

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2005-505048

(P2005-505048A)

(43) 公表日 平成17年2月17日(2005.2.17)

(51) Int. Cl.⁷
G06F 12/00

F I
G06F 12/00 510A

テーマコード(参考)
5B082

審査請求 未請求 予備審査請求 有 (全 56 頁)

(21) 出願番号 特願2003-533099 (P2003-533099)
 (86) (22) 出願日 平成14年9月27日 (2002. 9. 27)
 (85) 翻訳文提出日 平成16年3月29日 (2004. 3. 29)
 (86) 国際出願番号 PCT/US2002/030884
 (87) 国際公開番号 W02003/029955
 (87) 国際公開日 平成15年4月10日 (2003. 4. 10)
 (31) 優先権主張番号 09/968, 735
 (32) 優先日 平成13年9月28日 (2001. 9. 28)
 (33) 優先権主張国 米国 (US)
 (31) 優先権主張番号 10/120, 030
 (32) 優先日 平成14年4月9日 (2002. 4. 9)
 (33) 優先権主張国 米国 (US)

(71) 出願人 503353988
 ジーベル システムズ インコーポレイテッド
 アメリカ合衆国 カリフォルニア州 94
 404 サン マテオ ブリッジポイント
 パークウェイ 2207
 (74) 代理人 100082005
 弁理士 熊倉 禎男
 (74) 代理人 100067013
 弁理士 大塚 文昭
 (74) 代理人 100074228
 弁理士 今城 俊夫
 (74) 代理人 100086771
 弁理士 西島 孝喜

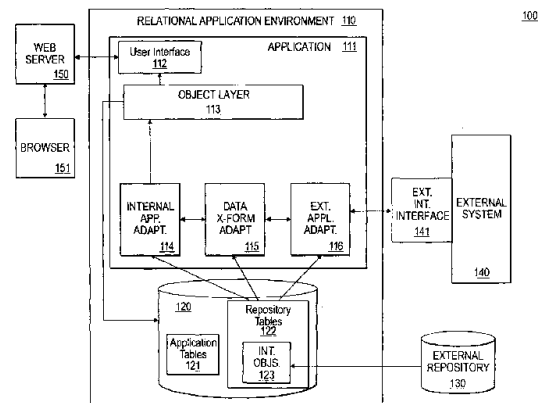
最終頁に続く

(54) 【発明の名称】 統合オブジェクトを企業のビジネスアプリケーションと共に用いるための方法及びシステム

(57) 【要約】

企業ビジネスアプリケーションと共に統合オブジェクトを用いるための方法及びシステムが開示されている(図1)。本方法は、第1企業アプリケーション用の第1フォーマットを有する外部オブジェクト(140)を、第2企業アプリケーション(111)にインポートするステップと、統合オブジェクト(123)を使って第1フォーマットの外部オブジェクト(140)を、第2企業アプリケーション(111)用にフォーマットされた第2フォーマットの外部オブジェクトに変換するステップと、第2ビジネスアプリケーション内の第2フォーマットの外部オブジェクトを用いるステップとを含む。

【選択図】 図1



【特許請求の範囲】

【請求項 1】

第 1 企業アプリケーション用の第 1 フォーマットを有する外部オブジェクトを、第 2 企業アプリケーションにインポートするステップと、
統合オブジェクトを使用して、前記第 1 フォーマットの外部オブジェクトを、前記第 2 企業アプリケーション用にフォーマットされた第 2 フォーマットの外部オブジェクトに変換するステップと、
前記第 2 ビジネスアプリケーション内の前記第 2 フォーマットの外部オブジェクトを使用するステップとを含むことを特徴とする方法。

【請求項 2】

前記統合オブジェクトを使用するステップは、
前記第 2 企業ビジネスアプリケーションと通信する外部アプリケーションアダプタ内の前記第 1 フォーマットの外部オブジェクトを受信するステップと、
前記外部アプリケーションアダプタによって、前記第 1 フォーマットの外部オブジェクトのローカル表現を生成するステップとを含むことを特徴とする請求項 1 に記載の方法。

【請求項 3】

前記統合オブジェクトを使用するステップは、
前記外部アプリケーションアダプタから、データ変換アダプタ内の前記ローカル表現を受信するステップと、
前記ローカル表現内に含まれる 1 又はそれ以上のフィールドを、前記第 2 企業アプリケーションに対応させるために名称変更するステップと、
前記ローカル表現から共通表現を作成するステップとを更に含むことを特徴とする請求項 2 に記載の方法。

【請求項 4】

前記統合オブジェクトを使用するステップは、
内部オブジェクト層と通信する内部アプリケーションアダプタ内の前記共通表現を前記データ変換アダプタから受信するステップと、
前記内部アプリケーションアダプタによって、前記共通表現から前記第 2 フォーマットの外部オブジェクトを生成するステップとを含むことを特徴とする請求項 3 に記載の方法。

【請求項 5】

第 1 企業アプリケーション用の第 1 フォーマットを有する内部オブジェクトを、第 2 企業アプリケーションにエクスポートするステップと、
統合オブジェクトを使用して、前記第 1 フォーマットの外部オブジェクトを、前記第 2 企業アプリケーション用にフォーマットされた第 2 フォーマットの外部オブジェクトに変換するステップと、
前記第 2 ビジネスアプリケーション内の前記第 2 フォーマットの外部オブジェクトを使用するステップとを含むことを特徴とする方法。

【請求項 6】

前記統合オブジェクトを使用するステップは、
内部オブジェクト層と通信する内部アプリケーションアダプタ内の前記第 1 フォーマットの内部オブジェクトを受信するステップと、
前記内部アプリケーションアダプタによって、前記第 1 フォーマットの内部オブジェクトの共通表現を生成するステップとを含むことを特徴とする請求項 5 に記載の方法。

【請求項 7】

前記統合オブジェクトを使用するステップは、
データ変換アダプタ内の前記共通表現を前記内部アプリケーションアダプタから受信するステップと、
前記共通表現内に含まれる 1 又はそれ以上のフィールドを、前記第 2 企業アプリケーションに対応させるために名称変更するステップと、
前記共通表現からローカル表現を生成するステップとを更に含むことを特徴とする請求項

10

20

30

40

50

6 に記載の方法。

【請求項 8】

前記統合オブジェクトを使用するステップは、

前記第 2 企業アプリケーションと関係付けられた外部アプリケーションアダプタと通信する外部アプリケーションアダプタ内の前記ローカル表現を、前記データ変換アダプタから受信するステップと、

前記外部アプリケーションアダプタによって、前記共通表現から前記第 2 フォーマットの外部オブジェクトを生成するステップとを更に含むことを特徴とする請求項 7 に記載の方法。

【請求項 9】

前記統合オブジェクトを前記第 2 企業アプリケーションへ提供するステップを更に備えることを特徴とする請求項 8 に記載の方法。

【請求項 10】

外部企業アプリケーションを記述した内部企業アプリケーション内外部メタデータを受信するステップと、

統合オブジェクトウィザードを使って、前記外部メタデータを統合オブジェクトに変換するステップとを含むことを特徴とする方法。

【請求項 11】

第 1 企業アプリケーション用の第 1 フォーマットを有する外部オブジェクトを、第 2 企業アプリケーションにインポートする手段と、

統合オブジェクトを使用して、前記第 1 フォーマットの外部オブジェクトを、前記第 2 企業アプリケーション用にフォーマットされた第 2 フォーマットの外部オブジェクトに変換する手段と、

前記第 2 ビジネスアプリケーション内の前記第 2 フォーマットの外部オブジェクトを使用する手段とを含むことを特徴とするシステム。

【請求項 12】

前記統合オブジェクトを使用する手段は、

前記第 2 企業ビジネスアプリケーションと通信する外部アプリケーションアダプタ内の前記第 1 フォーマットの外部オブジェクトを受信する手段と、

前記外部アプリケーションアダプタによって、前記第 1 フォーマットの外部オブジェクトのローカル表現を生成する手段とを備えることを特徴とする請求項 11 に記載のシステム。

【請求項 13】

前記統合オブジェクトを使用する手段は、

前記外部アプリケーションアダプタから、データ変換アダプタ内の前記ローカル表現を受信する手段と、

前記ローカル表現内に含まれる 1 又はそれ以上のフィールドを、前記第 2 企業アプリケーションに対応させるために名称変更する手段と、

前記ローカル表現から共通表現を生成する手段とを更に備えることを特徴とする請求項 12 に記載のシステム。

【請求項 14】

前記統合オブジェクトを使用する手段は、

内部オブジェクト層と通信する内部アプリケーションアダプタ内の前記共通表現を、前記データ変換アダプタから受信する手段と、

前記内部アプリケーションアダプタによって、前記共通表現から前記第 2 フォーマットの外部オブジェクトを生成する手段とを備えることを特徴とする請求項 13 に記載のシステム。

【請求項 15】

第 1 企業アプリケーション用の第 1 フォーマットを有する内部オブジェクトを、第 2 企業アプリケーションにエクスポートする手段と、

10

20

30

40

50

統合オブジェクトを使って、前記第1フォーマットの外部オブジェクトを、前記第2企業アプリケーション用にフォーマットされた第2フォーマットの外部オブジェクトに変換する手段と、

前記第2ビジネスアプリケーション内の前記第2フォーマットの外部オブジェクトを使用する手段とを備えることを特徴とするシステム。

【請求項16】

前記統合オブジェクトを使用する手段は、

内部オブジェクト層と通信する内部アプリケーションアダプタ内の前記第1フォーマットの内部オブジェクトを受信する手段と、

前記内部アプリケーションアダプタによって、前記第1フォーマットの内部オブジェクトの共通表現を生成する手段とを備えることを特徴とする請求項15に記載のシステム。 10

【請求項17】

前記統合オブジェクトを使用する手段は、

データ変換アダプタ内の前記共通表現を前記内部アプリケーションアダプタから受信する手段と、

前記共通表現内に含まれる1又はそれ以上のフィールドを、前記第2企業アプリケーションに対応させるために名称変更する手段と、

前記共通表現から、ローカル表現を生成する手段とを更に備えることを特徴とする請求項16に記載のシステム。

【請求項18】

20

前記統合オブジェクトを使用する手段は、

前記第2企業アプリケーションと関係付けられた外部アプリケーションアダプタと通信する外部アプリケーションアダプタ内の前記ローカル表現を、前記データ変換アダプタから受信する手段と、

前記外部アプリケーションアダプタによって、前記共通表現から前記第2フォーマットの外部オブジェクトを生成する手段とを更に備えることを特徴とする請求項17に記載のシステム。

【請求項19】

前記統合オブジェクトを前記第2企業アプリケーションへ提供する手段を更に備えることを特徴とする請求項18に記載のシステム。 30

【請求項20】

外部企業アプリケーションを記述した内部企業アプリケーション内外部メタデータを受信する手段と、

統合オブジェクトウィザードを使って、前記外部メタデータを総合オブジェクトに変換する手段とを備えることを特徴とするシステム。

【請求項21】

複数の命令を記憶したコンピュータ読み取り可能な記憶媒体であって、コンピュータが前記複数の命令を実行すると、前記コンピュータは、

第1企業アプリケーション用の第1フォーマットを有する外部オブジェクトを、第2企業アプリケーションにインポートするステップと、 40

統合オブジェクトを使って、前記第1フォーマットの外部オブジェクトを、前記第2企業アプリケーション用にフォーマットされた第2フォーマットの外部オブジェクトに変換するステップと、

前記第2ビジネスアプリケーション内の前記第2フォーマットの外部オブジェクトを使用するステップとを実行することを特徴とするコンピュータ読み取り可能な記憶媒体。

【請求項22】

追加命令を記憶したコンピュータ読み取り可能な記憶媒体であって、コンピュータが前記追加命令を実行すると、前記コンピュータは、

前記第2企業ビジネスアプリケーションと通信する外部アプリケーションアダプタ内の前記第1フォーマットの外部オブジェクトを受信するステップと、 50

前記外部アプリケーションアダプタによって、前記第1フォーマットの外部オブジェクトのローカル表現を生成するステップとを更に実行することを特徴とする請求項21に記載のコンピュータ読み取り可能な記憶媒体。

【請求項23】

追加命令を記憶したコンピュータ読み取り可能な記憶媒体であって、コンピュータが前記追加命令を実行すると、前記コンピュータは、前記外部アプリケーションアダプタから、データ変換アダプタ内の前記ローカル表現を受信するステップと、前記ローカル表現内に含まれる1又はそれ以上のフィールドを、前記第2企業アプリケーションに対応させるため名称変更するステップと、前記ローカル表現から共通表現を生成するステップとを更に実行することを特徴とする請求項22に記載のコンピュータ読み取り可能な記憶媒体。

10

【請求項24】

追加命令を記憶したコンピュータ読み取り可能な記憶媒体であって、コンピュータが前記追加命令を実行すると、前記コンピュータは、内部オブジェクト層と通信する内部アプリケーションアダプタ内の前記共通表現を、前記データ変換アダプタから受信するステップと、前記内部アプリケーションアダプタによって、前記共通表現から前記第2フォーマットの外部オブジェクトを生成するステップとを更に実行することを特徴とする請求項23に記載のコンピュータ読み取り可能な記憶媒体。

20

【請求項25】

複数の命令を記憶したコンピュータ読み取り可能な記憶媒体であって、コンピュータが前記複数の命令を実行すると、前記コンピュータは、第1企業アプリケーション用の第1フォーマットを有する内部オブジェクトを、第2企業アプリケーションにエクスポートするステップと、統合オブジェクトを使って、前記第1フォーマットの外部オブジェクトを、前記第2企業アプリケーション用にフォーマットされた第2フォーマットの外部オブジェクトに変換するステップと、前記第2ビジネスアプリケーション内の前記第2フォーマットの外部オブジェクトを使用するステップとを実行することを特徴とするコンピュータ読み取り可能な記憶媒体。

30

【請求項26】

追加命令を記憶したコンピュータ読み取り可能な記憶媒体であって、コンピュータが前記追加の命令を実行すると、前記コンピュータは、内部オブジェクト層と通信する内部アプリケーションアダプタ内の前記第1フォーマットの内部オブジェクトを受信するステップと、前記内部アプリケーションアダプタによって、前記第1フォーマットの内部オブジェクトの共通表現を生成するステップとを更に実行することを特徴とする請求項25に記載のコンピュータ読み取り可能な記憶媒体。

【請求項27】

追加命令を記憶したコンピュータ読み取り可能な記憶媒体であって、コンピュータが前記追加の命令を実行すると、前記コンピュータは、データ変換アダプタ内の前記共通表現を前記内部アプリケーションアダプタから受信するステップと、前記共通表現内に含まれる1又はそれ以上のフィールドを、前記第2企業アプリケーションに対応させるため名称変更するステップと、前記共通表現からローカル表現を作成するステップとを更に実行することを特徴とする請求項26に記載のコンピュータ読み取り可能な記憶媒体。

40

【請求項28】

追加命令を記憶したコンピュータ読み取り可能な記憶媒体であって、コンピュータが前記追加命令を実行すると、前記コンピュータは、

50

前記第2企業アプリケーションと関係付けられた外部アプリケーションアダプタと通信する外部アプリケーションアダプタ内の前記ローカル表現を、前記データ変換アダプタから受信するステップと、
前記外部アプリケーションアダプタによって、前記共通表現から前記第2フォーマットの外部オブジェクトを生成するステップとを更に実行することを特徴とする請求項27に記載のコンピュータ読み取り可能な記憶媒体。

【請求項29】

追加命令を記憶したコンピュータ読み取り可能な記憶媒体であって、コンピュータが前記追加命令を実行すると、前記コンピュータは、
前記統合オブジェクトを前記第2企業アプリケーションへ提供するステップを更に実行することを特徴とする請求項28に記載のコンピュータ読み取り可能な記憶媒体。

10

【請求項30】

複数の命令を記憶したコンピュータ読み取り可能な記憶媒体であって、コンピュータが前記複数の命令を実行すると、前記コンピュータは、
外部企業アプリケーションを記述した内部企業アプリケーション内の外部メタデータを受信するステップと、
統合オブジェクトウィザードを使って、前記外部メタデータを総合オブジェクトに変換するステップとを実行することを特徴とするコンピュータ読み取り可能な記憶媒体。

【請求項31】

第1企業アプリケーション用の第1フォーマットを有する外部オブジェクトを、第2企業アプリケーションへインポートする第2企業アプリケーションと、
前記第2企業アプリケーションに関係付けられた1つ又はそれ以上の統合オブジェクトとを備え、前記第2企業アプリケーションは、前記第1フォーマットの外部オブジェクトを、前記第2企業アプリケーション用にフォーマットされた第2フォーマットの外部オブジェクトに変換し、前記第2フォーマットの外部オブジェクトは、第2ビジネスアプリケーションに用いられることを特徴とするシステム。

20

【請求項32】

前記第2企業アプリケーションは、
前記第1フォーマットの外部オブジェクトを受信し、前記第2企業ビジネスアプリケーションと通信し、かつ前記第1フォーマットの外部オブジェクトのローカル表現を生成する外部アプリケーションアダプタを備えることを特徴とする請求項31に記載のシステム。

30

【請求項33】

前記第2企業アプリケーションは、
前記外部アプリケーションアダプタから前記ローカル表現を受信し、前記ローカル表現内に含まれる1又はそれ以上のフィールドを、前記第2企業アプリケーションに対応させるため名称変更し、かつ前記ローカル表現から共通表現を生成するデータ変換アダプタを更に備えることを特徴とする請求項32に記載のシステム。

【請求項34】

前記第2企業アプリケーションは、
前記データ変換アダプタから前記共通表現を受信する内部アプリケーションアダプタを更に備え、前記内部アプリケーションアダプタは、内部オブジェクト層と通信し、かつ前記内部アプリケーションアダプタによって前記共通表現から前記第2フォーマットの外部オブジェクトを生成することを特徴とする請求項33に記載のシステム。

40

【請求項35】

第1企業アプリケーション用の第1フォーマットを有する内部オブジェクトをエクスポートする第2企業アプリケーションと、
前記第2企業アプリケーションと関係付けられた1又はそれ以上の統合オブジェクトとを備え、前記前記第2企業アプリケーションは、前記内部オブジェクトを前記第1企業アプリケーション用にフォーマットされた1又はそれ以上の第1フォーマットの外部オブジェクトに変換することを特徴とするシステム。

50

【請求項 36】

前記第 2 企業アプリケーションは、

前記第 1 フォーマットの内部オブジェクトを受信する内部アプリケーションアダプタを備え、内部アプリケーションアダプタは、内部オブジェクト層と通信し、かつ前記内部アプリケーションアダプタによって前記第 1 フォーマットの内部オブジェクトの共通表現を生成することを特徴とする請求項 35 に記載のシステム。

【請求項 37】

前記第 2 企業アプリケーションは、

前記内部アプリケーションアダプタから前記共通表現を受信し、前記共通表現内に含まれる 1 又はそれ以上のフィールドを、前記第 2 企業アプリケーションに対応させるため名称変更し、かつ前記共通表現からローカル表現を生成するデータ変換アダプタを更に備えることを特徴とする請求項 36 に記載のシステム。

10

【請求項 38】

前記第 2 企業アプリケーションは、

前記データ変換アダプタから前記ローカル表現を受信する外部アプリケーションアダプタを備え、外部アプリケーションアダプタは、前記第 2 企業アプリケーションと関係付けられている外部統合インタフェースと通信し、かつ前記外部アプリケーションアダプタによって前記共通表現から前記第 2 フォーマットの外部オブジェクトを生成することを特徴とする請求項 37 に記載のシステム。

20

【請求項 39】

前記統合オブジェクトは、前記第 1 企業アプリケーションに提供されることを特徴とする請求項 38 に記載のシステム。

【請求項 40】

外部企業アプリケーションを記述する外部メタデータを受信する内部企業アプリケーションと、

前記外部メタデータを統合オブジェクトに変換する統合オブジェクトウィザードとを備えることを特徴とするシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、コンピュータシステムに関し、さらに詳しくは、統合オブジェクトを企業のビジネスアプリケーションと共に用いるための方法及びシステムに関するものである。

30

【背景技術】

【0002】

或るオブザーバーによれば、現代の会社の活力が情報であるとするれば、その動脈は、会社企業を取り巻くデータを動き易くする「アプリケーション間インタフェース」である。これは、最近「アプリケーションネットワーク」として知られるようになっている。

【0003】

一般的な組織のために、アプリケーションネットワークは、特別な (ad hoc) アプリケーション統合プログラムの集まりへと組織的に成長している。この異色の集合体は、新たなアプリケーションを実行する時間を増大させるので、ビジネスに非常に深刻な衝撃を与え、上級幹部が明確なビジネス像を描けないようにしてしまう。つまり、会社の動脈を詰まらせてしまうのである。競合会社がある中で生き残るためにアプリケーション統合が重要になってきたのは事実であるが、それにもかかわらず先行技術では、長期にわたって経費を会社に投資し、特別仕様のコード (custom code) を手作り (handcraft) 又は「改変 (hack)」することが許容されていた。同様に、長期のアプリケーション統合に関する決定が、個々のプロジェクトの規準だけに基づいた最低許容レベルで行われてきた。企業アプリケーション統合 (EAI) の効果的な解決法は、これらの問題の本質が決定的に困難であるが故にまだ見つかっていない。

40

【0004】

50

インターネットの出現、クライアント/サーバー・コンピュータ化、会社の合併及び買収、グローバル化及びビジネスプロセスのリエンジニアリングは、時には手動で、あるシステムが相当の旧式であったとしても異なるシステムが互いに通信させる新たな方法を常に求めるために、会社の情報技術（IT）部門で力をあわせてきた。その結果生じた混乱の中、不適切な通信システムはITの実行力が衰えビジネスが要求する速さで変化していった。

【0005】

ITの最近の傾向として、サポートするのに必要なアプリケーション間インタフェースの量が増大 - しばしば膨大な量である - することでこの問題が悪化する一方である。ごく最近では、企業のアプリケーションは、データウェアハウス及びエンタープライズリソースプランニング（ERP）のような機能を実行しており、電子商取引を容易にしている。したがってこれら3つの技術を簡潔に見直すことは冗長と感じてしまうかもしれないが、未だ解決されていない企業アプリケーション統合（EAI）の必要性を理解することに役立つであろう。

10

【0006】

データウェアハウス技術は、多くのオペレーションシステムからウェアハウス（倉庫）へ、定期的に移動されなければならない大量の真新しい履歴データを必要とする。ソースデータは、オンライントランザクション処理（OLTP）のために構築されるのが普通であり、一方で典型的なデータウェアハウスは、多次元分析処理（OLAP）フォーマットも備えている。したがって、ソースデータは、ウェアハウス（倉庫）へ移される際に、広範囲に集められ且つ再フォーマットにかけられなければならない。

20

【0007】

先行技術に従う典型的なデータウェアハウスは、（a）ソースデータを抽出するステップ、（b）その抽出されたデータをクリーニングするステップ、（c）抽出されクリーニングされたデータを多次元で集めるステップ、（d）ウェアハウスにロード（load）するステップ、の4つのステップで構成される。各ウェアハウスのソースは、特定データの抽出、クリーニング、集約、及びロードのルーチンを組立てる必要がある。フォレスト研究所によれば、平均的な大会社は約4つのデータウェアハウスを有していると推測している。2年以内に、この数が6に増えると予測されている。また、各ウェアハウスに含まれるデータの平均の量は、同じ2年以内に約130ギガバイトから約260ギガバイトへと量的に倍増すると予測される。

30

【0008】

このような大量のデータがこれまでの増加ペースで増大することに伴う問題は、ソースデータの質によって一層悪化する。METAグループが行った研究によると、現在、典型的なデータウェアハウスにロードされるデータの20%程度は劣悪な質のデータである。その回答者の約70%は手作業によりコード化された抽出処理、先浄処理、及びロード処理を用いていたことがその同じ研究で示されている。要求された集約プロセスに関して、事例証拠は、この機能だけを完了するのに50時間程度のコンピュータ時間が必要になる可能性があることを示している。この様なやり方でコーディングされたプログラムを維持するには相当な努力が必要になることが自明である。

40

【0009】

一方、典型的なERPシステムは、基本的に大型の統合実装アプリケーションであり、給与計算（payroll）、製造、汎用の原簿、及び人的資源の核となるビジネス機能をサポートする。しかしながら、ERPシステムを実行することは、多くの理由で途方にくれる処理になり得る。

【0010】

先ず最初に、会社は製品を購入するのであってソリューションを構築することはしない。これは、会社内の事業ユニットが、既存の他の方法ではなくその製品及びそれがどのように作用するかに適応しなければならないことを意味する。更に、今日のERPシステムは、会社における従来のソリューション全てを取り替えることはできない。このため、EP

50

Rシステムは、しかるべき所にある他の旧来のシステムと効率的に通信しなければならない。結局、1つのベンダーが組織のあらゆる需要を満足することができないのが普通なので、会社が1以上の完全に異なるERPシステムを採用することは珍しくはない。

【0011】

その結果、データをERPシステムから出し入れするためのオプションが、データウェアハウスに用いられる既知のアプローチを妨げる。各ERPシステムは、そのベンダーによって継続的に拡張(エンハンス)される所有権のあるデータモデルを有している。そのようなモデルを操作する抽出ルーチン又はロードルーチンを記述することは複雑なだけでなく、データの妥当性及び企業アプリケーションに固有のビジネスルールが無視されがちなので、ベンダーによって遠慮させられる。代わりに、ERPはビジネスオブジェクトレベルでの相互作用を要求し、それは汎用の原簿、予算又は支払い勘定のような特定のビジネスエンティティを扱う。

10

【0012】

各形態の電子商取引は、長年広く活用されてきた。基本的に、電子商取引は電子データ交換(EDI)と共に開始した。EDIは、会社が購入注文及び請求書(invoice)を電子的に通信できるようにし、現在の会社がEDIを使用してサプライチェーン管理を行うまでに発展を続けてきた。しかしながら、興味のある物を売買したりオークションしたりするのにごく最近のオンラインインターネットウェブサイトを爆発的に利用して初めて、頑強で効果的なEAIのための緊急を要する必要性が存在していた。

【0013】

限りある時間フレーム内で特定のビジネス目的を達成するようにアプリケーションが開発されてきている。典型的な大組織の場合、異なるチームの人々が、様々な種類のオペレーティングシステム、DBMS、及び開発ツールを使って、何百ものアプリケーションを開発している。それぞれの場合、他のアプリケーションとの統合を考慮せずに特定の要件を満たしている。

20

【0014】

アプリケーション統合に関して、幾つかの力強い傾向が市場を動かしている。例えば、ピア・ツー・ピアネットワークと分散処理における顕著な展開は、事業にとって協働者及び供給者を統合するのと同様に、自己の機能的部門を良好に統合できるようにしている。また、前述したインターネット/「イントラネット」/「エクストラネット」の爆発的増加は、バックエンド(back-end)の旧式アプリケーションとの統合を必要とする新しいクラスの「ヒューマン・アクティブ」アプリケーションに対する要求を刺激している。

30

【0015】

企業のアプリケーションソフトウェアパッケージ採用における世界的なすさまじい成長も、バックエンドの旧式アプリケーションとの統合を必要とする。結局、例えばIBMのMQシリーズのメッセージキューイング(message queuing)のようなメッセージ指向ミドルウェア(MOM)製品が徐々に普及してくるようになる。顧客がMOMとの単純な1対1アプリケーション接続の利点を理解すると、多数対多数アプリケーション統合への関心が著しく増大することになる。

【0016】

多数対多数アプリケーション統合(又は1対多数統合でも)に関してキーとなる問題は、全システムを接続するのに必要なデータ変換の数が極めて大きいことである。これは、システム数の二乗にオブジェクトタイプの数に掛けた程の大きさである。この問題を扱い易くするためには、オブジェクトの意味翻訳を形式翻訳と分離する方法を有し、その形式翻訳を実行するメタデータ駆動のツールを有することが望ましい。

40

【0017】

急速に変化する現在の環境において、世界中の多数の開発者は、カスタマイズされた複数のアプリケーション特有の翻訳スキームを埋め込む必要なく、本質的に異なるアプリケーションが互いに通信できる必要性を満たすシステムを開発することに努力を集中している。これはいまだ未解決な必要性として世界経済の緊急の課題に置かれている。

50

【0018】

本出願は、Jeffrey Michael Fischer及びMark Coyleによる2001年4月14日出願の米国仮特許出願第60/283,765号「統合オブジェクト」の出願日に関する恩典を請求する。

【0019】

(発明の詳細な説明)

統合オブジェクトを企業ビジネスアプリケーションと共に用いるための方法及びシステムが開示されている。或る実施形態では、本方法は、第1企業アプリケーション用の第1フォーマットを有する外部オブジェクトを第2企業アプリケーションに取り込む(import)するステップと、統合オブジェクトを使って第1フォーマットの外部オブジェクトを、第2企業アプリケーション用にフォーマットされた第2フォーマットの外部オブジェクトに変換するステップと、第2ビジネスアプリケーション内の第2フォーマットの外部オブジェクトを用いるステップとを含む。

10

【0020】

以下の説明では、分かり易くする目的で、本発明が完全に理解できるように特定の専門用語を使っている。しかしながら、本発明を実行するためにこれら特定用語の詳細が不要であることは当業者であれば明白であろう。

【0021】

以下の詳細な説明の幾つかの部分は、アルゴリズムの用語と、コンピュータメモリ内のデータビットを操作する記号表示とで示されている。これらのアルゴリズムの説明及び表現は、データ処理分野の当業者が作業の内容を別の当業者に最も効果的に伝えるために用いられる手段である。アルゴリズムは、ここではそして一般的には、所望の結果に到る自己矛盾のない一連のステップであると考えられている。前記ステップは、物理量の物理的操作を要求するものである。通常、必ずというわけではないが、これらの量は、記憶、移動、組み合わせ、比較、或いは操作され得る電気又は電磁の信号形態をとる。主として一般的な使用のために、これらの信号は、ビット、値、要素、記号、文字、用語、数などとして引用されることが好都合であると分かっている。

20

【0022】

しかしながら、これらの用語や同様の用語のすべては、適切な物理量と関連付けられているとともに、これら物理量に適用される単に都合の良いラベルであると理解されるべきである。以下の議論から明白なように、他に特定して述べていなければ、「処理」、「計算」、「演算」、「決定」、又は「表示」などの用語を用いた内容は、コンピュータシステム又は同様な電子計算装置に関する動作及び処理を言及し、物理(電子)量で示されるコンピュータシステムのレジスタ及びメモリ内のデータを、同様に物理量として示されるコンピュータシステムのメモリ若しくはレジスタ、又は情報記憶装置、伝送装置、若しくは表示装置等の別の装置内の他のデータに変換し且つ操作する。

30

【0023】

また、本発明は、ここでの働き(オペレーション)を実行するための装置にも関係している。要求された目的のためにこの装置が特別に構成されるようにしてもよいし、又はコンピュータに記憶されたコンピュータプログラムによって選択的に起動又は再構成される汎用コンピュータを備えていてもよい。そのようなコンピュータプログラムは、限定するわけではないが、フロッピー(登録商標)ディスク、光ディスク、CD-ROM、磁気光ディスクを含む何れかの種類のディスク、読み取り専用メモリ(ROM)、ランダムアクセスメモリ(RAM)、EPROM、EEPROM、磁気若しくは光カード、又は電子命令を記憶するのに適し、かつそれぞれがコンピュータシステムバスに連結される何れかの種類の媒体等のコンピュータ読み出し可能な記憶媒体内に記憶されることができる。

40

【0024】

ここで提示されるアルゴリズム及び表示は、本来的に特定のコンピュータ又は他の装置に関係付けられているものではない。様々な汎用システムを、ここでの教示に関するプログラムと共に用いることができたり、又は要求された方法のステップを実行するために更に

50

特別の装置を構築することが好ましいことが分かっている。様々なこれらのシステムに要求される構造は、以下の説明から明らかになるだろう。加えて、本発明は何れかの特別なプログラム言語を引用して記載されてはいない。ここで記載したように様々なプログラミング言語が本発明の内容を実行するために用いられ得ることを認識されたい。

【0025】

図1は、統合オブジェクトを実行するシステム100のブロック図を示す。システム100は、関連アプリケーション環境110を含んでいる。関連アプリケーション環境110は、2つの異なる企業アプリケーションシステムがデータを共有できるようにする。例えば、或る会社が或る基本的な企業アプリケーションシステム（基本システム）を使用し、かつその会社のサプライヤーが別の外部企業アプリケーションシステム（外部システム）を使用して、環境110は、前記外部システムからの企業データが前記基本システムに基本システムの特定フォーマットでアクセスできることを許可する。

10

【0026】

環境110内には2つの要素がある。アプリケーション111は環境110の能力を提供し、外部統合インタフェース141を介して外部システム140とインタフェースする。第2の要素は、関連データベース120である。

【0027】

アプリケーション111は、外部アプリケーションアダプタ116を備えている。外部アプリケーションアダプタ116は、インタフェース141のような外部統合インタフェースと通信する。アダプタ116は、外部データオブジェクトの意味を失うことなく、外部システム140のデータオブジェクトのローカル表現を作り出す。このローカル表現は、データ変換アダプタ115に送られる。外部システム140からもたらされたローカル表現内のフィールド名は、外部システム140のデータオブジェクトの共通表現を作成するため、環境110の命名取り決めに応じて名称変更される。更に、環境140と環境110のオブジェクト構造間の有意な差異は、前記データ変換アダプタによって解決される。前記共通表現が、内部アプリケーションアダプタ114に送られる。内部アプリケーションアダプタ114は、内部オブジェクト層113と直接通信する。

20

【0028】

オブジェクト層113は、内部アプリケーションアダプタ114からの共通表現を受信し、それは環境110と十分に統合可能である。この共通表現は、オブジェクト層113によって、関連データベース120内にオブジェクトとして記憶される。ユーザインタフェース112はオブジェクト層113の上に存在し、データを表示する。また、ウェブサーバー150を介してウェブブラウザ151によりそのデータを表示することもできる。

30

【0029】

関連データベース120は、アプリケーションテーブル121及びリポジトリテーブル122を備えている。アプリケーションテーブル121は、会計、注文、又は連絡情報等の実際のデータを含んでいる。リポジトリテーブル122は、統合オブジェクト123及びビジネスオブジェクトも同様に含んでいる。統合オブジェクト123は、外部システム140の外部データオブジェクトのフォーマットを記述したメタデータであり、外部リポジトリ130内に記憶されることができる。リポジトリテーブル122は、アプリケーションによって使用される他の構造と同様に統合オブジェクトを含みながら、アプリケーションテーブル121のフォーマットを記述したメタデータである。

40

【0030】

統合オブジェクト123はアダプタ114～116へ送られ、各アダプタは環境110内に外部データオブジェクトを統合するため、外部オブジェクト定義を読み出す。アプリケーション111は、リポジトリテーブル122からの静的な構成データを、アプリケーションテーブル121内に入っている動的データと併合し統合することによって、ブラウザ151を通してユーザが見るものを生成する。例えば、アプリケーション111は、HTMLコードを構築しブラウザ151を介して閲覧されるウェブサーバー150へ提供する。ウェブページは、コールセンターアプリケーション用の会計情報及びデータフィー

50

ルドを表示することができる。この表示される情報は、データが閲覧される企業アプリケーションとは異なる企業アプリケーションからのものであってもよい。

【0031】

前述システム100のフレームワークは、ユーザがプログラミングコードを書いたり、又はそれをするコンサルタントを雇うことなく、異なるデータフォーマットを有する複数のビジネスシステムを統合することを許可する。アダプタ114～116は、交換可能な構成要素として統合オブジェクト123を使用し、その結果、アダプタ114～116が再利用され得る。環境110は、統合オブジェクト123を使って自己のオブジェクト定義をエクスポート（export）することもでき、その結果、外部システム140は、環境110のデータオブジェクトを統合することもできる。更に、環境110は、（外部リポジトリ130内に在る場合）外部システム140のオブジェクト定義を取り出し、それを使って外部システム140のデータオブジェクトを統合することもできる。

10

【0032】

図2は、本発明の或る実施形態に従ったデータ統合システム200の高レベルのブロック図である。外部スキームのデータオブジェクトは外部リポジトリ210内にある。リポジトリ210は、それ自身のフォーマットスキームリポジトリ210内に、外部データオブジェクトを記述した外部統合オブジェクト定義が入っている。リポジトリ210は、統合オブジェクトウィザード220によってアクセス可能である。ウィザード220は、リポジトリ210からオブジェクト定義を抽出し、かつ局所（ローカル）化された統合オブジェクト230を生成する。

20

【0033】

ウィザード220は、アプリケーション111を実行するユーザ対話型ソフトウェアを表している。図2は、外部リポジトリ210からオブジェクト定義を読み出すとともに、統合オブジェクト230を生成する前記ウィザード220の機能を示している。統合オブジェクトは、アダプタ260によって処理された外部表現と、統合オブジェクトインスタンス240と、XML文書250との間のマッピングを宣言的に定義する。表1は、統合オブジェクト定義を記憶するためのデータベーススキーマの一実施形態を示すが、この他にも本発明の概念及び範囲内で実現でき考えられる実施方式はある。

【0034】

（表1）

30

カラム(C o l u m n)名	説明	データ型式	外部キー目 標テーブル
ID	このオブジェクトの固有のID	ID	
NAME	内部（標準）表現によるこのオブジェクトの名称	ストリング	
EXT__NAME	外部表現によるこのオブジェクトの名称	ストリング	
XML NAME	このオブジェクトのXMLタグ	ストリング	
ADAPTER__INFO	外部フォーマットと内部フォーマット間の翻訳のためにアダプタにより使用される追加情報	ストリング	

40

【0035】

統合オブジェクト230は、命名されかつ分類されたフィールドを含む「構成要素」より構成される。統合オブジェクトの構成要素は互いに親子関係にある。統合構成要素及びそれらのフィールドは、リポジトリ内の他の物理的構成要素とリンク付けられることができる。物理的オブジェクトに統合構成要素をリンクすることは、オブジェクトを外部システムと交換するときを利用するための標準的な表現をアダプタ114～116のために提供する。例えば、統合オブジェクト230の構成要素がテーブルの定義とリンクされる場合

50

、アダプタ 114 ~ 116 は、その出力データを同じ定義を有するテーブル又は別のテーブルに配置することができる。アダプタ 114 ~ 116 は、（統合オブジェクト定義内で参照された表を直接利用しない場合、）外部システムに統合オブジェクトの型式と、統合構成要素から実際のステージングテーブル（staging table）へのマッピングとを提供する。統合オブジェクトの標準的な内部メモリ表現が同様に生じる。

【0036】

統合構成要素の定義のそれぞれは、親への外部キーだけでなく独自のキーも含んでいる。これらのキーは、複数のフィールドを含んでいてもよいし重複していてもよい。表 2 は、統合構成要素の定義を記憶するためのデータベーススキーマの一実施形態を示しているが、この他にも本発明の概念及び範囲内で実現でき考えられる実施方式はある。

10

【0037】

（表 2）

カラム (Column) 名	説明	データ型式	外部キー目標テーブル
ID	この構成要素の固有の ID	ID	
NAME	内部（標準）表現によるこの構成要素の名称	ストリング	
EXT_NAME	外部表現によるこの構成要素の名称	ストリング	
XML_NAME	この構成要素の XML タグ	ストリング	
PAR_OBJ_ID	この構成要素が属する親オブジェクトの ID	ID	S_INT_OBJ
PAR_CMP_ID	この構成要素の親構成要素の型式の ID（ルートの場合は NULL）	ID	S_INT_COMP
ADAPTER_INFO	外部フォーマットと内部フォーマット間の翻訳のためにアダプタにより使用される追加情報	ストリング	

20

30

【0038】

テーブルが複数の構成要素を含み、かつそれぞれの行が唯 1 つの構成要素に属している非正規化テーブル（表）には、正規化された論理表現が提供される。リポジトリ 210 は、非正規化テーブルの最上段に正規化されたオブジェクトを示している。

【0039】

統合オブジェクト定義のレベルが図 2 に示されている。3 つのレベルは、統合オブジェクト定義 231、統合構成要素定義 232、統合構成要素フィールド定義 233 である。統合オブジェクト定義 231 は、データオブジェクトを外部システムの表現とローカルシステムの表現との間でマップする。図 2 に示すように、各定義レベルは、標準セクションとして 3 つのセクションから構成される。即ち、ローカルシステムが理解できるデータオブジェクトを記述する規範セクションと、外部システムが理解できるデータオブジェクトを記述する外部セクションと、さらに標準化されたフォーマットでデータオブジェクトを記述する XML セクションである。XML と説明したが、XML と同様の他のマークアップ言語も、本実施形態の範囲内にあるとみなされる。

40

【0040】

統合オブジェクト定義の第 2 レベルは、統合構成要素定義レベル 232 である。このレベルは、スカラーフィールドの集合を一緒にグループ分けしている。統合オブジェクト内の構成要素型式の集合は、ある「ルート」構成要素によって階層的に配置される。オブジェクトインスタンスの中には、各構成要素の型式に関する多くのインスタンスが存在し得る。統合オブジェクト定義の第 3 レベルは、統合構成要素フィールド定義 233 である。フ

50

フィールドは、ストリング、数、又は日付等の単純なスカラー値を定義する。表3は、統合構成要素フィールド定義を記憶するためのデータベーススキーマの一実施形態を示しているが、この他にも本発明の概念及び範囲内で実現でき考えられる実施方式はある。

(表3)

カラム名	説明	データ型式	外部キー目標テーブル
ID	このフィールドの固有のID	ID	
NAME	内部(標準)表現によるこのフィールドの名称	ストリング	
EXT_NAME	外部表現によるこのフィールドのネーム	ストリング	
XML_NAME	このフィールドのXMLタグ	ストリング	
PAR_CMP_ID	このフィールドの親構成要素の型式のID	ID	S_INT_C OMP
DATA_TYPE	標準表現によるこのフィールドのデータ型式(例えば、ストリング、ID、数、日付、通貨等)	ストリング	
EXT_DATA_TYPE	外部表現によるこのフィールドのデータ型式	ストリング	
LENGTH	標準表現によるこのフィールドの長さ	数	
EXT_LENGTH	外部表現によるこのフィールドの長さ	数	
ADAPTER_INFO	外部フォーマットと内部フォーマット間の翻訳のためにアダプタにより使用される追加情報	ストリング	

10

20

【0041】

必要であれば、フィールドレベル233は、スカラー値の標準表現と外部表現との間でフィールド名称マッピングを提供する。統合構成要素のインスタンス内で、所与のフィールドインスタンスが一度だけ出現する。

30

【0042】

統合オブジェクト230の標準セクションと外部セクションを用いて、アダプタ260は、外部システムのインスタンス表現と統合オブジェクトインスタンス240の間でマップすることができる。統合オブジェクト230のXMLセクションを利用して、XMLページ250を作成することができる。アダプタ260は、アダプタ116のような外部アダプタでもよいし、アダプタ114のような内部アダプタでもよい。

【0043】

図3は、統合オブジェクト330と統合オブジェクトインスタンス340の間の差異を示している。一般的に、統合オブジェクト330はメタデータで、即ち、特定のデータ集合の一般化された表現又はモデルである。図3に示したように、統合オブジェクトインスタンスは、統合オブジェクト330のフォーマット又は構造で編成された実際のデータである。

40

【0044】

図4は、本発明の一実施形態に従ったウィザード220を用いて統合オブジェクトを構築するための典型的なユーザインタフェース400を示している。ライン(line)410で、新規又は既存のプロジェクトが指定される。ライン420で、ベース(ソース)システムとインタフェースするためのビジネスアプリケーションが選択される。次に、ユーザは、ライン430で生成される新規の統合オブジェクトのためにモデル化されるであろうソ

50

ースオブジェクトを選択する。ライン440で、ユーザは新規統合オブジェクト用の固有の名称を入力する。同様のユーザインタフェースが構成要素の選択のために用いられ、2つのシステムの間で転送される。また、各構成要素の全ての子に対するアクセスもあり、しれは選択されたり又は選択されなかったりする。

【0045】

リポジトリメタデータの変更は珍しいことではなく、そして統合オブジェクト定義が外部オブジェクト表現の実際の構造と一致しない場合は、実行時に問題を引き起こす可能性がある。これらの問題を回避するためには、統合オブジェクト定義を、変更の可能性がある外部オブジェクト定義と「同期化させる」ツールを有することが望ましい。図5Aと5Bは、このツールの考えられる一実施形態を示している。

10

【0046】

図5Aは、統合オブジェクトの代表的な同期ダイアグラム500を示している。図5Aは、2つの統合オブジェクトを含む。統合オブジェクト510は、新規の内部メモリ統合オブジェクトである。統合オブジェクト520は、データベース内に記憶された既存の統合オブジェクトである。ダイアグラム500は、多数の構成要素を備えた新規のビジネスオブジェクト510を示す。ダイアグラム500に示すように、同期化のために対応するボックスのクリックによって各構成要素が選択される。既存オブジェクト520と新規オブジェクト510間の差異は、構成要素511（ビジネスアドレス）及び構成要素512（機会）に関して顕著である。この様に、ビジネスアドレス構成要素511及び機会構成要素512は、既存の統合オブジェクト520に加えられる新規の構成要素である。図5Bは、同期化された統合オブジェクト550を示している。構成要素「接触」560とその全ての子が更新されている。構成要素「ビジネスアドレス」551は、構成要素「機会」552と同様に新規である。また、同期化の構成要素として選択されなければ、構成要素は削除される。

20

【0047】

図6は、統合された多重プロセッサを表わすコンピュータシステム600を示し、そこでは本発明の要素が実行され得る。例えば、システム600は、サーバー150のアーキテクチャでもよいし、又は環境110が存在するハードウェアであってもよい。コンピュータシステム600の一実施形態は、情報を伝達するためのシステムバス620と、バス620に連結されている情報を処理するためのプロセッサ610とを備えている。さらに、コンピュータシステム600は、バス620に連結されたランダムアクセスメモリ（RAM）、又は（ここではメインメモリとして引用された）他の動的記憶装置625を備え、プロセッサ610により実行される情報及び命令を記憶する。

30

【0048】

また、メインメモリ625は、プロセッサ610が命令を実行する間に、一時的な変数又は他の中間情報を記憶するために用いられる。また、コンピュータシステム600は、プロセッサ610が使用する静的情報及び命令を記憶するバス620に連結された読み出し専用メモリ（ROM）及び/又は他の静的記憶装置626を含む。環境110は、リポジトリ130が記憶装置627で実行されるシステム600内に実装される。アダプタ114～116とリポジトリテーブル間の通信は、バス620を介して行われるのが普通である。ブラウザインタフェースは、表示装置643上に示される。

40

【0049】

また、磁気ディスク又は光ディスク等のデータ記憶装置627とそれに対応する装置は、情報及び命令を記憶するコンピュータシステム600に連結されることができる。コンピュータシステム600は、I/Oインタフェース630を介して、第2のI/Oバス650に連結することもできる。表示装置643、入力装置（例えば英数字入力装置642及び/又はカーソル制御装置641）を含め、複数のI/O装置を1/Oバス650に連結することができる。

【0050】

通信装置640は、ネットワークを介して他のコンピュータ（サーバー又はクライアント

50

)にアクセスするための装置である。通信装置640は、モデム、ネットワークインタフェースカード、又はイーサネット(登録商標)、トークンリング、若しくは他の型式のネットワークに連結するのに用いられる他の周知のインタフェース装置を備えている。

【0051】

図7は、統合オブジェクトを企業ビジネスアプリケーションと共に用いる方法の高いレベルのフローチャート700を示している。このプロセスは処理701で始まり、フローは処理ブロック702へ続く。ユーザは、統合対象のアプリケーションに関連する内部及び外部インタフェースの型式を選択する。

【0052】

フローは、処理ブロック703へ続き、ここでウィザード220を使って、外部アプリケーションのビジネスオブジェクトのための新規の統合オブジェクトを作成する。処理ブロック704で、必要であれば新規の統合オブジェクトが修正される。例えば、フィールド名称又は関係が変更される。

【0053】

処理ブロック705では、ウィザード220を実行してローカルアプリケーション用の統合オブジェクトを作成する。これらのローカル統合オブジェクトは、ローカルデータオブジェクトを外部システムにエクスポートできるようにする。フローは、処理ブロック706へ続き、前述したように、ここでユーザは新規のローカルデータオブジェクトを修正する。

【0054】

フローは、処理ブロック707へ続き、ここでデータマップを書き込む。データマップは、外部及び内部インタフェースの何れが、内部及び外部両方の新規の統合オブジェクトと対応するかを示す。処理ブロック708では、ユーザを介することなく内部及び外部両方のオブジェクトを統合する。フローはブロック709で完了する。

【0055】

統合オブジェクトを企業のビジネスアプリケーションと共に使用する方法及びシステムを開示した。本発明を特定の例とサブシステムに関して説明してきたが、本発明がこれら特定の例又はサブシステムに限定されるものではなく、他の実施形態にも拡張されることは当業者であれば自明であろう。本発明は、特許請求の範囲で特定されるようにこれら他の実施形態の全てを含むものである。

【0056】

添付図面は本発明の明細書の一部を成して本発明の好適な実施形態を示し、前記本発明の一般的な記述及び本発明の詳細な記述と共に本発明の原理を説明し教示する役目を果たす。

【図面の簡単な説明】

【0057】

【図1】統合オブジェクトを実行するシステム100のブロック図である。

【図2】本発明の一実施形態によるデータ統合システム200の高いレベルでのブロック図である。

【図3】統合オブジェクト330と統合オブジェクトインスタンス340の間の相違を示す図である。

【図4】本発明の一実施形態に従いウィザード220を使って統合オブジェクトを構築する代表的なユーザインタフェース400を示す図である。

【図5A】統合オブジェクトの典型的な同期ダイアグラム500を示す図である。

【図5B】同期された統合オブジェクト550を示す図である。

【図6】統合された多重プロセッサを表すコンピュータシステム300を示す図であり、本発明の要素がここで実行される。

【図7】企業ビジネスアプリケーションと共に統合オブジェクトを用いる方法の高いレベルでのフローチャート700を示す図である。

10

20

30

40

【 図 1 】

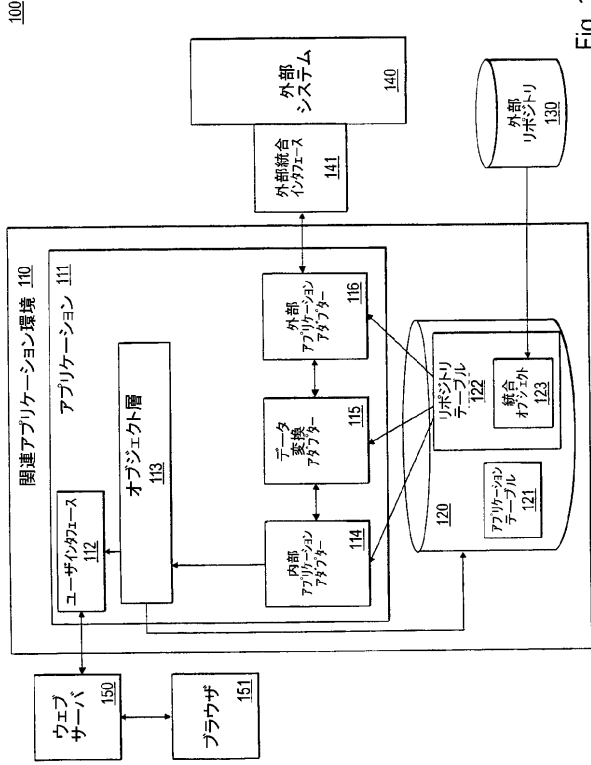


Fig. 1

【 図 2 】

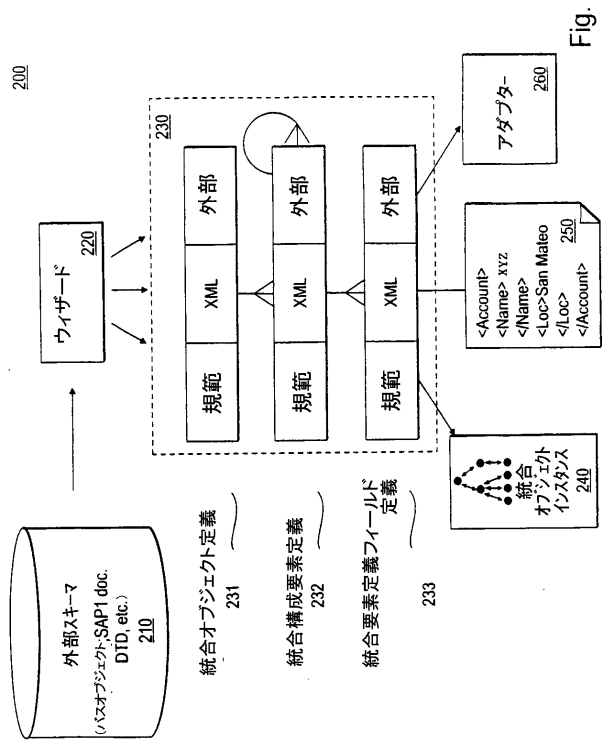


Fig. 2

【 図 3 】

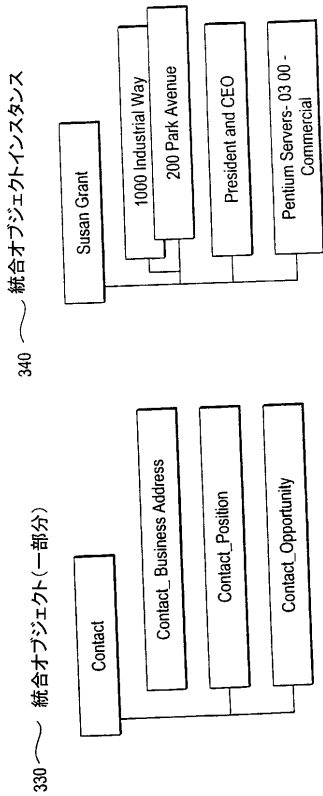


Fig. 3

【 図 4 】

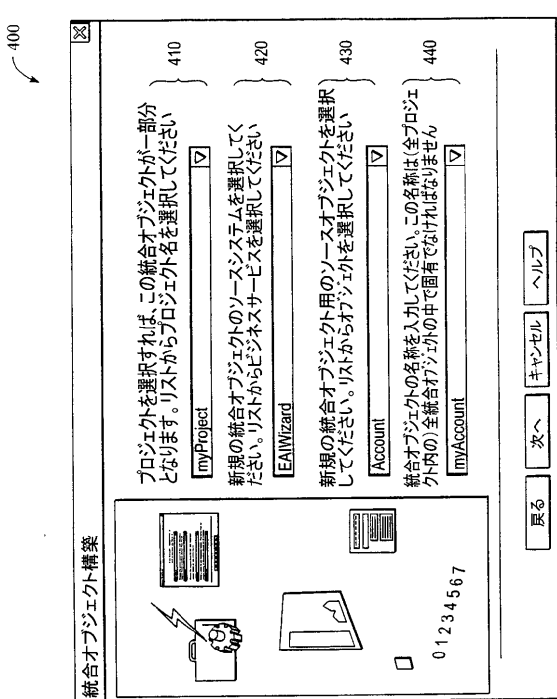
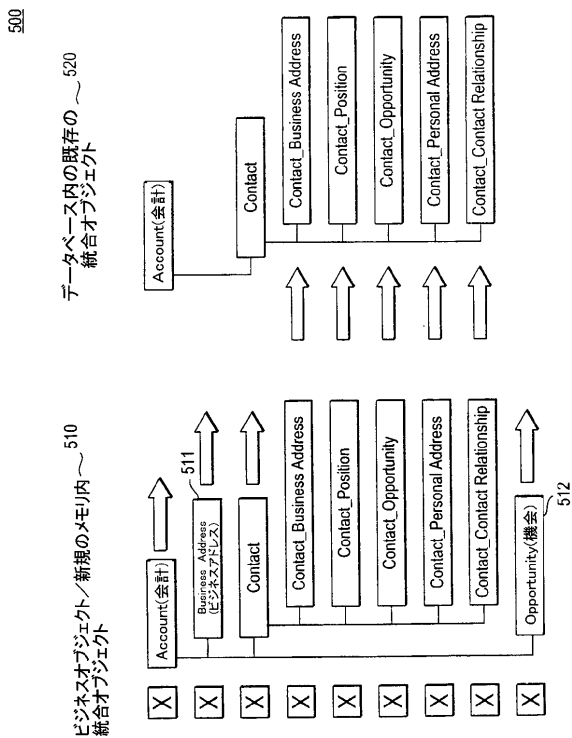


Fig. 4

【 図 5 A 】



【 図 5 B 】

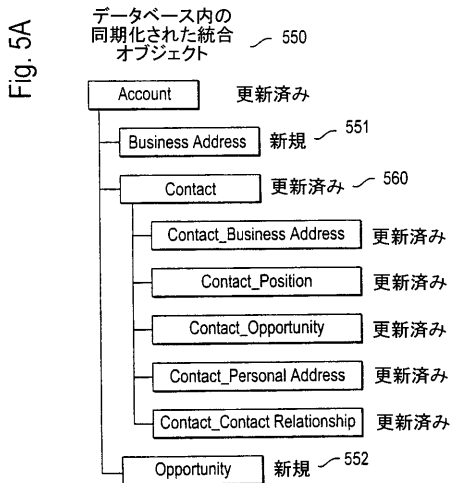


Fig. 5A

Fig. 5B

【 図 6 】

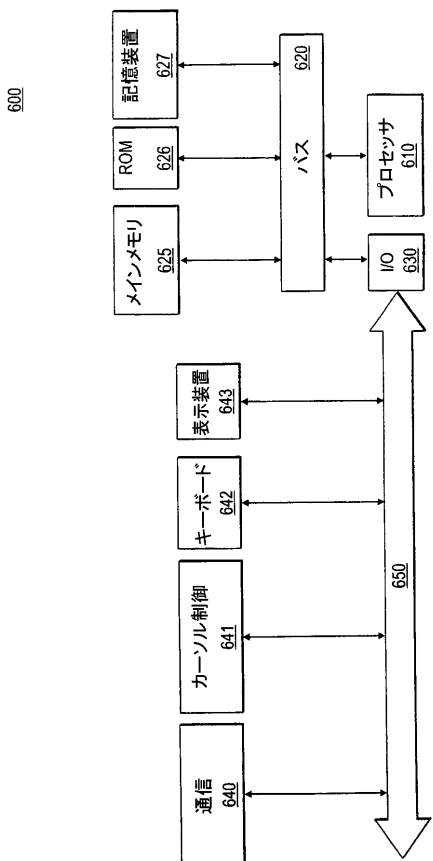


Fig. 6

【 図 7 】

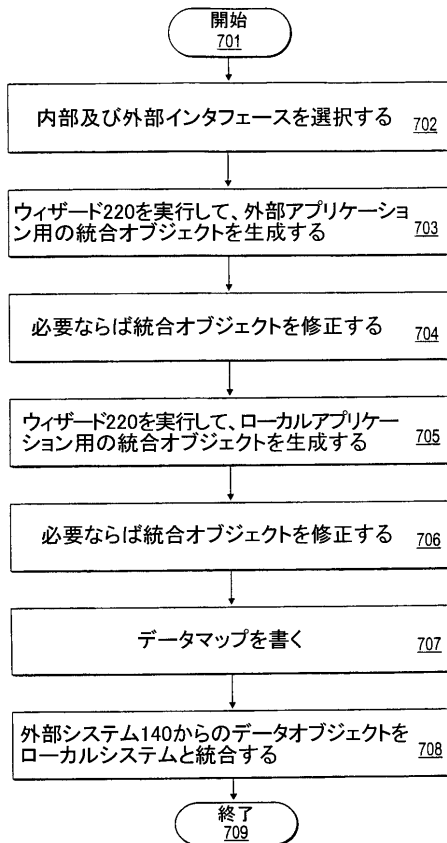


Fig. 7

【国際公開パンフレット】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
10 April 2003 (10.04.2003)

PCT

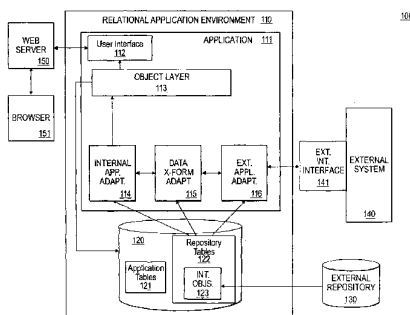
(10) International Publication Number
WO 03/029955 A1

- (51) International Patent Classification: G06F 7/00
- (21) International Application Number: PCT/US02/30884
- (22) International Filing Date: 27 September 2002 (27.09.2002)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data: 09/968,735 28 September 2001 (28.09.2001) US
10/120,030 9 April 2002 (09.04.2002) US
- (71) Applicant (for all designated States except US): SIEBEL SYSTEMS, INC. [US/US]; 2207 Bridgepointe Parkway, San Mateo, CA 94404 (US).
- (72) Inventors; and
(75) Inventors/Applicants (for US only): FISCHER, Jeffrey, Michael [US/US]; 1923 Mason Street, San Francisco, CA 94133 (US); COYLE, Mark [US/US]; 2 Townsend Street # 1-210, San Francisco, CA 94107 (US).
- (74) Agents: MALLIE, Michael, J. et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 7th floor, 12500 Wilshire Boulevard, Los Angeles, CA 90025 (US).
- (81) Designated States (national): AF, AG, AI, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LX, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.
- (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SI, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CI, CG, CL, CM, GA, GN, GQ, GW, ML, MR, NI, SN, TD, TG).

Published:
with international search report

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR USING INTEGRATION OBJECTS WITH ENTERPRISE BUSINESS APPLICATIONS



(57) Abstract: A method and system for using integration objects with enterprise business applications is disclosed (Fig.1). A method comprises importing external objects (140) having a first format for a first enterprise application into a second enterprise application (111); using integration objects (123) to transform the first format external objects (140) into second format external objects formatted for the second enterprise application (111); and using the second format external objects in the second business application.



WO 03/029955 A1

WO 03/029955 A1 

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 03/029955

PCT/US02/30884

**METHOD AND SYSTEM FOR USING INTEGRATION OBJECTS WITH
ENTERPRISE BUSINESS APPLICATIONS**

[0001] This application claims the benefit of the filing date of the following Provisional U.S. Patent Application: "INTEGRATION OBJECTS", application number 60/283,765, filed April 14, 2001, having as inventors: Jeffrey Michael Fischer and Mark Coyle.

FIELD OF THE INVENTION

[0002] This invention relates to computer systems, and more specifically to a method and system for using integration objects with enterprise business applications.

BACKGROUND OF THE INVENTION

[0003] According to one observer, if the lifeblood of today's corporations is information, then their arteries are the "inter-application interfaces" that facilitate movement of data around the corporate enterprise. This has more recently become known as an "application network".

[0004] For the typical organization, the application network has grown organically into a collection of ad hoc application integration programs. This menagerie has had a very serious impact on businesses as it increases the time for implementing new applications, prevents senior management from getting a clear picture of the business and, in short, clogs the corporate arteries. In spite of the fact that application integration has become crucial to a competitive corporation's survival, it has nevertheless been acceptable in the prior art to handcraft or "hack" custom code for such purposes at enormous long-term cost to the corporation. Long-term application integration decisions have, likewise, been made at the lowest possible levels based solely on individual project criteria. Because of the decidedly difficult nature of these problems, an effective enterprise application integration (EAI) solution has yet to be found.

WO 03/029955

PCT/US02/30884

[0005] The advent of the Internet, client/server computing, corporate mergers and acquisitions, globalization and business process re-engineering, have together forced corporate information technology (IT) departments to continually seek out new, and often manual, ways to make different systems talk to each other--regardless of how old some of those systems may be. In the ensuing chaos, inadequate communications systems have had a debilitating effect on IT's abilities to move as fast as the business needs it to.

[0006] Recent trends in IT have only exacerbated this problem by increasing--often by an order of magnitude--the amount of inter-application interfacing needed to support them. Most recently, enterprise applications have performed such functions as data warehousing and enterprise resource planning (ERP), and facilitated electronic commerce. A brief review of these three technologies would, therefore, be helpful in understanding the long-felt but as yet unresolved need for EAI.

[0007] Data warehousing techniques require large volumes of clean historical data that must be moved, on a regular basis, from many operational systems into the warehouse. Source data is usually structured for online transactional processing (OLTP), while the typical data warehouse also accommodates online analytical processing (OLAP) formats. Therefore, the source data must undergo extensive aggregation and reformatting as it is transferred to the warehouse.

[0008] A typical data warehouse according to the prior art is populated in four steps: (a) extracting the source data; (b) cleaning such extracted data; (c) aggregating the cleaned, extracted data in a number of dimensions; and (d) loading the warehouse. Each warehouse source requires the building of a specific data extraction, cleansing, aggregation, and load routine. Forrester Research estimates that the average large company has approximately four data warehouses. In two years, it is expected that this number will grow to six. The average amount of data contained in each warehouse is also expected to double in size in that same period--from about 130 gigabytes to about 260 gigabytes.

[0009] The problems associated with such large amounts of data growing at an ever-increasing pace is exacerbated by the quality of source data. According to a study conducted by the META Group, typical data warehouses are being loaded today with as much as 20% poor quality data. That same study indicates that about 70% of its respondents used extraction, cleansing and loading processes that were coded by hand.

WO 03/029955

PCT/US02/30884

With respect to the required aggregation processes, anecdotal evidence also reveals that as much as 50 hours of computer time may be required to complete this function alone. It is readily apparent that significant maintenance efforts would be involved with programs coded in such a manner.

[0010] On the other hand, typical ERP systems are essentially large, integrated packaged applications that support core business functions, such as payroll, manufacturing, general ledger, and human resources. Implementing an ERP system, however, can be an overwhelming process for a number of reasons.

[0011] First and foremost, the corporation is buying a product and not building a solution. This means that business units within the corporation must adapt to the product and how it works, not the other way around. Furthermore, today's ERP systems cannot replace all of a corporation's custom solutions. They must, therefore, communicate effectively with other legacy systems in place. Finally, it is not atypical for a corporation to employ more than one and completely different ERP system because a single vendor cannot usually meet every organizational need.

[0012] As a result, the options for getting data into and out of an ERP system preclude known approaches used for data warehousing. Each ERP system has a proprietary data model that is constantly being enhanced by its vendor. Writing extract or load routines that manipulate such models is not only complicated, but is also discouraged by the vendor since data validation and business rules inherent in the enterprise application are likely to be bypassed. Instead, ERPs require interaction at the business object level, which deals with specific business entities such as general ledgers, budgets or accounts payable.

[0013] Electronic commerce in one form or another has been around for many years. In essence, it got its start with electronic data interchange (EDI). EDI permitted companies to communicate their purchase orders and invoices electronically, and continued to develop such that today's companies use EDI for supply chain management. However, not until the more recent exploding use of online Internet websites to buy, sell, and even auction, items of interest has there been such a dire need for robust, effective EAI.

[0014] Applications get developed in order to accomplish a specific business objective in a measured time frame. In a typical large organization, different teams of

WO 03/029955

PCT/US02/30884

people using a wide assortment of operating systems, DBMSs and development tools develop hundreds of applications. In each case, the specific requirements are satisfied without regard for integration with any other applications.

[0015] Several powerful trends are driving the market for application integration. For example, significant developments in peer-to-peer networking and distributed processing have made it possible for businesses to better integrate their own functional departments as well as integrate with their partners and suppliers. The aforementioned Internet/"intranet"/"extranet" explosion is also fueling the demand for a new class of "human active" applications that require integration with back-end legacy applications. Tremendous growth around the world in the adoption of enterprise application software packages also requires integration with back-end legacy applications. Finally, message oriented middleware (MOM)--products such as IBM's MQSeries message queuing product--are becoming increasingly popular. Once customers realize the benefits of simple one-to-one application connectivity with MOM, their interest in many-to-many application integration increases significantly.

[0016] The key problem with many-to-many application integration (or even one-to-many integration) is that number of potential data transformations needed to connect all systems is very large. It may be as much as the square of the number of systems multiplied by the number of object types. To make this problem tractable, it is desirable to have methods to separate the semantic translation of objects from format translations and to have metadata-driven tools to perform the format translations.

[0017] In today's rapidly changing environment, the concerted efforts of thousands of developers worldwide are focused on developing a system that satisfies the need for disparate applications to communicate with each other, without the necessity of embedding multiple, customized application-specific translation schemes. This as yet unfulfilled need is grounded in the imperative of the global economy.

WO 03/029955

PCT/US02/30884

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] The accompanying drawings, which are included as part of the present specification, illustrate the presently preferred embodiment of the present invention and together with the general description given above and the detailed description of the preferred embodiment given below serve to explain and teach the principles of the present invention.

[0019] **Figure 1** illustrates a block diagram of a system 100 implementing integration objects.

[0020] **Figure 2** is a high-level block diagram of a data integration system 200 according to one embodiment of the invention.

[0021] **Figure 3** illustrates the difference between an integration object 330 and an integration object instance 340.

[0022] **Figure 4** illustrates an exemplary user interface 400 for building integration objects using wizard 220, according to one embodiment of the present invention.

[0023] **Figure 5A** illustrates an exemplary synchronization diagram 500 for integration objects.

[0024] **Figure 5B** illustrates a synchronized integration object 550.

[0025] **Figure 6** illustrates a computer system 300 representing an integrated multi-processor, in which elements of the present invention may be implemented.

[0026] **Figure 7** illustrates a high level flow diagram 700 of a method for using integration objects with enterprise business applications.

WO 03/029955

PCT/US02/30884

DETAILED DESCRIPTION

[0027] A method and system for using integration objects with enterprise business applications is disclosed. In one embodiment, the method comprises importing external objects having a first format for a first enterprise application into a second enterprise application; using integration objects to transform the first format external objects into second format external objects formatted for the second enterprise application; and using the second format external objects in the second business application.

[0028] In the following description, for purposes of explanation, specific nomenclature is set forth to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required in order to practice the present invention.

[0029] Some portions of the detailed descriptions, which follow are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

[0030] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into

WO 03/029955

PCT/US02/30884

other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0031] The present invention also relates to apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus.

[0032] The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear from the description below. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein.

[0033] **Figure 1** illustrates a block diagram of a system 100 implementing integration objects. System 100 includes a relational application environment 110. The relational application environment 110 allows two different business application systems to share data. For example, if a company uses one base enterprise application system, and the company's supplier uses a different external business application system, environment 110 allows enterprise data from the external system to be accessible to the base system, in the base system's specific format.

[0034] Within environment 110 are two elements. Application 111 provides the ability of environment 110 to interface with external system 140 via external integration interface 141. The second element is relational database 120.

[0035] Application 111 includes an external application adapter 116. The external application adapter 116 communicates with external integration interfaces, such as interface 141. Adapter 116 creates a local representation of the external system's 140

WO 03/029955

PCT/US02/30884

data objects without losing the meaning of the external data objects. The local representation is provided to data transformation adapter 115. The names of fields within the local representation brought from external system 140 will be renamed to correspond with the naming conventions of environment 110, in order to create a common representation of the external system's 140 data objects. In addition, any semantic differences between the object structures of environment 140 and environment 110 are resolved by the Data Transformation Adapter. The common representation is provided to an internal application adapter 114. The internal application adapter 114 communicates with the internal object layer 113 directly.

[0036] Object layer 113 receives the common representation from internal data adapter 114, which is fully integratable with environment 110. The common representation may be stored as an object in relational database 120 by object layer 113. A user interface 112 may reside on top of object layer 113 and display the data. The data may also be displayed by a web browser 151 via a web server 150.

[0037] Relational database 120 includes application tables 121 and repository tables 122. Application tables 121 contain real data, such as accounts, orders or contact information. Repository tables 122 include integration objects 123 and business objects as well. The integration objects 123 are metadata that describes the format of external data objects of external system 140 that may be stored in external repository 130. The repository tables 122 are metadata that describes the format of application tables 121, as well as other structures used by the application, including integration objects. The integration objects 123 are provided to adapters 114-116, which read the external object definitions in order to integrate the external data objects into environment 110. Application 111 generates what a user will see via browser 151, by merging and integrating static configuration data from repository tables 122 with the dynamic data contained in application tables 121. For example, application 111 builds the HTML code provided to the webserver 150 that is viewed via browser 151. The web page may display account information and data fields for a call center application. The displayed information may be from an enterprise application different from the one with which the data is viewed.

[0038] The framework of system 100 described above allows multiple business systems having different data formats to be integrated, without the user having to write

WO 03/029955

PCT/US02/30884

programming code or to hire a consultant to do so. The adapters 114-116 use the integration objects 123 as interchangeable components, so that the adapter 114-116 may be reused. Environment 110 may also be able to export its own object definitions using integration objects 123, so that external system 140 may integrate environment's 110 data objects. Furthermore, environment 110 may take external system's 140 object definitions (if they exist in external repository 130) and use them to integrate external system 140's data objects.

[0039] **Figure 2** is a high-level block diagram of a data integration system 200 according to one embodiment of the invention. Data objects of external schemes are resident within external repository 210. Repository 210 contains external integration object definitions that describe the external data objects in its own format schemes repository 210. Repository 210 is accessible by integration object wizard 220. Wizard 220 extracts the object definitions from repository 210 and creates a localized integration object 230.

[0040] Wizard 220 represents the user-interactive software that implements application 111. **Figure 2** shows the function of the wizard 220: it reads an object definition from external repository 210 and generates an integration object 230. The integration object declaratively defines a mapping between the external representation processed by adapter 260, integration object instance 240 and XML document 250.

Table 1 illustrates one embodiment of a database schema for storing integration object definitions, although other implementations of the schema are feasible and considered within the spirit and scope of the present invention.

Column Name	Description	Datatype	Foreign Key Target Table
ID	Unique Id for this object	ID	
NAME	Name for this object in internal (canonical) representation	String	
EXT_NAME	Name for this object in external representation	String	
XML_NAME	XML Tag for this object	String	
ADAPTER_INFO	Additional information used by adapter	String	

WO 03/029955

PCT/US02/30884

	for translation between external and internal formats		
--	---	--	--

***Table 1**

[0041] An integration object 230 is composed of "components," which contain named and typed fields. The components of an integration object are related to one another by parent-child relationships. Integration components and their fields can be linked to other physical components in its repository. The linking of integration components to physical objects provides a standard representation for adapters 114-116 to use when exchanging objects with external systems. For example, if the components of an integration object 230 are linked to table definitions, then an adapter 114-116 can place its output data in those tables or other tables with the same definition. The adapter 114-116 provides the external system with the integration object type and mapping from the integration components to the actual staging tables (if the tables referenced in the integration object definition were not used directly). A standard in-memory representation of integration objects occurs as well.

[0042] Each integration component definition may include a unique key as well as a foreign key to its parent. These keys may include multiple fields and may overlap. **Table 2** illustrates one embodiment of a database schema for storing integration component definitions, although other implementations of the schema are feasible and considered within the spirit and scope of the present invention.

Column Name	Description	Datatype	Foreign Key Target Table
ID	Unique Id for this component	ID	
NAME	Name for this component in internal (canonical) representation,	String	
EXT_NAME	Name for this component in external representation	String	
XML_NAME	XML Tag for this component	String	
PAR_OBJ_ID	ID of parent object that this component belongs to.	ID	S_INT_OBJ
PAR_CMP_ID	ID of parent component type for this	ID	S_INT_COMP

WO 03/029955

PCT/US02/30884

	component (may be NULL if root).		
ADAPTER_INFO	Additional information used by adapter for translation between external and internal formats	String	

*Table 2

[0043] De-normalized tables where a table includes multiple components, and each row belongs to only one component are provided normalized logical representation. The repository 210 represents a normalized object on top of the de-normalized table.

[0044] The levels of integration object definitions are shown in Figure 2. The three levels are the integration object definition 231, the integration component definition 232, and the integration component field definition 233. The integration object definition 231 maps a data object between an external system's representation and the local system's representation. Each definition level is composed of three sections as shown in Figure 2, a canonical section, that describes the data object so the local system can understand; an external section, that describes the data object so the external system can understand; and an XML section, that describes the data object in a standardized format. Although described as XML, other markup languages similar to XML are considered to be within the scope of this embodiment.

[0045] The second level of integration object definition is the integration component definition level 232. This level groups together a set of scalar fields. The set of component types in an integration object are arranged in a hierarchy, with one "root" component. Within the object instance, there may be multiple instances of each component type. The third level of integration object definitions is the integration component field definition 233. A field defines a simple scalar value, such as a string, a number, or a date. Table 3 illustrates one embodiment of a database schema for storing integration component field definitions, although other implementations of the schema are feasible and considered within the spirit and scope of the present invention.

WO 03/029955

PCT/US02/30884

Column Name	Description	Datatype	Foreign Key Target Table
ID	Unique Id for this field	ID	
NAME	Name for this field in internal (canonical) representation,	String	
EXT_NAME	Name for this field in external representation	String	
XML_NAME	XML Tag for this field	String	
PAR_CMP_ID	ID of parent component type for this field	ID	S_INT_COMP
DATA_TYPE	Data type of this field (e.g. String, ID, Number, Date, Currency, etc) in canonical representation.	String	
EXT_DATA_TYPE	Data type of this field in external representation.	String	
LENGTH	Length of this field in canonical representation	Number	
EXT_LENGTH	Length of this field in external representation	Number	
ADAPTER_INFO	Additional information used by adapter for translation between external and internal formats	String	

*Table 3

[0046] If necessary, the field level 233 provides field name mapping between the canonical and external representations of the scalar value. Within the integration component instance, a given field instance may appear only once.

[0047] The canonical and external sections of integration object 230 may be used by adapter 260 to map between an external system's instance representation and integration object instance 240. The XML sections of integration object 230 may be used to generate XML page 250. Adapter 260 may be an external adapter, such as adapter 116 or an internal adapter, such as adapter 114.

WO 03/029955

PCT/US02/30884

[0048] **Figure 3** illustrates the difference between an integration object 330 and an integration object instance 340. Generally, an integration object 330 is metadata; that is, it is a generalized representation or model of a particular set of data. An integration object instance is actual data organized in the format or structure of the integration object 330 as seen in **Figure 3**.

[0049] **Figure 4** illustrates an exemplary user interface 400 for building integration objects using wizard 220, according to one embodiment of the present invention. At line 410 a new or existing project may be designated. At line 420 a business application for interfacing with the base (source) system is selected. A user then selects a source object that will be modeled for the new integration object being created at line 430. At line 440, the user enters a unique name for the new integration object. A similar user interface is used for selection of components to transfer between the two systems. There is also access to all children of each component that may be selected, or not.

[0050] Alteration of repository metadata is not uncommon, and it can cause problems at runtime if an integration object definition does not match the actual structure of the external object representation. To avoid these problems, it is desirable to have a tool to "synchronize" the integration object definition with a potentially changed external object definition. **Figures 5A and 5B** illustrate one possible embodiment of this tool. **Figure 5A** illustrates an exemplary synchronization diagram 500 for integration objects. **Figure 5A** includes two integration objects. Integration object 510 is a new in-memory integration object. Integration object 520 is an existing integration object stored in a database. Diagram 500 shows a new business object 510 with a number of components. Each component may be selected for synchronization by clicking on its corresponding box, as seen in diagram 500. The differences between the existing object 520 and new object 510 are noticeable with component 511 (business address) and component 512 (Opportunity). Thus, business address component 511 and opportunity component 512 are new components that are added to the existing integration object 520. **Figure 5B** illustrates a synchronized integration object 550. The component "contact" 560 and all its children have been updated. Component "Business Address" 551 is new, as well as, component "Opportunity" 552. A component may also be deleted, by not selecting that component for synchronization.

WO 03/029955

PCT/US02/30884

[0051] Figure 6 illustrates a computer system 600 representing an integrated multi-processor, in which elements of the present invention may be implemented. For example, system 600 may be the architecture of server 150 or the hardware in which environment 110 exists. One embodiment of computer system 600 comprises a system bus 620 for communicating information, and a processor 610 coupled to bus 620 for processing information. Computer system 600 further comprises a random access memory (RAM) or other dynamic storage device 625 (referred to herein as main memory), coupled to bus 620 for storing information and instructions to be executed by processor 610. Main memory 625 also may be used for storing temporary variables or other intermediate information during execution of instructions by processor 610. Computer system 600 also may include a read only memory (ROM) and/or other static storage device 626 coupled to bus 620 for storing static information and instructions used by processor 610. Environment 110 may be implemented in system 600 where the repository 130 is implemented in storage 627. Communication between the adapters 114-116 and repository tables is typically made via bus 620. Browser interface may be shown on display 643

[0052] A data storage device 627 such as a magnetic disk or optical disc and its corresponding drive may also be coupled to computer system 600 for storing information and instructions. Computer system 600 can also be coupled to a second I/O bus 650 via an I/O interface 630. A plurality of I/O devices may be coupled to I/O bus 650, including a display device 643, an input device (e.g., an alphanumeric input device 642 and/or a cursor control device 641).

[0053] The communication device 640 is for accessing other computers (servers or clients) via a network. The communication device 640 may comprise a modem, a network interface card, or other well-known interface device, such as those used for coupling to Ethernet, token ring, or other types of networks.

[0054] Figure 7 illustrates a high level flow diagram 700 of a method for using integration objects with enterprise business applications. The process begins at block 701 and flow continues to processing block 702. A user chooses the type of internal and external interfaces that are associated with the applications to be integrated.

[0055] Flow continues to processing block 703, where wizard 220 is used to create new integration objects for the external application's business objects. At

WO 03/029955

PCT/US02/30884

processing block 704, the new integration objects are modified if necessary. For example, field names or relationships may be altered.

[0056] At processing block 705, wizard 220 is run to create integration objects for the local application(s). These local integration objects allow local data objects to be exported to an external system. Flow continues to processing block 706, where the user may modify the new local data objects, as described above.

[0057] Flow continues to processing block 707, where a data map is written. The data map indicate which external and internal interfaces correspond to the new integration objects, both internal and external. At processing block 708, both internal and external data objects may be integrated without user intervention. Flow completes at block 709.

[0058] A method and system for using integration objects with enterprise business applications is disclosed. Although the present invention has been described with respect to specific examples and subsystems, it will be apparent to those of ordinary skill in the art that the invention is not limited to these specific examples or subsystems but extends to other embodiments as well. The present invention includes all of these other embodiments as specified in the claims that follow.

WO 03/029955

PCT/US02/30884

CLAIMS

We claim:

- 1). A method, comprising:
importing external objects having a first format for a first enterprise application into a second enterprise application;
using integration objects to transform the first format external objects into second format external objects formatted for the second enterprise application;
and
using the second format external objects in the second business application.
- 2). The method of claim 1, wherein using integration objects, comprises:
receiving the first format external objects in an external application adapter, wherein the external application adapter communicates with the second enterprise business application; and
creating a local representation of the first format external objects by the external application adapter.
- 3). The method of claim 2, wherein using integration objects further comprises:
receiving the local representation in a data transformation adapter from the external application adapter;
renaming one or more fields contained in the local representation to correspond to the second enterprise application; and
creating a common representation from the local representation.
- 4). The method of claim 3, wherein using integration objects comprises:
receiving the common representation in an internal application adapter from the data transformation adapter, wherein the internal application adapter communicates with an internal object layer; and
creating the second format external objects from the common representation by the internal application adapter.

WO 03/029955

PCT/US02/30884

- 5). A method, comprising:
exporting internal objects having a first format for a first enterprise application to a second enterprise application;
using integration objects to transform the first format external objects into second format external objects formatted for the second enterprise application;
and
using the second format external objects in the second business application.
- 6). The method of claim 5, wherein using integration objects, comprises:
receiving the first format internal objects in an internal application adapter, wherein the internal application adapter communicates with an internal object layer; and
creating a common representation of the first format internal objects by the internal application adapter.
- 7). The method of claim 6, wherein using integration objects further comprises:
receiving the common representation in a data transformation adapter from the internal application adapter;
renaming one or more fields contained in the common representation to correspond to the second enterprise application; and
creating a local representation from the common representation.
- 8). The method of claim 7, wherein using integration objects further comprises:
receiving the local representation in an external application adapter from the data transformation adapter, wherein the external application adapter communicates with an external application adapter associated with the second enterprise application; and
creating the second format external objects from the common representation by the external application adapter.

WO 03/029955

PCT/US02/30884

- 9). The method of claim 8, further comprising:
providing the integration objects to the second enterprise application.
- 10). A method, comprising:
receiving external metadata in an internal enterprise application, wherein the
external metadata describes an external enterprise application; and
transforming the external metadata into integration objects using an integration
object wizard.
- 11). A system, comprising:
means for importing external objects having a first format for a first enterprise
application into a second enterprise application;
means for using integration objects to transform the first format external objects
into second format external objects formatted for the second enterprise
application; and
means for using the second format external objects in the second business
application.
- 12). The system of claim 11, wherein using integration objects, comprises:
means for receiving the first format external objects in an external application
adapter, wherein the external application adapter communicates with the
second enterprise business application; and
means for creating a local representation of the first format external objects by
the external application adapter.
- 13). The system of claim 12, wherein using integration objects further
comprises:
means for receiving the local representation in a data transformation adapter
from the external application adapter;
means for renaming one or more fields contained in the local representation to
correspond to the second enterprise application; and

WO 03/029955

PCT/US02/30884

means for creating a common representation from the local representation.

14). The system of claim 13, wherein using integration objects comprises:

means for receiving the common representation in an internal application adapter from the data transformation adapter, wherein the internal application adapter communicates with an internal object layer; and

means for creating the second format external objects from the common representation by the internal application adapter.

15). A system, comprising:

means for exporting internal objects having a first format for a first enterprise application to a second enterprise application;

means for using integration objects to transform the first format external objects into second format external objects formatted for the second enterprise application; and

means for using the second format external objects in the second business application.

16). The system of claim 15, wherein using integration objects, comprises:

means for receiving the first format internal objects in an internal application adapter, wherein the internal application adapter communicates with an internal object layer; and

means for creating a common representation of the first format internal objects by the internal application adapter.

17). The system of claim 16, wherein using integration objects further comprises:

means for receiving the common representation in a data transformation adapter from the internal application adapter;

means for renaming one or more fields contained in the common representation to correspond to the second enterprise application; and

means for creating a local representation from the common representation.

WO 03/029955

PCT/US02/30884

- 18). The system of claim 17, wherein using integration objects further comprises:
means for receiving the local representation in an external application adapter from the data transformation adapter, wherein the external application adapter communicates with an external application adapter associated with the second enterprise application; and
means for creating the second format external objects from the common representation by the external application adapter.
- 19). The system of claim 18, further comprising:
means for providing the integration objects to the second enterprise application.
- 20). A system, comprising:
means for receiving external metadata in an internal enterprise application, wherein the external metadata describes an external enterprise application;
and
means for transforming the external metadata into integration objects using an integration object wizard.
- 21). A computer-readable medium having stored thereon a plurality of instructions, said plurality of instructions when executed by a computer, cause said computer to perform:
importing external objects having a first format for a first enterprise application into a second enterprise application;
using integration objects to transform the first format external objects into second format external objects formatted for the second enterprise application;
and
using the second format external objects in the second business application.

WO 03/029955

PCT/US02/30884

- 22). The computer-readable medium of claim 21 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:
receiving the first format external objects in an external application adapter, wherein the external application adapter communicates with the second enterprise business application; and
creating a local representation of the first format external objects by the external application adapter.
- 23). The computer-readable medium of claim 22 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:
receiving the local representation in a data transformation adapter from the external application adapter;
renaming one or more fields contained in the local representation to correspond to the second enterprise application; and
creating a common representation from the local representation.
- 24). The computer-readable medium of claim 23 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:
receiving the common representation in an internal application adapter from the data transformation adapter, wherein the internal application adapter communicates with an internal object layer; and
creating the second format external objects from the common representation by the internal application adapter.
- 25). A computer-readable medium having stored thereon a plurality of instructions, said plurality of instructions when executed by a computer, cause said computer to perform:
exporting internal objects having a first format for a first enterprise application to a second enterprise application;

WO 03/029955

PCT/US02/30884

using integration objects to transform the first format external objects into second format external objects formatted for the second enterprise application;
and
using the second format external objects in the second business application.

- 26). The computer-readable medium of claim 25 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:
receiving the first format internal objects in an internal application adapter, wherein the internal application adapter communicates with an internal object layer; and
creating a common representation of the first format internal objects by the internal application adapter.
- 27). The computer-readable medium of claim 26 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:
receiving the common representation in a data transformation adapter from the internal application adapter;
renaming one or more fields contained in the common representation to correspond to the second enterprise application; and
creating a local representation from the common representation.
- 28). The computer-readable medium of claim 27 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:
receiving the local representation in an external application adapter from the data transformation adapter, wherein the external application adapter communicates with an external application adapter associated with the second enterprise application; and
creating the second format external objects from the common representation by the external application adapter.

WO 03/029955

PCT/US02/30884

- 29). The computer-readable medium of claim 28 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:
providing the integration objects to the second enterprise application.
- 30). A computer-readable medium having stored thereon a plurality of instructions, said plurality of instructions when executed by a computer, cause said computer to perform:
receiving external metadata in an internal enterprise application, wherein the external metadata describes an external enterprise application; and
transforming the external metadata into integration objects using an integration object wizard.
A system, comprising:
a second enterprise application that imports external objects having a first format for the first enterprise application into a second enterprise application; and
one or more integration objects associated with the second enterprise application that transforms the first format external objects into second format external objects formatted for the second enterprise application; wherein the second format external objects are used by the second business application.
- 31). The system of claim 31, wherein the second enterprise application, comprises:
an external application adapter that receives the first format external objects, wherein the external application adapter communicates with the second enterprise business application, and creates a local representation of the first format external objects.

WO 03/029955

PCT/US02/30884

32). The system of claim 32, wherein the second enterprise application further comprises:

a data transformation adapter that receives the local representation from the external application adapter, wherein the data transformation adapter renames one or more fields contained in the local representation to correspond to the second enterprise application, and creates a common representation from the local representation.

33). The system of claim 32, wherein the second enterprise application further comprises:

a data transformation adapter that receives the local representation from the external application adapter, wherein the data transformation adapter renames one or more fields contained in the local representation to correspond to the second enterprise application, and creates a common representation from the local representation.

34). The system of claim 33, wherein the second enterprise application further comprises:

an internal application adapter that receives the common representation from the data transformation adapter, wherein the internal application adapter communicates with an internal object layer, and creates the second format external objects from the common representation by the internal application adapter.

35). A system, comprising:

a second enterprise application that exports internal objects having a first format for a first enterprise application; and
one or more integration objects associated with the second enterprise application that transform the internal objects into one or more first format external objects formatted for the first enterprise application.

WO 03/029955

PCT/US02/30884

- 36). The system of claim 35, wherein the second enterprise application, comprises:
an internal application adapter that receives the first format internal objects, wherein the internal application adapter communicates with an internal object layer, and creates a common representation of the first format internal objects by the internal application adapter.
- 37). The system of claim 36, wherein the second enterprise application further comprises:
a data transformation adapter that receives the common representation from the internal application adapter;
wherein the data transformation adapter renames one or more fields contained in the common representation to correspond to the second enterprise application, and creates a local representation from the common representation.
- 38). The system of claim 37, wherein the second enterprise application further comprises:
an external application adapter that receives the local representation in from the data transformation adapter, wherein the external application adapter communicates with an external integration interface associated with the second enterprise application, and creates the second format external objects from the common representation by the external application adapter.
- 39). The system of claim 38, wherein the integration objects are provided to the first enterprise application.
- 40). A system, comprising:
an internal enterprise application that receives external metadata, wherein the external metadata describes an external enterprise application; and

WO 03/029955

PCT/US02/30884

an integration object wizard that transforms the external metadata into integration objects.

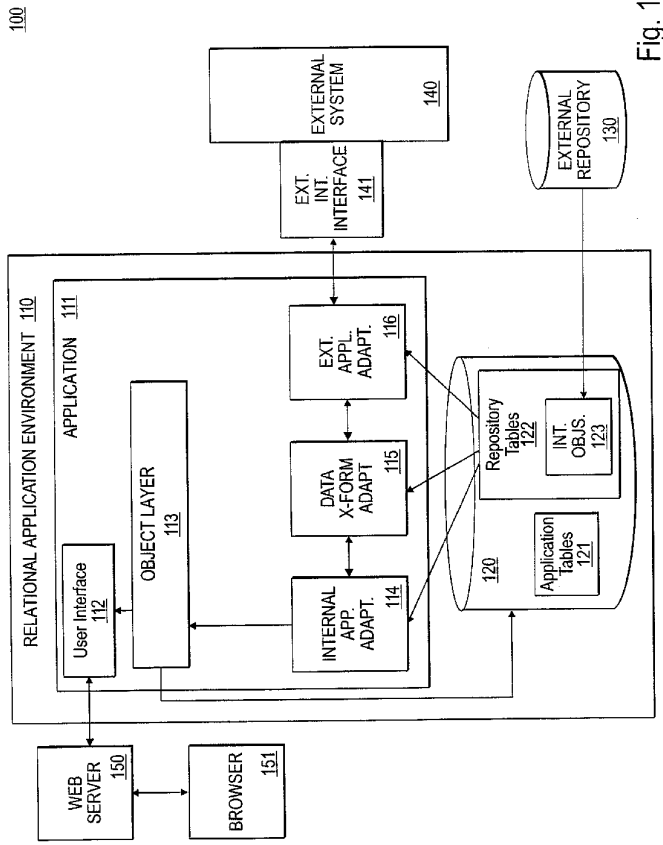


Fig. 1

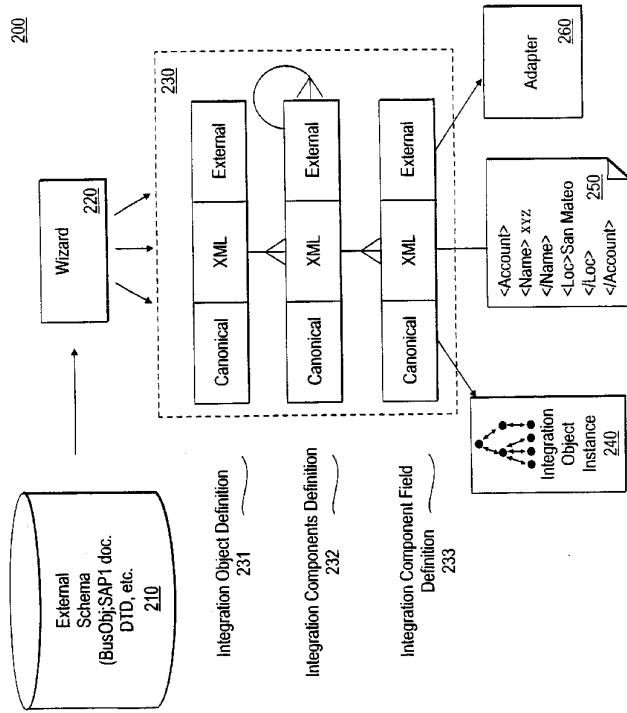


Fig. 2

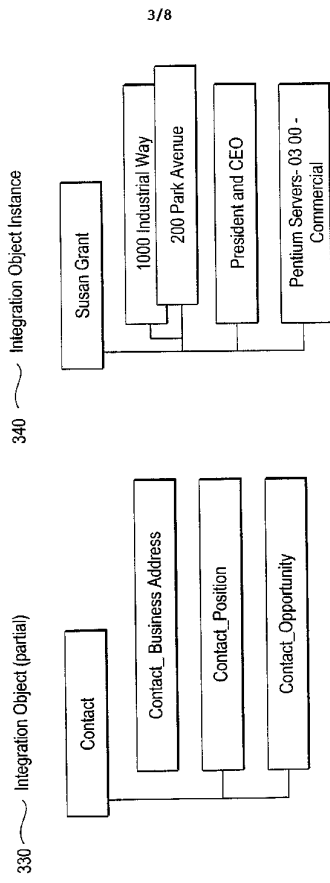
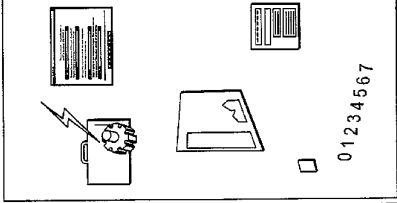


Fig. 3

400

Integration Object Builder



0 1 2 3 4 5 6 7

410 Choose the project this integration Object will be part of. Select a project name from the list.
myProject

420 Choose the source system of the new integration Object. Select a business service from the list.
EAIWizard

430 Choose the source object for the new integration Object. Select an object from the list.
Account

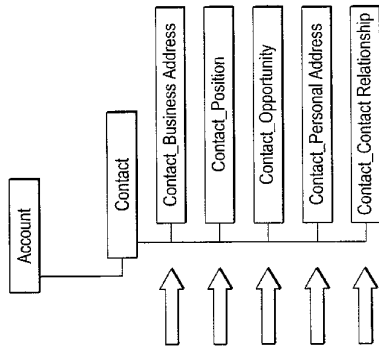
440 Enter a name for the integration Object. This name must be unique among all integration Objects (in all projects)
myAccount

Back Next Cancel Help

Fig. 4

500

Existing Integration Object ~ 520
in Database



Business Object/New In-Memory ~ 510
Integration Object

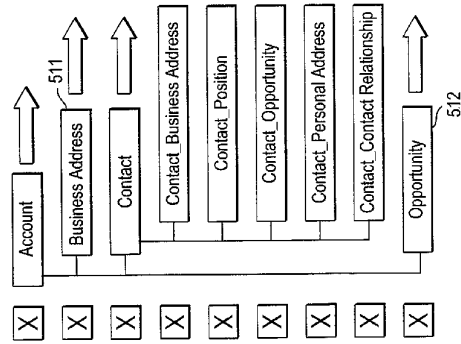


Fig. 5A

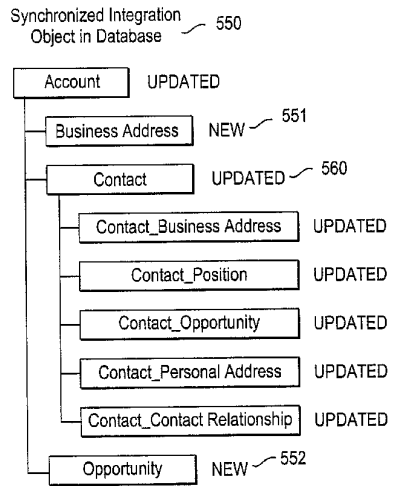


Fig. 5B

WO 03/029955

PCT/US02/30884

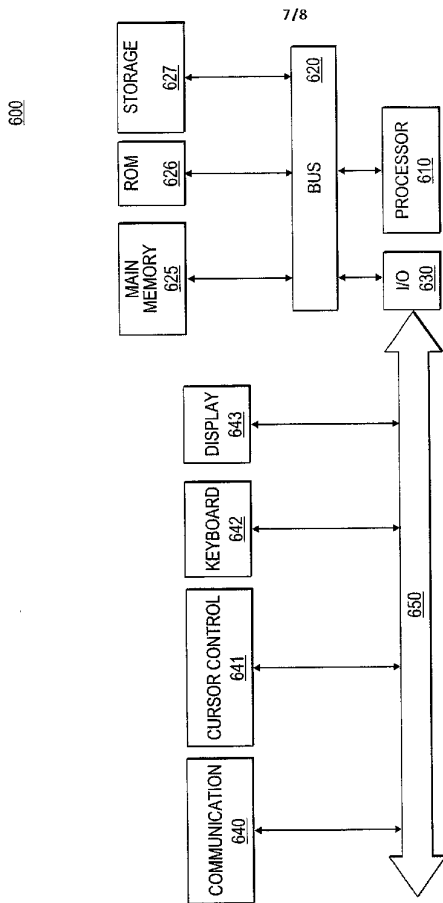


Fig. 6

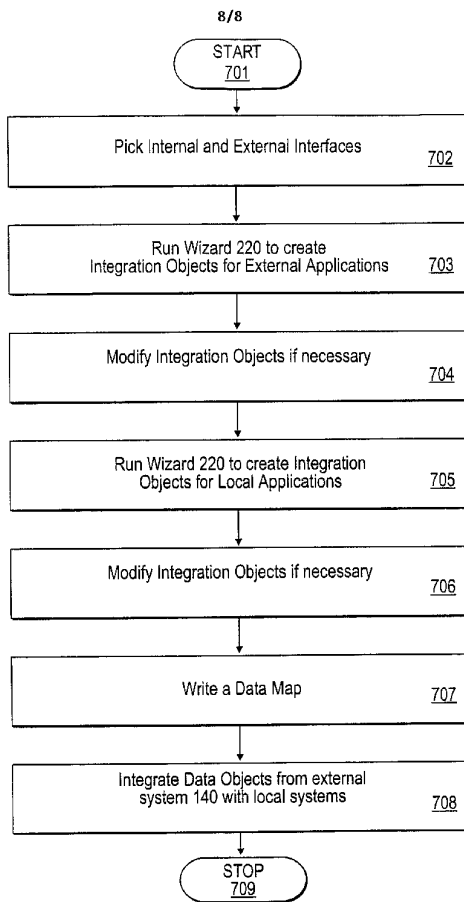


Fig. 7

【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US02/30884
A. CLASSIFICATION OF SUBJECT MATTER		
IPC(7) : G 06 F 7/00 US CL : 707/103 According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) U.S. : 707/1, 103, 104, 200		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,958,004 A (HELLAND et al) 28 September 1999 (28.09.1999), columns 6-9.	1-40
Y	US 6,125,383 A (GLYNIAS et al) 26 September 2000 (26.09.2000), columns 8-11.	1-40
Y	US 5,542,078 A (MARTEL et al) 30 July 1996 (30.07.1996), columns 5-10.	1-40
Y	US 5,809,507 A (CAVANAUGH) 15 September 1998 (15.09.1998), columns 8-10.	1-40
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
Special categories of cited documents:		
A document defining the general state of the art which is not considered to be of particular relevance	*I*	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
E earlier application or patent published on or after the international filing date	*X*	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*Y*	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
O document referring to an oral disclosure, use, exhibition or other means	*Z*	document member of the same patent family
P document published prior to the international filing date but later than the priority date claimed		
Date of the actual completion of the international search 14 November 2002 (14.11.2002)	Date of mailing of the international search report 05 DEC 2002	
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230	Authorized officer Alvin Oberley Peggy Harwood Telephone No. 703-305-3665	

Form PCT/ISA/210 (second sheet) (July 1998)

フロントページの続き

(81) 指定国 AP(GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, N O, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW

(72) 発明者 フィッシャー ジェフリー マイケル

アメリカ合衆国 カリフォルニア州 9 4 1 3 3 サン フランシスコ メゾン ストリート 1
9 2 2

(72) 発明者 コイル マーク

アメリカ合衆国 カリフォルニア州 9 4 1 0 7 サン フランシスコ # 1 - 2 1 0 タウンゼ
ント ストリート 2

Fターム(参考) 5B082 GA02