

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5031113号
(P5031113)

(45) 発行日 平成24年9月19日(2012.9.19)

(24) 登録日 平成24年7月6日(2012.7.6)

(51) Int.Cl.		F I		
G 0 6 F 13/00	(2006.01)	G 0 6 F 13/00	6 5 0 B	
G 0 6 F 15/00	(2006.01)	G 0 6 F 15/00		
G 0 6 F 12/00	(2006.01)	G 0 6 F 12/00	5 3 3 J	

請求項の数 8 (全 33 頁)

(21) 出願番号	特願2011-70885 (P2011-70885)	(73) 特許権者	500046438 マイクロソフト コーポレーション アメリカ合衆国 ワシントン州 9805 2-6399 レッドモンド ワン マイ クロソフト ウェイ
(22) 出願日	平成23年3月28日(2011.3.28)	(74) 代理人	110001243 特許業務法人 谷・阿部特許事務所
(62) 分割の表示	特願2001-511554 (P2001-511554) の分割	(72) 発明者	レイモンド・イー・オジー アメリカ合衆国 マサチューセッツ州 O 1944 マンチェスター ハーパー ス トリート 50
原出願日	平成12年6月28日(2000.6.28)	(72) 発明者	ジャック・イー・オジー アメリカ合衆国 ニューハンプシャー州 03036 チェスター ノース ポンド ロード 87
(65) 公開番号	特開2011-175654 (P2011-175654A)		
(43) 公開日	平成23年9月8日(2011.9.8)		
審査請求日	平成23年3月28日(2011.3.28)		
(31) 優先権主張番号	09/356,148		
(32) 優先日	平成11年7月19日(1999.7.19)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】ピア・ツー・ピア型コラボレーションシステムで使用するエンドポイントに称呼を割り当てるための分散型の方法

(57) 【特許請求の範囲】

【請求項1】

ピア・ツー・ピア型コラボレーションシステムで使用するエンドポイントに称呼を割り当てるための分散型の方法において、前記方法は、

テレスペースに加入するメンバーを招待するステップと、

前記テレスペースの複数のメンバーの各々の各エンドポイントへユニークな称呼を割り当てるステップであって、前記ユニークな称呼は、前記テレスペースに前記複数のメンバーの各々が加入した順序を表わす値と、前記招待されたメンバーについては、前記招待されたメンバーを招待したメンバーのエンドポイントを表わす前記ユニークな称呼とを含む、割り当てるステップと、

割り当てられた前記ユニークな称呼を保存するステップと

を含むことを特徴とする方法。

【請求項2】

異なるエンドポイントの複数の前記ユニークな称呼の各々は招待する側のメンバーの連鎖を表わすことを特徴とする請求項1に記載の方法。

【請求項3】

前記ユニークな称呼は、前記テレスペースの創始メンバーを表わす第1の順位と、前記創始メンバーによって前記テレスペースへ加入するように招待されたメンバーを表わす少なくとも第2の順位とを含む複数の順位を含むことを特徴とする請求項1に記載の方法。

【請求項4】

前記割り当てるステップは

A) 前記テレスペースの作成時に、テレスペースの創始メンバーに対応するエンドポイントが、第1の順位の数字を含むユニークな称呼をそれ自体に割り当てるステップと、

B) 前記テレスペースの作成に続いて、前記創始メンバーにより、前記テレスペースへ前記創始メンバーが招待した前記テレスペースの新規メンバーに対応する複数のエンドポイントの各々に対して、前記テレスペースの創始メンバーの前記第1の順位の数字と、前記新規メンバーが前記テレスペースへ加入した順序を表わすシーケンシャル順位である前記新規メンバーのエンドポイントの称呼の第2の順位の数字とを含むユニークな称呼を割り当てるステップと

を含むことを特徴とする請求項1に記載の方法。

10

【請求項5】

前記割り当てるステップは、前記招待する側のメンバーの各々が、招待する側のエンドポイントが前記テレスペースへ招待する新規のテレスペース・メンバーの各々にユニークな称呼を割り当てるステップを含むことを特徴とする請求項1に記載の方法。

【請求項6】

ピア・ツー・ピア型コラボレーションシステムにおいて使用するエンドポイントへの称呼を割り当てるための分散型の方法において、前記方法は、

テレスペースの各々の招待する側のメンバーが、前記招待する側のメンバーが順次前記テレスペースへ招待する新規のテレスペース・メンバーの各々の各エンドポイントにユニークな称呼を割り当てるステップであって、前記ユニークな称呼は、前記テレスペースに前記新規のテレスペース・メンバーが加入した順序を表わす値と、前記テレスペースへの加入順序に基づく前記新規のテレスペース・メンバーを招待したメンバーのエンドポイントを表す前記ユニークな称呼とを含む、割り当てるステップと、

20

前記招待する側のメンバーが前記テレスペースへ招待する複数のテレスペース・メンバーに対して、メンバーの各称呼は前記テレスペース内でユニークなものであることを保証するステップと

を含むことを特徴とする方法。

【請求項7】

前記保証するステップは、前記招待する側のメンバーが前記テレスペースへ招待する複数のテレスペース・メンバーに対して、メンバーの各称呼が前記コラボレーションシステムの中でユニークなものであることを保証することを含むことを特徴とする請求項6に記載の方法。

30

【請求項8】

前記割り当てるステップは、前記ユニークな称呼を割り当てるために疑似乱数ジェネレータを使用することを含むことを特徴とする請求項6に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明はインターネットなどのネットワークを介して通信できるように相互に接続されたコンピュータに関し、さらに詳しくは分散型データ・モデルに準拠してデータを調整あるいは維持するための分散型コンピュータシステムに関する。

40

【背景技術】

【0002】

インターネットは、数百万のユーザ間での通信及び対話(interaction)のためのダイナミックな公共の環境を作り出した。ビジネスにおいては、インターネットによりベンダとメーカ、メーカとディストリビュータ、ディストリビュータとカスタマ、その他の関係が再定義された。個々の企業の社内にある機密ネットワークへインターネット技術が拡張され、これらは「イントラネット」あるいは「プライベート・インターネット」と呼ばれているが、会社のディレクター及び会社のネットワーク・インフラストラクチャを使って、新しい形で社員間、またワークグループ間でのドキュメントや情報の共有が

50

できるようになった。オンライン・サービスたとえば電子掲示板やチャットルーム、電子商取引また製品のテクニカルサポートなどが、インターネットを介して運営されているウェブ(WWW (World Wide Web))上で利用できる。

【0003】

インターネットは、その中心に、サーバ・クライアント・アーキテクチャがあり、これは個々のクライアント(即ちインターネット・コンテンツのユーザ)がたとえばブラウザなどのコンピュータで実行可能なアプリケーションを介してサーバ(インターネット・コンテンツのプロバイダ)とインタフェースし、ウェブ・サイトからドキュメントを入手する。ブラウザはソフトウェア・プログラムで、パーソナル・コンピュータからインターネット・ドキュメントを要求、受信(たとえばダウンロード)、解釈、表示することができるようにし、また一般にインターネットを散策(navigate)できる。ウェブ・サーバは典型的には外部プログラムを実行するための標準インタフェースを備え、最も一般的なインタフェースはCommon Gateway Interface (CGI)である。ウェブ・サイトはドキュメントの集合で、通常はホームページと、これにリンクされるドキュメントで構成され、クライアントから離れたリモートな場所にあるサーバ上に配置されている。ドキュメントはコンパウンド・ドキュメントであっても良く、これはデータ、グラフィックス、ビデオ、サウンド、および/またはその他の形式のメディアならびに他の文書へのリンクで構成される。基本的に、WWWは相互接続されたドキュメントの、または、さらに正確にはインターネット上の様々なサイトに配置されているドキュメント・オブジェクトのウェブ(蜘蛛の巣: web)である。

【0004】

WWW上のドキュメント・オブジェクトの形式にはドキュメントとスクリプトとがある。スクリプトは実行可能なプログラムまたはファイルに格納された一組のコマンドであり、これがウェブ・サーバにより実行され、ドキュメントを作成し、そのドキュメントがウェブ・ブラウザへ返される。スクリプトによる代表的なアクションには、ライブラリ・ルーチンまたはその他のアプリケーションを実行してファイルまたはデータベースから情報を取り出すこと、または選択されたハイパーテキスト・リンクに対応する文書を取得することが含まれる。スクリプトは、たとえば、ユーザがウェブ・ブラウザ上で特定のハイパーテキスト・リンクを選択した時に、ウェブ・サーバ上で実行される。

【0005】

インターネット技術の基礎となっているものは標準化の進歩であり、これにはパーソナル・コンピュータのハードウェア、ソフトウェア、ネットワーク・プロトコル、インフラストラクチャの標記方法(たとえばUniform Resource LocatorまたはURL)が含まれる。URLはWWWにある全てのドキュメント・オブジェクトについてその位置のアドレスを提供する。URLはドキュメント・オブジェクトを唯一無二の形で(以下、ユニークに)参照し、しばしばインターネット・プロトコルを用いたアクセス・アルゴリズムを定義する。

【0006】

本明細書で使用されている表現として、「インターネット・プロトコル」は、インターネット上で送信するためのドキュメントを含むメッセージを生成するために一般に受け入れられて使用されたり、またはインターネット上でこうしたメッセージを送信するための、それぞれ現行のまたは将来の言語表記および現行または将来の通信プロトコルであると定義される。これらの言語表記には、たとえば、現時点ではHypertext Markup Language (HTML)やeXtensible Markup Language (XML)が含まれる。これらの通信プロトコルはたとえば現時点では、Hypertext Transfer Protocol (HTTP)、TCP/IP、FTP、GOPHERなどを含む。当業者はこれらの言語表記やプロトコルに詳しいであろう。

【0007】

HTMLはハイパーテキスト・ドキュメントを記述するために使用される言語で、マル

10

20

30

40

50

チメディア・コンテンツや他のドキュメントへのリンクを含む。HTMLドキュメントはマークアップ要素（以下、マークアップ・エレメント）の階層的セットを含み、多くのエレメントが開始タグとこれに続くコンテンツ、さらにそれに続く終了タグを備える。タグはアングル・ブラケット（<と>）で囲み、ドキュメントがどのような構造になっているか、ドキュメントをどのように表示するか、またハイパーテキスト・リンクのリンク先とラベルなどを表わす。マークアップ・エレメントのためのタグが存在しており、これらはたとえばタイトル、ヘッダーや、ボールド、インタリックなどのテキスト属性、リスト、パラグラフ境界、他のドキュメントへの外部リンク、また同じ文書の他の部分への内部リンク、グラフィック画像、及びその他多くのドキュメント機能などである。HTMLを使ったプログラミングに関しては多くの成書が流通している。

10

【0008】

XMLはインターネット・プロトコルに含まれるもう一つの言語表記である。XMLは、少なくとも現行バージョンでは、HTMLと互換性があることを補足するものである。コンピュータに格納されていて、XMLドキュメントと呼ばれるデータ・オブジェクトのクラスを記述する、またこれらのオブジェクトを処理するプログラムの動作を記述する標準的な方法である。XMLドキュメントはエンティティ（entities）と呼ばれる格納単位から構成され、これにはテキストデータまたはバイナリデータのどちらかが含まれる。テキストは文字から構成され、その幾つかはドキュメントの文字内容を構成し、別の幾つかはマークアップ（markup）を構成する。マークアップはドキュメントの記述たとえば格納レイアウトや論理構造などを記述する（以下、エンコードする）。XML処理装置と呼ばれるソフトウェアモジュールを使用してXMLドキュメントを読み出し、その内容や構造へのアクセスを提供する。XMLに関するさらに詳しい情報はXML仕様バージョン1.0への参照で得ることができ、<HTTP://www.w3.org/XML>で利用可能であり、また本明細書の参照に含めてある。

20

【0009】

ウェブ・サーバとウェブ・ブラウザは代表的にはHTTPメッセージプロトコルおよびその基盤となるインターネットのTCP/IPデータトランスポート・プロトコルを用いて通信する。HTTPにおいて、ウェブ・ブラウザはウェブ・サーバへの接続を確立してからHTTP要求メッセージをサーバに送出する。要求メッセージに呼応して、ウェブ・サーバは認証をチェックし、たとえばURLで識別されるドキュメントをダウンロードするなどの要求された動作を実行し、要求された動作で得られたHTMLドキュメントあるいはエラーメッセージのどちらかを含むHTTP応答メッセージを返す。返されたドキュメントは単にウェブ・サーバ上に格納されているスタティック（静的）ファイルであったり、要求メッセージに呼応して呼び出されたスクリプトを使用して動的に生成されることがある。

30

【0010】

インターネットを活用するために、インターネット・プロトコルに準拠したツールや資源が開発されており、これには企業で必須のアプリケーションたとえばeメール（email）が含まれる。eメールというのは電子メールのことで、これを用いて選択したアドレスで電子的にドキュメントを送受信する。インターネットを使う対話の大半が、「ドキュメントの送受信」モデルに従ったeメールの使用や、その他のブラウザを使うメディアであると予想されていた。おそらくはこのモデルのために、ユーザはインターネットを本質的に「ピア・ツー・ピア」的なものであり、これによって個人が他の個人によって提供されたドキュメントにアクセスできるものであり、しかも、もっと大きな権限を有する者による干渉を受けないでできるものであると見なすことが多い。

40

【発明の概要】**【発明が解決しようとする課題】****【0011】**

インターネットはユーザに娯楽を提供したり通信の有用な方法を提供する上でダイナミックかつフレキシブルなものであるが、ユーザの要求の全部を満たすことはできない。ユ

50

ーザはますますインターネットを通して対話するようになるが、もっと在来型の方法たとえばマルチメディア（電話、ファックス、ホワイトボード）のものやマルチテンポラル（リアルタイム、オーバーナイト・メール）なもの、またその他形式にとらわれない通信手段を使うことにより、インターネット以外での対話も引き続き行われている。

【0012】

共有されるプライベート空間での個人や中小グループ間でのパーソナルでプライベートな通信やその他共有される相互の活動にインターネットに基づくパラダイムを拡張することが望ましい。このような対話は、参加者のパーソナルコンピュータ間またはその他のネットワーク接続可能なデバイス間で即時的、直接的、機密的に行なわれるのが望ましく、これらの対話は、通信が傍受されたり機密性が侵害されたりするような第三者のウェブ・サイトへとインターネット上のトラフィックが集中させるようなサーバ・ボトルネックとは何ら関わりを持たないのが良い。

10

【0013】

また様々な遠隔地（以下、リモートサイト）にいるユーザがピア・ツー・ピアの形式でドキュメントを共有し編集できるようにする一方で、ローカルサイトにおいてドキュメントの一貫したコピーを維持することのできる技術を提供するのも望ましい。このような技術ではユーザの誰もがドキュメントに関する変更要求を発行でき、変更要求が発行された時点でサイトがネットワークから切断されても当該変更が調整された方法で全てのローカルサイトにあるコピーに対して非同期的に行なうことができる。この技術は全てのサイトにわたって高いレベルでのデータ一貫性を提供することにより各々のサイトが様々な地理的に分散したユーザによる閲覧および/または編集のために基本的に同じドキュメントを提示することができるようになる。

20

【課題を解決するための手段】

【0014】

本発明はパーソナルコンピュータ・システム、通信装置、またはその他のネットワーク接続可能なデバイスで動作可能で、遠隔地に居る他のユーザと共有して特定のタスクたとえば「チャット」、ゲーム、またはビジネスアプリケーションを実行するためのアクティビティとよばれる装置に関するものである。アクティビティはデルタと呼ばれるデータ変更要求をユーザとの対話呼応して生成するためのツールを含む。アクティビティはデータ・モデルに準拠して不揮発性で永続的であることが望ましいメモリにデータを保存するためのデータ変更エンジンも備える。データ変更エンジンはツールからデルタを受信して、そのデルタにその要求を実行するアクティビティ特有のコマンドを提供し、デルタの実行に起因するデータ変更をツールへ通知するためのツール・エンドを含む。データ変更エンジンはダイナミクス・マネージャからデータ変更指示を受信してからデルタを実行する、即ちデルタのコマンドを実行して要求された変更をデータに加えるためのダイナミクス・マネージャ・エンドも含む。

30

【0015】

別の観点においては、本発明は「テレスペース」と呼ばれる共有プライベート空間で人間及び小グループ間での通信及びその他の共有活動及び相互活動を提供するためのシステムに関する。本発明のシステムにおいては、テレスペースの参加者またはメンバーは、ネットワーク接続可能なデバイスを介して対話し、このデバイスはたとえばインターネットなどのネットワーク上で互いに通信してテレスペース・データのローカルコピーを個々に格納する。各々のテレスペースはテレスペースのメンバーのネットワーク接続可能なデバイスの各々の上で動作可能な1つ以上のアクティビティの具現化されたもの（以下、インスタンシエーション）である。各々のアクティビティはテレスペースのメンバーとの対話に呼応してデータ変更要求またはデルタを開始するための1つ以上のツールを含む。各々のアクティビティはまた、ツールとは別に、共通データ・モデルに準拠したテレスペース・データのローカルコピーを保持するための1つ以上のデータ変更エンジンも含む。このデータ・モデルはたとえば、アクティビティに固有であって、テレスペース全体にわたるものであることが望ましい。各々のネットワーク接続可能なデバイスもデルタに呼応し

40

50

て、要求された変更をデータのローカルコピーに適用することによりデータ変更エンジンに指示を出してデルタを実行させるためと、様々なネットワーク接続可能なデバイスからのデルタの実行を調整するためのダイナミクス・マネージャを含む。

【 0 0 1 6 】

さらに別の観点においては、本発明は望ましくはインターネット・プロトコルに従って通信するように接続された望ましくはネットワーク接続可能な別の場所にある複数のデバイスを含むネットワーク化されたシステムにおいて実施できる。ネットワーク接続可能な各のデバイスは (a) 共通データ・モデルに準拠してアクティビティに関連するデータのローカルコピーを格納するためのメモリ、 (b) 前述のデータ変更エンジンに対応する1つ以上のツールを各々が含む1つ以上のアクティビティ、 (c) デルタが関係するテレスペースに参加するネットワーク化されたシステムのネットワーク接続可能な全てのデバイスの間でデルタを共有させるための通信マネージャ、 (d) ローカルに生成されたデルタと遠隔地で生成された (以下、リモート生成された) デルタの両方の実行を調整するためのダイナミクス・マネージャ、を含む。ネットワーク化されたシステムのネットワーク接続可能なデバイスのどれででもユーザの対話を開始してその結果としてのデルタを生成することができ、このときのデルタは望ましくはネットワーク上のデバイスの各々に送信される。ダイナミクス・マネージャはローカルのデータ変更エンジンの動作を指示し、望ましくはデルタの実行に優先順位を付けてネットワーク化されたシステム全体でのデータの実質的な一貫性を維持する。ネットワーク化されたシステムは様々なリモートサイトにいるユーザに対して、たとえばデータを共有して編集したり、ピア・ツー・ピア型の形式でその他の活動を独立して実行できるようにする一方で、ネットワークから切断されている時でも各々のネットワーク接続可能なデバイスで使用するデータの実質的に一貫したコピーを保持することができるようにする。

【 0 0 1 7 】

さらに別の観点においては、本発明は個人間の対話のためのアクティビティに基づくコラボレーションシステム (A B C システム) として実装可能である。 A B C システムはたとえば公衆ネットワーク (たとえばインターネットまたはワイドエリアネットワーク (W A N)) 経由またはプライベート・ネットワーク (ローカル・エリア・ネットワーク (L A N)) またはイントラネット) 経由で、ケーブル、光ファイバー、赤外線 (I R) またはその他の形式の通信リンクにより前述のネットワーク化されたシステムを形成するように接続可能なネットワーク接続可能な複数のデバイスを含む。ネットワーク接続可能な各々のデバイスは前述のメモリと、1つ以上のダイナミクス・マネージャを含むフレームワークを装備している。ネットワーク接続可能なデバイスは公衆ネットワーク・インフラストラクチャ、パーソナル・デスクトップ・アプリケーション、及びインフラストラクチャ (時に「パーソナル・インターネット」と呼ばれることがある)、また適用可能なら企業内イントラネットともプラグイン的に互換性を有するものが望ましい。 A B C システムは動的に挿入可能型アクティビティ・コンポーネント (以下、ダイナミックなスナップイン・アクティビティ・コンポーネント) を使用するのが望ましく、各々がたとえば「チャット」、ドキュメント編集、ゲーム、などの特定のタスクを実行する。アクティビティ・コンポーネントはたとえばシュリンクラップ製品として利用できたりまたはたとえばインターネットなどのネットワーク上からダウンロードできるソフトウェアを含むことができる。アクティビティ・コンポーネントはフレームワークと共通の A P I (アプリケーション・プログラミング・インタフェース) を介して動作可能である。したがって、フレームワークはアクティビティ・コンポーネントの形をとるアプリケーションのためのプラットフォームであると見なすことができまたそのように機能する。

【 0 0 1 8 】

本発明のさらに別の観点では、 A B C システム全体で高レベルのデータ一貫性を維持するための分散技術に関する。各々のピア・ユニットにおいて、ダイナミクス・マネージャはデータ変更要求を実行する順序を決定してデータの一貫性を促進 (p r o m o t e) するデータ変更要求優先順位スキームを実現する。データ変更要求優先順位方式の代表的な

10

20

30

40

50

実施態様においては、各々のデータ変更要求は要求シーケンス番号情報と従属情報とを有する（含むのが望ましい）。ダイナミクス・マネージャは適当なエンジンに対してこの情報に応じてデータ変更を行なわせ、ロールバック（変更取り消し（undo））させ、またやり直し（変更の再実行（redo））を行なわせることができる。従属情報はたとえば単一の優先データ変更要求を指定することができ、この場合ダイナミクス・マネージャは優先するデータ変更要求に対応するピア・ユニットで実際に処理されてからその優先要求に従属するデータ変更要求を処理するように保証する。同じ優先要求を従属先とするデータ変更要求間で従属性の衝突がある場合、従属性衝突解決スキームを使用できる。たとえば、日付またはテレスペースのメンバーシップのスポンサー、エンドポイントの特性（たとえばジョブのタイトル、ワークグループ、または部門）、要求された変更の性質またはドキュメント位置、またはその他の根拠に基づいて優先権を付与することができる。

10

【0019】

つまり、本発明はそれぞれ別個の商業製品やサービスとして実現可能で、これには個々のアクティビティ・コンポーネント、ネットワーク接続可能な個々のデバイス用のフレームワークおよび/またはネットワーク化されたシステムまたはABCシステム、ならびにシステムを構成するデバイス間の対話を行なうための通信サービスの提供を含む。

【図面の簡単な説明】

【0020】

【図1】図1は、従来のコンピュータ・システムの代表的アーキテクチャのブロック図である。

20

【図2】図2は、図1のコンピュータ・システムで実行される従来のアプリケーションのブロック図で、制御モジュールはディスプレイまたはその他のユーザ・インタフェースとインタフェースすること及びデータ・モデルにしたがってデータを維持することを担当する。

【図3】図3は、インターネット・ベースのシステムのブロック図で、WWW用のクライアントサーバ・システムと、本発明の実施態様によるパーソナル・ウェブのためのピア・ツー・ピアシステムとの両方を示す。

【図4】図4は、本発明の実施態様による図1のコンピュータ・システムで実装されるABCシステムの一部のブロック図である。

【図5】図5は、図1のコンピュータ・システムで実装されるABCシステムの一部のブロック図で本発明の実施態様による代表的テレスペース・アプリケーションを示す。

30

【図6】図6は、図5のチェス・テレスペースで実装されるABCシステムの一部のブロック図である。

【図7】図7は、図4のABCシステムの一部を含むフレームワーク700のブロック図である。

【図8】図8は、ピア・ユニット間の通信に係る図4のABCシステムの部分の実施態様のブロック図である。

【図9】図9は、図4のABCシステムで処理するデルタの代表的フォーマットを示す略図である。

【図10】図10は、図4のABCシステムで処理するデルタをサポートするメモリ構造のブロック図である。

40

【図11A】図11Aは、本発明の実施態様による図4のABCシステムによるデルタ処理の方法を示すフローチャートである。

【図11B】図11Bは、本発明の実施態様による図4のABCシステムによるデルタ処理の方法を示すフローチャートである。

【図12】図12は、図11の方法において従属性の衝突を解決するためのフローチャートである。

【発明を実施するための形態】

【0021】

A. 術語

50

最初に、本明細書で使用している幾つかの術語と略語を紹介するのが有用であろう。これには以下を含む：アクティビティ、クライアント、通信マネージャ、コンポーネント、コントローラ、デルタ、デバイス、ダイナミクス・マネージャ、エンジン、フレームワーク、アイデンティティ、メンバー、ナビゲーション、人、サーバ、テレスペース、ツール、URL、XML。

【0022】

アクティビティ (Activity) は、あるユーザが他のユーザと対話するために実際にはABCシステムで何をするかを表わす。術語「アクティビティ」は状況によって二つの意味のどちらかを取ることができる。「アウトサイド・ビュー」からは、ABCシステムとコンピュータ・ユーザとの間のランタイムの対話を意味する、たとえばこの種のアクティビティでは共有ホワイトボードに描画するのをユーザに許可したり、または二人以上のユーザがリアルタイムでチャットするのを許可する。この意味でのアクティビティはまたバックグラウンド・サービス (即ちエージェント) であっても良い。これはローカルデバイスまたはサーバデバイス上でユーザの側で動作し、たとえばアーカイブサーバ・エージェント、または受信FAXエージェントなどである。ソフトウェア設計者の「インサイド・ビュー」からは、「アクティビティ」はユーザのパーソナル・コンピュータまたはその他の形態の装置上で動作することができ決まったタスクを実行するモジュール化されたプログラムであってコンピュータで実行可能なプログラムを表わす。この意味でのアクティビティはツールとエンジンという2種類のコンポーネントからなり、これらは設計時に統合されて「アクティビティ・テンプレート」を作成する。アクティビティ・テンプレートはたとえばシュリンクラップ・ソフトウェアとして、またはウェブからダウンロードできるソフトウェアとして、など多くの方法で配布することができる。ワープロ、スプレッドシートなどに使用するため広く配布されるプログラムはABCシステムで使用できるように設計されたバージョンであって、ABCシステムプログラマ・インタフェースに適合するアクティビティ・テンプレートの形で販売されるバージョンを有するものとして考えている。

【0023】

クライアント (Client) は、ユーザのパーソナル・コンピュータ、通信機器、またはその他の形のデバイスで、ヒトとの対話ができ、ネットワークたとえばインターネットに接続可能なものを表わす。ユーザのパーソナル・コンピュータ上で実行されるバックグラウンド・プログラムは「クライアント・サービス・コントローラ」と呼ばれる。ユーザのパーソナル・コンピュータ上で実行されるフォアグラウンドのユーザ・インタフェース (UI) プログラムは「クライアントUIコントローラ」と呼ばれる。

【0024】

通信マネージャ (Communications Manager) は、しかるべき宛先とのデルタの送受信を指示するメカニズムを表わす。通信マネージャはたとえば、コンピュータで実行可能なプログラムとして実装することができ、これが、ネットワークたとえばインターネット上で送信するためにツール/エンジンのペアで作成されてローカルに開始したデルタを別のリモート・パーソナル・コンピュータへまたは他の形のネットワーク接続可能なデバイスへ送信する、またはそのリモートデバイスがネットワークに接続されていない場合にはリレー装置へ送信するもので、またこれはネットワークたとえばインターネット上で受信されるリモートで生成されたデルタをダイナミクス・マネージャへ送信する。

【0025】

コンポーネント (Component) は、コンピュータで実行可能なプログラムとアクティビティ内部で使用されるその資源を表わす。全てのコンポーネントはURLで識別される。ウェブがコンポーネントのグローバル・リポジトリで、ABCシステムの適当なコンポーネント・プラットフォームへ確実にダウンロード可能なバージョンを備えるものとして考えている。

【0026】

10

20

30

40

50

コンピュータ (Computer) は、以下で定義するような装置を表わす。

【0027】

コントローラ (Controller) は、トップレベルにあるプログラムであってコンピュータで実行可能なプログラムを表わし、これはABCシステムの「内部」機能を駆動する。コントローラは任意のプラットフォームたとえばUNIX (登録商標) のデーモン・コントローラまたはインテル社製マイクロプロセッサ搭載コンピュータ・システム上で動作するマイクロソフト・ウィンドウズ (TM) 用コントローラとしてチューニングされる。一般的な2種類のコントローラとしては、一般にユーザからは見えないバックグラウンド・プロセスとして動作するサービス・コントローラと、ユーザが対話的に制御するユーザ・インタフェース (UI) コントローラがある。

10

【0028】

デルタ (Delta) は、1つまたはそれ以上のツールからエンジン宛てのデータ変更要求 (即ちデータに対して希望する変更についての通知またはプロンプト) を内包する自己内包型のデータユニットを表わす。ツールはユーザとの対話に回答してダイナクス・マネージャによるデルタ作成を開始し、後述するように、ダイナクス・マネージャの制御下で非同期実行のためにこれらをエンジンに送信する。デルタは制御情報を提供するヘッダ部分と、要求が関連するデータに関する情報を提供するペイロード部分とを含む特定のフォーマットを持っている。個々のデルタは1つまたはそれ以上のペイロードを持つことができ、多数のペイロードを使用する場合には各々をユニークなデバイス機能またはユーザの役割を有するテレスペース・メンバーあてに指定できる。ツールはユーザへの表示用またはその他の形でのプレゼンテーションを行なうためエンジンから潜在的データ変更に関する情報を要求し、デルタが表示を変更させた時に非同期的に通知される。

20

【0029】

デバイス (Device) は、たとえばパーソナル・コンピュータ (pc) や通信機器など、本明細書では文脈からして特に別のものが示されない限りは、通常ネットワーク接続可能な、即ちネットワーク接続可能な他のデバイスとネットワーク上たとえばインターネット上でインターネット・プロトコルを使用して通信することができるような物理的ハードウェア装置を表わす。全てのデバイスにはユニークな識別コードがABCシステムによって割り当てられ、URLを有し、デバイスを使用する者とは区別される。デバイスは潜在的に多くのテレスペースをホスティングする。デバイスは、たとえば、pcであったり、セルラホン、ファックス、ハンドヘルド・オーガナイザ、ケーブルテレビのセットトップ・ボックス、エンターテイメント用電子ユニット (たとえば電子玩具やゲームと、ステレオ装置、またはこれら以外の同様にネットワーク接続可能な機器、装置、製品など) のことがある。

30

【0030】

ダイナクス・マネージャ (Dynamics manager) は、ABCシステムのフレームワークの一部を表わし、デルタの作成を支援したりデルタの実行を調整する。ダイナクス・マネージャはたとえば、ユーザのパーソナル・コンピュータ上でまたはその他の形態を有しネットワーク接続可能なデバイス上で動作するコンピュータで実行可能なプログラムとして実装することができる。

40

【0031】

エンドポイント (Endpoint) は、デバイスと人のユニークなペアを表わす。ABCシステムでは、ある1つのデバイスの多数のユーザの中から一人の人をユニークに識別する方法として、またはその同じ人によって使用される多数のコンピュータの中から1つのデバイスや複数のデバイスを識別する方法として、この概念を使用する。

【0032】

エンジン (Engine) は、ABCシステム内部の永続的格納及びデータ構造の管理及び変更を実現するものであってアクティビティの「下半分」と呼ばれるものを表わす。エンジンはコンポーネントとしてパッケージされ、それ自体にはユーザ・インタフェースがなく、ユーザ・インタフェースについては対応するツールに依存し、実質的に移植可能

50

である。エンジンは対応するツールを機能させるためにのみ存在するのが望ましい。エンジンはユニバーサル同期 (universal synchrony) した状態で動作するのが望ましい。

【 0 0 3 3 】

フレームワーク (Framework) は、ABCシステムの「内部動作」を表わし、ダイナミクス・マネージャを始めそのサブシステムのほとんどのものを含んでいるが、ユーザ・インタフェースはない。フレームワークはユーザのパーソナル・コンピュータ上またはその他の形態のネットワーク接続可能なデバイス上で動作することのできるものでコンピュータで実行可能なプログラムであり論理的には高レベルのコントローラとアクティビティの間に位置する。

10

【 0 0 3 4 】

アイデンティティ (Identity) は、一般に「人 (person) 」と同義語だが、一般にURLで表わされるもので他の人がある人を認知できる1つの名前を指す。この概念が使用される理由としては、ABCシステムではある人が多くのアイデンティティまたは別名を持つことができ、またたとえば多くの異なるURLを持てるという考え方を採用しているためである。

【 0 0 3 5 】

メンバー (Member) は、テレスペース内部の参加者またはテレスペースの加入者を表わし、一般にテレスペースのメンバーシップを表わす場合にはエンドポイントの人の部分 (デバイス部分と区別して) と同義語である。

20

【 0 0 3 6 】

ナビゲーション (Navigation) はURL間を移動する動作を表わし、ABCシステムではウェブ・ブラウザによるウェブページ間、ウェブサイト間のナビゲーションと類似している。

【 0 0 3 7 】

人 (Person) は、人間、または人間の代理として作用するサーバを指し、一般にテレスペース内部のアクティビティに参加している者を指す。各々の人は1つまたはそれ以上のアイデンティティを有し、その各々はURLでユニークに識別される。

【 0 0 3 8 】

サーバ (Server) は、通信チャンネルを介する場合を除き普通は人間との直接対話ができないがサービス・コントローラのもとでバックグラウンド・プログラムとしてのみ実行することができるデバイスを指す。

30

【 0 0 3 9 】

テレスペース (Telespace) は、仮想空間であって、1つ以上のアクティビティに参加するために人びとが集まり、人びとが何らかのものを共有する仮想空間を指す。テレスペースとアクティビティの結果はユーザのパーソナル・コンピュータ上またはその他の形態のネットワーク接続可能なデバイス上のメモリに永続的に格納される。テレスペースは、一般に、あるユーザのデバイスと他の人達のデバイスの間で同期を保っている。テレスペースは「メンバーシップ」の論理的単位またアクティビティへのアクセスを表わす。テレスペースは1つ以上のアクティビティのインスタンシエーション (instantiation) である。

40

【 0 0 4 0 】

ツール (Tool) は、アクティビティのユーザ・インタフェースを実現するものであってアクティビティの「上半分」と呼ばれるものを指す。ツールはコンポーネントとしてパッケージされ、ユーザのジェスチャーに応答してデルタの作成を開始し、個々のアクティビティの対応するエンジンと対話して特定のタスクを実行する。

【 0 0 4 1 】

URLは、universal resource locator (汎用資源識別子) の略語で、ドキュメントなどの資源の構造化されたユニークなアドレスであり、また場合によっては資源を取り扱うときの方法を表わす。URLは本明細書において実質的に全

50

ての永続的オブジェクト、ユーザから見えるオブジェクト、及び外部プログラムから見えるオブジェクトについて使用される。

【0042】

XMLは、前述の通り、拡張マークアップ言語 (eXtended Markup Language) の略語で、ウェブ上で使用することを想定したSGMLから派生した標準化された構造化されたデータ・フォーマットである。状況によってはABCシステムで使用されるメモリ内のオブジェクト構造を表わすこともあり、これはXML標準のセマンティクス (semantics) に準拠するものである。多くのXMLの概念が本明細書で使用されており、たとえばドキュメント、エレメント、タグ、属性、値、コンテンツ、エンティティ、リンク、ポインタなどがある。XMLは本明細書ではたとえばドキュメント内のデータを構造化するために使用される。

10

【0043】

B. 従来コンピュータ・システム

図1は、代表的なコンピュータ・システム100の従来型のシステム・アーキテクチャを図で表わしたものであり、開示された本発明をこれに実装することができる。図1に例示したコンピュータ・システムは説明の目的のみ議論するが、本発明を制限するものとして見なすべきではない。本発明はさらにコンピュータ・システムとして従来考えられてきたデバイスに制限されるものではなく、これはたとえばビデオゲーム・コンソール、パーソナル機器、またはケーブルテレビのセットトップボックスなどを含む各種のネットワーク接続可能なデバイスのいずれにおいても本発明を実装できるためである。以下の説明は特定のコンピュータ・システムを記述する際に共通に使用される術語を示しているが、説明される概念は図1に図示したものと異なるアーキテクチャを有する他のコンピュータ・システムにも等しく適用されるものである。たとえば、ケーブルテレビのセットトップボックスは大容量格納機能を含まないがデジタル信号プロセッサを内蔵したビデオチューナを含んでいる。

20

【0044】

コンピュータ・システム100は、従来のマイクロプロセッサを含む中央演算処理装置 (CPU) 105、一時的な情報格納用のランダム・アクセス・メモリ (RAM) 110、永久的な情報格納用のリード・オンリー・メモリ (ROM) 115を含む。メモリ・コントローラ120はシステムRAM110を制御するために提供される。バス・コントローラ125はバス130を制御するために提供されており、割り込みコントローラ135は他のシステム・コンポーネントからの様々な割り込み信号を受信して処理するために使用される。

30

【0045】

大容量格納はディスク142、CD-ROM147またはハードディスク装置152によって提供できる。データやソフトウェアはリムーバブル・メディアたとえばディスク142やCD-ROM147を経由してクライアント・コンピュータ100と交換できる。ディスク142はディスクドライブ装置141に挿入することができ、ディスクドライブ装置141はコントローラ140によってバス130へ接続されている。同様に、CD-ROM147はCD-ROMドライブ装置146に挿入することができ、CD-ROMドライブ装置146はコントローラ145によってバス130へ接続されている。最後に、ハードディスク152は固定ディスクドライブ装置151の一部であり、これはコントローラ150によってバス130へ接続されている。

40

【0046】

クライアント・コンピュータ100へのユーザ入力は多数のデバイスによって提供され得る。たとえば、キーボード156とマウス157はキーボード及びマウス・コントローラ155によってバス130へ接続される。オーディオ・トランスデューサ196はマイクロホン及びスピーカの両方として機能することができるもので、オーディオ・コントローラ197を介してバス130へ接続されている。他の入力デバイスたとえばペンおよび/またはタブレットや、音声入力用マイクロホンなどを適当なコントローラとバス130

50

経由でクライアント・コンピュータ100へ接続することができるのは当業者には明らかである。DMAコントローラ160はシステムRAM110へのダイレクト・メモリアクセスを実行するために提供される。視覚表示はビデオコントローラ165によって生成され、これがビデオ・ディスプレイ170を制御する。

【0047】

コンピュータ・システム100はまたクライアント・コンピュータ100がバス191経由でネットワーク195へ相互接続できるようにするネットワーク・アダプタ190も含む。ネットワーク195はローカル・エリア・ネットワーク(LAN)、広域ネットワーク(WAN)、またはインターネットであり、多数のネットワーク・デバイスを相互接続する汎用通信回線を使用する。

【0048】

コンピュータ・システム100は一般にオペレーティング・システム・ソフトウェアによって制御調節される。コンピュータ・システム制御機能のなかでも、オペレーティング・システムはシステム資源の割り当てを制御しプロセスのスケジュール、メモリ管理、ネットワーク及びI/Oサービスなどのタスクを実行する。

【0049】

C. データ・モデルを制御する従来の方

図2は「モデル・ビュー・コントローラ」として周知の技術を使ったデータ・モデルを制御する従来システム200を示す。システム200において、データ・モデル・モジュール202はたとえばデータベースなどのデータをデータ・モデルにしたがって格納し維持する。コントローラ・モジュール204はローカルに開始したイベント206(たとえばユーザ入力など)やまたは外部で開始したイベント208(たとえばリモートのコンピュータ・システムから受信したイベント通知など)に応じて、「現在値をセットする」コマンドをデータ・モデル・モジュール202に指示して格納しているデータへの変更を行なわせる。コントローラ・モジュール204はデータ変更を出力装置たとえばディスプレイ206などに通知する。ディスプレイ206はデータ・モデル・モジュール202から変更されたデータを取得してデータの表示を更新し、現在のデータがユーザから観察できるようにする。前述の技術は厳密にイベント駆動型であり動作においてリニアである。典型的には、データへの変更をコントローラ・モジュール204に指示させる内部または外部のイベントはコントローラ・モジュール204のFIFO(先入れ先出し)型バッファに格納され、次いでイベント通知がコントローラモジュールに到達した順序で処理される。また典型的には、グローバルな一貫性問題、即ちシステム200内のモデルデータのコピーが他のリモートシステムにあるデータと一致していることを保証する対応がシステム200には行なわれていない。

【0050】

システム200で例示したようなデータ・モデルを制御する従来のアプローチは、コンピュータ・システムがメインフレームとパーソナル・コンピュータとを問わず相互に隔離されて動作していた時代には適切だった。インターネットの発達とともに、コンピュータ・ユーザはコラボレーションするときのもっと優れた方法を要求しており、本発明では、データ・モデルに対してデータ変更エンジンによって行なわれる変更ダイナミクス・マネージャが介在する「モデル介在型制御」と呼べるようなものを提供する。

【0051】

D. コンピュータ間動作のインターネット・パラダイム

図3はインターネット環境300を示したもので、本発明が有利に使用できるものである。ユーザの観点からすると、従来のクライアントサーバ・ビュー302において、個別のクライアント306、308はインターネット309上でサーバ310と別々に通信する。各々のクライアント306、308はインターネット309経由でサーバへ要求を発信し、各々について、サーバ上に格納されたドキュメントをインターネット309上でクライアントへ提供することにより、サーバ310がこの要求に応答する。各々のサーバ310とクライアント306、308は、図1に図示したようなパーソナル・コンピュータ

10

20

30

40

50

として実現でき（サーバの場合には幾つかのUIコンポーネントはオプションとなる）、適当なプログラムたとえばクライアント・ブラウザ311および/または後述するようなその他の通信インタフェースの実行が可能である。インターネット使用のピア・ツー・ピア・ビュー312においては、ユーザのコンピュータ・システムはピア・ユニット314A~Dを構成し、インターネット経由の通信は一つのピア・ユニットから別のピア・ユニットへ明らかに介在するものがなくても配向できる。

各々のピア・ユニット314A~Dは図1に図示したようなパーソナル・コンピュータとして、またはその他の形態のネットワーク接続可能なデバイスとして実現可能である。本発明はクライアント306、308、またはピア・ユニット314A~Dのいずれかで有利なるように実現可能であるが、このような説明はシステムのピア・ツー・ピア・ビューでの実装が中心となる。

10

【0052】

さらに詳細には以下で説明するが、このようなピア・ツー・ピア通信は直接またはリレー・デバイス316経由で行なうことができる。リレー・デバイス316は「ストア・アンド・フォワード」型が望ましく、これはインターネットから一時的に切断されているピア・ユニット314へ向けられたメッセージを格納し、その後再接続した時にそのピア・ユニットへメッセージを転送することができるものである。

【0053】

E. ABCシステムのアーキテクチャと動作

図4はピア・ユニット314A~Dの一つ、たとえばピア・ユニット314Aにおいて実現されるABCシステム400を示す。ピア・ユニット314A上のABCシステムはフレームワーク402、少なくとも一つのテレスペース404、及びユーザ・インタフェース406を備える。フレームワーク402は多数のテレスペース404にサービスするためのプラットフォームを提供するのが望ましい。フレームワーク402は、アクティビティが動作しフレームワーク・コンポーネントと通信するプログラマ・インタフェースを備えるモジュラー構造が望ましい。

20

【0054】

フレームワーク402はユーザ・インタフェース・マネージャ408、アイデンティティ・マネージャ410、テレスペース・マネージャ412、アクティビティ・マネージャ414、格納・マネージャ416、コントローラ・サービス・マネージャ418、ダイナミクス・マネージャ420、通信マネージャ422を含む。

30

【0055】

ユーザ・インタフェース(UI)マネージャ408は多数のユーザ・インタフェース・コントローラ(図示していない)の共有サービスの管理を担当する。UIマネージャ408はディスプレイ・ウィンドウの窓ガラスの向こうにあるアクティビティのレイアウトを管理し、それ以外にユーザ・インタフェースとして期待する「ルック・アンド・フィール」を提供する。UIマネージャ408はアクティビティのナビゲーション(たとえばgo(行く)、next(次)、previous(前))なども管理しナビゲーション履歴を保持する。

【0056】

アイデンティティ・マネージャ410はあるテレスペース・メンバーのアイデンティティの保存を担当する。前述したように、アイデンティティは名前であり、また対応するURLであり、これにより各々のユーザが他のユーザによって認識される。個々のユーザは1つまたは多数のアイデンティティを持つことがある。アイデンティティ・マネージャ410はアイデンティティのレコードまたはテーブルを望ましくはXMLで保持する。アイデンティティ・マネージャ410はまたテレスペース・メンバーのURLとこれらに対応するデバイスのURLのレコードまたはテーブルを望ましくはXMLで保存する。これ以外に、独立したメンバー・マネージャを実装することができる。

40

【0057】

テレスペース・マネージャ412はピア・ユニット314Aで開くことができるテレス

50

ペース404の各々の管理を担当する。各々のテレスペース404は1つまたはそれ以上のアクティビティのインスタネーション (i n s t a n t i a t i o n) である。各々のテレスペース404は対応するアクティビティ・マネージャ414を有している。

【0058】

アクティビティ・マネージャ414は(a)テレスペースへ新規アクティビティを追加する、(b)テレスペースで既存のアクティビティを開く、(c)アクティビティ・テンプレートの新しいバージョンからテレスペースのアクティビティを更新する。新規のアクティビティを追加するには、アクティビティ・マネージャ414にアクティビティ・テンプレートのURLを提供してテンプレートを開き、テンプレート情報(たとえばコンポーネントURL)を抽出してテレスペースへ伝搬する。テンプレートはあるテレスペースについての初期アクティビティ構成を定義する。ユーザは後で必要に応じてテレスペース404へさらにアクティビティを追加できる。追加した後、アクティビティはテレスペースの「一部」となりテレスペースのメンバー全員から見えるようになる。テレスペースはアクティビティ・マネージャを識別してアクティビティ・マネージャとデータを結合するタグを備える。望ましくは、各々のドキュメントはローカル・レジストリをこれにリンクし、XMLタグの名前がレジストリに保存されるようにして、拡張可能な、プラットフォームに依存しない方法でドキュメントとこれに対応するテレスペースの間のマッピング(参照ポイントまたは連係)を表現するようにする。各々のテレスペース・メンバーはその人のピア・ユニット314A~Dで利用できるテレスペースのフレームワークとアクティビティ・テンプレートを有する。

【0059】

各々のアクティビティはツール424などのツールと、エンジン426などのエンジンとを含む。ツール424はアクティビティにユーザ・インタフェース(UI)機能を提供してメンバーとUI406経由で対話する。UIを介した対話はたとえばキーボード156経由で(図1)またはマウス157経由で(図1)開始されたUIイベントを含む。このようなUIイベントに回答して、ツール424はこれに対応するエンジン426に対してデータ・モデル変更を実行するように要求し、データ変更が起こった時に非同期的にUIを更新する非同期データ変更通知をエンジン426に予約する。ツール424もコントローラ・サービス・マネージャ418の指示の下に提供されるバックグラウンドサービスと対話するためのアプリケーション・プログラム・インタフェース(API)を実装している。エンジン426はテレスペース404をサポートするデータの維持と変更、および/またはツールを介して得られたユーザとの対話からの結果の維持と変更を担当する。後述するように、エンジン426は永続的モデルデータを変更することと非同期データ変更通知をツール424に送出することをどちらもダイナミクス・マネージャ420の指示制御下で行なうことができる。格納・マネージャ416は格納されたデータへのアクセスを制御する。

【0060】

アクティビティ・テンプレートの作成について、ソフトウェア・デベロッパはフレームワーク内で使用するツールとエンジンを書いたり取り込んだりできる。アクティビティ・テンプレートはアクティビティを含むツール・コンポーネント及びエンジン・コンポーネントの永続的表現である。アクティビティ・テンプレートはたとえばシュリンク・ラップしたソフトウェアとして配布したりたとえばインターネット上で遠隔地のサーバからピア・ユニット314Aにダウンロードしたりできる。アクティビティ・コンポーネントはウェブ・ドキュメントであると考えることができ、URL経由で永続的に表現される。アクティビティ・テンプレートそれ自体がURLを有するのが望ましく、これによりアクティビティ設計の変更を追跡できるようになる。アクティビティ・テンプレートは単一のアクティビティ・テンプレートかまたはアクティビティ集合体テンプレートであり得る。単一のアクティビティ・テンプレートはたとえば「チャット」などの一つのアクティビティにだけに関連する。それに対してアクティビティ集合体テンプレートは「チャット及びアウトライン」などといったアクティビティの集合体に関連する。

【 0 0 6 1 】

使用するときには、ABCシステム400はアイデンティティ・マネージャ410経由でメンバーのアイデンティティを取得し、テレスペース・マネージャを開き、テレスペース・マネージャに対してURL経由でテレスペースを開くように要求し、テレスペース・マネージャにアクティビティ・マネージャを要求し、さらに、通常はアクティビティのURLを使って、アクティビティ・マネージャがアクティビティを開く。次に、ABCシステム400はメンバーがテレスペースを使用して特定のアクティビティによって提供される共有された特定のタスクを実行することができるようにする。

【 0 0 6 2 】

図5はピア・ユニット314A~D上のABCシステム500の代表的なマルチ・テレスペースの使用法を示す。ABCシステム500が分散システムであることは理解されるべきである。そこでここでは、ピア・ユニット314Aだけのコンポーネントの説明が提供されているが、ピア・ユニット314B~Dも同様のコンポーネントを備えているので、これらについては別個の説明は不要である。

【 0 0 6 3 】

ピア・ユニット314Aについて図示したように、ABCシステム500はピア・ユニット314Aがメンバーとして登録されている複数の代表的テレスペースを含み、これには自動車設計テレスペース502、チャット・テレスペース504、チェスで遊ぶテレスペース506が含まれている。また図示してあるように、ピア・ユニット314はテレスペース502、504、506の操作中に使用される複数のコンポーネントを有している。ピア・ユニット314Aは、メンバーとして登録されているテレスペース502、504、506の各々をリストし記述してあるテレスペース・レコード508と、これらのテレスペースに参加するテレスペース・メンバーのアイデンティティの各々をリストし記述してあるメンバー・レコード512とを保持するメモリ510を備えている。UI512はユーザが開始したテレスペースへの変更及びメンバー・レコード508、514への変更を受信できる。

【 0 0 6 4 】

テレスペース502、504、506の各々は図示したアクティビティ522、524、526のうちの異なる一つを例示している（これとは別の場合には、図示したアクティビティ522、524、526は単一のテレスペースに常駐することもあるが、この場合単一のテレスペースの全てのメンバーがアクティビティの各々へのアクセスを行なうことになる）。各々のアクティビティ522、524、526は各々のアクティビティに対応する特定のツール532、534、536であって、各々のエンジン542、544、546の制御下にメモリ510に格納された対応するデータ・モデル552、554、556に対して変更を行うためのツール532、534、536を含み、これらのツールはそれぞれテレスペース502、504、506の中に永続的に保存される。たとえば、自動車設計テレスペース502はCAD（computer-aided design）ソフトウェアの形でツール532を含むことができるアクティビティ522のインスタンスであり、チェスで遊ぶためのテレスペース506はソフトウェアのチェスゲームの形を取るツール536を含むことができるアクティビティ526のインスタンスである。

【 0 0 6 5 】

ユーザはユーザ・インタフェース514経由でテレスペース502、504、506のアクティビティ522、524、526を遂行し、ユーザ・インタフェース514はツール532、534、536とのインタフェースを持つことによりユーザの要求をツールに提供する。この要求に回答して、エンジン542、544、546がダイナミクス・マネージャ501の制御下でメモリ510内の対応するデータ・モデル552、554、556の永続状態を変更し、現在の状態を反映させる。たとえば、チェスで遊ぶためのテレスペース506でのデータの永続的状态は、そのチェス盤と、盤上のチェスの駒の位置を含むことができる。ユーザ要求はチェスの駒の動きを指定でき、これがデータの変更、即ちデルタを構成する。また現在の状態はデルタの実行と駒の動きの完了を反映する。

10

20

30

40

50

【 0 0 6 6 】

図6はABCシステム500の動作を示したもので、ピア・ユニット314Aに配置され図5を参照して説明したコンポーネントに関するものである。エンジン546はアクティビティ526の特定の集中タスクを推進する上で実装するときの候補となるアクションまたはオプション（たとえばチェスの駒の動き）をツール536に提示する。ツール536はユーザ・インタフェース514経由でテレスペース506のメンバー（メンバー・レコード512に記録されているメンバー）であるユーザと対話し、アクティビティのビュー（画像）を提示する（たとえば、チェス盤面のビューと、おそらくは動きのリスト）。ユーザのジェスチャーと呼ばれる（たとえばビショップをXからYへ動かすなど）ものであってユーザ・インタフェース514から入力されたユーザ入力に応答して、ツール536はデルタを生成してシステム内に格納することにより、ユーザによる選択をメモリ内に記録する。デルタは要求された変更の単位であり、好ましくは、コンテナ・オブジェクトの形である。コンテナ・オブジェクトはオブジェクト指向プログラミング技術に馴染みのある者にはには周知である。ツール536はデルタにおいて要求された変更の粒度（granularity）の決定を担当する。デルタ・コンテナ・オブジェクトはたとえば特定のチェスの駒の識別及びチェスの駒の動きを構成する盤面のもとの位置と目標位置といった、ユーザの指定したデータに加えて、変更を実行するための1つまたはそれ以上のコマンドを保持（内蔵）することができる。生成時には、デルタはコマンドが何も入っていない空の状態、変更を行なうためにエンジン546によって適当なコマンドを埋め込まれるまで、それ自身が単に変更要求を構成しているだけである。デルタにおいてエンジン特有の方法（これはツール536には分かっていると良い）で変更を行なうのに必要とされるコマンドを記録するのはエンジン546の責任である。

10

20

【 0 0 6 7 】

さらに詳しく説明すると、線「a」で示したように、ツール536はダイナミクス・マネージャ501によるデルタ生成を要求することからデルタの作成を開始する。ダイナミクス・マネージャ501はデルタを作成して線a'でツール536へデルタを返し、フォールトリカバリ（障害復旧）が必要な場合に備えて作成したデルタを記録する。

【 0 0 6 8 】

図6の線「a」で示してあるように、ツール536はツール・インタフェース602を介してエンジン546のツール・エンド604へデルタの制御を渡し、これによりエンジンを起動する（この説明は時としてデルタが渡されるまたはデルタへの制御を一つのコンポーネントから別のコンポーネントへ渡すべきであることを表わす。望ましくは、デルタはメモリ内に割り当てられ、ポイントがXMLエレメントとして渡されるオブジェクトである。）。エンジンのツール・インタフェース602はデルタにコーディングされている要求された変更を行なうことができるコマンドを作成するための、または、言い換えればユーザの意図にしたがうように逐次的な形で適当なコマンド、代表的にはエンジンに特有なコマンドでデルタを埋めるための、一組のアプリケーション・プログラム・インタフェースを表出する。次にエンジン546はこの段階ではコマンドで埋められているデルタの制御をツール536に戻す。

30

【 0 0 6 9 】

図6の線「b」で示してあるように、ツール536は実行のために埋められたデルタの制御をダイナミクス・マネージャ501へ渡す（即ち送出する）。ダイナミクス・マネージャ501は他のピア・ユニット314B～Dから線「b」経由で到着するデルタも受信することができる。ダイナミクス・マネージャ501は、ピア・ユニット314B～Dにおけるダイナミクス・マネージャとともに、チェスゲームのテレスペースに参加している全てのメンバーについて、ABCシステムの一方のエンドから別のエンドへ、データ・モデル556へ行なわれた変更の一貫性を維持する責任がある。

40

【 0 0 7 0 】

ダイナミクス・マネージャ501はキュー構造612とダイナモ・プロセス614とを含み、好ましくはコンピュータで実行可能なプログラムがダイナミクス・マネージャの主

50

論理を含む。ダイナミクス・マネージャ501とそのダイナモ・プロセス614は通信マネージャ622と双方向通信する。ダイナミクス・マネージャ501は受信したデルタを順序付ける目的でキューに入れてキュー構造612にして、次にダイナモ・プロセス614がキューに入っているデルタを処理するときにはキュー構造にサービスし、エンジン546に指示して適当な順序でデルタを実行させる。さらに詳しく説明すると、ダイナミクス・マネージャ501のダイナモ・プロセス614はインタフェース615を介して図6の線「c」を介してエンジン546のダイナミクス・マネージャ・エンド616へ向けて、「デルタ・コマンド実行」メッセージの形で制御信号を送信する。デルタコマンド実行メッセージはエンジン546に指示して、メモリ510に含まれており、テレスペースに永續するデータ・モデル556への要求された変更を行なうことで、ツール536から受信したデルタの実行を遂行する。

10

【0071】

ダイナミクス・マネージャ501のダイナモ・プロセス614はまた、通信マネージャ622経由で他のエンドポイントへ向けてローカルに開始した全デルタを配布し、通信マネージャ622はネットワークたとえば626上でピア・ユニット314B~Dへデルタを送信する。ピア・ユニット314B~D各々では、各々のダイナミクス・マネージャが受信したデルタを自分のローカル・キュー構造へとキュー入れし、つぎに特定の順序で各々のエンジンへデルタを渡して実行させる。

【0072】

一つのダイナミクス・マネージャはアプリケーションによって一つのテレスペースまたは多数のテレスペースをサポートできるので、多数のエンジンに指示して様々なテレスペース用のデルタを実行させる責任をとらせることもできる。多数のテレスペースを持つシステムでは、図5に図示したように、デルタの情報を調べることで、あるデルタをどのエンジンが実行すべきかを、ダイナミクス・マネージャ501が決定する。デルタはエンジンのうちの特定の一つに同定されまたデルタのペイロードの中に入っているエンジン識別子またはコードのタグを付けられたコマンドを含む。キューに入れられたデルタがダイナミクス・マネージャで処理される時に、ダイナミクス・マネージャはエンジン・コードと参照されたデルタとを関連付け、適当なエンジンによるデルタの実行を指示する。

20

【0073】

最後に、図6の線「d」で示されているように、エンジン546は「データ変更通知」メッセージを送出してツール536にデルタの実行を通知し、またデータ・モデル556におけるデータの新しい現在状態を通知する。デルタ変更通知は値または参照を使って行うことができる。つまり、通知は新しいデータの値それ自体を含むことができたり、またはメモリ内の新しいデータへのポインタを含むことができる。この通知に回答して、ツール536はたとえばチェス番のグラフィック上での動きを表示することにより、ユーザが利用可能な変更をユーザ・インタフェース514に行なわせる。ユーザがディスプレイ170を観察している場合(図1)、ユーザは、デルタの実行によって行なわれたチェス・ゲームへの変更を見ることができるが、それ以外の場合には、変更が行なわれた時点でユーザが変更を「見る」ことなくメモリ510内のデータに変更が行なわれる。

30

【0074】

望ましくは、ピア・ユニット314Aのデータ・モデル556に格納されているデータ・モデル556へ変更がエンジン546によって行なわれるのと実質的に同時に、他のピア・ユニット314B~Dもローカルに格納しているデータに対して同様の変更を行なって一貫した方法でデルタの意図を反映する。ピア・ユニット314A~Dのどれかがネットワークたとえばインターネットに接続されていないような状況が起こり得る。ピア・ユニット314Aが切断されているときには、通信マネージャ622はデバイス・プレゼンス・デテクタ430を使用してピア・ユニット314Aが接続されているかどうかを確認し、接続されていない場合には、接続が復元される時まで送出されるデルタを送出デルタストア628に格納しておき、接続された時点で、格納したデルタを送信する。1つ以上の宛先ピア・ユニット314B~Dがネットワークから切断されている場合、ピア・ユニ

40

50

ット314Aの通信マネージャ622はプロセスを続けて他のピア・ユニットが切断されているか否かにかかわらず送出デルタをリレー316(図3)に送信する。リレー・デバイス316(図3)はこのような送出デルタを受信し、必要に応じて送出デルタストア318(図3)に格納し、ピア・ユニット314B~Dが再接続された時点で宛先へ転送する。動作中には、通信マネージャ622は全ての送出デルタをリレー316のURLへ送信することができ、リレーは搭載されている参照テーブル320を使用してデルタを中継する場合に使用する宛先ピア・ユニット314B~DのエンドポイントURLを確認する。リレー・サービスは実際にはインターネット・サービス・プロバイダ(ISP)またはその他のインターネット組織によって提供され得るものである。

【0075】

ツール536とエンジン546の間で、ユーザ・インタフェース・アクティビティはツール536経由でのみ直接行なわれ、データ・モデルの変更はエンジン546経由でのみ直接行なわれることが理解される。専門用語を使った表現では、ツール536はUIを「保有」しており、エンジン546はデータ・モデルを「保有」している。これは図2に図示したような現行の代表的アプリケーションとは対照的である。たとえば、今日のスプレッドシート・プログラムでは、代表的にはツールの機能とエンジンの機能を組み合わせしており、本発明の前述の実施態様のようにこれらを分離するのではない。ツールをエンジンから分離することにより、従ってデータ・モデルからUIを分離することにより、ダイナミクス・マネージャはたとえばデータの一貫性を維持する目的で、様々なピア・ユニットから発せられたデルタの間に介在し仲介できる。さらに、エンジンからツールを切り離すことでデルタの実行処理を非同期的に行なえるようになる。これはダイナミクス・マネージャ501とエンジン546がテレスペースの中にある全てのメンバーについてデータの一貫性を維持するような方法で各々の責任を遂行するのに時間がかかるため有用である。また、他のピア・ユニット314B~Dから到着するデルタがローカルに開始されたデルタの実行のタイミングに影響することがあることからも有用である。

【0076】

図7は個別に実装されたアクティビティ(図示していない)で使用するフレームワーク700の実施態様を示す。フレームワークは、たとえばフレームワーク上で動作可能なアプリケーション独自のスナップイン(プラグイン)であるコンピュータ・プログラム製品として、個々のアクティビティとは独立して、販売またはライセンスすることができる。一般的に言うと、フレームワークはテレスペースに関してコラボレーションをホスティングする。図示したフレームワーク700は1つ以上のダイナミクス・マネージャ702、1つの通信マネージャ704、1つのデータ構造テンプレート706を含む。データ構造テンプレート706はアクティビティ・デベロッパが使用するための原始型オブジェクト(オブジェクト・プリミティブ)を、望ましくはXMLフォーマットで含むことができる。フレームワーク700はたとえばコンピュータで読み取り可能な媒体706として実装可能で、媒体上にダイナミクス・マネージャ702と通信マネージャ704を含むコンピュータで実行可能なコードと、データ構造テンプレート700のオブジェクト・プリミティブを含むコンピュータで読み取り可能なデータとが格納される。

【0077】

F. 通信マネージャ、プレゼンス・サーバ、及びリレー

図8はネットワークたとえばインターネット803経由で多数のピア・ユニット802A~Dの間での通信ができるように装備したABCシステム800を示す。各々のピア・ユニット802A~Dはピア・ユニット314Aに関連して前述したコンポーネントを含むことができる(図8に図示したピア・ユニットの個数は単に例示を目的として選択したものであり、特定の実装の構成により左右され、また時間とともに変化することがある)。各々のピア・ユニット802A~Dはツール805A~Dによって開始されエンジン807A~Dにより実行されるデルタ処理を調節するためのダイナミクス・マネージャ804A~Dを含み、またピア・ユニット802A~Dの間の通信を調整するための通信マネージャ806A~Dを含む。通信マネージャ806A~Dの役割はネットワークたとえば

10

20

30

40

50

インターネット803上でインターネット・プロトコルたとえばTCP/IPなどを使用してメッセージ(デルタを含む)を送受信することを含む。各々のピア・ユニット802A~Dはとくに機能の上で通信マネージャ830A~Dの動作サポートにおいてデータを格納するためのメモリ808A~Dを含む。

【0078】

前述のように、通信マネージャ806A~D各々はそれ自体のピア・ユニットと他のピア・ユニットとの間の通信を全て管理する責任がある。たとえば、通信マネージャ822はピア・ユニット802Aとピア・ユニット802B~Dの間の双方向通信を管理する。通信はピア・ユニット802Aからピア・ユニット802B~Dの1つまたはそれ以上へ送出される、またはピア・ユニット802B~Dの一つからピア・ユニット802Aへ送出されるどちらかのメッセージを含むことができる。宛先ピア・ユニット802B~D(前者の場合)またはピア・ユニット802A(後者の場合)がネットワークたとえばインターネットに接続されている限り(「オンライン」状態である)直接かつ仲介なしに送信を行なうことができる。前述のように、デバイス・プレゼンス・マネージャ812を用いて想定した宛先がオンラインになっているか、または一時的に切断されている(「オフライン」状態である)どうかを確認することができる。宛先がオフラインになっている場合、通信はリレー814(前述の通り)を経由して行なうことができ、リレーは宛先がオンラインに復帰した時に宛先へのメッセージを転送する。

10

【0079】

つまり、ピア・ユニット802Aがピア・ユニット802Cへのメッセージを送出しようとして、ピア・ユニット802Cがオンラインであるとプレゼンス・サーバ812から通信マネージャ806Aに通知されている場合には、通信マネージャ806Aはネットワークたとえばインターネット803経由で直接ピア・ユニット802CのURL宛てにメッセージを送出する。ピア・ユニット802Cでは通信マネージャ806Cがメッセージを受信して、メッセージをダイナミクス・マネージャ804Cへ渡す。

20

【0080】

逆に、ピア・ユニット802Aがピア・ユニット802Cへのメッセージを送出しようとして、ピア・ユニット802Cがオフラインであるとプレゼンス・サーバ812から通信マネージャ806Aに通知されている場合には、通信マネージャ806Aはネットワークたとえばインターネット803経由でリレー814のURLへメッセージを送出する。リレー814は高性能ファイルサーバとして実装可能で、ピア・ユニット802Cがオンラインに復帰するまで中継メッセージを保存しておき、復帰時にメッセージをピア・ユニット802CのURLへ転送する。ピア・ユニット802Cでは前述のようにメッセージを受信して、通信マネージャ806C経由でダイナミクス・マネージャ804Cへ渡す。

30

【0081】

したがって、プレゼンス・サーバ812はABCシステム800内ではピア・ユニット802A~Dがオンラインかどうかをモニタする目的で、また他のピア・ユニットの接続状態をピア・ユニットの各々に通知する目的で、使用することができる。この目的を満たすためには、プレゼンス・サーバ812はたとえば図1のコンピュータ・システム100に示したように構成することができる。

40

【0082】

動作中、ピア・ユニット802A~Dの各々はプレゼンス・サーバ812に自分の接続状態を通知する責任、即ちユニットがオンライン中かまたはこれからオフラインになるとしてるところかを通知する責任がある。別の方法としては、プレゼンス・サーバ812は時々ピア・ユニット802A~Dをポーリングしてこれらの接続状態の情報を取得することができる。この情報はプレゼンス・サーバ812内で望ましくは揮発性メモリ816内の接続ディレクトリ818に保存する。接続ディレクトリ818は初期状態では空で、ピア・ユニット802A~Dが接続状態の状態をプレゼンス・サーバ812に提供するまで、たとえば接続状態の通知を提供するまで、空のままである。そして、状態に関する情報の受信時に、接続ディレクトリ818は通知してきたピア・ユニットのURLと接続

50

状態の情報を保存する。

【 0 0 8 3 】

プレゼンス・サーバ 8 1 2 はピア・ユニット 8 0 2 A ~ D に対して接続状態予約サービスを提供することもできる。プレゼンス・サーバ 8 1 2 は各々の接続しているピア・ユニット 8 0 2 A ~ D に他の各ピア・ユニット 8 0 2 A ~ D の接続状態を通知し接続状態の変化も通知する。プレゼンス・サーバ 8 1 2 は (a) メッセージを送信「したい」ピア・ユニット 8 0 2 A ~ D から状態変化通知の要求があった時点で、(b) ネットワークたとえばインターネット 8 0 3 へピア・ユニットが接続する時点で、(c) 時々、たとえばピア・ユニットのいずれかの状態に変化があった時点で、または (d) 上記の組み合わせのいずれかの時点で、接続しているピア・ユニット 8 0 2 A ~ D へ接続状態通知を送信できる。望ましくは、全ての通信はインターネット・プロトコルに準拠して行なう。これらのプロトコルにはとくに前述のものが含まれる。

10

【 0 0 8 4 】

各ピア・ユニット 8 0 2 A ~ D はまたメンバーシップ状態サービスに加入することができ、そうすることでそれ以後はメンバーとなっている各テレスペースのメンバーシップの現状に基づいて知ることができる。このサービスを有効にするためには、プレゼンス・サーバ 8 1 2 はプレゼンス・サーバによってサポートされる各メンバーについてのレコードを格納するため揮発性メモリ 8 1 6 にメンバー・ディレクトリ 8 2 2 を保持しておく。各ピア・ユニットはプレゼンス・サーバに時々、あるいは、これ以外ではメンバーシップ状態に何らかの変更があった時点で、現在のメンバーシップ状態を提供する。

20

【 0 0 8 5 】

例示すると、ピア・ユニット 8 0 2 A がピア・ユニット 8 0 2 B へたとえばデルタなどのメッセージを送信したい場合、ピア・ユニット 8 0 2 A はプレゼンス・サーバ 4 3 0 にアクセスしてピア・ユニット 8 0 2 B の URL を取得し、これの接続状態の情報を得る。ピア・ユニット 8 0 2 A が予約サービスに加入していると仮定すると、ピア・ユニット 8 0 2 B の状態についての通知を受信することになる。つまりダイナミクス・マネージャ 8 0 4 A は宛先ピア・ユニット 8 0 2 B がネットワークに接続しているかどうかについての知識がなくとも通信マネージャ 8 0 6 A へメッセージを渡すことができる。

【 0 0 8 6 】

G . 分散データ一貫性モデルとデルタ処理

前述したように、ピア・ユニット 8 0 2 A ~ D 内のダイナミクス・マネージャ 8 0 4 A ~ D はローカルに生成したデルタやリモートで生成したデルタを受信することと各々のエンジン 8 0 7 A ~ D で実行する正しい順序を維持することで、テレスペースに参加している A B C システム 8 0 0 のピア・ユニット 3 1 4 A ~ D の全部にわたってデータ・モデルの一貫性を高いレベルで維持できるようにすることが役割である。どれか一つのピア・ユニットから発信されたデルタが順不同に受信および / または実行されたり、または一時的にネットワークからピア・ユニットが切断されているため大幅に遅延した後で受信および / または実行されるため、こうすることは一層困難なものになる。さらに、デルタは別々で一貫性のないローカルデータの状態に基づいて生成されることがある。

30

【 0 0 8 7 】

理論的には、データの一貫性は 3 つの属性を使って記述することができ、その各々は何らかの程度でコラボレーション・システムである程度達成できるものであり、それらは即ち因果関係の保存、意図の保存、および収束性である。さらに詳しく説明すると、これらの一貫性の属性は以下のように記述することができる。

40

【 0 0 8 8 】

1 . 因果関係の保存 (即ちシーケンス (順序) の一貫性) は、デルタ A がデルタ B の原因となる場合、デルタ A は最初に行われる必要があり、デルタの作成を生じたシーケンスがその実行において保存されるようにすることが必要とされる。言い換えれば、この属性を満たすには従属動作の実行順序が自然界の原因 / 結果の順序と同じである必要がある。

50

【 0 0 8 9 】

2. 意図の保存（即ちセマンティック（意味）上の一貫性）はデルタの意味とその作成の裏に隠れている意図が保存される必要がある。言い換えれば、この属性を満たすには、デルタの何らかの対について、たとえばデルタAとデルタBで、一方の実行は他方の意図を保存しなければならない。これを達成するには、データ状態に対する変更をユーザが要求した時点でのデータ状態をシステムで記録する。たとえば、共有ドキュメントの場合、変更「x」が要求された時、次の単語が「y」でありこれは「x」が実行された時にも同じである。言い換えれば、各ピア・ユニットに格納されたデータに対してデルタを実行する効果が元となるピア・ユニットに格納されたデータに対する同じデルタの実行の効果と同じでなければならない。

10

【 0 0 9 0 】

3. 収束性は全てのピア・ユニット802A~Dの各々によって維持されるデータのコピーは同じデルタを実行した後に同一でなければならないことを要求する。これは「完全順序」（total ordering）として公知である。

【 0 0 9 1 】

ABCシステム800は高レベルのデータ一貫性を維持するのに必要な情報を記録しておき、実行が他のデルタの先行実行に依存するデルタについてのデルタ実行順序の制約を強制する。たとえば、ピア・ユニット802Aのダイナミクス・マネージャ804Aでのデルタ受信の順序毎にデルタが大文字A、B、C、Dで指定されている時、デルタA、B、C、Dは因果関係を保存するように受け取った順序で正しく実行することが可能である。これは、たとえばデルタAとCがピア・ユニット802Aからローカルに開始されたものでデルタBとDがピア・ユニット802Dからリモートで到着したものである場合でも同じである。ただし、場合によっては、受信した順序でこれらのデルタを実行すると、順序が乱れた変更が行なわれて因果関係の違反が起こることになる。たとえばローカルに生成されたデルタとリモートで生成されたデルタがたどる通信経路でのレイテンシ（待ち時間latencies）が異なっているため、デルタが交互に入れ違いになって受信される結果を招くことがある。さらに、因果関係の保存が実現されたとしても、結果は意図の保存に関する属性を侵害することがあり、変更の意味がなくなる場合もある。たとえば、デルタBは文章「A cow has legs」（牛には脚がある）について「脚」の前に「4本の」を挿入したいとする。デルタAがデルタBでの変更について知識がなく、同じ文章の中で「脚」を「頭」に変更したい場合にデルタBで文脈を変更するためにデルタBの前に実行されてしまうと、結果は「A cow has four a head」（牛には4つの頭がある）になる。実際、この結果は意味がなく、またデルタAがデルタBより後に処理されることにより発生する結果にも意味がないのである。収束性については、ピア・ユニット802A~Dでのデータのコピーの「同一性」（即ち複製）が破られる状況には複数のものがある。たとえば、ネットワーク上の通信障害によりデルタがあるピア・ユニット802A~Dで受信されなくなることがあり、そのために異なるデルタ実行結果が発生する場合がある。不可能でなくとも現実のシステムにおける完全収束、因果関係の完全保存、意図の完全保存を実現するのは困難であるが、高いレベルのデータ一貫性がピア・ユニット間えのコラボレーションには望ましい。

20

30

40

【 0 0 9 2 】

高レベルのデータ一貫性を実現するためには、各々のダイナミクス・マネージャ804A~Dは、所定のフォーマットのデルタを作成し、デルタと各々のメモリにデータ一貫性を維持するのに使用するある種の情報を記録するのが望ましい。さらに詳しく説明すると、各ダイナミクス・マネージャ804A~Dが作成する各々のデルタにシーケンス番号（Seq. No.）を割り当て、特定のピア・ユニット802A~Dにおいてダイナミクス・マネージャによって作成された全デルタの中でのデルタのシーケンシャルな位置を表わす。ダイナミクス・マネージャ804A~Dは次にデルタに割り当てたシーケンス番号と一緒にエンドポイント識別子（EP）を記録し、これはデルタの発生の元となったエンドポイント（ピア・ユニットと人）を指定しているものである。ダイナミクス・マネージャ

50

ヤ 804A ~ D はまたデルタに作成時刻で単一の従属性を指定する情報を記録する。デルタに記録された従属性情報は、たとえば他の一つのデルタの発生元となるエンドポイントのシーケンス番号と EP の形態を取るが、このときの他の一つのデルタは、たとえば、通常はそのデルタについて作成したピア・ユニットで注目しているデルタと同じアクティビティで注目しているデルタの作成の直前にエンジン 807A ~ D によって実行された最後のデルタである。

【0093】

ある種のデルタは、たとえばテレスペース・データの状態がその実行に重要でないため、先行するデルタに依存しない。これらのデルタは「独立デルタ」と呼ばれる。独立デルタでは、このような従属性シーケンス番号を提供する必要がない。各ダイナミクス・マネージャ 804A ~ D は受信したデルタのヘッダから一貫性情報を抽出し、デルタ情報の残りと一緒にこの情報をメモリに保持する。

10

【0094】

図 9 は本発明の実施態様によるデルタ 900 のフォーマットを示しており、これは上記で説明した一貫性情報を使うことができるものである。デルタ 900 はヘッダ 902 と、これに続くペイロード 904 を持っている。ヘッダ 902 はデルタを識別し、一貫性情報を含み、他の「ハウスキーピング (house-keeping)」情報を含むこともある。ペイロード 904 は 1 つ以上のコマンド・フィールド 908 を有し、これがエンジン 807A ~ D の内の一つによってデルタに記録されたコマンドを指定する。

【0095】

ヘッダ 902 は一貫性情報、たとえば次のようなものを指定する：

Seq. No. @ EP No. : Seq. No. @ EP No.

20

【0096】

ここで、最初の Seq. No. @ EP No. (コロンの前のもの) は、発生元となるピア・ユニットでの、即ち URL で識別されたピア・ユニット 314A での、現在のデルタのシーケンス番号であり、コロンは「~ が以下に依存する」と読むものであり、コロンの後の第 2 の Seq. No. @ EP No. は従属しているシーケンス番号であって、これはたとえば同じピア・ユニットたとえば URL で指定されたピア・ユニット 314A で生成された直前のデルタを指定するもので、ここで現在のデルタが従属する先のデルタを参照する。

30

【0097】

図 10 はデルタからたとえばダイナミクス・マネージャ 804A によって抽出された情報を格納するための代表的なメモリ構造 1000 を示している。ダイナミクス・マネージャ 804A は次のようなレコードを永続的に保持する：即ち、A) デルタ・ログ 1002 であり、これはテレスペース内部でダイナミクス・マネージャによって実行されたデルタの全部の記録、即ちデルタ 1、デルタ 2、... デルタ n の記録であり、n はテレスペースにおいてダイナミクス・マネージャ 804A により一番最後に実行されたデルタに対応するカウントであり、B) デルタ・格納 1004 であり、これはこのようなデルタの内容を格納し、またたとえばデルタ・ログ 1002 に実装できるものであり、C) ベクトル 1006 であり、これはピア・ユニット 802A のダイナミクス・マネージャ 804A がテレスペースの他のピア・ユニット 802B ~ D 各々から受信したデルタの最大シーケンス番号とそのテレスペースについてダイナミクス・マネージャ 804A で生成された最大のシーケンス番号を指定するアレイであり、D) エンドポイント (EP) ログ 1008 であり、これはピア・ユニット 314A もメンバーに加入している先のテレスペースのメンバーであるエンドポイントのリストで、各々のエンドポイント識別子を提供するものであり、E) データ・モデル空間 1010 であり、これは各々のアクティビティにおいて使用するアクティビティ特有のデータ・モデルに準拠したデータ用の格納を提供するものであり、F) ローカル・デルタ・キュー 1012 であり、G) 到着デルタ・キュー 1014 であり、H) ホールド・デルタ・キュー 1016 である。

40

【0098】

50

ベクトル1006は高いレベルのデータ一貫性を維持する際に使用される情報を格納するための観点からして重要であることからして、さらに説明を行なう。ベクトル1006はたとえば各受信データに関して次のような一貫性情報を保存する：即ち、(a)発生元のエンドポイントの識別子(UID)、これはテレスペース全体から見てユニークなものでそのエンドポイントの識別である。(b)エンドポイントから見たときの相対的なシーケンス番号、これは各エンドポイントから受信したテレスペース内のデルタのその時点でのカウントで、基本的には、受信デルタの順序付けの目的を有するものでピアであるエンドポイントのタイムスタンプに相当する。(c)デルタの従属シーケンス番号、これはエンドポイントがデルタを作成した時点で、そのピア・ユニットで実行されたシーケンスにおける(即ち「ユーザが見た」現在状態)何らかの先行デルタに基づいていたことを示す。好適実施態様においては、これらの情報の全部が受信デルタから抽出できる。

10

【0099】

動作中には、たとえばダイナミクス・マネージャ804Aによってデルタを受信した場合、ダイナミクス・マネージャは受信デルタがローカルに発生したかリモートで発生したかによって、受信デルタをローカル・デルタ・キュー1012又は到着デルタ・キュー1014のどちらか一方に置く。実装によってはしるべきキューの中のデルタの各々へのポイントだけを格納することができる。デルタ・キュー1012、1014はたとえばFIFO(先入れ先出し)ワークロード・バッファとして実装される(別の方法では、ある種の実装では到着デルタとローカル・デルタの両方を保持するために単一のキューを使用することができる)。ダイナミクス・マネージャ804Aはキュー1012、1014から先入れ先出し方式でデルタを処理する。実装によってはたとえば負荷のバランスを取るためおよび/またはレイテンシを制御するために、ローカル・データ又は到着デルタのキュー1012、1014の一方又は他方からのデルタを処理するための適当な優先順位アルゴリズムを使用することができる。ローカル・キューにあるローカルデルタより先に到着キューにある全ての到着デルタを処理してデルタ処理に置ける平均的レイテンシを制限するのが望ましい。別の方法では、ローカルデルタが優先的に処理される逆の順序の場合にはピア・ユニットがネットワークへ再接続した時に到着デルタ・キューが長くなる場合に長いレイテンシを回避できる。

20

【0100】

図11Aはデルタ処理メソッド1100の代表的実装を図示したもので、前述の順位付け及びデータ一貫性情報を使用する。ステップ1102において、ダイナミクス・マネージャ804Aは次の、又は「現在の」デルタを、ローカル・デルタ・キュー1012と到着デルタ・キュー1014の一方から取り出す。ステップ1106で、ダイナミクス・マネージャ804Aはベクトル1006をチェックし、シーケンス及び従属性情報から、現在のデルタが独立かどうか、また対応する従属性シーケンス番号を持っていないかどうかを決定する。そうだとした場合には、ステップ1108で、図11Bのサブルーチン1110を呼び出すかまたは起動することにより、受信した順序で現在のデルタをすぐに処理する。サブルーチン1110のステップ1112で、ダイナミクス・マネージャ804Aは、デルタのUID、シーケンス番号、従属シーケンス番号をデルタ・ログに記録することにより、デルタ・ログへデルタを登録または同化する。次に、ステップ1114がデルタを実行させる。その後、ステップ1116ではタイムスタンプ・ベクトル1006を更新する。そのため、ダイナミクス・マネージャ804Aはヘッダ情報を抽出し、ベクトル1006の中のそれに対応するピア・ユニットのエンドポイント番号と最大シーケンス番号を更新する。次に、サブルーチンは、ステップ1128に戻って図の11Aのデルタ処理を継続する。ステップ1106のテストで、現在のデルタが独立ではないと判断した場合、ステップ1118で、ダイナミクス・マネージャ804Aはベクトル1006の従属シーケンス番号(デルタの先頭になっているもので指定されている)をチェックし、現在のデルタが従属している先のデルタが処理されているかどうか判断する。対応するエンドポイントについてシーケンス番号がベクトル1006に格納されている番号より小さければ、これはすでに処理されている(理論的にはこの情報をデルタ・ログに格納してもい

30

40

50

いのだが、デルタログはそのサイズを現実的な範囲内に保つため時々ページするのが望ましい)。未だ処理されていない場合には、ステップ 1 1 2 2 で、現在のデルタはホールド・デルタ・キュー 1 0 1 6 に置かれる。現在のデルタの従属先であるデルタがすでに処理されている場合には、現在のデルタは以下で詳細に説明するように処理することができる。

【 0 1 0 1 】

ステップ 1 1 2 4 で、ダイナミクス・マネージャ 8 0 4 A はデルタ処理の順序を決定し、前述したように図 1 1 B のサブルーチン 1 1 1 0 を次に関連させる。デルタ・ログ 1 0 0 2 のデルタ・シーケンス番号で示されて判るように現在のデルタがすでに実行されたデルタに従属していて、同じ優先順位のデルタに従属する受信デルタが他にない場合、それ以上の一貫性を心配することもなく、現在のデルタはすぐに処理される。一方、現在のデルタがデルタ・ログ 1 0 0 2 に登録されている別の係属デルタと同じ従属シーケンス番号即ち Seq. No. @ EP を持っている場合には、これらのデルタの間で衝突条件が発生する。

10

【 0 1 0 2 】

本実施態様においては、衝突の発生時におけるデルタの実行順序はデルタの発生元のエンドポイントに依存するが、他の実装ではこれ以外に別の優先順位スキーマまたは初期設定順序を使用して衝突を解決することができる。実装されたときには、テレスペースのメンバーであるエンドポイントには（たとえばテレスペースの第 1 のエンドポイント・メンバーによって）それぞれユニークなエンドポイント番号が割り当てられ、この番号がたとえばエンドポイントがもともとテレスペースに参加した順序に対応する。つまり、ピア・ユニット 3 1 4 A がそのテレスペースで最初のものであったとすると、これには EP 1 と番号が付けられ、それに続く新しいメンバーは階層的番号スキーマにしたがって番号を受信して、たとえばテレスペースへ新しいメンバーを招待したスポンサーであるエンドポイントを反映する。つまり、エンドポイント EP 1 が新しく 2 人のメンバーをテレスペースへ招待した場合、これらは EP 1 . 1 と EP 1 . 2 として番号が付けられる。次に、エンドポイント EP 1 . 1 が新しく 3 人のメンバーを招待して加入させた場合、新しいメンバーは EP 1 . 1 . 1、EP 1 . 1 . 2、EP 1 . 1 . 3 と番号が付けられる。エンドポイント番号の桁は左から右へ第一位、第二位、などと表わすことができる。つまり、EP 1 . 1 . 3 では、一番左側の番号「1」が第一位、二番目の位置の桁の「1」が第二位であり、「3」が第三位である。

20

30

【 0 1 0 3 】

前述の番号付けスキーマは、ピア・ツー・ピアで、完全分散型コラボレーション・システムに使用するエンドポイントへユニークな番号を割り当てるのに特に適した特別な性質を有しているので、メンバーは何時でもメンバーシップに加入したり脱退することができる。または何時でもテレスペースに接続したりテレスペースから一時的に切断したりできる。前述のように、このスキーマはテレスペースメンバーの各々の各エンドポイントにユニークな番号を割り当てる。あるメンバーの各番号はそのメンバーがテレスペースに参加した順序を表わし、またまた別のテレスペースメンバー（「招待する側のメンバー」）によって加入するように招待された（または何らかの支援された）各メンバーについては、招待した側のメンバーも表わしている。これらの新しいメンバーによってさらに多くのメンバーに加入するように招待すると、割り当て番号はさらに桁数を追加して招待メンバーの家系図または連鎖を表わすようになる。したがって、割り当て番号はテレスペースの創始メンバー以後にテレスペースへ追加された各メンバーの家系図 (ancestry) を表わすものと考えられる。つまり、割り当て番号の数字が複数の場合には、第一位はテレスペースの創始メンバーを表わし、第一位以降の各位（即ち第二位、第三位など）は先行する位の数字で指定される招待する側のメンバーによってテレスペースへの参加を招待されたメンバーを表わしている。招待する側のメンバーはテレスペースの創始メンバーであったり、または中程度に古参の別のメンバーでテレスペース・メンバーシップを後援する特権があるメンバーであったりする。一つの実施態様においては、各々招待する側のメンバー

40

50

はそれが後援するメンバーにテレスペース番号を割り当てる責任がある。

【 0 1 0 4 】

このような番号付けスキーマだと、図 1 1 A のステップ 1 1 2 6 で、ダイナミクス・マネージャ 3 0 4 A は従属性の衝突を解決し、完全な順位を維持する。図 1 2 に図示しているように、従属性衝突の解決を行なうのは、各位毎の数字によって E P 番号を比較するステップ 1 2 0 2 と、その比較に基づいて処理順位を決定するステップ 1 2 0 4 で、最低の位に最低の番号を持ったデルタが最初に処理されるように順序が決められる。次に、ステップ 1 2 0 6 では、デルタのどれかが順序を外れて先に処理されていないかを確認する。順序から外れて処理されたデルタがない場合には、ステップ 1 2 0 8 で次のデルタを処理する。一方、順序を外れて処理されたものがある場合は、ステップ 1 2 1 0 でそのデルタの実行の直前のデータ状態まで処理を戻してから、ステップ 1 2 1 2 で正しい順序でデルタの処理を「やり直し」する。

10

【 0 1 0 5 】

このように、この従属性衝突を解決するスキーマではたとえば第一位に「1」のある全てのエンドポイント番号が第一位に「2」や「3」またはもっと大きな値を取るエンドポイント番号より先行して処理されることを保証する。デルタが第一位で同じ番号の値を持っている時には、順位が特定されるまで第二位の数字を比較するなどして、「実行」または必要なら「取消」「再実行」動作をデルタに対して行なうことで、デルタ実行の順位を明らかにする。

【 0 1 0 6 】

たとえば、デルタ・ログ 1 0 0 2 が例示の目的で次のような形のデルタ及び関連情報を含んでいたとする

```
" Seq. No. AA @ EP. No. BB : Seq. No. CC
  @ EP. No. DD "
```

これはエンドポイント番号 BB にあるシーケンス番号 AA がエンドポイント番号 DD のシーケンス番号 CC に従属している」と読みとることができる。本明細書では例示の目的でデルタは A から C までとして表わし、その指定にあわせてアルファベット順で受信されるものとする。デルタ・ログ 1 0 0 2 は次のようになる：

```
Delta " A " - - Seq. No. 1 @ EP 1.2 : Seq.
No. 0 @ EP 1
```

30

```
Delta " B " - - Seq. No. 2 @ EP 1.2 : Seq.
No. 1 @ EP 1.2
```

```
Delta " C " - - Seq. No. 1 @ EP 1 : Seq. No.
. 1 @ EP 1.2
```

【 0 1 0 7 】

この例では、デルタ A はエンドポイント 1 . 2 からのシーケンス番号 1 を持っており、これはエンドポイント 1 からのシーケンス番号 0 に従属している。デルタ B と C の両方も同じエンドポイントからの同じデルタに従属しており、そのデルタと言うのはデルタ A である。デルタ B はエンドポイント 1 . 2 で生成されたもので、デルタ C はエンドポイント 1 で生成されたものである。本実施態様によれば、3 つのデルタ全てが従属デルタであり、これらは次のような順序で処理される。最初にデルタ A が処理されるが、これは受信したこれらのデルタの最初のものであったためである。またしたがってダイナミクス・マネージャは適当なエンジンにデルタ A によって指定されたデータ変更を実行させる。次に、デルタ B (次に受信したデルタ) が処理され、適当なエンジンにデルタ B により指定されたデータ変更を実行させる。次いで、ダイナミクス・マネージャはデルタ C を処理し、デルタ B と従属性の衝突があること、またデルタ C はデルタ B (即ち E P 1 . 2) より小さいエンドポイント番号 (即ち E P 1) を持っていることが判る。その結果、デルタ B は「取り消される」ことになり、またはさらに詳しく説明すると、格納されているデータに対して行なわれたデータ変更が破棄されて、そのデータ変更を実行する前の状態にまでデータを復元または戻すようにする。次に、デルタ C が処理されデルタ C によって指定され

40

50

たデータ変更が行なわれる。最後に、デルタ B が処理されてデータ変更がもう一度行なわれる。このように、この例での要求されたデータ変更の実行順序は、復元とやり直しの後で、デルタ A、次にデルタ C、そしてデルタ B の順序となる。

【 0 1 0 8 】

前述の例は第一位と第二位の番号だけのあるエンドポイント番号に関係したもののだが、例示した原理はさらに多くのもっと大きい次数を持った他のエンドポイント番号にも等しく良く適用され、前述のように大きな次数の比較が衝突の解決には必要である。したがって、従属性の衝突があった際の優先順位は、前述のように、メンバーシップについてのスポンサーシップの順位、そしてメンバーシップの日付に基づく年代順で割り当てられる。

【 0 1 0 9 】

別の方法としては、従属性の衝突時の優先順位はエンドポイントの特性（たとえば業務のタイトル、ワークグループ、または部門）、要求された変更の性質（たとえばパラグラフの消去または章の追加）、データ中の変更位置（たとえばドキュメント・タイトルの変更またはドキュメントの要約の変更）、またはその他の根拠に基づいたものでも良い。優先順位を付与するためのこうした基準の各々の裏にある原理を例示するには、衝突の発生時には、たとえば何らかの特性を備えたエンドポイントからのデルタを、他の特性を備えたものより先に、たとえば製品設計チームからのデルタなどをマーケティング・チームからのデルタより先に処理する、またはその逆を行なうことができる。

【 0 1 1 0 】

さらに、データ変更のロールバック（変更取り消し）ややり直し（時として UNDO および REDO と呼ばれる）はオーバーヘッドの観点では高価であるから、実装によってはデルタ処理の優先順位決についての別の形態を提供することが望ましいこともある。実装する場合には、たとえばアクティビティによって他の優先順位決定方法を提供することがある。ある種のアクティビティは衝突解決または優先順位の決定方法が一種類であることがあり、また別のアクティビティは別のタイプのもを備えることがあるが、これはたとえば特定のアクティビティについてのデータ一貫性の重要性に依存することがある。チャット・ルームのようなリアルタイムのアクティビティでは、応答時間を短縮するために変更の取り消しとやり直しを避けることがあるが、これはとくに実行順序があまり重要ではないためである。

【 0 1 1 1 】

他の実装では、デルタの実行を、たとえば編集中のドキュメントのある部分に対する変更は何らかの形で関連するデルタのクラスターの実行、またはある種類の変更に何らかの形で関連するデルタのクラスターの実行とすることで、ほぼ同時に全ての変更を行なえるようにすることがある。さらに、ベクトル 1 0 0 6 に格納されたエンドポイントについての最大シーケンス番号より所定の値だけ若いシーケンス番号を持っているような過剰に順番が遅くなったデルタはやり直しが多すぎるので、スラッシング (thrashing) を防止するため実装によっては破棄されることがある。

【 0 1 1 2 】

図 1 1 A に戻って、ステップ 1 1 2 8 では、ダイナミクス・マネージャ 3 0 4 A は現在のデルタの優先処理に依存しているためこれまでに処理されていなかったデータがホール
ド・デルタ・キュー 1 0 1 6 にあるかどうかを判定する。このようなデルタがある場合、メソッドはステップ 1 1 2 4 に戻って処理を行なう。このようなデルタがステップ 1 1 2 4、1 1 2 6、1 1 2 8 のループで全て処理された後、メソッド 1 1 0 0 はステップ 1 1 0 2 に戻る。

【 0 1 1 3 】

データの一貫性を最適化するためのデルタ処理の前述の方法は本明細書で説明しているような実装でとくに有利であり、その場合には、A) ピア・ユニットはネットワークから一時的に切断されることがあるからローカル・デルタが順序を外れて処理されるような長い遅延の後までデルタを受信しないことがある、B) エンドポイントはメンバーシップのリストに動的に加入したり脱退したりすることがある、C) データの一貫性に関するオー

10

20

30

40

50

バヘッドは最小限に押えられピア・ユニット間に分散される。

【0114】

さらに別のエンドポイント・ナンバリング・スキーマは本発明の特定の実施態様において有利に用いることができる。たとえば、擬似乱数ジェネレータを使ってユニークなテレスペースメンバー識別子を割り当てるようなスキーマを実現できる。乱数ジェネレータが十分にランダムな数を生成しシード（種）が充分ユニークなものであれば一意性を保証できる。たとえば、ネットワーク・アダプタのMACアドレスをジェネレータのシードとして使用し32バイト値を作成することでテレスペース内に限らずABCシステム内でも一意性を保証できる。

【0115】

アクティビティを構成する各ツール・エンジンの組み合わせが別々のダイナミクス・マネージャに対応できることと、多数のダイナミクス・マネージャを提供してた数の同時に実行するアクティビティに対応できる点で、ABCシステムはマルチタスク、マルチスレッドにすることもできる。テレスペースとダイナミクス・マネージャの間の一対一の関係ではデルタ処理で最小の遅延が導入されるが、スレッドが窮乏する（starvation）とエラー状態またはフォールト状態を起してしまうことがある。他方、単一のダイナミクス・マネージャで多くのテレスペースをサポートすると渋滞（congestion）の問題が起こることがある。各コンポーネントの最適な個数はハードウェアの用途、参加するピア・ユニットの個数、トラフィック条件によって左右される。

【0116】

ABCシステムは複数のコンピュータ・システム（CS）を含み、その各々が1つ以上のテレスペース（TS）を備え、各テレスペースが1つ以上のアクティビティ・コンポーネントを具現化（instantiation）し、各アクティビティ・コンポーネントには1つ以上のツール（T）とエンジン（E）のペアを有し、これら各々の動作はダイナミクス・マネージャ（DM）によって調整される。つまり、ABCシステムは1つ以上のダイナミクス・マネージャを備えた多数のテレスペースを有することができる。したがって、ABCシステムはたとえば次のように構成することができる：

ABCシステム =

$$\begin{aligned} CS_1 &= DM_1 + A_1, & \text{ここで } A_1 &= T_1 + E_1 \\ CS_2 &= DM_2 + A_2, & \text{ここで } A_2 &= T_2 + E_2 \\ &\dots & & \\ &\dots & & \\ CS_n &= DM_n + A_n, & \text{ここで } A_n &= T_n + E_n \end{aligned}$$

【0117】

ここで全部のアクティビティ A_1 、 A_2 、 A_n は1つ以上のテレスペースに含まれるものとし、「 n 」は正の整数である。つまり、例を挙げると、様々なテレスペースのインスタンスエシジョンの例は次のようになる：

$$\begin{aligned} TS & A_1, & \text{または} \\ TS & A_1 + A_2, & \text{または} \\ TS & A_1 + A_2 + A_m \end{aligned}$$

あるいは

$$\begin{aligned} TS_1 & A_1 \\ TS_2 & A_2 \\ &\dots \\ TS_m & A_m \end{aligned}$$

【0118】

ここで記号「 $\{ \}$ 」は「 \sim を含む集合」を表わし、「 m 」は正の整数である。したがって、各テレスペースは1つまたはそれ以上のアクティビティを具現化し、フレームワークは単一のテレスペースを有することができる。別の方法では、各テレスペースは1つ以上のアクティビティを具現化し、フレームワークが多数のテレスペースを有することができる

10

20

30

40

50

【0119】

前述した実施態様のコンポーネントをソフトウェアで実装したものは、たとえばコンピュータで読み取り可能な媒体、たとえば図1のディスク142、CD-ROM147、ROM115、またはハードディスク装置152などの有形の媒体上に固定されるか、またはモデムまたはその他のインタフェース・デバイスたとえば媒体191上のネットワーク195に接続した通信アダプタ190を介して送信することができるかのどちらかのコンピュータ用命令及びルーチンを含む。媒体191は光通信回線または有線通信回線を含みこれに制限されない有形の媒体であるか、またはマイクロ波、赤外線、またはその他の通信技術を含みこれに制限されない無線技術で実現される。これはまたインターネットであってても良い。一連のコンピュータ用命令が本明細書で本発明に関連して前述し多機能の全部または一部を実現する。多くのコンピュータ・アーキテクチャまたはオペレーティング・システムで使用される多数のプログラミング言語でこのようなコンピュータ用命令を書くことができることは当業者には理解されよう。さらに、このような命令は現在または将来の、半導体、磁気、光、またはその他のメモリ・デバイスを含みこれに制限されない何らかのメモリ技術を用いて保存することができ、あるいは現在または将来の、光、赤外線、マイクロ波、またはその他の送信技術を含みこれに制限されない何らかの通信技術を用いて送信される。このようなコンピュータ・プログラム製品が、印刷したまたは電子化された説明を伴ったリムーバブル媒体、たとえばシュリンクラップ・ソフトウェアとして配布される、コンピュータ・システムのたとえばシステムのROMまたは固定ディスク上にプリロードされる、またはたとえばインターネットまたはWWW(World Wide Web)などのネットワーク上のサーバまたは電子掲示板システムから配布されることを想定している。

10

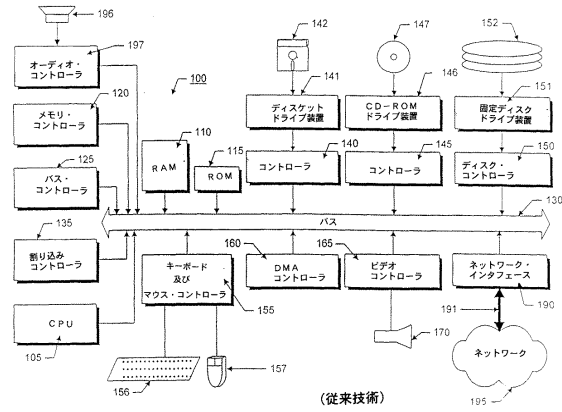
20

【0120】

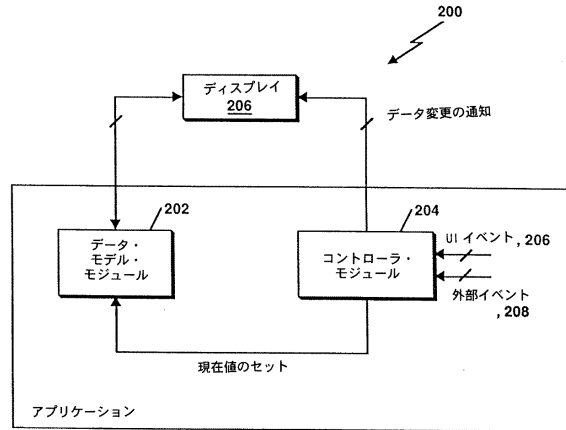
本発明の代表的実施態様を開示したが、本発明の利点の幾つかを実現する様々な変化や変更を本発明の精神と範囲から逸脱することなく成し得ることは当業者には明らかであろう。同じ機能を実行する他のコンポーネントが適宜置き換えられることができることは当業者には明らかであろう。さらに、本発明の方法は適当なプロセッサ用命令を用いるソフトウェアで全て実装したもの、またはハードウェア論理とソフトウェア論理とを組み合わせ用いて同じ結果を実現しようとするハイブリッド型で実装したもののどちらかで実現可能である。さらに、メモリの容量、特定の機能を実現するために使用される論理および/または命令の特定の構成、ならびに本発明のコンセプトに対するその他の変更はすべて添付の請求の範囲に含まれることを意図している。

30

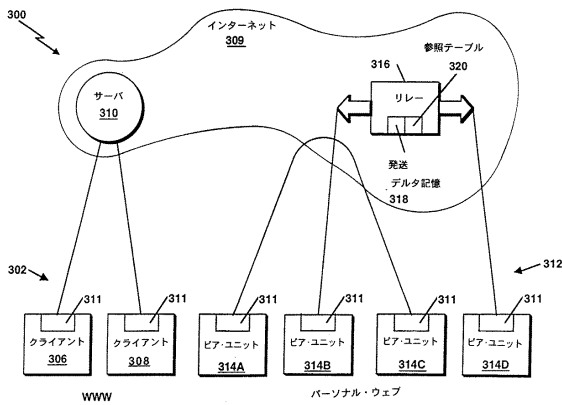
【図1】



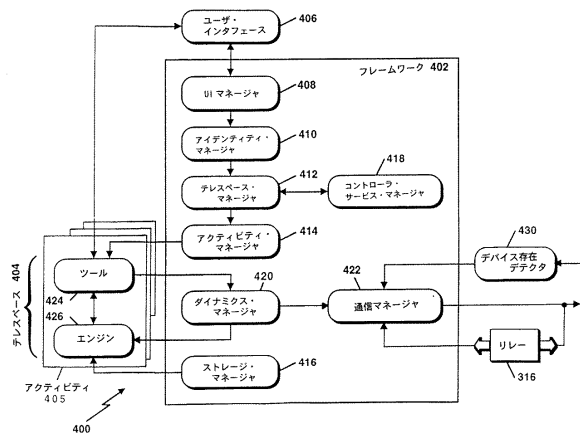
【図2】



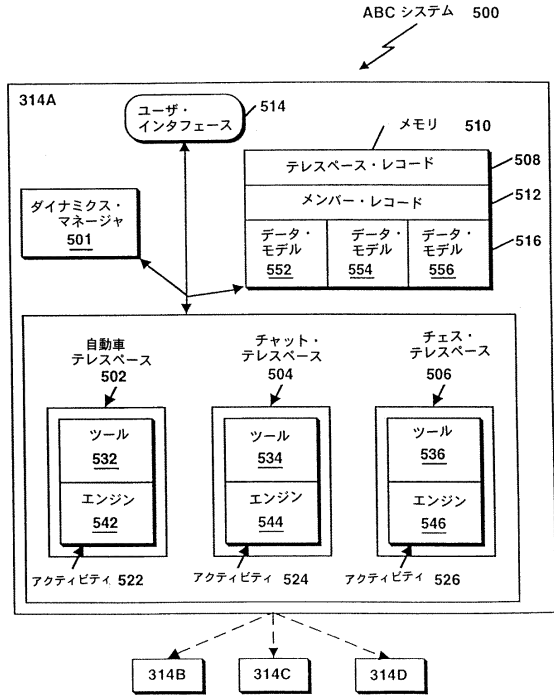
【図3】



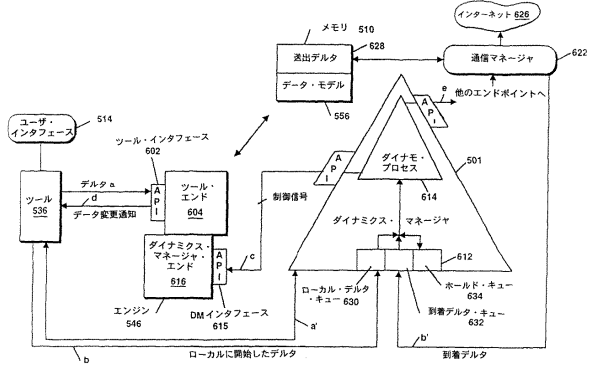
【図4】



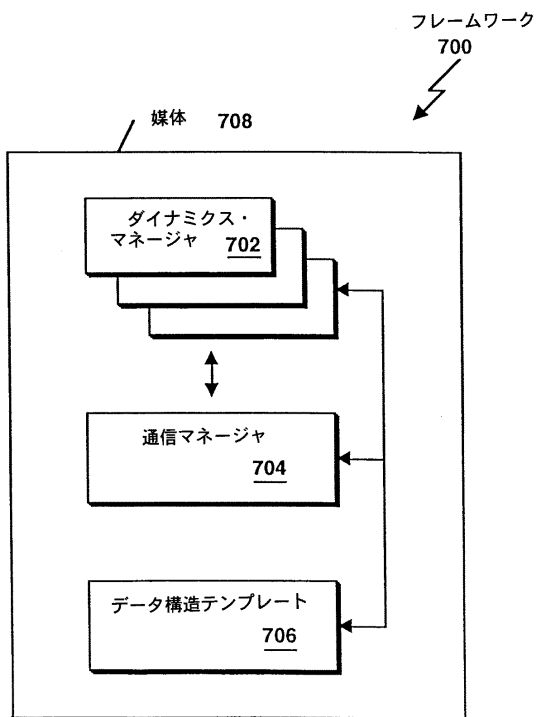
【図5】



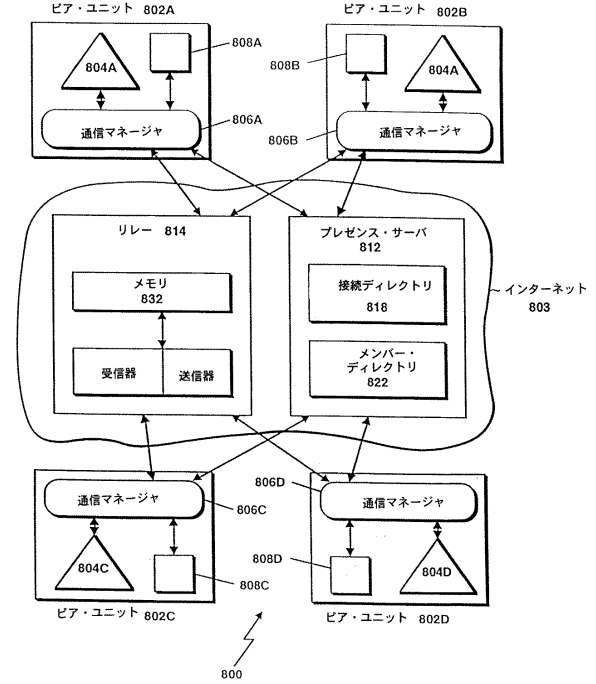
【図6】



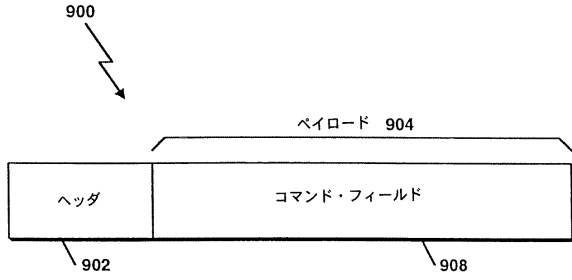
【図7】



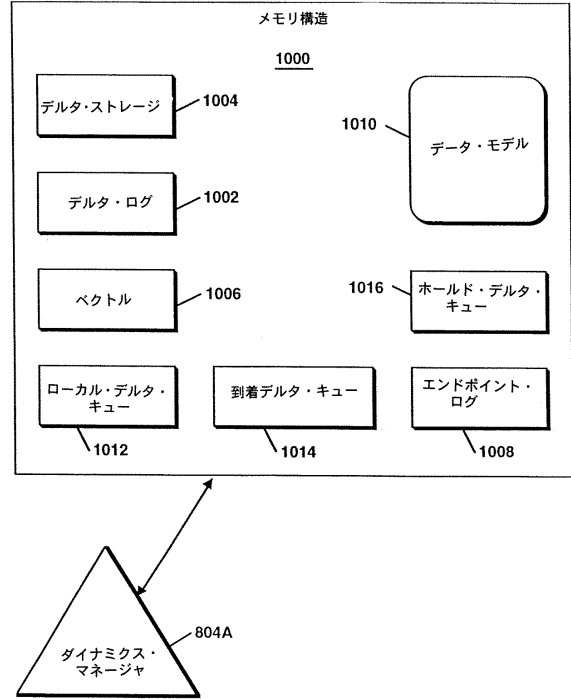
【図8】



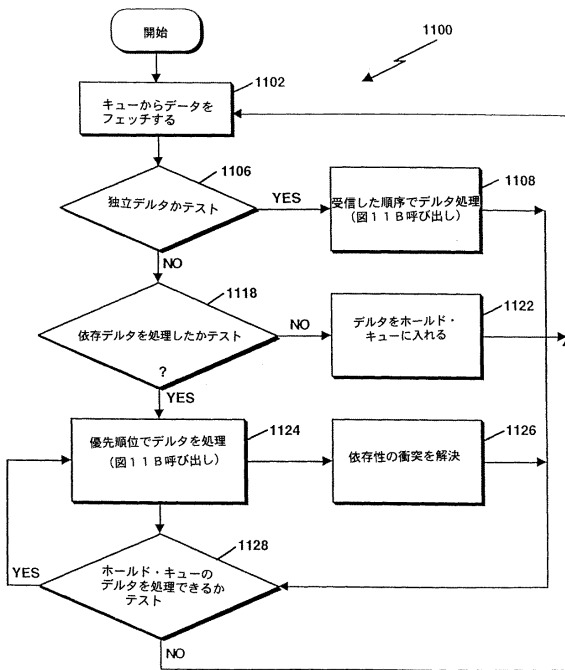
【図9】



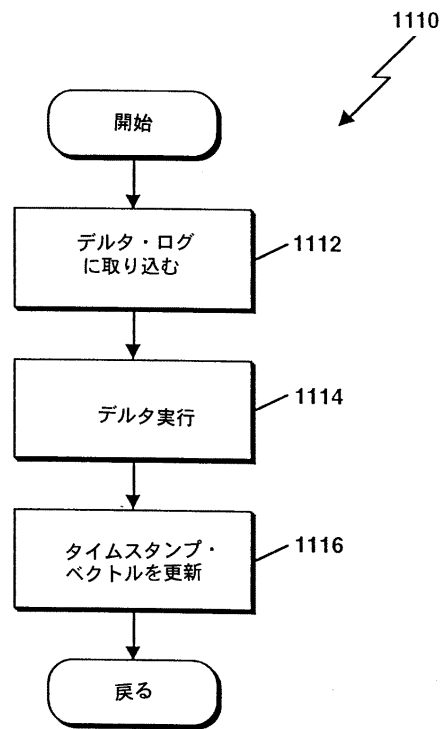
【図10】



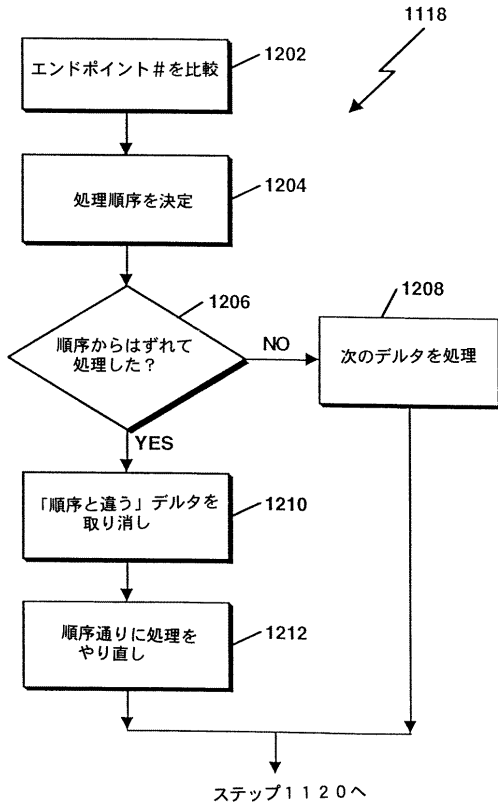
【図11A】



【図11B】



【図 12】



フロントページの続き

審査官 木村 雅也

- (56)参考文献 特開平10-500548(JP,A)
特開平07-135530(JP,A)
名嘉村 盛和,ペトリネットの最近の応用,電気学会論文誌C Vol.114-C No.9
 ,日本,社団法人電気学会,1994年 8月20日,第114-C巻,第898頁-第906頁
- (58)調査した分野(Int.Cl.,DB名)
- | | |
|------|-------|
| G06F | 13/00 |
| G06F | 12/00 |
| G06F | 15/00 |