリからナビゲートしてコンテナ組織体に帰り、それをインポートすることを可能にするURLを含む。メッセージはまた、ある局面では、デモンストレーション組織体のURLを含む。これが、公開ディレクトリをブラウズしているユーザが「今それを試みる」ことを可能にする。彼らは、デモンストレーション組織体にログインし、機能性を徹底的に検査することができる。ある局面では、メッセージには、パッケージに関する記述データ(例えば名前、説明、その他)、およびパッケージに含まれるオブジェクトのリストが含まれる。ディレクトリはこの情報を用いて、ディレクトリをブラウズしてインポートするパッケージを捜しているユーザに、詳細な情報を提供する。公開ディレクトリに関する追加の詳細について、下記に述べる。

#### 【0058】

##### パッケージのインポート

ある組織体へパッケージをインポートしインストールするために、インポートユーザは、パッケージエクスポートプロセスによって生成されたURLへ、ディレクトリを通じてか、またはソースユーザからのメッセージを介してかのいずれかでナビゲートする。このURLは、特定のエクスポートされたアプリケーションおよびパッケージを同定する独自のキーを含む。インポートユーザは公開ディレクトリをブラウズしてこのURLを発見したかもしれないし、または、エクスポートユーザが単にそのユーザへURLをEメールしたかもしれない。インポートユーザがURLをクリックすれば、彼らはアクセスし、閲覧し、パッケージをインポートすることができる。

#### 【0059】

一つの局面では、インストールは、インストールウィザードで行なわれる一つまたは複数の工程を有する多段階プロセスである。例えば一つの局面では、工程には、ユーザがインストールしたいものを検査し確認するためのパッケージコンテンツ表示画面を提供する工程、インストーラの組織体中の既存のプロファイルのためのセキュリティを構成する工程、パッケージコンテンツをインポートする工程、および意図したユーザに対してアプリケーションを配置する工程が含まれる。インポートユーザはまた、インストールパッケージ中の任意のアイテムをカスタマイズすることを選択してもよい。

#### 【0060】

ある局面では、パッケージのインポートおよびインストールプロセス中に、下記工程の

10

20

30

40

50

いくつかまたはすべてが、行なわれる必要があるかもしれないし、または生じるかもしれない：

ユーザIDおよびパスワードの入力により受信者組織体にログインする。これはパッケージをインポートすることになる受信者組織体用のユーザIDである。

任意で、エクスポーターは、パスワードでパッケージを保護した可能性がある。その場合は、インポートユーザは、それをインポートできる前にパッケージパスワード(これは受信者組織体にログインするために必要なユーザパスワードとは異なるパスワードである)を入力しなければならない。

パッケージ中のオブジェクト名が受信者組織体の中のセットアップデータと衝突する場合は、インポートプロセスが失敗するであろう。インポートユーザは、受信者組織体内の衝突するオブジェクトのオブジェクト名を変更し、インポートプロセスを再開することができる。

10

インポートプロセスの間、矛盾するメタデータのアップデートを防止するために、受信者組織体はロックされる。

インポートプロセスは、インポートユーザがパッケージをインポートする適切な組織体の権限を持っていることを確かめる検査を行う。

インポートユーザは、パッケージ中のソース組織体特異的参照から、受信者組織体に適切な値へのマッピングを定義するように求められる。例えば、インポートユーザは、ユーザID、プロフィールまたは役割を指定するように促されるであろう。

セットアップデータが、「展開」モードで受信者組織体にコピーされる。これによって、受信者組織体内のユーザにそれを配置する前に、インポートユーザが、アプリケーションが正確に機能することを実証することが可能になる。

20

インポートプロセスは、パッケージを悪意のある機能について精査する。例えば、それは、第三者ウェブサイトヘータを送信する可能性のある、任意のウェブ統合リンク(WIL)を検査することができる。

もしパッケージ定義中に特定されれば、インポートユーザは、パッケージ中のいかなるセットアップデータも、それをインポートした後に変更することができない。例えば、もしパッケージ定義中に特定されれば、インポートユーザは、それをインポートした後に、カスタムオブジェクトにフィールドを追加または消去することができない。カスタムオブジェクト編集画面の機能が、受信者組織体でのオブジェクトの任意の編集を許可する前に、パッケージ定義テーブルを検査することにより、これを実施する。

30

#### 【0061】

インポートユーザは、任意でパッケージを「アンインストール」することができる。システムは、どのメタデータオブジェクトがパッケージに属するかという追跡記録を保持する(例えばパッケージデータベーススキーマを通じて)ので、これを実施することができる。

#### 【0062】

ある局面では、パッケージをアップグレードできる。例えば、もし公表者/エクスポートユーザがソースパッケージを変更すれば、インポートユーザは、加入者が最初にパッケージをインポートして以来作成してきた任意のデータ行を保存しながら、公表者によって行なわれた変更を彼らの組織体に取り入れることを選択することができる。ある局面によれば、一つまたは複数のフラグをパッケージ定義中にセットして、パッケージのカスタマイズができるかどうか、どの程度できるか、およびアップグレードできるかを決定することができる。一つの局面では、「管理可能」フィールドが、特定のオブジェクトのカスタマイズがアップデートに従うかどうかを識別するために提供される。例えば、パッケージまたはパッケージ中のオブジェクトに管理済みとマークされた場合は、ユーザは、パッケージまたはオブジェクトをカスタマイズすることを認められ、パッケージのアップグレードの際に、これらのカスタマイズは変更されないことになる。別の局面では、「制御」フィールドが、公表者および/または加入者がオブジェクトを改変してもよいかどうかを識別するために提供される。別の局面では、誰かがオブジェクトを変更できるかできない

40

50

かを識別するために、「不変」フィールドが提供される。例えば、パッケージが公表された後は、誰もそれを修飾することができないように、ソースユーザが不変フラグをセットすることができる。アップグレードの際に、アップグレードプロセスは、カスタマイズを維持する範囲を決定するために、これらの各フィールド(存在する場合は)の検査を実行する。

#### 【0063】

##### アプリケーションディレクトリ

本発明はまた、アプリケーションの中央ディレクトリを提供し；ソースユーザは中央ディレクトリにパッケージを登録することができる。一つの局面では、中央ディレクトリには、任意の人による使用を意図した公表されたパッケージを含む公開部分、および一般的な使用を意図しないパッケージ(例えばソースユーザによって選択されたインポートユーザが使用することを意図したパッケージ)を含むプライベート部分が含まれる。そのディレクトリは、他のユーザが公表されたアプリケーションをブラウズし、自分の組織体へどれをインストールしたいかを定めることを可能にする。一つの局面では、中央ディレクトリはカテゴリ階層によって組織され、それにより、ユーザが、自分が関心を示すアプリケーションの型のカテゴリによって、探索しブラウズすることを可能にする。一つの局面では、ディレクトリは、ユーザが「今それを試みる」ことを可能にする、即ち、ユーザは、組織体へそれをインストールする前に、パッケージを含むデモンストレーション組織体を見ることができる。別の局面ではディレクトリは、提出物が公開ディレクトリに現われる前に、受理可能であると保証する、自動承認プロセスを提供する。別の局面ではディレクトリは、アプリケーションの(インポート)ユーザが、アプリケーションの有用性および品質について投票することを可能にする評価システムを含む。この投票は、公開ディレクトリに現われ、他のユーザが見ることができる。

#### 【0064】

ある局面では、ディレクトリは、JSPページ、JSTL、JSPタグライブラリ、JSPタグおよびJavaクラスを用いて構築される。一つの局面では、ディレクトリによって用いられるデータを、一人または複数のディレクトリ管理者によって管理される組織体オブジェクトに保存する。アプリケーションのソースユーザは、各アプリケーションのディレクトリエントリオブジェクト中に保持される記述的情報を入力し維持するためにディレクトリを用いることができる。ディレクトリエントリは、関連ステータスフィールドを有し、「公表」または「公開」のステータスを有するアプリケーションのみがビジターに提供される。「プレビュー」ステータスは、公開される前に、ウェブに現われるであろう結果を、ソースユーザおよびディレクトリ管理者が見ることを可能にする。一つの局面では、値のカテゴリピックリストに基づいて、ソリューションカテゴリによるアプリケーションの画面表示がダイナミックに与えられる。アプリケーションは、複数の選択ピックリストを用いて所属するカテゴリのタグを付される。新しいカテゴリを作成するために、ピックリストおよびカテゴリラベルにいつでも新しい値を追加できる。

#### 【0065】

##### 役割および公表モデル

サイトをブラウズするビジターを除いて、ディレクトリサイトの他のほとんどすべての使用は、ログインするかまたは有効なセッションIDを持つことのいずれかをユーザに要求すべきである。これらのユーザは、種々の役割を果たすであろうし、これらの役割に応じて作業するための種々の権限を有してもよい。例えば、三つの基礎的な役割には、開発者、公表者、およびインポータが含まれよう。

#### 【0066】

開発者は、公表者役および編集権限の作成を通じて、別の信頼する人にアプリケーションを公表する責任を手渡すことができるべきである。公表者(それはデフォルトでは開発者自身である)は、アプリケーションについて記述するオリジナルのディレクトリエントリを作成すること、それを後日改変すること、支援する他の公表者を加えること、あるいはディレクトリからエントリを消去することでさえ権限を与えられる可能性がある。本明

10

20

30

40

50

細書に用いるソースユーザは、開発者と公表者のどちらかまたは両方であり得る。

【0067】

ディレクトリにログインする人のユーザIDを用いて、各アプリケーションについてユーザが有する役割および権利を決定する。ユーザは、ディレクトリが保持されている組織体に実際にアクセスするべきではない。ユーザIDはしかし、ユーザを一意的に同定し、必要な場合に情報を自分自身の組織体から取り出すために、用いられる。

【0068】

開発者は、セットアップウィザードを通じてアプリケーションおよびそのパッケージのオリジナル作成者である。ウィザードの一つ(任意での)の工程は、公表するアプリケーションをディレクトリに提出することである。そこで、開発者は、自身が情報を入力するか、または公表者およびディレクトリエントリの維持者の役割を務める他の人を指名してもよい。所与のディレクトリエントリの公表者は、ディレクトリ中のアプリケーションに関する記述的情報を入力し維持することに責任を負うユーザである。各アプリケーションのディレクトリエントリは、オリジナルの提出者が与えた権限を有する指定された公表者のセットを有する。これらの権限は、フィールドを編集すること、ステータスを変更すること、ディレクトリからアプリケーションを消去すること、あるいは公表者(例えば同じかより劣った権限を有する)として他の者を加えることでさえ権利を与える。インポータは、任意のインポートユーザであって、例えばアプリケーションのインポートプロセスを自分の組織体中で展開するために開始した、組織体またはテナントのシステム管理者である。インポートプロセスの間に、どの組織体が、どのアプリケーションを、誰によってインポートしたかを示すレコードが、データベースシステムに作成される。これを、アプリケーションがどのくらい評判がよいかを示すために、ならびにもしそれが望ましければ、コメントおよび評価を、アプリケーションをインストールした組織体のユーザのみに制限するために用いてもよい。個人は様々な時に多面的役割に参加してよい；すなわち、ユーザは、一つのアプリケーションの開発者であり、また別のアプリケーションのインポートユーザであってよい。

【0069】

ディレクトリ情報のレビュー

公表者は、アプリケーションのディレクトリエントリを作成し、説明、サムネイル、スクリーンショットなどの入力関連情報、および他の情報を入力する。公開される前に、公表者はいつでもこの情報をレビューしてもよい。情報が、閲覧用に公開利用可能な準備ができた場合は、公表者は状態を「提出」に変更する。一つの局面では、これによってディレクトリの管理者によるアプリケーションのレビューが開始され、管理者はさらなる情報その他をリクエストしてもよい。一旦アプリケーションがレビューを完了したら、ディレクトリ管理者は、その状態を「公開」に変更する。これにより、アプリケーションはディレクトリ内での公開閲覧用に利用可能になる。ビジネスプロセスに依存して、公開状態は、さらに公表者によるそれ以上の変更をロックアウトしてもよい。この場合は、万一公表者が公開された後にディレクトリエントリをアップデートするような必要があれば、公開閲覧からエントリを消去するように(例えば、状態をニューに変更し戻すことによって)、ディレクトリ管理者にリクエストを提出することができよう。公表者が変更の権限および再度のレビューをリクエストすることなく、エントリを改変することを許可するのが望ましい場合は、公開エントリが変更されたことを単に示すために、ディレクトリエントリ中の改変フラグを用いることができる。

【0070】

ユーザの評価およびコメントによる評判の管理

一つの局面では、各アプリケーションについての他者の考えをビジターに認識させるために、閲覧ユーザはアプリケーションに評価を与え、およびコメントの形の個人的レビューを提供してもよい。一つの局面では、実際にアプリケーションをインストールした組織体のユーザのみが、コメントおよび/または評価を提供してもよい。この資格付与を、例えばアプリケーションのインストールを始めたシステム管理者によって作成されたインポ

10

20

30

40

50

トレコードを見ることにより、推定することができよう。

#### 【0071】

評価を、コミュニティがどのようにアプリケーションを評価するかの公正な反映に保つために、ユーザが複数の記入をすることによってシステムをもてあそばないように一定の保護を与える必要がある。いずれにしても、ユーザは、アプリケーションの評価を自分が当初に評価したものから変更したいと望む可能性がある。一つの局面では、評価システムは、人が投票するように処理することにより遂行される；各ユーザは、アプリケーションに関連した可能な五つ星評価の内の一に投票し、各アプリケーションは、個々の可能な評価に対する投票の得点を保持する。そうすれば、各アプリケーションの平均評価を決定し、さらに、投票のより有益なヒストグラムを作ることは些細なことになる。ユーザはいつでも投票を変更してもよい、しかし、何回評価をしても、彼らは、一つのアプリケーション当たり1回のみ投票することになるだけである。各ユーザが評価またはコメントしたアプリケーションに対する彼らの現在の評価の記録を、各ユーザごとに維持する。

10

#### 【0072】

##### ディレクトリデータモデル

図3は、一つの態様によるアプリケーションのディレクトリ用のデータモデルを示す。示すように、この態様によるディレクトリデータモデルには以下の5つの主要なオブジェクトが含まれる：ディレクトリエントリ/DirectoryEntry、カテゴリページ/CategoryPage、公表者/Publisher、インポート/Import、およびユーザレビュー/UserReview。より少ないかまたはより多くのオブジェクトを用いてもよいことは当然である。

20

#### 【0073】

ディレクトリエントリオブジェクトは、公表のために提出された個々のアプリケーションそれぞれに関係する情報を保持する。情報は、アプリケーション提出または変更の間に、このテーブルにウェブサイトにより書き込まれ、適切な内部ディレクトリサポート担当者によってレビューされ、変更される可能性がある。図4は、一つの態様によるディレクトリエントリオブジェクトおよび他のオブジェクトの定義を示す。

#### 【0074】

ディレクトリエントリオブジェクト中の各エントリは、ディレクトリレコードのステータスを表示するステータスフィールドを伴う。その値は、外部入力および編集機能とレビューおよびプレビューの内部プロセスとの結合を可能にする。ステータス値には以下ものが含まれよう：

30

- ・新規/New：最初に最小の情報のみで作成された
- ・提出された/Submitted：エントリは、公表前の内部レビューの準備済み
- ・プレビュー/Preview：内部サイトプロバイダ担当者がサイトとの関連での情報をレビューすることを許可する。
- ・公開/Public：公表可能、公開閲覧の準備済み
- ・休止中/Inactive：もはやディレクトリで見れない(削除された)

状態を公開に変更する場合、アプリケーションの開発者および他の公表者に例えばEメールによって通知し、および公表日付フィールドまたは変数を設定する。

#### 【0075】

40

一つの局面によれば、各エントリは、アプリケーションがリストされるソリューションカテゴリを表わす、多重選択ピクリストフィールドに関連付けられる。ディレクトリはダイナミックにカテゴリに適応し、したがっていつでもカテゴリを追加してよい。読みやすさのために、一つの局面では、ピクリスト値が、カテゴリおよび/またはサブカテゴリをそれぞれの名前へコード化する。

#### 【0076】

評価(例えば一つ星～五つ星)のためのユーザ投票全体を、平均を計算することができるフィールドに保存する。さらにシステムは、各アプリケーションの評価を決定して表示することができ、例えば、いくつのアプリケーションが一つ星、二つ星、その他と評価されたかを数える。

50

## 【 0 0 7 7 】

カテゴリページオブジェクトを用いてカテゴリ階層を形成し、ディレクトリによるカテゴリページの動的生成を推進する。このオブジェクトはまた、ページのタイトルとともに、ページに注目される選択として表示するいくつかのアプリケーションのリストを保持する。

## 【 0 0 7 8 】

公表者オブジェクトは、アプリケーションのディレクトリエントリを作成して維持することに責任を負うユーザ(例えばソースユーザ)のユーザIDを、そのディレクトリエントリを編集する権限と共に保持する。権限レベルの例には、以下のものが含まれる：

- ・編集/Edit：ある人にディレクトリエントリ中の変更可能なフィールドを編集する権利を与える。
- ・削除/Delete：ユーザに、ディレクトリエントリを削除する権利を与える。
- ・ユーザ追加/AddUser：ユーザに、同一のまたはより限定的な権利を有する追加のユーザを加える権利を与える。

一つの局面では、最高から最低までの権限の順序は、削除/Delete、ユーザ追加/AddUser、編集/Editである。

## 【 0 0 7 9 】

インポートオブジェクトは、任意の特定の組織体へ、どのアプリケーションのインポートを開始したかについての記録として働く。このオブジェクトは、ディレクトリエントリのID(あるいはパッケージID)、アプリケーションをインポートするシステム管理者、システム管理者の所属組織体、およびインポートの日付を保持する。このオブジェクトは、インポート数に基づいたアプリケーションのランク付けを可能にし、望ましい場合にはユーザを、自分の組織体にインポートされたアプリケーションのみについてコメントしおよび評価するように制限するために用いることができる。

## 【 0 0 8 0 】

ユーザレビューオブジェクトは、特定のアプリケーションエントリの指定ユーザによってなされたコメントおよび評価を保持する。誰がアプリケーションにコメントおよび評価を加えることができるかについて制限を課すこともできる(認証されたユーザのみに許可する、またはさらにアプリケーションをインポートした組織体のユーザのみに制限するなどの)。ある実施には、いかなるそのような制限も必要でない。

## 【 0 0 8 1 】

カテゴリページオブジェクトを用いる、カテゴリの実施ソリューションカテゴリによってアプリケーションのリストを見ることができるとは、ディレクトリに関するユーザの有用な要求である。任意のアプリケーションが任意の数のカテゴリ中に現われ、カテゴリは好ましくは入れ子である。さらに、企業規模または市場区分などによるデータの他の閲覧を可能にするために、種々のカテゴリ化方式によってアプリケーションをリストすることが望ましい。

## 【 0 0 8 2 】

一つの局面によれば、カテゴリを、各アプリケーションエントリにタグのアプリケーション(多重選択ピクリストからの)として割り当てる。アプリケーションのためにディレクトリエントリを調製する作業をしているユーザに、どのカテゴリがアプリケーションに当て嵌まるかを示すチェックボックスのセットの形の現行のカテゴリ化方式が提示される。システムは、アプリケーションのディレクトリエントリに必要なピクリスト値を割り当てることになる。

## 【 0 0 8 3 】

ディレクトリ管理者が手動でエントリを作成し、どのアプリケーションがどのカテゴリにあるかを示すレポートを掲載するのを容易にするために、一つの局面では、各アプリケーションエントリは、利用可能なカテゴリの多重選択ピクリストを含むカテゴリフィールドを含む。読みやすさのために、サブカテゴリを表わすピクリスト値が、サブカテゴリへの経路を示すべきである。即ち「消費者+ゲーム」値は、アプリケーションが消費者

10

20

30

40

50

カテゴリの下のサブカテゴリであるゲームに存在することを示唆する。多値の場合は、アプリケーションが複数のカテゴリに置かれる。

【0084】

カテゴリおよびそれらの階層は、カテゴリページを生成する目的で作成されるので、特定のカテゴリ階層の階層的表示には、カテゴリページオブジェクトのセットを用いる。一つの局面では、カテゴリページオブジェクトは、各カテゴリおよび/または利用可能なサブカテゴリに対して存在する。カテゴリページオブジェクトを、ノード自身のカテゴリおよびその親ノードのカテゴリを指定するフィールドを通じて相互にリンクして、ツリーを形成する。ルートノードは、親を有しないカテゴリページによって示される(例えば図5参照)。ツリーの横断走査をクエリが遂行する。例えば、全ソリューションと呼ばれる特定の階層のトップレベルカテゴリのラベルを見出すために、クエリは、以下のようなものになるであろう：

ラベルをカテゴリページ c から選択(親ノード c = 「全ソリューション」)。

(select Label from CategoryPage c where ParentNode c = 'All-Solutions')

【0085】

一つの局面では、表示ノードはまた、JSPページ(指定されたカテゴリに分類されるアプリケーションのリストを与える)によって用いられる追加情報も所持する。この追加情報には、関連するアプリケーションの総数、ページタイトルラベル、ページ本体のテキスト、およびページ上の主なアプリケーションを維持するためのフィールドが含まれる。

【0086】

要約すると、ディレクトリがサポートするユーザ機能には、以下のものが含まれるであろう。

ビジター(閲覧およびインポートするユーザ)用

- ・ビジネス分野(カテゴリ)、制作者、提出日、評価、または他の基準によってフィルタされソートされたアプリケーションリストを閲覧する
- ・アプリケーションの説明(詳細ページ)を見る
- ・アプリケーションの詳細な仕様/プロファイル(オブジェクト番号、オブジェクト名、ウェブ統合リンク(WIL)、その他)を見る
- ・集計されたアプリケーションのユーザ評価を見る
- ・ビジネス分野によってまたは全体の「強調されている」アプリケーションを見る
- ・トップにランクされたアプリケーションを見る
- ・最も人気のあるアプリケーションを見る
- ・個々のアプリケーションについての評価を提出/変更する(任意で制限)
- ・個々のアプリケーションに関連したコメントを記入する(任意で制限)
- ・アプリケーション特異的なデモンストレーション組織体中でアプリケーションを試みる
- ・アプリケーションを自分の組織体へインポートする

【0087】

開発者(エクスポートユーザ)用

- ・開発者が開発したアプリケーションパッケージを閲覧する
- ・関連する記述的情報と共に公表のためのアプリケーションを提出する
- ・アプリケーションをディレクトリから除去する
- ・アプリケーションの説明を編集し/アップデートする
- ・プロバイダレビューによってアプリケーションが承認された場合に、Eメール通知を受取る
- ・他のユーザに、アプリケーションの管理および公表に関する責任を委任する
- ・ディレクトリエントリの添付ファイルとして、アプリケーションに関連する、画像、pdf、および他のドキュメントをアップロードする

【0088】

その他

代りの態様の一つでは、エクスポートされたパッケージを保存するためにデータベースス

10

20

30

40

50



キーマの中で新しい型の組織体(例えばコンテナ組織体)を用いるのではなく、エクスポートされたパッケージは、データベース(例えば、Oracleデータベース)中のバイナリ大規模オブジェクト、あるいはフラットファイルフォーマットに保存されたテキストまたはバイナリデータとして実行される。しかし、アップグレードスクリプトは、フラットファイルに対してうまく作動しない。したがって、一つの態様では、データベースシステム(例えば salesforce.com サービス)内の別個の「隠された」組織体としてパッケージが実行される。これにより、有利にアップグレードスクリプトを公開して、これらのエクスポートされた組織体をアップグレードすることが、可能になる。

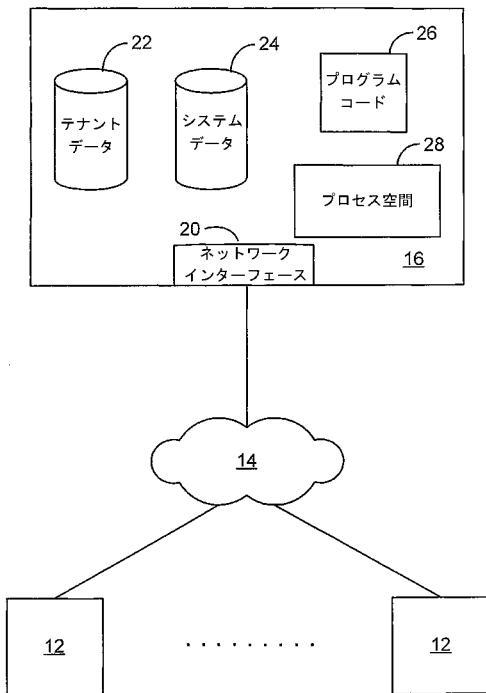
【0089】

一つの局面では、パッケージ中のすべてのオブジェクトの主キーを含むパッケージテーブルを保存する代わりに、パッケージに含まれるすべてのセットアップデータに一意的なパッケージIDに対する外部キーを保存することを行なう。しかし、パッケージテーブルアプローチが好ましい、それは、実行時にどのオブジェクトがパッケージに含まれていたかを、より効率的に速く決定するからである。インストールされたパッケージを基本システムからまたは相互に区別するために、実行時にパッケージにどのオブジェクトが含まれていたかを決定することが望ましいであろう。例えば、カスタムオブジェクト編集画面では、読取りオンリーとしてインポートされたカスタムオブジェクトを改変することはできないが、一方で他のすべてのカスタムオブジェクト(インポートされたのではない)を修飾することができる。

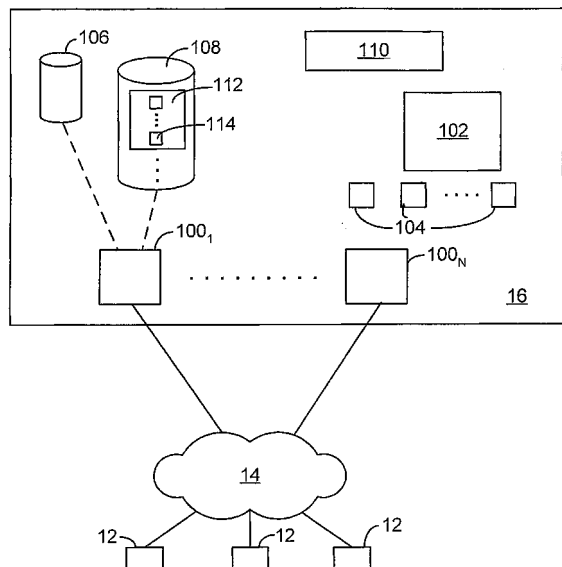
【0090】

本発明を、例として、および具体的な態様により、記述したが、本発明は、開示された態様に限定されることはないことは当然である。反対に、当業者には明白であろう様々な改変および類似の改作を包含することを意図する。したがって、添付した特許請求の範囲には、そのような改変および類似の改作をすべて包含するように、最も広い解釈を与えるべきである。

【図1】



【図2】

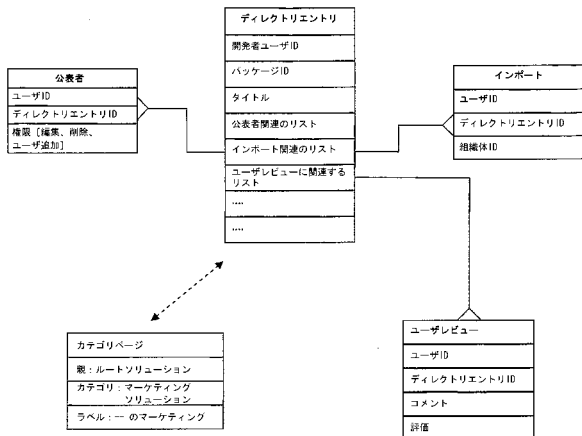


10

20

【図3】

オブジェクト関係



【図4-1】

フィールド名	説明	型	例	ユーザー編集	可視
タイトル	アプリケーション名	テキスト	リクルート部隊	X	X
バージョン	バージョン	テキスト	1.03		
パッケージ名	パッケージ名	テキスト	パッケージ-v1		X
パッケージID	パッケージID	ID	a00300000000raOY		
全数	開発者数	ID	005300000000e67Y		
開発者	開発者ID	ID	005300000000e67Y		
会社名	会社名	テキスト	Acme, Inc.	X	X
開発者名前	開発者名前	テキスト	Bob Jones	X	X
ステータス	エントリステータス	ピククリスト	公開		X
公開日	公開する日付	日付	03/04/2005		X
更新	ユーザーにより変更されたレコード	チェックボックス	T		
要約	短い行説明	テキスト領域	リクルートおよび仕事の管理	X	X
オーバービュー	アプリケーションの長い説明	長いテキスト領域	Lorem ipsum dolor sit Lorem ipsum dolor sit	X	X
特徴リスト	特徴のリスト	長いテキスト領域	Lorem ipsum dolor sit	X	X
カテゴリ	ソリューションカテゴリ	多重選択ピククリスト	販売手数料、マーケティング	X	X
アプリケーションロケール	アプリケーションのロケール	多重選択ピククリスト	en, fr, (ISO-639/3166)	X	X
エントリロケール	説明のロケール	ピククリスト	en, (ISO-639/3166)	X	X
サムネイル	画像URL	URL	http://foo.com/a.jpg	X	X
より多くの情報ページ	より多くの情報ページへのリンク	URL	http://foo.com/readme	X	X
認証済	(認証済、未決)のうち一つ	ピククリスト	認証済		
市場	オプション市場エントリへのリンク	URL	http://www.salesforce.com/partners/solution-detail.jsp?id=xxx	X	X
システム要求	他のシステム要求	テキスト領域	AppFactory, webforce	X	X
編集リスト	アプリケーション編集作業中	多重選択ピククリスト	all, pe		X
スクリーンショット	単一のスクリーンショットへのリンク	URL	http://foo.com/b.jpg	X	X
プレゼンテーション	プレゼンテーションへのリンク	URL	http://foo.com/demo	X	X
開発者ノート	開発者からのオプションのノート	テキスト領域	このアプリケーションはAppFactoryの支援で構築 (http://appfactory.comを参照のこと)	X	X
試用ユーザー名	アプリケーション試用のためのログインユーザー名	テキスト	user-001@mfi.org		
試用ユーザーパスワード	アプリケーション試用のためのログインパスワード	テキスト	Good4U		
一つ星	一つ星投票の数	数	0		X
二つ星	二つ星投票の数	数	22		X
三つ星	三つ星投票の数	数	40		X
四つ星	四つ星投票の数	数	234		X
五つ星	五つ星投票の数	数	2		X

【図4-2】

フィールド名	説明	型	例
ディレクトリエントリ関連リスト			
オブジェクト	説明	型	例
发布者	編集/削除を許可されたユーザー	ルックアップ	
インポート	アプリケーションのインポート	ルックアップ	
ユーザーレビュー	ユーザーによるコメントおよび投票	ルックアップ	
詳細にストグラム	詳細システムへの投票	ルックアップ	
カテゴリページ	これが注目のアプリケーションであるというカテゴリページのリスト	ルックアップ	

フィールド名	説明	型	例
发布者	編集するエントリ	ルックアップID	a00300000000raOY
ディレクトリエントリ	ユーザー	ID	005300000000e67Y
ユーザー	发布者のEメールアドレス	Eメール	joe@hotmail
編集	編集権限	チェックボックス	T
削除	削除権限	チェックボックス	F
ユーザー追加	ユーザー追加権限	チェックボックス	T

フィールド名	説明	型	例
ディレクトリエントリ	編集するエントリ	ルックアップID	a00300000000raOY
組織体	アプリケーションがインポートされた組織体	組織体ID	a003000000002raOY
ユーザー	ユーザーID	ID	005300000000e67Y
日付	インポートの日付	日付	01-02-2005

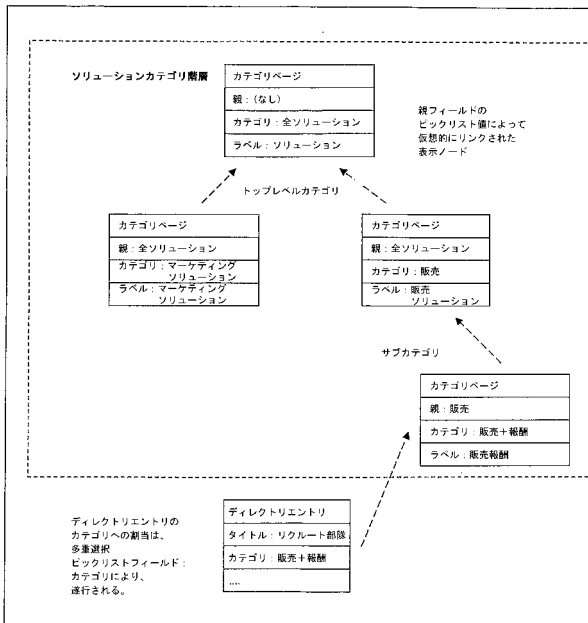
  

フィールド名	説明	型	例
ディレクトリエントリ	編集するエントリ	ルックアップID	a00300000000raOY
ユーザー	ユーザー識別子	ID	005300000000e67Y
ユーザーエイリアス	ユーザーに与えられたエイリアス	テキスト	Joe Cool
ユーザーEメール	オプションのEメールアドレス	Eメール	joe@hotmail
コメント	ユーザーが加えたコメント	テキスト領域	これは素晴らしい!
評価	星評価	ピククリスト	5

フィールド名	説明	型	例
親ノード	親への文字列参照	ピククリスト(ソリューションカテゴリとの適合)	全ソリューション
カテゴリ	ソリューションカテゴリ	ピククリスト(ソリューションカテゴリとの適合)	金融
ラベル	ページタイトル	テキスト	金融サービス
本体	ページに表示するテキスト	テキスト領域	一の様々なソリューションがある
カウント	カテゴリ中のアプリケーションの数	数	23
注目アプリ1	第1の注目すべきアプリケーションエントリ	ルックアップID	a00300000000raOY
注目アプリ2	第2の注目すべきアプリケーションエントリ	ルックアップID	a003000000002raOY
注目アプリ3	第3の注目すべきアプリケーションエントリ	ルックアップID	a0030000000023raOY

【図5】



【 6 】

- Personal Setup
  - My Personal Information
  - Email
  - Mobile Users
  - Logout
  - Office Edition
- Customforce
  - Customize
  - Extend
    - Custom Apps
    - Custom Objects
    - Custom Tabs
    - Custom Settings
  - Integrate
- Administration Setup
  - Manage Users
  - Company Profile
  - Security Controls
  - Communication Templates
  - Translations Workbench
  - Data Management
- Order Center

**Multiforce Projects** [Help](#)

This page allows you to manage Multiforce projects.

- You can create local projects which can then be published to the Multiforce Directory
- You can install projects from the Multiforce Directory or directly from third parties

**Current Projects** [New](#)

Action	Project Name	Description	Status	Source
<a href="#">Edit</a> <a href="#">Del</a>	Bugforce	Bug tracking application	Active	Local
<a href="#">Edit</a> <a href="#">Del</a>	Custom Objects reports	Custom reports we are developing for clients	Inactive	Local
<a href="#">Edit</a> <a href="#">Del</a>	Competition tab	Custom tab for tracking competitors	Published	Local

**Installed Projects** [Install from Directory](#)

This page only visible if you have "Create Projects" Add to project button is controlled by new "Create Projects" perm

Action	Project Name	Description	Status	Source
<a href="#">Edit</a> <a href="#">Uninstall</a>	Public Release application	Public Release application	In	Local
<a href="#">Edit</a> <a href="#">Uninstall</a>	Customer Service Dashboard	Dashboards for cases & solutions	Installed	Third Party

**Installation History**

Date	User	Project Name	Action	Source
4/11/2005 10:46 AM	admin@bluewin.com	Project	Installed	Multiforce Directory
4/10/2005 10:46 AM	eric@bluewin.com	Project	Uninstalled	Local
4/5/2005 10:45 AM	eric@bluewin.com	Project	Installed	Local
3/21/2005 10:41 AM	lee@bluewin.com	Customer Service Dashboards	Installed	Third Party

[Download full installation history log >](#)

【 7 】

- Personal Setup
  - My Personal Information
  - Email
  - Mobile Users
  - Logout
  - Office Edition
- Customforce
  - Customize
  - Extend
    - Custom Apps
    - Custom Objects
    - Custom Tabs
    - Custom Settings
  - Integrate
- Administration Setup
  - Manage Users
  - Company Profile
  - Security Controls
  - Communication Templates
  - Translations Workbench
  - Data Management
- Order Center

**Project: Bugforce** [Help](#)

[Edit](#) [Delete](#) [Publish](#)

Project Name: Bugforce Security Rating: Low Risk

Status: Active [\(Change\)](#) Deployment Status: In Development [\(Change\)](#)

Source: Local

Created by: Admin User, 4/11/2005 10:46 AM Modified by: Admin User, 4/11/2005 10:46 AM

Description: Bug tracking application

**Project Items** [What's New?](#)

Action	Name	Type
<a href="#">Del</a>	Bugforce	Custom app
<a href="#">Del</a>	Bugs	Custom tab
<a href="#">Del</a>	Release	Custom tab
<a href="#">Del</a>	Del	Custom object
<a href="#">Del</a>	Release	Custom object
<a href="#">Del</a>	Close Bugs	Report
<a href="#">Del</a>	Release Type	Dashboard
<a href="#">Del</a>	Bug Notification HTML	Email Template

【 8 】

```

<!-- PROJECT -->
<entity name="Project" keyPrefix="030"
  access="orgHasTabSet" editAccess="UserPermissions.CustomizeApplication"
  apiAccess="orgHasPrivateApi"
  javaPackageRoot="core.project" usesNewUIStyle="true" motifName="Setup"
  isDeletableViaEntityObjects="true"
  dbSchemaName="core"
  comment="All Project definition are managed by this entity."/ >

  <field name="Id" fieldType="PRIMARYKEY"/>
  <field name="Name" fieldType="NAME" columnType="TEXT" maxLength="80"/>
  <field name="Description" columnType="MULTILINETEXT" maxLength="1000"/>

  <field name="Overview" fieldType="FOREIGNKEY" domain="Document"/>
  <field name="Status" columnType="STATILENUM" name="ProjectStatus" dbValueRequired="yes"/ >
  <field name="Source" fieldType="FOREIGNKEY" domain="Organization"/ >

  <field name="CreateDate" fieldType="CREATEDATE"/>
  <field name="CreatedBy" fieldType="CREATEDBY"/>
  <field name="LastModifiedDate" fieldType="LASTUPDATE"/>
  <field name="LastModifiedBy" fieldType="LASTUPDATEBY"/>
  <field name="SystemModstamp" fieldType="SYSTEMSTAMP"/>
</entity>

<!-- PROJECT MEMBER -->
<entity name="ProjectMember" keyPrefix="034"
  access="orgHasTabSet" editAccess="UserPermissions.CustomizeApplication"
  apiAccess="orgHasPrivateApi"
  javaPackageRoot="core.project" usesNewUIStyle="true" motifName="Setup"
  isDeletableViaEntityObjects="true"
  plsqlPackage="cProject" dbSchemaName="core"
  plsqlGetDetail="get_project_member" plsqlGetEdit="get_project_member_edit"
  plsqlInsertDetail="insert_project_member" plsqlUpdateDetail="update_project_member"
  plsqlApiGetList="api_project_member_get_list"
  plsqlApiGetBulkList="api_project_mbr_get_bulklist"
  plsqlDelete="delete_project_member"
  comment="All Project Members for a specific Project are managed by this entity." >

  <field name="Id" fieldType="PRIMARYKEY"/ >
  <field name="Project" fieldType="FOREIGNKEY" domain="Project" dbValueRequired="yes"
    updateNormallyToDb="no" alternateKeyIndex="0" parentRelationshipName="Project"
    childRelationshipName="ProjectItems"/ >

  <field name="Subject" fieldType="FOREIGNKEY" dbValueRequired="yes"
    domain="Project,TabSet,CustomEntityDefinition,CustomFieldDefinition,Scoutrol,EmailTemplate,Layout,
    ListLayout,ProfileLayout,Document,Folder,Report,Dashboard,Filter,User,Profile,CustomEntity
    DataTemplate"/ >

  <field name="OriginalSubject" fieldType="FOREIGNKEY" dbValueRequired="yes"
    domain="Project,TabSet,CustomEntityDefinition,CustomFieldDefinition,Scoutrol,EmailTemplate,Layout,
    ListLayout,ProfileLayout,Document,Folder,Report,Dashboard,Filter,User,Profile,CustomEntity
    DataTemplate"/ >

  <field name="CreateDate" fieldType="CREATEDATE"/>
  <field name="CreatedBy" fieldType="CREATEDBY"/>
  <field name="LastModifiedDate" fieldType="LASTUPDATE"/>
  <field name="LastModifiedBy" fieldType="LASTUPDATEBY"/>
  <field name="SystemModstamp" fieldType="SYSTEMSTAMP"/>
</entity>

```

---

 フロントページの続き

- (74)代理人 100093908  
弁理士 松本 研一
- (74)代理人 100116894  
弁理士 木村 秀二
- (74)代理人 100130409  
弁理士 下山 治
- (74)代理人 100134175  
弁理士 永川 行光
- (74)代理人 100172591  
弁理士 江嶋 清仁
- (74)代理人 100131886  
弁理士 坂本 隆志
- (72)発明者 ブルックス デヴィッド  
アメリカ合州国 カリフォルニア州 サン ノゼ ペリエ レーン 3859
- (72)発明者 ジャシク ベンジ  
アメリカ合州国 カリフォルニア州 サン フランシスコ 17ス アヴェニュー 1215 #  
2
- (72)発明者 ベザラ エリック デヴィッド  
アメリカ合州国 カリフォルニア州 オークランド 62nd ストリート 375
- (72)発明者 チャスマン ダグラス  
アメリカ合州国 ニューヨーク州 ピッツフォード アンバー ヒル ドライブ 33
- (72)発明者 ハンスマ スコット  
アメリカ合州国 カリフォルニア州 サン フランシスコ 25ス ストリート 4626
- (72)発明者 タッカー ルイス ウィリー  
アメリカ合州国 カリフォルニア州 サン フランシスコ 19ス ストリート 3964
- (72)発明者 メイソン ティモシー  
アメリカ合州国 カリフォルニア州 サン フランシスコ 6ス アベニュー 1465
- (72)発明者 ウォン サイモン  
アメリカ合州国 カリフォルニア州 サン カルロス ダートマウス アベニュー 531
- (72)発明者 ツォ ティエン  
アメリカ合州国 カリフォルニア州 サン フランシスコ ブキャナン 1962
- (72)発明者 グロス アダム  
アメリカ合州国 カリフォルニア州 サン フランシスコ 7ス アベニュー 1367 #2
- (72)発明者 タム スティーブン  
アメリカ合州国 カリフォルニア州 サン フランシスコ ページ ストリート 161

審査官 池田 聡史

- (56)参考文献 国際公開第01/065435(WO, A1)  
特表2005-503596(JP, A)

- (58)調査した分野(Int.Cl., DB名)  
G06F 12/00