

(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2005-535947
(P2005-535947A)

(43) 公表日 平成17年11月24日(2005.11.24)

(51) Int.C1.⁷**G06F 12/00****G06F 9/52**

F 1

G06F 12/00

G06F 12/00

G06F 12/00

G06F 9/46

テーマコード(参考)

5B082

511C

514K

475C

審査請求 有 予備審査請求 有 (全 18 頁)

(21) 出願番号 特願2004-509796 (P2004-509796)
 (86) (22) 出願日 平成15年4月4日 (2003.4.4)
 (85) 翻訳文提出日 平成16年11月26日 (2004.11.26)
 (86) 國際出願番号 PCT/EP2003/003529
 (87) 國際公開番号 WO2003/102778
 (87) 國際公開日 平成15年12月11日 (2003.12.11)
 (31) 優先権主張番号 02012080.4
 (32) 優先日 平成14年5月31日 (2002.5.31)
 (33) 優先権主張国 歐州特許庁 (EP)

(71) 出願人 390009531
 インターナショナル・ビジネス・マシーンズ・コーポレーション
 INTERNATIONAL BUSINESS MACHINES CORPORATION
 アメリカ合衆国10504 ニューヨーク
 州 アーモンク ニュー オーチャード
 ロード
 (74) 代理人 100086243
 弁理士 坂口 博
 (74) 代理人 100091568
 弁理士 市位 嘉宏
 (74) 代理人 100108501
 弁理士 上野 剛史

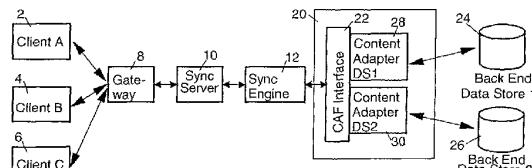
最終頁に続く

(54) 【発明の名称】異なるタイプのバックエンド・データ・ストアにアクセスするためのシステムおよび方法

(57) 【要約】

【課題】同期エンジンが、移動デバイスとバックエンド・データ・ストアとの間で、そのバックエンド・ストアのアーキテクチャおよびデータ・フォーマットとは無関係にデータを同期化できるようにする、フレームワークを提供する。

【解決手段】このフレームワークは、バックエンド・データ・ストアからの同期データにアクセスするコンテンツ・アダプタを導入する。これらのアダプタはデータをバックエンド・データ・ストアに依存しない表現に変換し、一般的な方法で異なるバックエンドにアクセスする必要のあるすべてのアプリケーションまたはモジュールは、これを使用することができる。競合の検出および解決を目的とする汎用同期エンジンは、この種のモジュールの一例である。コンテンツ・アダプタを使用することのできる他のアプリケーションは、通知フレームワークまたはポータル、およびすべての他のデータ集約アプリケーションである。



【特許請求の範囲】**【請求項 1】**

クライアント(2、4、6)への接続およびバックエンド・データ・ストア(24、26)への接続を有する中央同期サーバ(10)を使用することによって、異なる前記クライアント(2、4、6)間でデータを交換するためのシステムであつて、前記クライアントは、同期化されるデータを作成するためのプログラムと、前記中央同期サーバ(10)との同期化を実行するための同期エンジン(12)とを有し、前記システムは、更なるコンポーネントである

前記同期エンジン(12)との単一バックエンド中立インターフェース(CAF-インターフェース；22)と、

前記単一のバックエンド中立インターフェース(22)および前記割り当てられたバックエンド・データ・ストア(24、26)とのインターフェースを有するバックエンド依存部分を備える、少なくとも1つのコンポーネント(コンテンツ・アダプタ；28、30)と、

によって特徴付けられる、システム。

【請求項 2】

前記コンポーネント(28、30)は抽象化バックエンド非依存部分をさらに有し、前記抽象化バックエンド非依存部分は、すべてのバックエンド依存部分が使用するための共通の機能を提供する、請求項1に記載のシステム。

【請求項 3】

各タイプのバックエンド・データ・ストア(24、26)に独自のコンポーネント(28、30)が割り当てられる、請求項2に記載のシステム。

【請求項 4】

前記データの交換がデータの同期化である、請求項1に記載のサーバ。

【請求項 5】

前記バックエンド・データ・ストア(24)および前記クライアントの更新を永続的にバッファするためのキャッシュ(50)をさらに有し、前記コンポーネント(28、30)が、前記キャッシュへ更新をバッファするのを制御し実行するための、ならびにバッファされた更新を前記それぞれのクライアントおよび前記バックエンド・データ・ストア(24)に複製するための、キャッシュ機構を有する、請求項2に記載のシステム。

【請求項 6】

前記キャッシュ機構がバックエンド・モニタ(60)を有する、請求項5に記載のシステム。

【請求項 7】

前記キャッシュ機構がキャッシュ・モニタ(70)をさらに有する、請求項5に記載のシステム。

【請求項 8】

前記キャッシュ機構がバックエンド・マネージャ(80)をさらに有する、請求項6に記載のシステム。

【請求項 9】

前記キャッシュ機構は、各タイプのバックエンド・データ・ストア(24)に対し、そのバックエンド依存部分(60'、70'、80')および抽象化バックエンド非依存部分(60''、70''、80'')を備えた独自のバックエンド・モニタ、キャッシュ・モニタ、およびバックエンド・マネージャを提供する、請求項6に記載のシステム。

【請求項 10】

前記キャッシュ機構が固定記憶(40)をさらに有する、請求項5に記載のシステム。

【請求項 11】

前記キャッシュ・モニタ(70)が、前記キャッシュから前記バックエンド・データ・ストア(24)への更新を、バッチ・モードまたは連続するトリクル・モードで複製する、請求項7に記載のシステム。

【請求項 12】

前記バックエンド・モニタ(60)は、前記キャッシュ(50)と前記バックエンド・データ・ストア(24)との間の更新を、バッチ・モードまたは連続するトリクル・モードで複製する、請求項6に記載のシステム。

【請求項 13】

前記キャッシュ(50)および前記バックエンド・データ・ストア(24)がデータベースである、請求項5に記載のシステム。

【請求項 14】

前記クライアントがモバイル・クライアントである、請求項1に記載のシステム。

【請求項 15】

同期プロトコルがSyncMLである、請求項4に記載のシステム。

【請求項 16】

請求項1から15に記載のシステムを使用することによってデータを同期化する方法であつて、

前記クライアントから同期セッション要求を受信するステップと、

前記同期サーバに対して前記クライアントを認証するステップと、

前記クライアントから更新を受信するステップと、

バックエンド・モニタを使用して、前記CAFインターフェースを介してバックエンド・データ・ストアに対して前記クライアントを認証するステップと、

データ・オブジェクトを作成し、前記同期サーバによって前記クライアントから受信した前記更新にそれらを記入するステップと、

前記CAFインターフェースを呼び出して、前記データ・オブジェクトを引き渡すステップと、

前記バックエンド・データ・ストアに割り当てられた前記コンポーネントの適切なバックエンド特有部分を選択するステップと、

前記データ・オブジェクトCAFをバックエンド特有フォーマットに変換するステップと、

バックエンド特有部分を呼び出してデータ・オブジェクトをそこに引き渡すことにより、前記更新を実行するステップと、

を有する方法。

【請求項 17】

前記バックエンド特有部分が、前記バックエンド・データ・ストアに割り当てられた前記抽象化バックエンド非依存部分から継承される、請求項16に記載の方法。

【請求項 18】

前記データ・オブジェクトが、前記クライアント要求をバックエンド特有部分に渡すために使用される、請求項16に記載の方法。

【請求項 19】

前記データ・オブジェクトがメタ・データを含む、請求項18に記載の方法。

【請求項 20】

前記クライアントと前記中央同期サーバとの間で使用される同期プロトコルがSyncMLであり、前記中央同期サーバによって受信される前記更新はXML文書として提示される、請求項16に記載の方法。

【請求項 21】

デジタル・コンピュータの内部メモリに格納されたコンピュータ・プログラムであつて、前記コンピュータ上で実行された場合に請求項16から20に記載の方法を実行するためのソフトウェア・コードの一部を含む、コンピュータ・プログラム。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、異なるクライアント間でデータを交換または同期化するための方法およびシ

10

20

30

40

50

システムに関し、より詳細には、異なるタイプのバックエンド・データ・ストアとリンクする中央同期サーバを使用したクライアント間でのデータの同期化に関する。

【背景技術】

【0002】

同期とは、たとえばノートブック・カレンダと携帯情報端末（PDA）カレンダとの間などの、異なるクライアント間でデータを整合させておくこと、および、これらのクライアント側で異なる識別子を備えた異なるデータ・フォーマットでデータを格納することと定義することが可能である。同期データは、完全なコンピュータ・プログラムまたはこれらプログラムの一部からなるものとすることもできる。

【0003】

中央同期サーバは、それを介して異なるクライアント間でデータを交換できるものとして使用されることが多い（図1を参照）。中央同期サーバは、通常は負荷の高いキャリア（たとえばYahoo）によって使用されるものである全データをローカルに格納する（すなわち、独自所有のデータ形式を使用する場合）か、DB2、Domino、またはExchangeのようなバックエンド・データ・ストアに直接アクセスするかの、いずれかが可能である（図2を参照）。

【0004】

バックエンド・データ・ストアは、情報にアクセスし格納するための専用のプロトコルおよびメカニズムを有する。リレーショナル・データベースはODBCを共通インターフェースとして使用するが、たとえばパーソナル情報管理（PIM）データを格納するデータベースは、通常、所有権を主張できる方法でアクセスされ、コンテンツの非常に特有なデータ構造に依拠するものである。その結果、ウェブ・サーバ・ベースのアプリケーションは、特有のデータベースおよびアクセスされる特定タイプのコンテンツに依存することになる。新しいデータベースおよび新しいタイプのコンテンツを既存のソフトウェアに採用するには、かなりの努力が必要である。

【0005】

たとえば、多くの異なるデータベース製造業者および複数のマルチメディア・コンテンツに関係する大規模な同期ソリューションの開発は、これらの状況の下ではかなり困難である。複数のクライアント・デバイスを同期化するための論理を含む同期エンジンは、サポートされているタイプの各コンテンツおよび接続された各データベースに適合させる必要がある。実際、今日の同期エンジンは、データを含むバックエンド・ストアに大きく依存している。精巧な同期エンジンを実施するためにかなりの額を投資しても、1つまたは少数のバックエンド・システムが恩恵を受けるだけである（図3を参照）。

【0006】

異なるバックエンド・システムに集められて格納されたデータを送信することによって通知システムがユーザに通知したい場合にも、同じ問題が生じる。異なるシステムからのデータを総合するポータルも、情報にアクセスするためにシステムから独立した方法を必要とする。

【0007】

米国特許第5974238号には、それぞれが少なくとも1つの共通データ・セットを含む複数のデータ・セットを有し、それぞれのコンピュータが共通データ・セットのコピーを有する、ハンドヘルド・コンピュータに格納されたデータとホスト・コンピュータとの間で動的な同期を実行するための装置が記載されている。ハンドヘルド・コンピュータは、プロセッサと、通信ポートと、データ同期エンジンとを有する。データ同期エンジンは、擬似キッシュと、擬似キッシュに接続された1つまたは複数のタグとを有する。データがメイン・メモリに書き込まれた場合には必ず、および／または、関連する擬似キッシュ・タグが無効化された場合には、データが同期化される。プロトコル・セットに厳密に準拠することによって、誰がデータを所有しているか、誰がデータのコピーを有するか、および誰がデータを修正したかを、システムが常に把握しているため、データの首尾一貫性が達成される。データ同期エンジンは、コピー内のいかなる相違点をも解決し、

ホスト・コンピュータおよびハンドヘルド・コンピュータ内に共通データ・セットの同一コピーを格納できるようにする。

【0008】

この従来技術の特許は、バックエンド・データ・ストアを使用する同期アーキテクチャは対象としていない。したがって、バックエンド・データ・ストアに関する上記の問題は、当該特許によって対処も解決もされていない。

【特許文献1】米国特許第5974238号

【発明の開示】

【発明が解決しようとする課題】

【0009】

したがって、本発明の目的は、従来技術のソリューションの欠点を回避しながら、異なるタイプのバックエンド・データ・ストアにリンクされた中央同期サーバを使用するアーキテクチャでデータを交換または同期化するための、新しい方法およびシステムを提供することである。

【課題を解決するための手段】

【0010】

当該目的は、独立した請求項の機能によって解決される。他の好ましい実施形態は、従属する請求項に規定されている。

【0011】

本発明は、同期エンジンが移動デバイスとバックエンド・データ・ストアとの間で、そのバックエンド・ストアのアーキテクチャおよびデータ・フォーマットとは無関係にデータを同期化できるようにするフレームワーク（枠組み）を開示するものである。このフレームワークは、バックエンド・データ・ストアから同期データにアクセスするコンテンツ・アダプタを導入する。これらのアダプタはデータをバックエンド・データ・ストアから独立した表現に変換し、一般的な方法で異なるバックエンドにアクセスする必要のあるすべてのアプリケーションまたはモジュールは、これを使用することができます。競合の検出および解決を目的とする汎用同期エンジンは、この種のモジュールの一例である。コンテンツ・アダプタを使用することのできる他のアプリケーションは、通知のフレームワークまたはポータル、およびデータを集約するすべての他のアプリケーション、である。いずれのバックエンド・データ・ストア特有の問題もコンテンツ・アダプタのバックエンドに依存した部分によって処理され、その部分は、第三者によって容易に提供され、フレームワークにプラグインすることが可能である。

【0012】

本発明については、添付の図面を使用してより詳細に説明される。

【発明を実施するための最良の形態】

【0013】

中央同期サーバを使用する異なるクライアント間での同期化は、通常は以下のステップからなる同期プロトコルに基づくものである。

【0014】

事前同期化：実際の同期化を準備するために、発生前に実行しなければならない何らかのアクション。これらのアクションは、クライアント機能の認証、許可、および決定というグループに分類される。認証は、サーバが請求の対象となるものであること、およびクライアントが請求の対象となるものであることを保証する。許可は、クライアントが要求されたアクション（たとえば削除し、更新し、または新規エントリ作成のみをする）を実行できるかどうかをチェックする。最終的にサーバは、クライアントへのデータの流れを最適化するようにデバイス機能（たとえば最大バッファ・サイズ）を決定する。

【0015】

同期化：これは、同期データが交換される部分である。2つの同期パートナー間では、データ・エントリのすべてのローカルIDが双方のパートナーの知っているグローバルIDにマッピングされる。したがって、あらゆるパートナーは、ローカルからグローバルへ

10

20

30

40

50

IDをマッピングするためのマッピング・テーブルを有する。次に、更新された、新規の、または削除された、エントリのみが交換される。双方のパートナーが同じデータ・エントリを更新すると、競合が発生することになる。この更新の競合は、更新をマージ（統合）するよう試みる、エントリを重複させる、一方のエントリを他のエントリに勝たせる、または単に何も実行せずに競合を報告してユーザが解決できるようにする、といった異なる試行で解決することができる。

【0016】

事後同期化：事後同期化では、マッピング・テーブルの更新、未解決の競合の報告、などのすべてのクリーンアップ・タスクが実行される。

【0017】

広く使用されている同期プロトコルが、SyncMLである。SyncMLは、トランスポート（移送）・プロトコルとは独立した同期化のためにXMLベースのトランスポート・プロトコルを提供するものである。各同期メッセージはXML文書である。典型的なSyncMLシステムが図4に示されており、アプリケーションB（たとえば携帯情報端末）がそのクライアント同期エンジンを介してSyncMLフレームワークに同期データを送信している。SyncMLフレームワークは、API呼出し（たとえば、更新、作成）およびデータを、有効なSyncML文書に変換し、これをサーバに送信する。サーバ側では、SyncMLフレームワークが文書を受信し、これを解析した後、コマンドおよびデータをサーバ同期エンジンに送信し、その後サーバ同期エンジンがアプリケーションA（たとえばノートブックのLotus Notesカレンダー）と対話する。

10

20

【0018】

図5は、モバイル（移動式）・クライアントと、これらと異なるバックエンド・データ・ストアのタイプとの間の通信アーキテクチャで使用される、本発明のコンテンツ・アダプタ・フレームワークの基本アーキテクチャを示す図である。

【0019】

異なるモバイル・クライアント2、4、6は、無線または有線のゲートウェイ8を介しウェブ・サーバ10を通じて同期エンジン12にアクセスし、さらに当該同期エンジン12はCAF20を介して異なるバックエンド・データ・ストアのタイプ24、26と対話する。CAF20は、新しいバックエンド・データ・ストアを容易に追加するために、単一のバックエンド中立インターフェース（CAFインターフェース22）を介して異なるバックエンド・データ・ストア24、26のデータにアクセスする基盤を提供する。CAF20は、少なくとも単一のCAFインターフェース22と、1つまたは複数のコンテンツ・アダプタ28、30とからなる。

30

【0020】

CAFインターフェース22は、同期サーバ10がバックエンド・データにアクセスするための単一のインターフェースを表すものであり、その結果コンテンツ検索が同期サーバから分離される。同期エンジン12は、CAFインターフェース22を介して、特定のバックエンド・データ・ストア24、26からのコンテンツに独立してアクセスすることができる。好ましいことに、CAFインターフェース22と同期サーバ10との間でデータを交換するために、データ・フォーマットとしてデータ・オブジェクトが使用される。

40

【0021】

基本的にコンテンツ・アダプタは、すべてのデータ・ストア特有の依存関係を与えるものである。

【0022】

本発明の好ましい実施形態では、各コンテンツ・アダプタ28、30が抽象化バックエンド非依存部分およびバックエンド依存部分を含む。バックエンド依存部分は、データ・ストア特有のすべての依存関係を含む。これによって同期化されたバックエンド・データへのアクセスが実施され、そのデータのデータ・ストアに依存しない表現が作成されて、CAFインターフェース22を使用して同期エンジン12またはアプリケーション層に提供される。CAF特有のプロセス流れは、コンテンツ・アダプタ28、30のバックエン

50

ド非依存部分によって管理される。バックエンド非依存部分は、たとえばキューイング・メカニズム、通信処理など、すべてのバックエンド・データ・ストアに共通の機能を提供する。

【0023】

アプリケーション（たとえば同期エンジン）が使用することのできるセマンティック（意味的）情報を含めるために、共通データ・オブジェクトのクラス階層が定義される。データ・オブジェクトの特殊なサブクラスは、たとえばアドレス、カレンダ、マルチメディア情報、リレーションナル・データベースといった、サポートされたデータ・タイプの典型的なプロパティを記述する。特定の種類の情報に関するデータ・オブジェクトを記述するプロパティは、アドレス・ブック情報に関する vCard (ビジネス・カード情報を交換するための標準フォーマット)、またはカレンダ・エントリに関する vCal (カレンダ情報を交換するための標準フォーマット)などの、一般標準から取得することができる。XML を適用することによって、特定データベースから独立した顧客特有のデータベースを表すことさえも可能である。10

【0024】

最終的にフレームワーク 20 は、高ボリューム・システムまたは低速バックエンド・システムのために、同期エンジン 12 とバックエンド・データ・ストア 24、26 との間のキャッシュ機構を容易に統合するためのインフラストラクチャを提供する（図 6 を参照）。20

【0025】

コンテンツ・アダプタは、高速読み取り / 書込みアクセスを提供し、異なるバックエンド・システム（たとえば Domino、DB2、Exchange）に適応可能であり、複数の SyncML メッセージをサポートし、常に一貫したデータを有し、異なるコンテンツ・フォーマットに適応可能である。20

【0026】

C AF の基本アーキテクチャによって実行される方法は、以下のように簡単に要約することができる。クライアントは、同期サーバを使用して同期セッションを要求する。サーバはクライアントを認証し、同期セッションを受け入れる。クライアントはサーバに更新を送る。同期サーバはデータ・オブジェクトを作成し、クライアントから受信した更新に記入する。その後同期サーバは C AF インターフェースを呼び出し、データ・オブジェクトを渡す。C AF は、コンテンツ・アダプタの適切なバックエンド特有部分を選択する。30

【0027】

C AF は、バックエンド特有部分を呼び出してデータ・オブジェクトを渡す。コンテンツ・アダプタのバックエンド特有部分は、データ・オブジェクトをバックエンド特有フォーマットに変換し、バックエンド特有 API (アプリケーション・プログラミング・インターフェース) を呼び出す。40

【0028】

従来技術と比較される本発明の主な利点の 1 つは、アクセス特有のバックエンド・データベースを呼出し側アプリケーションから独立させることである。C AF は異なるデータベース・プロバイダからのコンポーネントを統合し、高水準インターフェースを介したそれらのデータベース機能へのアクセスを提供する。40

【0029】

コンテンツ・アダプタは、共通インターフェースを使用することによって、複数のバックエンド・システムを備えたアプリケーションの相互運用性を保証する。たとえば同期エンジンは、特定データベースの独自所有コマンドに依存しない。さらにコンポーネントは、データベースとアプリケーションとの間で交換されるコンテンツの複雑さを隠すものである。これによって、プログラミングの努力およびソリューションの複雑さが大幅に軽減される。C AF は、バックエンド・システム・プロバイダが、そのバックエンド・データ・ストアにアクセスしている様々なアプリケーションに対して 1 つのインターフェースのみを開発できるようにするものもある。これによって、双方の当事者（アプリケーショ50

ン・プロバイダおよびバックエンド供給業者)が莫大な時間および費用を節約することになる。

【0030】

C AFによって以下のことことが可能になる。

- ・同期エンジンは同じプロトコルおよびA P Iを使用して異なるバックエンド・システムと対話することができる。

- ・バックエンド・システム・プロバイダは、それらのバックエンド・システムに向けて調整されたそれら専用のコンテンツ・アダプタを作成することができる。

- ・同期エンジン、C AF、およびバックエンド・システムの間に、効率のためにバックエンド接続を再使用するための認証を処理するセッションを提供する。
10

- ・キャッシュ・システムを利用して、同期エンジンとC AFの通信の短いレイテンシー(待ち時間)を達成することができる。

- ・コンポーネントのロードの均衡および障害時保管の配分を可能にする。

【0031】

図6は、C AFの好ましい実施を示す図である。C AF100は、C AFインターフェース22、コンテンツ・マネージャ30、およびキャッシュ機構を有する。

【0032】

C AFインターフェース22は、同期サーバにバックエンド・データ・ストア24にアクセスするための単一のインターフェースを提供し、その結果、同期サーバ10からコンテンツ検索が分離される。
20

【0033】

コンテンツ・マネージャ30は、認証およびバックエンド管理要求(たとえば同期アンカーの取得)をバックエンド・マネージャ80に転送し、固定記憶40を使用して新しいデータをキャッシュ50に書き込み、固定記憶40を介してキャッシュ50から更新を取得する。検索および実行コマンドがバックエンド上で実行される。

【0034】

キャッシュ機構は、永続キャッシュ50と、更新をキャッシュ50にバッファリングし、バッファリングされた更新をそれぞれのクライアントと同期させるためのメカニズムとを、提供する。永続キャッシュ50は、OracleまたはIBM DB2のようなリレーショナル・データベースとすることが可能であり、たとえばJDBC呼び出しを介してアクセスすることができる。好ましくは、キャッシュ機構は、キャッシュ・モニタ70、バックエンド・モニタ60、バックエンド・マネージャ80、および固定記憶40からなる。バックエンド・マネージャ80は、そのバックエンド特有部分80'(コンテンツ・アダプタ)を備えた抽象化バックエンド・マネージャ80"を含み、キャッシュ・モニタ70は、そのバックエンド特有のキャッシュ・モニタ部分70'(コンテンツ・アダプタ)を備えた抽象化キャッシュ・モニタ70"を含み、バックエンド・モニタ60は、そのバックエンド特有のバックエンド・モニタ部分60'(コンテンツ・アダプタ)を備えた抽象化バックエンド・モニタ60"を含む。
30

【0035】

キャッシュ・モニタ70は、主にすべての新しいデータをキャッシュからバックエンド・データ・ストアに複製するために使用される。バックエンド要件に応じて、バッチ(一括)またはトリクル(細流)などの異なる複製方針を採用することができる。主な目的が、正規のバックエンド・クライアントの代わりに同期クライアントをより適切にサポートすることである場合は、バッチ・モードが好ましい。キャッシュ・モニタ70のバックエンド依存部分70'は、それぞれのバックエンド・データに特有であり、バックエンド・データ・ストアの機能(たとえばDB2、Domino)を活用するものでなければならない。さらに、同期オブジェクトをコンテンツ・ストレージ特有のフォーマット(たとえばLotus DominoまたはMS Exchange)にも変換する。
40

【0036】

バックエンド・モニタ60は、バックエンド・データ・ストア24内で発生する更新(50

たとえば、正規の L o t u s N o t e s クライアントがデータベースを更新する)を、同期サーバの外部からキャッシュ 5 0 にトリクルする。これによって同期クライアントは、各同期セッションに対する完全な複製のオーバヘッドを必要とせずに、常に最新のバックエンド・データと同期をとることができる。バックエンド・モニタ 6 0 のバックエンド特有部分 6 0 ' は、各バックエンド・データ・ストアに特有であり、コンテンツ・ストレージ特有のフォーマット(たとえば L o t u s D o m i n o または M S E x c h a n g e)を C A F 同期オブジェクトに変換する。

【 0 0 3 7 】

バックエンド・モニタは、全体のシステム性能を最適化するために、アグレッシブ(積極的な)更新またはレイジー(のろい)更新を含む様々な更新ポリシーを有することができる。

【 0 0 3 8 】

バックエンド・マネージャ 8 0 は、バックエンド・データ・ストアの管理機能へのアクセスを提供する。好ましくは C A F をサポートするために、以下の機能、すなわちユーザ認証の妥当性検査、認証済みユーザのアクセス許可の検索、現在のバックエンド特有のタイムスタンプ(現在の「同期アンカー」)の検索、および C A F が変更を監視したい U R I の追加 / 削除が、提供される。バックエンド・マネージャ 8 0 のバックエンド特有部分 8 0 ' は、各バックエンド・データ・ストアに特有であり、コンテンツ・ストレージ特有のフォーマット(たとえば L o t u s D o m i n o または M S E x c h a n g e)を C A F 同期オブジェクトに変換する。

【 0 0 3 9 】

同期されることになるデータは、リモート・コンテンツ・ストア(バックエンド)に直接格納するか、または性能上の理由でサーバ上に(ローカルに)持続的にキャッシュすることができる。固定記憶 4 0 は、キャッシュとして固定記憶媒体を使用してバックエンド・データ・ストアへの読み取り / 書込みアクセスを最適化するが、当該アーキテクチャは、固定記憶がキャッシュ 5 0 およびバックエンド・モニタを介してバックエンド・データ・ストアに直接接続するのを妨げるものではない。

【 0 0 4 0 】

上記 C A F 実施における通信の流れは、以下のように簡潔に要約することができる。

【 0 0 4 1 】

コンテンツ・マネージャ 3 0 は、同期エンジン 1 2 から要求を受信し、データの検出または格納を必要とする場合にはそれらを固定記憶 4 0 に転送し、タイムスタンプが必要であるか認証が要求される場合には、それらをバックエンド・マネージャ 8 0 に転送する。可能であれば、バックエンド・レジストリはすべての使用可能なバックエンド・マネージャ 8 0 およびモニタ 6 0 、 7 0 を含み、コンテンツ・マネージャ 3 0 および固定記憶 4 0 がこれにアクセスする。キャッシュ・モニタ 7 0 は、更新済みのデータを固定記憶 4 0 から取得し、これらをバックエンド・フォーマットに変換して、当該データをキャッシュ・モニタのバックエンド依存部分 7 0 ' を使用してバックエンド・データ・ストア 2 4 に転送する。バックエンド・モニタ 6 0 のバックエンド依存部分 6 0 ' は、バックエンド・データ・ストア 2 4 から更新を受信し、これらをデータ・オブジェクトに変換した後、固定記憶 4 0 に転送する。

【 0 0 4 2 】

キャッシュが使用できない場合、または所与のデータベースにバックエンド 2 4 への直接アクセスが指定されていない場合、コンテンツ・マネージャ 3 0 は g e t U p d a t e s 呼出しをバックエンド・モニタ 6 0 に転送し、更新されることになる項目をキャッシュ・モニタ 7 0 に転送する。どちらのモニタも、提供されたユーザ I D およびパスワードを使用してバックエンドにアクセスする。

【 0 0 4 3 】

C A F を使用した同期エンジンの認証

同期エンジンには、C A F への要求を認証する際に以下の 2 つのオプションがある。

10

20

30

40

50

1. 各コマンドに要求されたバックエンド・ユーザのIDおよびパスワードを提供する
 2. ユーザIDおよびパスワードを提供することによって、1つのバックエンドに関するCAFに認証トークンを要求する。この同期セッションに関するCAFへのそれぞれの後続コマンドは、このトークン(LTPA トークンと同様)を使用して認証することができる。読み出し専用アクセス、読み出し/書き込みアクセス、または無制限の読み出しみきみアクセス、というトークン・タイプがサポート可能であるが、書き込みはその都度バックエンド側で認証しなければならない。

【0044】

バックエンド・データ・ストアへのCAFの認証

要求されたバックエンド・データ・ストアへの認証レベルは、各バックエンド・データ・ストアに関するアクセス制御リストに格納され、同期アダプタによってチェックされる。CAFはこのリストに依存して、同期サーバを介して自身のデータを更新するすべてのユーザに有効なグループ・ユーザおよびグループ・パスワードを使用するか、あるいはユーザごとに行なうか、のいずれかによって、自身をバックエンド・データ・ストアに認証させる。

10

【0045】

図7は、図6の好みの実施におけるCAFのインターフェース(I/F)を示す図である。

【0046】

同期エンジン12は、CAFインターフェース22を使用して、一般的な方法でバックエンド・データ・ストアにアクセスする。汎用同期エンジン12とCAFインターフェース22との間で効率良くデータを交換するために、好みのことにCAFインターフェース22は、データの交換には未加工値2進コード化スキームを使用する。これらの未加工データは、SyncObj内にActionOnDataオブジェクトとしてCAFメタ・データと共に組み込まれる。CAFメタ・データは、以下の通りである。

20

- ・Timestampは、SyncObjの現在の同期アンカーを表すものである。これはクラスjava.sql.Timestampのオブジェクトとして格納される。
- ・ActionTypeは、データ・アクション・タイプのSyncObjを表す。可能な値はクラス定数(CREATE、UPDATE、DELETE)内に定義される。これはショート(短)値として格納される。
- ・GUIDは、SyncObjのバックエンド特有のグローバルIDを表す。これはjava.lang.String値として格納される。
- ・UUIDは、SyncObjのCAF特有のキャッシュIDを表す。これはjava.lang.String値として格納される。
- ・databaseURIは、SyncObjが属するバックエンド・データベースのURIを表す。これはクラスcom.ibm.caf.URIのオブジェクトとして格納される。
- ・userIDは、SyncObjが属する同期化を開始したクライアントのユーザIDを表す。これはjava.lang.String値として格納される。

30

【0047】

CAFのインターフェース特有のプロセス流れ

抽象化モニタ・クラスは、バックエンド特有モニタをCAFアーキテクチャに統合するための基本機能を提供するものである。新しいモニタ・クラスを作成するために継承を介して使用することのできる、抽象化キャッシュ・モニタおよび抽象化バックエンド・モニタと呼ばれる2つの抽象化クラスが提供される(図8を参照)。

40

【0048】

抽象化クラスは、以下の機能を提供する。

- ・様々な更新ポリシー(トリクル/バッチ更新)に対するキューリング・メカニズム
- ・バックエンド・レジストリでの登録/登録解除
- ・システム基本設定コンポーネントでの更新に関する読み取り及び登録

50

・ C A F 固定記憶（キャッシュ）との通信の処理

【 0 0 4 9 】

C A F 特有のプロセス流れは、本特許明細書で定義されたインターフェースを介して、抽象化バックエンド・モニタ 6 0 および抽象化キャッシュ・モニタ 7 0 によって管理される。バックエンド・データ・ストア 2 4 との直接通信は、抽象モニタ・クラスから継承されたバックエンド特有のコンポーネント 7 0 '、6 0 ' で実施される（図 9 を参照）。

【 0 0 5 0 】

各モニタは、様々な更新およびデータ伝播ポリシーを実行可能にする内部キューを実施する。使用シナリオに応じて、更新済みのデータを、グループごとまたは長い時間間隔でバックエンドに送信することが必要な場合がある。

10

【 0 0 5 1 】

抽象化モニタ・コンポーネントは、以下の更新ポリシーを提供する構成可能なキューイング・メカニズムを実施する。

【 0 0 5 2 】

量トリガ：

量トリガは内部キューのサイズを監視する。一定の構成可能なしきい値に達すると、集められた項目を伝播する。この更新ポリシーを使用して、特定モニタ内にバッチ（しきい値 > 1 項目）またはトリクル（しきい値 = 1 項目）の更新メカニズムを構成することができる。

【 0 0 5 3 】

20

間隔トリガ：

間隔トリガは、最後の伝播からの経過時間を監視する。一定の構成可能な時間間隔に達すると、集められた項目を送信する。

【 0 0 5 4 】

複合トリガ：

複合トリガは、前述の両方のポリシーを利用する。当該トリガのうちの 1 つが活動化されると必ず、集められた項目を伝播する。

【 0 0 5 5 】

30

セッション処理

セッション処理の原則は、主に、C A F 処理全体の一部であるキャッシュ・モニタに適用される。バックエンド・モニタ内のセッション処理は、バックエンド特有の実施に完全に依存する。

【 0 0 5 6 】

40

キャッシュ・モニタには、C A F キャッシュを使用するパターンと使用しないパターンという 2 つの異なる使用パターンがある。キャッシュが使用される場合、セッション処理はまったくサポートされない。更新は、キャッシュからバックエンドへと非同期で複製される。バックエンド・データベースへのアクセスがアドミニストレータ（管理者）的なユーザ・アカウントを介して実行されるため、発生するすべての更新に対して接続のプールが可能である。キャッシュが使用されない場合、beginConnect() および endConnect() の方法を使用してセッション処理が実行される。すべての更新は「ユーザごと」にグループ化され、これによって各ユーザを所与の証明書と同期させるための接続のプールが可能になる。

【 0 0 5 7 】

図 8 は、好ましくは本発明の C A F によって適用されるセッション認証を使用した、双方向の同期化に関する同期流れを示す図である。

【 0 0 5 8 】

クライアントを、更新済み項目 A、B、および F、削除済み C、ならびに新しく作成された項目 D として、クライアントとサーバとの間の双方向の同期化が実行される。外部クライアント（たとえば Notes クライアント）を介して、バックエンド上で E が作成され、B、C、および F が更新された。

50

【0059】

クライアントからのパッケージ1は、バックエンド・データ・ストアに関する証明書を送信する。同期エンジンは、これらの証明書を検証のためにCAFに転送する。CAFは担当のバックエンドに証明書の検証を依頼し、同期セッションに有効な認証トークンを同期エンジンに戻す。このトークンを、この同期セッションのためのCAFに対する任意の今後の要求に含めることが必要である。

【0060】

パッケージ3で、クライアントはその更新を同期サーバに送信する。同期エンジンは、認証トークンを提示することによってCAFにバックエンド更新を要求する。次に同期エンジンはリストを比較し、競合を解決し、更新済みのエントリをCAFおよびクライアントに読み込む。CAFはこれらの更新をそのキャッシュに格納し、後にこれらの変更をバックエンド・システムに複製する。

【0061】

CAFとの対話のない同期エンジンは、マッピング・テーブル情報を使用してパッケージ5を処理する。

【図面の簡単な説明】

【0062】

【図1】本発明の基礎とすることができる簡略化された同期アーキテクチャを示す図である。

【図2】バックエンド・データ・ストアへの直接アクセスを伴う従来技術の同期アーキテクチャを示す図である。

【図3】異なるバックエンド・データ・ストアを備えた従来技術の同期アーキテクチャを示す図である。

【図4】本発明によって好ましく実施することができるSynchronML通信プロトコルを示す図である。

【図5】本発明のコンテンツ・アダプタ・フレームワーク(CAF)の基本アーキテクチャを示す図である。

【図6】CAFの好ましい実施を示す図である。

【図7】CAFインターフェースを示す図である。

【図8】クライアント、同期エンジン、CAF、およびバックエンド・システムの間の通信の流れを示す図である。

【図9】CAFによって使用される継承モデルを示す図である。

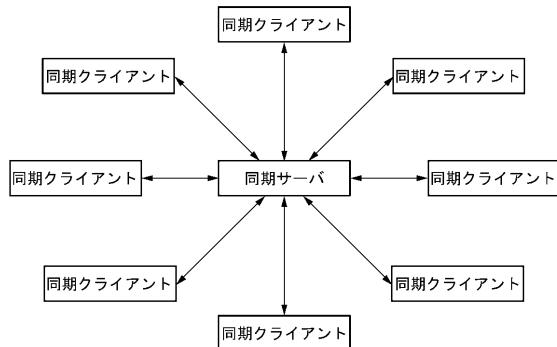
【図10】CAF特有のプロセスの流れを示す図である。

【符号の説明】

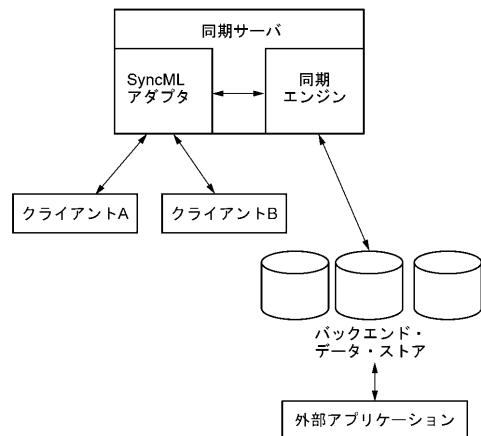
【0063】

2、4、6	クライアントA	
8	ゲートウェイ	
10	中央同期サーバ	
12	同期エンジン	
22	CAF - インターフェース	
24、26	バックエンド・データ・ストア	
28、30	コンテンツ・アダプタ	
40	持続ストア	
50	キャッシュ	
60	バックエンド・モニタ	
70	キャッシュ・モニタ	
80	バックエンド・マネージャ	

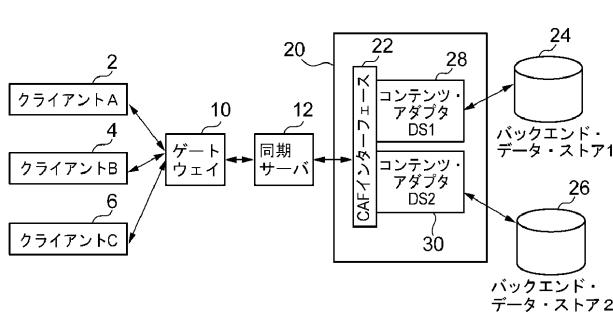
【 図 1 】



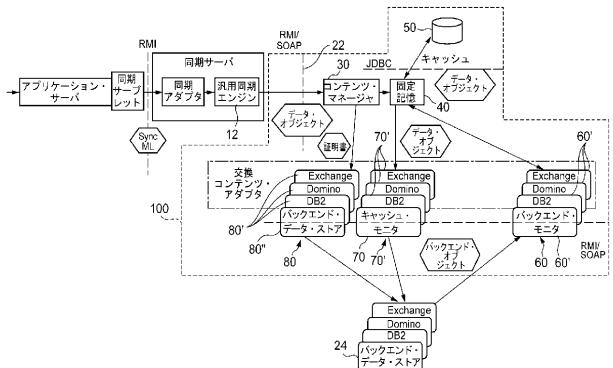
【 図 2 】



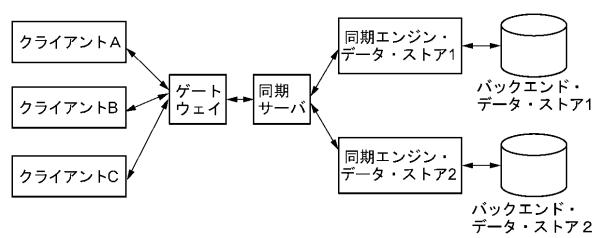
【図5】



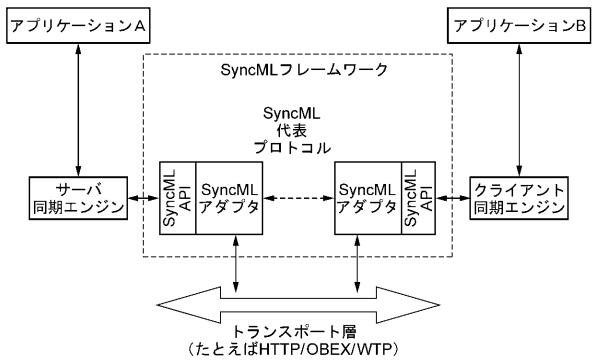
【 四 6 】



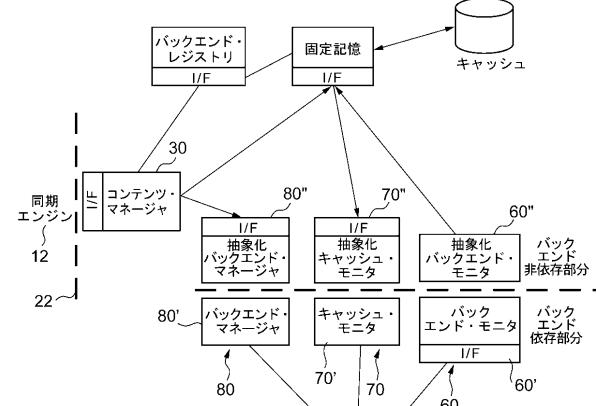
【 図 3 】



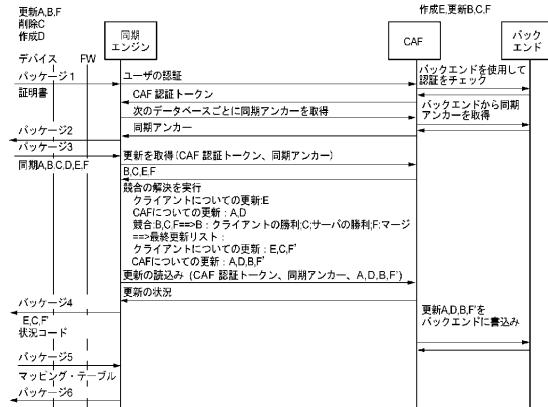
【 図 4 】



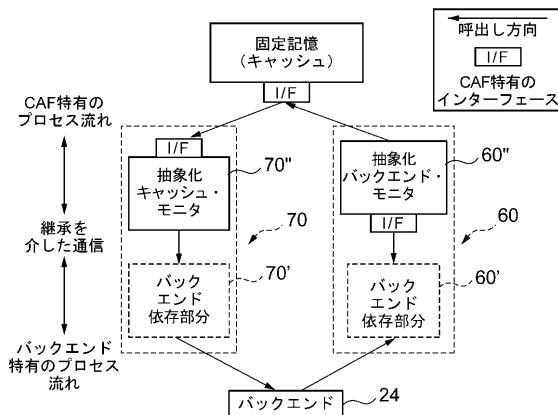
【 図 7 】



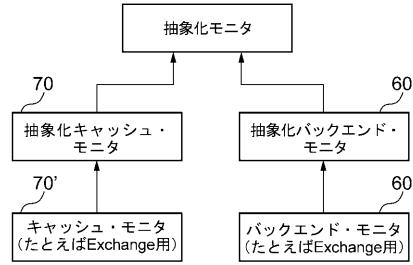
【図8】



【図10】



【図9】



【国際調査報告】

INTERNATIONAL SEARCH REPORT

International Application No.
PCT/EP 03/03529

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/30

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)
IPC 7 G06F

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 practical, search terms used)

EPO-Internal, WPI Data, INSPEC, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>BUTRICO M ET AL: "Enterprise data access from mobile computers: an end-to-end story" RESEARCH ISSUES IN DATA ENGINEERING, 2000. RIDE 2000. PROCEEDINGS. TENTH INTERNATIONAL WORKSHOP ON SAN DIEGO, CA, USA 28-29 FEB. 2000, LOS ALAMITOS, CA, USA, IEEE COMPUT. SOC, US, 28 February 2000 (2000-02-28), pages 9-16, XP010377083 ISBN: 0-7695-0531-7 the whole document</p> <p>---</p> <p>-/-</p>	1-21

Further documents are listed in the continuation of box C.

Patent family members are listed in annex.

* Special categories of cited documents :

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *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)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

T 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

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

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.

& document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the International search report
8 April 2004	12/05/2004
Name and mailing address of the ISA European Patent Office, P.B. 5018 Patentlaan 2 NL - 2280 HV Rijswijk Tel (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Warry, L

INTERNATIONAL SEARCH REPORT

International Application No
PCT/EP 03/03529

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 341 316 B1 (KLOBA DAVID D ET AL) 22 January 2002 (2002-01-22) column 1, line 60 -column 2, line 5 column 8, line 15 -column 9, line 62 column 12, line 6 -column 13, line 2 column 20, line 12 -column 22, line 12 figure 1A figure 3B	1-21
A	WO 01/90933 A (HODGSON KEVIN ;OPENWAVE SYSTEMS INC (US)) 29 November 2001 (2001-11-29) page 2, line 15 -page 5, line 6 page 6, line 15 -page 12, line 5 page 24, line 10 - line 23	1-21
A	JONSSON A ET AL: "SYNCML-GETTING THE MOBILE INTERNET IN SYNC" ON - ERICSSON REVIEW, ERICSSON. STOCKHOLM, SE, no. 3, 2001, pages 110-115, XP001081067 ISSN: 0014-0171 the whole document	1-21
A	3GPP: "Discussion of Synchronization Standards (Release 4)" 3GPP ARIB TR-T12-27.903 V4.0.0, 'Online! March 2001 (2001-03), pages 1-11, XP002276574 Valbone, France Retrieved from the Internet: <URL:www.arib.or.jp> 'retrieved on 2004-04-08! the whole document	1-21

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/EP 03/03529

Patent document cited in search report		Publication date		Patent family member(s)		Publication date
US 6341316	B1	22-01-2002	US	2002046296 A1		18-04-2002
			AU	4674400 A		10-04-2001
			GB	2371902 A		07-08-2002
			WO	0118688 A2		15-03-2001
			US	6553412 B1		22-04-2003
			US	6421717 B1		16-07-2002
			US	2001047394 A1		29-11-2001
			US	2001051979 A1		13-12-2001
			US	2002052781 A1		02-05-2002
			US	2002052916 A1		02-05-2002
WO 0190933	A	29-11-2001	AU	6902901 A		03-12-2001
			WO	0190933 A1		29-11-2001
			EP	1285364 A1		26-02-2003

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,HU,IE,IT,LU,MC,NL,PT,RO,SE,SI,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,LV,MA,MD,MG,MK,MN,MW,M X,MZ,NI,NO,NZ,OM,PH,PL,PT,RO,RU,SC,SD,SE,SG,SK,SL,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,YU,ZA,ZM,ZW

(72)発明者 ハンスマントーヴェ

ドイツ連邦共和国 7 1 1 5 5 アルトドルフ ビアケンシュトラッセ 3 0 / 1

(72)発明者 ヘッパー、シュテファン

ドイツ連邦共和国 7 2 0 7 4 チュービングン ドルファッカーシュトラッセ 2 4

(72)発明者 メルク、ロタール

ドイツ連邦共和国 7 1 0 9 3 ヴァイル・イム・シェーンブッフ キルヒヴィーゼンシュトラッセ
5

(72)発明者 シュトーバー、トーマス

ドイツ連邦共和国 7 1 0 3 2 ベーブリングン シュバルトヴェーク 8

F ターム(参考) 5B082 GA02 HA02 HA03