

(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2008-538428

(P2008-538428A)

(43) 公表日 平成20年10月23日(2008.10.23)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 9/54 (2006.01)	G06F 9/46 480C	5B082
G06F 12/00 (2006.01)	G06F 12/00 547H	

審査請求 未請求 予備審査請求 未請求 (全 32 頁)

(21) 出願番号 特願2008-505310 (P2008-505310)
 (86) (22) 出願日 平成18年3月9日 (2006.3.9)
 (85) 翻訳文提出日 平成19年9月6日 (2007.9.6)
 (86) 国際出願番号 PCT/US2006/008275
 (87) 国際公開番号 W02006/110237
 (87) 国際公開日 平成18年10月19日 (2006.10.19)
 (31) 優先権主張番号 11/102,039
 (32) 優先日 平成17年4月8日 (2005.4.8)
 (33) 優先権主張国 米国 (US)

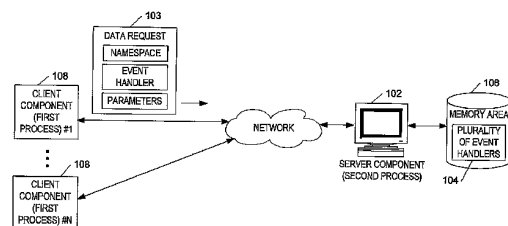
(71) 出願人 500046438
 マイクロソフト コーポレーション
 アメリカ合衆国 ワシントン州 9805
 2-6399 レッドモンド ワン マイ
 クロソフト ウェイ
 (74) 代理人 100077481
 弁理士 谷 義一
 (74) 代理人 100088915
 弁理士 阿部 和夫
 (72) 発明者 ジョージ ペレイラ
 アメリカ合衆国 98052 ワシントン
 州 レッドモンド ワン マイクロソフト
 ウェイ マイクロソフト コーポレーシ
 ョン内
 Fターム(参考) 5B082 BA08

最終頁に続く

(54) 【発明の名称】 ネットワークアプリケーションプログラム間で要求されたデータを生成および通信するためのシステムおよび方法

(57) 【要約】

ドキュメントフラグメントをウェブサーバからサーバ配置クライアントアプリケーションに配信する。本発明は、名前空間領域、イベントハンドラ領域、およびパラメータ領域を含むデータ要求のデータ構造を含む。第1プロセスはポピュレートしたデータ構造を第2プロセスに送信する。第2プロセスは、第2プロセスによって保守され、かつポピュレートされたデータ構造内で識別される論理を実行して、結果データを生成する。第2プロセスは、ユーザに表示するかまたは第1プロセスによって他の処理をさせるために、結果データを第1プロセスに送信する。



【特許請求の範囲】**【請求項 1】**

それぞれが自身に関連付けられた名前空間を有する複数のイベントハンドラであって、前記複数のイベントハンドラの各々は自身に関連付けられた 1 つまたは複数のパラメータをさらに有することを特徴とする複数のイベントハンドラと；および

フォーマット済みデータ要求を表すデータ構造であって、前記データ構造は前記名前空間を表す名前空間値を格納する名前空間領域と、前記複数のイベントハンドラの 1 つに対応するイベントハンドラ値を格納するイベントハンドラ領域であって、前記複数のイベントハンドラの 1 つは前記名前空間領域内の前記名前空間値が表す名前空間に関連付けられることを特徴とするイベントハンドラ領域と、前記パラメータの 1 つに対応するパラメータ値を格納するパラメータ領域であって、前記パラメータの 1 つは前記イベントハンドラ領域内の前記イベントハンドラ値が表す前記イベントハンドラに関連付けられることを特徴とするパラメータ領域とを含むことを特徴とするデータ構造

10

を格納する 1 つまたは複数のコンピュータ読取可能な記録媒体と；

第 2 のプロセスにより配置される第 1 のプロセスであって、前記第 1 プロセスはユーザまたはプロセスからのデータ要求を受信すること；

前記受信した要求に関連付けられた名前空間値を識別すること；

イベントハンドラ値を前記受信した要求と前記識別した名前空間値との関数として選択すること；

パラメータ値を前記受信した要求と前記選択したイベントハンドラ値との関数として決定すること；

20

前記名前空間領域内の前記識別した名前空間値、前記イベントハンドラ領域内の前記選択したイベントハンドラ値、および前記パラメータ領域内の前記決定したパラメータ値を格納することで、前記データ構造をポピュレートする (populate) こと；および

前記ポピュレートしたデータ構造を前記第 2 プロセスに送信すること

のためのコンピュータ実行可能命令を実行するよう構成されることを特徴とする第 1 のプロセスと；および

前記ポピュレートしたデータ構造を前記第 1 プロセスから受信すること；

前記受信したデータ構造の関数として、前記コンピュータ読取可能な記録媒体に格納した前記複数のイベントハンドラからイベントハンドラを選択すること；

30

前記選択したイベントハンドラを実行して結果データを生成すること；および

前記生成した結果データを前記第 1 プロセスに送信すること

のためのコンピュータ実行可能命令を実行するよう構成される第 2 のプロセスを含むことを特徴とする通信システム。

【請求項 2】

前記フォーマット済みデータ要求は、名前空間領域と前記名前空間値とを名前と値の対として備え、前記イベントハンドラ領域と前記イベントハンドラ値とを名前と値の対として備え、前記パラメータ領域と前記パラメータ値とを名前と値の対として含むことを特徴とする請求項 1 に記載のシステム。

【請求項 3】

前記コンピュータ読取可能な記録媒体と前記第 1 プロセスとは、サーバ配置個人情報管理クライアントと関連付けられることを特徴とする請求項 1 に記載のシステム。

40

【請求項 4】

前記コンピュータ読取可能な記録媒体と前記第 2 プロセスとは、個人情報管理サーバと関連付けられることを特徴とする請求項 1 に記載のシステム。

【請求項 5】

前記パラメータ値は、拡張可能マークアップ言語に従ってフォーマットしたデータを含むことを特徴とする請求項 1 に記載のシステム。

【請求項 6】

分散クライアントアプリケーションは前記第 1 プロセスを実行し、サーバアプリケーシ

50

ョンは前記第 2 プロセスを実行し、前記分散クライアントアプリケーションは少なくとも前記サーバアプリケーションの機能の一部を提供することを特徴とする請求項 1 に記載のシステム。

【請求項 7】

前記第 2 プロセスはさらに、前記コンピュータ読取可能な記録媒体に格納した前記複数のイベントハンドラを有する構成データベースを構築するためのコンピュータ実行可能命令を実行するよう構成されることを特徴とする請求項 1 に記載のシステム。

【請求項 8】

前記複数のイベントハンドラを格納する手段をさらに含む請求項 1 に記載のシステム。

【請求項 9】

前記第 1 プロセスを実行する手段と前記第 2 プロセスを実行する手段とをさらに含む請求項 1 に記載のシステム。

【請求項 10】

前記第 2 プロセスはさらに、

前記パラメータ値を検証すること；および

前記パラメータ値を自身に関連付けられたデータ型に変換すること

のためのコンピュータ実行可能命令を実行するよう構成されることを特徴とする請求項 1 に記載のシステム。

【請求項 11】

あるプロセスから別のプロセスヘデータを生成および配信するためのコンピュータ実施方法であって、前記コンピュータ実施方法は、

第 1 プロセスによりユーザからのデータ要求を受信すること；

第 1 プロセスにより前記受信した要求に関連付けられた名前空間を識別すること；

前記第 1 プロセスにより、前記受信した要求の関数として前記識別した名前空間に関連付けられたイベントハンドラを決定すること；

前記第 1 プロセスにより、前記受信した要求の関数として前記決定したイベントハンドラに関連付けられたパラメータをポピュレートすること；

前記第 1 プロセスにより、前記識別した名前空間、前記決定したイベントハンドラ、および前記ポピュレートしたパラメータを有するフォーマット済み要求を生成すること；

前記第 1 プロセスから、前記生成したフォーマット済み要求を第 2 のプロセスに送信すること；

前記第 2 プロセスにより、前記第 1 プロセスから前記フォーマット済み要求を受信すること；

前記第 2 プロセスにより、前記受信したフォーマット済み要求の関数として、メモリ領域内に格納した複数のイベントハンドラから 1 つのイベントハンドラを選択すること；

前記第 2 プロセスにより、前記選択したイベントハンドラを実行して結果データを生成すること；および

前記第 2 プロセスから、前記生成した結果データを前記第 1 プロセスに送信することを含むことを特徴とするコンピュータ実施方法。

【請求項 12】

前記第 1 プロセスにより前記送信した結果データを受信することをさらに含む請求項 1 に記載のコンピュータ実施方法。

【請求項 13】

前記第 1 プロセスにより、ユーザ定義名前空間、前記ユーザ定義名前空間に関連付けられたユーザ定義イベントハンドラ、および前記ユーザ定義イベントハンドラのユーザ定義パラメータをユーザから受信すること；および

前記第 2 プロセスとともに前記第 1 プロセスにより、前記受信したユーザ定義名前空間、前記ユーザ定義イベントハンドラ、および前記ユーザ定義パラメータを登録することをさらに含む請求項 11 に記載のコンピュータ実施方法。

【請求項 14】

10

20

30

40

50

前記第 2 プロセスは、前記受信したユーザ定義名前空間、前記ユーザ定義イベントハンドラ、および前記ユーザ定義パラメータを、前記第 2 プロセスのインフラストラクチャを変更せずに登録することを特徴とする請求項 1 3 に記載のコンピュータ実施方法。

【請求項 1 5】

1 つまたは複数のコンピュータ読取可能な記録媒体は、請求項 1 1 に記載のコンピュータ実施方法を実施するためのコンピュータ実行可能命令を有することを特徴とする請求項 1 1 に記載のコンピュータ実施方法。

【請求項 1 6】

ネットワーク環境で要求されたデータを生成および通信するためのコンピュータ実行可能コンポーネントを有する 1 つまたは複数のコンピュータ読取可能な記録媒体であって、前記コンポーネントは、

- ユーザからデータ要求を受信すること；
- 前記受信した要求に関連付けられた名前空間を識別すること；
- 前記受信した要求の関数として前記識別した名前空間に関連付けられたイベントハンドラを決定すること；
- 前記受信した要求の関数として前記決定したイベントハンドラに関連付けられたパラメータをポピュレートすること；
- 前記識別した名前空間、前記決定したイベントハンドラ、および前記ポピュレートしたパラメータを有するフォーマット済み要求を生成すること；および
- 前記生成したフォーマット済み要求をサーバコンポーネントに送信すること；

のためのサーバ配置クライアントコンポーネントを備え；および

- 前記送信した要求を前記サーバ配置クライアントコンポーネントから受信すること；
- 前記受信した要求の関数として、メモリ領域内に格納した複数のイベントハンドラから 1 つのイベントハンドラを選択すること；
- 前記選択したイベントハンドラを実行して結果データを生成すること；および
- 前記生成した結果データを前記サーバ配置クライアントコンポーネントに送信すること；

のためのサーバコンポーネント

を備えることを特徴とする 1 つまたは複数のコンピュータ読取可能な記録媒体。

【請求項 1 7】

前記メモリ領域内に格納した前記複数のイベントハンドラを有する構成データベースを構築するためのレジストリコンポーネントをさらに備える請求項 1 6 に記載のコンピュータ読取可能な記録媒体。

【請求項 1 8】

前記サーバコンポーネントは、前記受信した要求の関数として、前記構成データベースからイベントハンドラを選択することを特徴とする請求項 1 7 に記載のコンピュータ読取可能な記録媒体。

【請求項 1 9】

前記サーバ配置クライアントコンポーネントは、ネットワーク上で前記サーバコンポーネントと通信する個人情報管理クライアントを備えることを特徴とする請求項 1 6 に記載のコンピュータ読取可能な記録媒体。

【請求項 2 0】

前記サーバコンポーネントは個人情報管理サーバを備えることを特徴とする請求項 1 6 に記載のコンピュータ読取可能な記録媒体。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

本発明の実施形態は、コンピュータ通信の領域に関する。特に本発明の実施形態は、名前空間、名前空間内のイベント、およびイベントに対するパラメータを有するスキーマ形態のデータ要求に関する。

10

20

30

40

50

【背景技術】

【0002】

一部のサーバ配置ウェブベースメールアプリケーションにより、中央に位置するメールボックスにユーザはアクセス可能になる。これらのアプリケーションは、インターネット等のネットワークから電子メール (email) を送受信する機能、スケジュールを予約する機能、他の個人情報管理 (PIM) のタスクを実施する機能も提供する。これらのアプリケーションの一部はブラウザ内で実行し、他の PIM アプリケーションに対してコンパニオンである。上記のクライアントウェブアプリケーションは、PIM 機能を実施可能とするために大量のクライアントサーバ間のトラフィックを必要とする。例えば、ユーザが新しい電子メールをチェックするために受信箱内の電子メール一覧をリフレッシュするとき、受信箱内の新しい電子メールの一覧を抽出するためにウェブ要求がサーバに送信される。

10

【発明の開示】

【発明が解決しようとする課題】

【0003】

一部のクライアントアプリケーションに対して、クライアントサーバ間のトラフィックはハイパーテキスト転送プロトコル (HTTP) および (例えば、HTTP の POST 要求における単純化した名前と値の対に基づく) 他の様々なスキーマに対するウェブを使った分散オーサリングとバージョン管理 (WebDAV) 拡張の混成である。これらの従来の技術のシステムにおいて複数のプロトコルをデータアクセスに使用することは実装を複雑にし、保守性を低下させる。これらの従来の技術のシステムはさらに、要求に対して、首尾一貫しない不十分なスキーマを使用している。一部の従来の技術のシステムで使用されるような名前 - 値形式は非常に制限されたものであり、堅固に定形化したデータあるいはより複雑な構造または配列を容易にはサポートしていない。さらに、名前 値形式は名前と値の対に対するセパレータとして改行を必要とするため、特定の名前と値の対に関する全てのデータを、送信前に改行なしで符号化し受信時に復号化することが必要となる。一部の既存システムにおけるこれらの余計なステップは処理を遅延させ、複雑度を増加させる。一部の既存ウェブクライアントアプリケーションのさらに別の問題は、クライアントアプリケーションに様々な機能を提供するサーバサイドコードが無秩序で整理されていないことである。そのため、新たな要求ハンドラを既存システムのサーバサイドコードに追加することは困難である。

20

30

【0004】

従って、これらおよび他の欠点を解決するため、クライアントサーバ通信に対して改善したシステムが望まれる。

【課題を解決するための手段】

【0005】

本発明の実施形態は、ネットワークアプリケーションプログラム間で要求されたデータを生成および通信するためのプロトコルおよびスキーマを含む。1 実施形態において、本発明はさらに、要求されたデータを生成する要求ハンドラの開発者が利用可能なアーキテクチャ、およびアプリケーションプログラミングインタフェース (API) を含む。開発者は、API を用いてカスタマイズした (custom) 要求ハンドラを作成できる。

40

【0006】

本発明のクライアントサーバの実施形態では、本発明によるクライアントが、本発明の矛盾なく定義されたスキーマに従って、データ要求をサーバに送信する。その要求は要求ハンドラの名前空間、要求ハンドラ名前空間内のイベントハンドラ、およびイベントハンドラの 1 つまたは複数のパラメータを識別する。本発明は、サーバ内の要求ハンドラ名前空間にグループ化されるイベントハンドラのインフラストラクチャを含み、また、サーバは、それぞれのクライアントの要求に応じて複数フォーマットでクライアントに応答を配信する。サーバのインフラストラクチャは、要求を解析し、特定名前空間内の特定イベントハンドラを識別し、識別したイベントハンドラに関連付けられるハンドラのコードを要

50

求パラメータの関数として実行する。

【 0 0 0 7 】

本発明は、能力は高いが容量が小さく拡張性がある。データ要求に対して首尾一貫したスキーマを使用することで、アプリケーションプログラム間の相互作用が簡略化され、要求ハンドラの保守性が改善される。本発明のスキーマは複雑なデータ構造をサポートする。本発明は、要求されたデータを生成および配信するインフラストラクチャへの変更なしに、要求ハンドラの追加のサポートをする。

【 0 0 0 8 】

本発明の 1 態様によると、システムは、それぞれが自身に関連付けられた名前空間を有する複数のイベントハンドラを格納するための 1 つまたは複数のコンピュータ読取可能な記録媒体を含む。複数のイベントハンドラの各々はさらに、自身に関連付けられた 1 つまたは複数のパラメータを有する。コンピュータ読取可能な記録媒体は、データに対するフォーマット済み要求を表すデータ構造を格納する。データ構造は、名前空間を表す名前空間値を格納する名前空間領域、複数のイベントハンドラの 1 つに対応するイベントハンドラ値を格納するイベントハンドラ領域を含む。複数のイベントハンドラの 1 つは、名前空間領域内の名前空間値が表す名前空間と関連付けられる。また、データ構造は、パラメータの 1 つに対応するパラメータ値を格納するパラメータ領域も含む。前記パラメータの 1 つは、イベントハンドラ領域内のイベントハンドラ値が表すイベントハンドラに関連付けられる。システムは、また第 2 のプロセスが配置する第 1 のプロセスも含む。第 1 プロセスはコンピュータ実行可能命令を実行するよう構成される。このコンピュータ実行可能命令は、ユーザまたはプロセスからデータ要求を受信し、受信した要求に関連付けられた名前空間値を識別し、受信した要求および識別した名前空間値の関数としてイベントハンドラ値を選択し、受信した要求および選択したイベントハンドラ値の関数としてパラメータ値を決定する。第 1 プロセスはさらに、識別した名前空間値を名前空間領域内に格納すること、選択したイベントハンドラ値をイベントハンドラ領域内に格納すること、および決定したパラメータ値をパラメータ領域内に格納することで、データ構造をポピュレートする。第 1 プロセスはさらに、ポピュレートしたデータ構造を第 2 プロセスに送信する。第 2 プロセスはコンピュータ実行可能命令を実行するよう構成され、このコンピュータ実行可能命令は、ポピュレートしたデータ構造を第 1 プロセスから受信し、受信したデータ構造の関数として、コンピュータ読取可能な記録媒体上に格納した複数のイベントハンドラから 1 つのイベントハンドラを選択し、選択したイベントハンドラを実行して結果データを生成し、生成した結果データを第 1 プロセスに送信する。

10

20

30

【 0 0 0 9 】

本発明の別の態様によると、コンピュータ実施方法は、あるプロセスから別のプロセスへデータを生成および配信する。コンピュータ実施方法は、第 1 プロセスによりユーザからデータ要求を受信すること、および受信した要求に関連付けられた名前空間を識別することを含む。前記方法は、第 1 プロセスにより、識別した名前空間に関連付けられたイベントハンドラを、受信した要求の関数として決定することも含む。前記方法はさらに、第 1 プロセスにより、決定したイベントハンドラに関連付けられたパラメータを、受信した要求の関数としてポピュレートすること、および識別した名前空間、決定したイベントハンドラ、およびポピュレートしたパラメータを有するフォーマット済み要求を生成することをさらに含む。前記方法は、第 1 プロセスから、生成したフォーマット済み要求を第 2 プロセスに送信することも含む。前記方法は、第 2 プロセスにより、第 1 プロセスからフォーマット済み要求を受信すること、および受信したフォーマット済み要求の関数として、メモリ領域に格納した複数のイベントハンドラから 1 つのイベントハンドラを選択することも含む。コンピュータ実施方法は、第 2 プロセスにより、選択したイベントハンドラを実行して結果データを生成すること、および生成した結果データを第 2 プロセスから第 1 プロセスに送信することも含む。

40

【 0 0 1 0 】

本発明のさらに別の態様によると、 1 つまたは複数のコンピュータ読取可能な記録媒体

50

は、要求されるデータをネットワーク環境で生成および通信するためのコンピュータ実行可能コンポーネントを有する。前記コンポーネントはサーバ配置クライアントコンポーネントを含み、そのサーバ配置クライアントコンポーネントはデータ要求をユーザから受信し、受信した要求に関連付けられた名前空間を識別し、識別した名前空間に関連付けられたイベントハンドラを、受信した要求の関数として決定し、決定したイベントハンドラに関連付けられたパラメータを、受信した要求の関数としてポピュレートし、識別した名前空間、決定したイベントハンドラ、およびポピュレートしたパラメータを有するフォーマット済み要求を生成し、生成したフォーマット済み要求をサーバコンポーネントに送信する。前記コンポーネントはサーバコンポーネントも含み、そのサーバコンポーネントは、サーバ配置クライアントコンポーネントからの送信された要求を受信し、メモリ領域に格納した複数のイベントハンドラから、受信した要求の関数としてイベントハンドラを選択し、選択したイベントハンドラを実行して結果データを生成し、生成した結果データをサーバ配置クライアントコンポーネントに送信する。

10

【0011】

または、本発明は様々な他の方法および装置を備えることができる。

【0012】

他の特徴は、以降本明細書中で部分的に明らかにし部分的に指摘する。

【発明を実施するための最良の形態】**【0013】**

実施形態において、本発明は、図1に示すように、データに対するクライアントアプリケーションからの要求に回答して、サーバ配置クライアントアプリケーション、コンポーネント、プロセス等にデータを配信するサーバアプリケーション、コンポーネント、プロセス等を含む。サーバコンポーネント102上で、データ要求103等のデータ要求に回答するコードまたはルーチンがイベントハンドラ（例えば、複数のイベントハンドラ104）等に分割される。1実施形態において、イベントハンドラ104は論理的に名前空間に組織される。特に本発明は、名前空間と実行すべきイベントハンドラとに関する情報、およびそのハンドラに対するパラメータを含むデータ要求103等の要求を構築することを含む。本発明はさらにサーバ102内にインフラストラクチャを含み、それにより開発者はデータ要求に対する回答を配信する要求ハンドラを実装することができる。

20

【0014】

本発明は、一例として分散個人情報管理アプリケーションにおけるカレンダービューを含む。カレンダービューは所与の日付に対する予約の一覧を示す。サーバ102は、データベースに問い合わせて所定の時間範囲における予約を抽出することにより、カレンダー内のこのビューをリフレッシュする。次に、抽出した予約は、開始時間と各予約の期間によって、クライアント（例えば、クライアントコンポーネント108）内の正しい位置に描かれる。以前のシステムでは、上記のリフレッシュ操作は以下のように達成された。すなわち、クライアントコンポーネント108が、ウェブを使った分散オーサリングとバージョン管理（WebDAV）拡張をハイパーテキスト転送プロトコル（HTTP）に対して使用して、拡張可能マークアップ言語（XML）フォーマットで返すべきデータを問い合わせ、拡張スキーマ定義（XSD）変換を使用してHTMLをXMLデータから生成し、前記HTMLをユーザに表示する。上記のシステムは複雑であり、保守が困難である。

30

40

【0015】

しかしながら本発明を用いると、カレンダーのリフレッシュは、クライアントコンポーネント108がフォーマット済みデータ要求（例えば、データ要求103）を送信し、カレンダーデータを定義するコードストリーム（例えば、JavaScript）を受信することで達成される。カレンダーデータは部分的にサーバ102でレンダリングされる。クライアントコンポーネント108は前記コードを実行し、スクリプトを使用してHTMLマークアップを生成する。大部分のカレンダーデータおよび表示情報がサーバ102で事前に計算されるため、クライアントコンポーネント108の負荷が低減される。本発明の別の例では、サーバ102は、サーバ102でレンダリングされたHTMLを返すことでPIM

50

アプリケーションのメールビューをリフレッシュする。本発明は、特定フォーマットでデータを返すことに限定されない。むしろ、本発明は、HTML、XML、テキスト、JavaScriptペイロード等の、クライアントコンポーネント108が要求する任意の形態でデータを返す。

【0016】

図1を再度参照すると、ブロック図の例は、クライアントコンポーネント#1からクライアントコンポーネント#Nまで等の1つまたは複数のクライアントコンポーネント108、プロセス等とネットワーク(例えば、インターネット)経由で通信するサーバコンポーネント102等のサーバコンポーネント、プロセス等を示す。図1のシステムは、メモリ領域106等の1つまたは複数のコンピュータ読取可能な記録媒体を含む。メモリ領域106は、それぞれが自身と関連付けられた名前空間を有する複数のイベントハンドラ104を格納する。複数のイベントハンドラ104の各々は、自身と関連付けられた1つまたは複数のパラメータをさらに有する。フォーマット済みデータ要求103を表すデータ構造も、コンピュータ読取可能な記録媒体に格納される。図1において、データ要求はクライアントコンポーネント108からサーバコンポーネント102に送信されることを示している。データ要求103を表すデータ構造は、名前空間を表す名前空間値を格納する名前空間領域を含む。データ構造はさらにイベントハンドラ値を格納するイベントハンドラ領域を含み、イベントハンドラ値はメモリ領域106に格納した複数のイベントハンドラ104の1つに対応し、メモリ領域106は名前空間領域内の名前空間値が表す名前空間と関連付けられる。データ構造はさらに少なくとも1個のパラメータ値を格納するパラメータ領域を含み、前記パラメータ値はイベントハンドラ領域内のイベントハンドラ値が表すイベントハンドラと関連付けられるパラメータに対応する。

10

20

【0017】

図1の例では、クライアントコンポーネント108は第1プロセスであり、サーバコンポーネント102は第2プロセスである。第1プロセスは分散クライアントアプリケーションであるか、そうでなければ第2プロセスにより配置されてサーバコンポーネント102の機能の一部または全てを提供する。例えば、第1プロセスはサーバ配置個人情報管理クライアント等のサーバ配置クライアントアプリケーションが可能であり、サーバコンポーネント102は個人情報管理サーバが可能である。しかしながら、本発明は第1プロセスと第2プロセスとの間のクライアントサーバ関係に限定されない。例えば、第1プロセスと第2プロセスとの間の関係はピアツーピアでも可能である。

30

【0018】

1実施形態では、第1プロセスは第2プロセスからデータを要求するためのコンピュータ実行可能命令を実行するよう構成される。特に、第1プロセスはデータ要求をユーザまたはプロセスから受信する。例えば、前記要求は最近受信した電子メールをチェックするための要求、カレンダー表示を更新するための要求、また任意の他のデータ要求を含むことができる。第1プロセスは、受信した要求に関連付けられた名前空間値を識別し、受信した要求と識別した名前空間値との関数としてイベントハンドラ値を選択する。選択したイベントハンドラはユーザからの特定要求に対して特有であり、識別した名前空間値に関連付けられた複数のイベントハンドラ104の1つである。選択したイベントハンドラは1つまたは複数のパラメータを有する。第1プロセスは、選択したイベントハンドラの1つまたは複数のパラメータに対するパラメータ値を決定する。パラメータ値は受信した要求からのデータを使用して決定される。パラメータ値は任意のタイプのデータを含むことができ、第2プロセスが認識する任意のフォーマット(例えば、拡張可能マークアップ言語)で具体化することができる。第1プロセスは、識別した名前空間値を名前空間領域に格納すること、選択したイベントハンドラ値をイベントハンドラ領域に格納すること、および決定したパラメータ値をパラメータ領域に格納することで、データ要求103を表すデータ構造をポピュレートすることができる。第1プロセスは、ポピュレートしたデータ構造(例えば、データ要求103)を第2プロセスに送信する。

40

【0019】

50

1 実施形態において、ポピュレートしたデータ構造は一連の名前と値の対を含む。例えば、名前空間領域と名前空間値は名前と値の対であり、イベントハンドラ領域とイベントハンドラ値は名前と値の対であり、パラメータ領域とパラメータ値は名前と値の対である。

【0020】

第2プロセスは、第1プロセスに対して要求されるデータを生成および配信するためのコンピュータ実行可能命令を実行するよう構成される。1実施形態において、第2プロセスは第1プロセスからポピュレートしたデータ構造（例えば、データ要求103）を受信し、メモリ領域106に格納した複数のイベントハンドラ104からイベントハンドラを選択する。選択したイベントハンドラは、受信したデータ要求103で指定したイベントハンドラに対応する。第2プロセスは、第1プロセスから受信したデータ要求103のパラメータを用いて、選択したイベントハンドラを実行し、結果データを生成する。第2プロセスは、生成した結果データを第1プロセスに送信する。

10

【0021】

1実施形態において、第2プロセスは、構成データベースを問い合わせることで適切なイベントハンドラを選択する。その構成データベースは、利用可能な名前空間、イベントハンドラ（例えば、複数のイベントハンドラ104）、およびパラメータに関する情報を取り込む。例えば、第2プロセスは、第1プロセスからのデータ要求103を受信する前に構成データベースを構築するためのレジストリコンポーネントを含む。例えば、複数のイベントハンドラ104を、名前空間により構成データベース内で組織化することができる。

20

【0022】

本発明は、複数のイベントハンドラ104を格納する手段、第1プロセスを実行する手段、および第2プロセスを実行する手段を含む。（例えば、図に示すような）データ構造、ユーザインタフェース、アプリケーションプログラム、アプリケーションプログラミングインタフェース（API）、コンピュータ実行可能命令、ファームウェア等のハードウェアおよびソフトウェアは、複数のイベントハンドラ104を格納する手段、第1プロセスを実行する手段、および第2プロセスを実行するための手段を構成する。

【0023】

1実施形態において、本発明はPOST動詞またはGET動詞を有するHTTP要求を使用して実装される。要求は以下の情報を含む。すなわち、（例えば、URLのクエリ文字列パラメータ‘ns’で示される）名前空間、（例えば、URLのクエリ文字列パラメータ‘ev’で示される）イベント名、および1つまたは複数の要求パラメータである。パラメータはPOST要求に対するXML、またはGET要求に対する名前と値の対のいずれかである。

30

【0024】

本発明のGET要求（例えば、データ要求103）の例を以下に示す。

URL: /mail/ev.owa?ns=CalendarView&ev=Refresh&d=11142004T00:00:00Z

【0025】

この例において、クライアント（例えば、クライアントコンポーネント108）はサーバ（例えば、サーバコンポーネント102）に依頼して“CalendarView”名前空間における“Refresh”ハンドラを実行する。クライアントコンポーネント108は、2004年11月14日午前12時の値を有する“d”という名前の日付型のパラメータを渡す。サーバコンポーネント102は実行状態に応じ、このビューに対してマークアップまたはJavaScriptペイロードを返す。

40

【0026】

本発明のPOST要求（例えば、データ要求103）の例を以下に示す。

URL: /mailserver/ev.owa?ns=Messaging&ev=SaveMessage

50

```

B O D Y :
< p a r a m s >
  < s u b j e c t > T e s t   e m a i l < / s u b j e c t >
  < t o > < i t e m > u s e r A @ p a g e A . n e t < / i t e m > < i t e m >
u s e r B @ p a g e A . n e t < / i t e m > < / t o >
  < b o d y > T e s t   s o m e t h i n g < / b o d y >
< / p a r a m s >

```

【 0 0 2 7 】

この例では、クライアントコンポーネント 1 0 8 はサーバコンポーネント 1 0 2 に依頼して ' M e s s a g i n g ' 名前空間内の ' S a v e M e s s a g e ' イベントハンドラを実行する。イベントパラメータは s u b j e c t (文字列)、t o (文字列の配列) および b o d y (文字列) である。このイベントに対する応答は、成功またはエラーを示す H T T P の結果である。この場合、返されるマークアップは全く不要である。パラメータは X M L フォーマット形式またはサーバが理解可能な任意の他の言語フォーマットでよい。

10

【 0 0 2 8 】

次に図 2 を参照すると、フローチャート例は、要求されたデータを通信するためのクライアントおよびサーバプロセスの動作を示す。図 2 のフローチャートは、あるプロセスから別のプロセスへデータを生成および配信するためのコンピュータ実施方法を反映している。前記方法は、図 1 のクライアントコンポーネント 1 0 8 等の第 1 プロセスにより、2 0 2 においてユーザからのデータ要求を受信すること、および 2 0 4 において受信した要求に関連付けられた名前空間を識別することを含む。前記方法はさらに、識別した名前空間に関連付けられたイベントハンドラを、2 0 6 において受信した要求の関数として決定すること、および、決定がなされたイベントハンドラに関連付けられたパラメータを、2 0 8 において受信した要求の関数としてポピュレートすることを含む。また、前記方法は、2 1 0 において、識別した名前空間を有するフォーマット済み要求 (例えば、図 1 のデータ要求 1 0 3)、決定されたイベントハンドラ、およびポピュレートしたパラメータを、生成することを含む。前記方法は、2 1 2 において、前記第 1 プロセスから第 2 プロセス (例えば、サーバコンポーネント 1 0 2) へと生成したフォーマット済み要求を送信する。

20

30

【 0 0 2 9 】

本発明の前記方法は、2 1 4 において、第 2 プロセスにより、フォーマット済み要求を第 1 プロセスから受信することを含む。前記方法は、2 1 5 において、受信した要求に基づき名前空間を選択する。また、前記方法は、2 1 6 において、受信フォーマット済み要求の関数としての第 2 プロセスにより、1 つまたは複数のコンピュータ読取可能な記録媒体 (例えば、図 1 のメモリ領域 1 0 6) に格納された複数のイベントハンドラ (例えば、図 1 の複数のイベントハンドラ 1 0 4) から 1 つのイベントハンドラを選択することを含む。前記方法は、2 1 7 において、受信した要求内のパラメータを解析および検証する。前記方法はさらに、2 1 8 において、選択されたイベントハンドラを実行して結果データを生成すること、および、2 2 0 において、第 1 のプロセスに対して生成した結果データを第 2 プロセスから第 1 プロセスへ送信することを含む。第 1 プロセスは、結果データを受信し、ある例においてはデータをユーザに表示する。

40

【 0 0 3 0 】

開発者または他のユーザは、ユーザ定義の名前空間、イベントハンドラ、および / またはパラメータを生成することができる。本発明の前記方法はさらに、ユーザから第 1 プロセスにより、ユーザ定義名前空間、前記ユーザ定義名前空間に関連付けられたユーザ定義イベントハンドラ、および / または前記ユーザ定義イベントハンドラのユーザ定義パラメータを受信することを含む。第 1 のプロセスは、受信したユーザ定義名前空間、ユーザ定義イベントハンドラ、および / または第 2 プロセスを有するユーザ定義パラメータを登録する。第 2 プロセスは、第 2 プロセスのインフラストラクチャを変更せずに、受信したユ

50

ーザ定義名前空間、ユーザ定義イベントハンドラ、およびユーザ定義パラメータを登録する。すなわち、第2プロセス（例えば、サーバコンポーネント）を実行するコードは、登録によっては変化しない。第2プロセスのインフラストラクチャは、登録されたユーザ定義名前空間、登録されたイベントハンドラ、または登録されたパラメータに対して、第2プロセスが他の名前空間、イベントハンドラ、およびパラメータ値にアクセスするのと同様にアクセスする。

【0031】

1実施形態では、1つまたは複数のコンピュータ読取可能な記録媒体は、図2に示すようにコンピュータ実施方法を実施するためのコンピュータ実行可能命令を有する。

【0032】

次に図3を参照すると、ブロック図の例はサーバコンポーネント内モジュールを示す。モジュールは、サーバコンポーネントが生成した要求スレッド302内で呼び出される。1実施形態では、イベントハンドラはIHttpHandlerインタフェースを実装するHTTPハンドラである。本発明は、ハンドラクラスが要求を処理するコードを実行可能にするProcessRequestメソッドをこのインタフェース内で呼び出す。HTTPハンドラファクトリはIHttpHandlerFactoryインタフェースを実装するクラスである。上記のオブジェクトはHTTPハンドラオブジェクトのインスタンスを生成するために使用され、このインスタンスは要求を処理するために使用される。図3の例では、EventHandlerFactory304と名付けたHTTPハンドラファクトリクラスを示す。本発明（例えば、HTTPアプリケーション306）は、このクラスのオブジェクト（例えば、EventHandlerBaseインスタンス308）をインスタンス化してIHttpHandlerFactoryインタフェースの“GetHandler”メソッドを呼び出し、要求の名前空間に基づいて適切なHTTPハンドラクラス（例えば、EventHandlerBaseサブクラス）のインスタンスを生成する。HTTPアプリケーション306およびEventHandlerBaseインスタンス308は、HTTPコンテキスト314にアクセスする。

【0033】

EventContext312オブジェクトは、要求に関する情報（例えば、要求の名前空間、イベント名およびパラメータ）を含むデータ構造である。EventHandlerRegistry310は、名前空間とアプリケーションが実行するそれらのイベントハンドラとに関する情報に加え、サポートされるカスタムデータ構造と列挙体とに関する情報のリポジトリとしての役割を果たすクラスである。アプリケーションは、アプリケーション開始時にこのリポジトリ310を使用してEventHandlerBaseクラスを登録する。本発明は（例えば、リフレクションを使用して）クラスを走査し、名前空間、名前空間内のイベント、および前記イベントのパラメータを決定する。この情報は記憶され、後に要求処理およびパラメータ解析中に使用される。

【0034】

本発明による要求の実行フローの例を図3に示す。本発明は、本発明に従ってデータ要求を受信し、EventHandlerFactoryクラスのインスタンスを生成し、生成したインスタンス上でGetHandlerを呼び出す。EventHandlerFactoryのGetHandler実装はクエリ文字列を参照して名前空間とイベント名を把握し、クエリ文字列内の名前空間とイベント名とに対応するハンドラクラスに対して、EventHandlerRegistry310を検索する。本発明は、要求されたイベントハンドラを発見した場合には、そのイベントハンドラのインスタンスを生成する。また、EventContext312も生成する。本発明はイベントハンドラ上でProcessRequestを呼び出す。ProcessRequestは、（例えば、要求がGet動詞かまたはPOST動詞かに関わらず）要求自身に特有な解析クラスを使用して要求のパラメータを解析する。解析クラスは要求内のパラメータを解析し、EventHandlerRegistry310を使用して任意のタイプの変換を行ってスキーマが正確であることを保証する（例えば、このハンドラが要求する全てのパラメー

10

20

30

40

50

タが設定されていることを保証する)。パラメータはこの時点でコンテキスト内のコレクションに格納できる。本発明は、イベントハンドラに関連付けられたイベントハンドラコードを実行する(例えば、ハンドラクラス内のメソッドを実行する)。このコードはそのコンテキスト内のパラメータにアクセスし、処理を実行し、応答を要求エンティティに対して、要求エンティティが指定する任意のフォーマットで書き込む。

【0035】

次に図4を参照すると、ブロック図の例は、EventHandlerBase402から継承するイベントハンドラクラスのサンプル集合を示す。EventHandlerBase402はIHttpHandlerインタフェース404を実装する。CalendarEventHandler406、MessagingEventHandler408、およびMessageViewEventHandler410はEventHandlerBase402から継承する。開発者または他のユーザは、名前空間を生成し、前記名前空間内のイベントを識別し、前記イベントに対するパラメータを定義することにより、図4に示すようなカスタムイベントハンドラを生成することができる。ユーザ定義名前空間、イベントハンドラ、および/またはパラメータはユーザ選択のもの(例えば、既存名前空間内の新規イベントハンドラ)か、またはユーザ生成のもの(例えば、新規名前空間)であることができる。1実施形態では、ユーザ定義イベントハンドラはダイナミックリンクライブラリとして実装される。

【0036】

開発者または他のユーザは、以下に示すような属性を使用することで名前空間のスキーマを定義することができる。

```
[EventNamespace("Namespace1")]
internal class TestEventHandler: EventHandlerBase
{
    [Event("Event1")]
    [EventParameter("Param1", typeof(string))]
    [EventParameter("Param2", typeof(int))]
    public void Event1()
    {
        string param1 =
            (string)Context.GetParamValue("Param1")
    };
    int param2 = (int)Context.GetParamValue("Param2");
    Writer.Write("{0} + {1} = {2}", param1, param2, param1 + param2);
    }
}
```

【0037】

この例において、EventNamespace属性はワイヤー上の名前空間の名前を定義し、Event属性はワイヤー上のイベントの名前を定義し、EventParameter属性はパラメータの名前、タイプおよびパラメータがオブショナルかどうかを定義する。ハンドラを使用するため、ハンドラの開発者はEventHandlerBase402から継承するクラスを書き、パラメータとreturn voidをもたないメソッドとしてイベントを書き、上記の適切な属性を使用してスキーマを定義し、サーボコンポーネントを有するクラスを登録する。

【0038】

本発明のデータ構造により、開発者はパラメータとして使用できる複合型を定義するこ

10

20

30

40

50

とができる。データ構造の例を以下に示す。

```
[ EventStruct ( " r " ) ]
internal class RecipientWellItem
{
    [ EventField ( " dn " ) ]
    public string DisplayName ;
    [ EventField ( " em " , true , " oof@pageA.net "
) ]
// true = optional field
    public string Email ;
    [ EventField ( " t " ) ]
    public int Type ;
    internal RecipientWellItem ( )
    {
        DisplayName = " " ;
        Email = " " ;
        Type = 0 ;
    }
}
```

10

【 0 0 3 9 】

20

上記のデータ構造の例は以下に示すように符号化される。

```
< r dn = " display name " em = " email address "
t = " 1 " / >
```

【 0 0 4 0 】

本発明のサーバコンポーネントが特定のデータ構造に従う要求を受信するとき、本発明は前記特定データ構造のクラスのインスタンスを生成し、要求からのデータでそれをポピュレートする。

【 0 0 4 1 】

動作環境の例

図 5 はコンピュータ 1 3 0 の形態の汎用目的コンピューティングデバイスの 1 例を示す。本発明の 1 実施形態において、コンピュータ 1 3 0 等のコンピュータは、例示した他の図面、および本明細書で説明した他の図面における使用に適切である。コンピュータ 1 3 0 は 1 つまたは複数のプロセッサまたは演算装置 1 3 2 およびシステムメモリ 1 3 4 を有する。図示した実施形態において、システムバス 1 3 6 は、システムメモリ 1 3 4 を含む様々なシステムコンポーネントを演算装置 1 3 2 に結合する。バス 1 3 6 は、数種のバス構造の任意の 1 つまたは複数を表し、そのバス構造は、メモリバスまたはメモリコントローラ、周辺バス、加速グラフィックポート、および演算装置または様々な任意のバスアーキテクチャを使用するローカルバスを含む。例をあげるがこれに限定されず、上記のアーキテクチャは業界標準アーキテクチャ (I S A) バス、マイクロチャネルアーキテクチャ (M C A) バス、拡張 I S A (E I S A) バス、ビデオ電子標準協会 (V E S A) ローカルバス、およびメザニンバスとしても知られる周辺コンポーネント相互接続 (P C I) バスを含む。

30

40

【 0 0 4 2 】

コンピュータ 1 3 0 は典型的には少なくとも何らかの形態のコンピュータ読取可能な記録媒体を有する。コンピュータ読取可能な記録媒体は、揮発性および不揮発性媒体の両方、取外し可能および取外し不可能媒体の両方を含み、コンピュータ 1 3 0 がアクセス可能な任意の利用可能媒体であってもよい。例をあげるがこれに限定されず、コンピュータ読取可能な記録媒体はコンピュータ記憶媒体および通信媒体を備える。コンピュータ記憶媒体は揮発性および不揮発性、取外し可能および取外し不可能媒体を含み、それらはコンピュータ可読命令、データ構造、プログラムモジュールまたは他のデータ等の情報を格納す

50

るための任意の方法または技術で実装される。例えば、コンピュータ記憶媒体は R A M、R O M、E E P R O M、フラッシュメモリまたは他のメモリ技術、C D - R O M、デジタル多用途ディスク (D V D) または他の光ディスク記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置または他の磁気記憶デバイス、または所望の情報を格納するために使用できてコンピュータ 1 3 0 がアクセス可能な任意の他の媒体を含む。典型的には、通信媒体はコンピュータ可読命令、データ構造、プログラムモジュール、あるいは搬送波または他のトランスポート機構等の変調データ信号内の他のデータを具現化し、任意の情報配信媒体を含む。当業者は変調データ信号に精通しており、変調データ信号は 1 つまたは複数のその特性集合を有し、信号内の情報を符号化するように変化する。有線ネットワークまたは直接配線通信等の有線媒体、および音響、R F、赤外線、および他の無線媒体等の無線媒体は通信媒体の例である。上記の任意の組合せもコンピュータ読取可能な記録媒体の範囲内に含まれる。

10

【 0 0 4 3 】

システムメモリ 1 3 4 は、取外し可能および / または取外し不可能、揮発性および / または不揮発性メモリの形態でコンピュータ記憶媒体を含む。図示した実施形態において、システムメモリ 1 3 4 は読取専用メモリ (R O M) 1 3 8 およびランダムアクセスメモリ (R A M) 1 4 0 を含む。起動中などにコンピュータ 1 3 0 内の要素間での情報転送を支援する基本ルーチンを含む基本入出力システム 1 4 2 (B I O S) は、典型的には R O M 1 3 8 に格納される。典型的には、R A M 1 4 0 は、演算装置 1 3 2 が直接アクセス可能および / または演算装置 1 3 2 上で現在動作中のデータおよび / またはプログラムモジュールを含む。例として、図 5 にオペレーティングシステム 1 4 4、アプリケーションプログラム 1 4 6、他のプログラムモジュール 1 4 8、およびプログラムデータ 1 5 0 を示すが、これに限定されることはない。

20

【 0 0 4 4 】

コンピュータ 1 3 0 は、また他の取外し可能 / 取外し不可能、揮発性 / 不揮発性コンピュータ記憶媒体も含むことができる。例えば、図 5 は、取外し不可能、不揮発性の磁気媒体を読み書きするハードディスクドライブ 1 5 4 を示す。また、図 5 は、取外し可能な不揮発性の磁気ディスク 1 5 8 を読み書きする磁気ディスクドライブ 1 5 6 と C D - R O M または他の光媒体等の取外し可能、不揮発性の光ディスク 1 6 2 に読み書きする光ディスクドライブ 1 6 0 を示す。動作環境の例で利用できる他の取外し可能 / 取外し不可能、揮発性 / 不揮発性のコンピュータ記憶媒体は、これらに限らないが、磁気テープカセット、フラッシュメモリカード、デジタル多用途ディスク、デジタルビデオテープ、半導体 R A M、半導体 R O M 等を含む。ハードディスクドライブ 1 5 4、磁気ディスクドライブ 1 5 6 および光ディスクドライブ 1 6 0 は、典型的には、インタフェース 1 6 6 等の不揮発性のメモリインタフェースによりシステムバス 1 3 6 に接続される。

30

【 0 0 4 5 】

ドライブまたは他の大容量記憶デバイスおよび、上述した図 5 に示す関連コンピュータ記憶媒体は、コンピュータ読取可能命令、データ構造、プログラムモジュール、およびコンピュータ 1 3 0 に関する他のデータを記憶する。図 5 において、例えば、ハードディスクドライブ 1 5 4 は、オペレーティングシステム 1 7 0、アプリケーションプログラム 1 7 2、他のプログラムモジュール 1 7 4、およびプログラムデータ 1 7 6 を格納するものとして示されている。これらのコンポーネントは、オペレーティングシステム 1 4 4、アプリケーションプログラム 1 4 6、他のプログラムモジュール 1 4 8、およびプログラムデータ 1 5 0 と同一であるかまたはそれらとは異なってもよいことに注意されたい。オペレーティングシステム 1 7 0、アプリケーションプログラム 1 7 2、他のプログラムモジュール 1 7 4、およびプログラムデータ 1 7 6 は、ここでは異なる数字が与えられ、少なくともそれらが異なるコピーであることを示している。

40

【 0 0 4 6 】

ユーザは、キーボード 1 8 0 およびポインティングデバイス 1 8 2 (例えば、マウス、トラックボール、ペン、またはタッチパッド) 等の入力デバイスまたはユーザインタフェ

50

ース選択デバイスを通してコンピュータ130に命令と情報を入力することができる。他の入力デバイス（不図示）は、マイクロフォン、ジョイスティック、ゲームパッド、パラボラアンテナ、スキャナ等を含むことができる。これらおよび他の入力デバイスは、システムバス136に接続されたユーザ入力インタフェース184を通して演算装置132に接続されるが、パラレルポート、ゲームポート、またはユニバーサルシリアルバス（USB）等の他のインタフェースおよびバス構造により接続されてもよい。モニタ188または他のタイプのディスプレイデバイスも、ビデオインタフェース190等のインタフェース経由でシステムバス136に接続される。モニタ188に加えて、コンピュータはしばしばプリンタおよびスピーカ等の他の周辺出力装置（不図示）を含み、それらを出力周辺インタフェース（不図示）を通して接続することができる。

10

【0047】

コンピュータ130は、リモートコンピュータ194等の1つまたは複数のリモートコンピュータに対する論理接続を使用してネットワーク環境で動作することができる。リモートコンピュータ194はパーソナルコンピュータ、サーバ、ルータ、ネットワークPC、ピアデバイスまたは他の共通ネットワークノードであることができ、典型的には、コンピュータ130に関する上記の要素の多くまたは全てを含む。図5に示す論理接続は、ローカルエリアネットワーク（LAN）196および広域ネットワーク（WAN）198を含むが、他のネットワークを含んでもよい。LAN136および/またはWAN138は、有線ネットワーク、無線ネットワーク、それらの組合せ等であることができる。上記のネットワーク環境は、職場、企業規模のコンピュータネットワーク、イントラネット、およびグローバルコンピュータネットワーク（例えば、インターネット）で一般的である。

20

【0048】

ローカルエリアネットワーク環境で使用する際には、コンピュータ130はネットワークインタフェースまたはアダプタ186を通してLAN196に接続される。広域ネットワーク環境で使用する時、典型的には、コンピュータ130は、モデム178、またはインターネット等のWAN198上で通信を確立する他の手段を含む。モデム178は、内部または外部でもよく、ユーザ入力インタフェース184または他の適切な機能経由でシステムバス136に接続される。ネットワーク環境では、コンピュータ130に関して示したプログラムモジュール、またはその一部をリモートメモリ記憶デバイス（不図示）に格納することができる。例えば、図5はリモートアプリケーションプログラム192がメモリデバイス上にあるものとして示すが、これに限定されることはない。示したネットワーク接続は例であって、コンピュータ間で通信リンクを確立する他の手段を使用してもよい。

30

【0049】

一般的に、コンピュータ130のデータプロセッサは、コンピュータの様々なコンピュータ読取可能な記録媒体に異なる時点で格納された命令手段によりプログラムされる。プログラムおよびオペレーティングシステムは、典型的には、例えばフロッピーディスクまたはCD-ROM上に配信される。そこから、プログラムおよびオペレーティングシステムはコンピュータのセカンダリメモリにインストールまたはロードされる。実行時に、プログラムおよびオペレーティングシステムは、少なくとも部分的にはコンピュータのプライマリ電子メモリにロードされる。本明細書で説明する本発明は、これらおよび他の様々なタイプのコンピュータ読取可能な記録媒体を含み、上記媒体は、マイクロプロセッサまたは他のデータプロセッサと関連して以下で説明するステップを実施する命令またはプログラムを含む。本発明は、本明細書で説明する方法および技術に従ってプログラムされるときには、コンピュータそれ自身も含む。

40

【0050】

説明目的で、オペレーティングシステム等のプログラムおよび他の実行可能プログラムコンポーネントを本明細書で離散ブロックとして示す。しかしながら、上記のプログラムおよびコンポーネントはその時々でコンピュータの様々な記憶コンポーネントに存在し、コンピュータのデータプロセッサにより実行されることは理解される。

50

【 0 0 5 1 】

コンピュータ 130を含むコンピューティングシステム環境の例に関連させて説明するが、本発明は多数の他の汎用目的または特殊目的のコンピューティングシステム環境または構成で動作できる。コンピューティングシステム環境は、本発明の使用範囲または機能範囲に関する限定を示唆する意図はない。さらに、コンピューティングシステム環境は、動作環境の例で示したコンポーネントの任意の1つまたは組合せに関する依存性または要件を有すると解釈されるべきではない。本発明での使用に適した周知のコンピューティングシステム、環境、および/または構成の例は、これらに限らないが、パーソナルコンピュータ、サーバコンピュータ、ハンドヘルドまたはラップトップデバイス、マルチプロセッサシステム、マイクロプロセッサベースのシステム、セットトップボックス、プログラム可能家電機器、携帯電話、ネットワークPC、ミニコンピュータ、メインフレームコンピュータ、任意の上記システムまたはデバイスを含む分散コンピューティング環境等を含む。

10

【 0 0 5 2 】

プログラムモジュール等の1つまたは複数のコンピュータまたは他のデバイスが実行するコンピュータ実行可能命令の一般的なコンテキストで、本発明を説明することができる。一般的に、プログラムモジュールはこれらに限らないが、ルーチン、プログラム、オブジェクト、コンポーネント、および特定のタスクを実施するかまたは特定の抽象データ型を実装するデータ構造を含む。本発明は、通信ネットワークを通してリンクされるリモート処理デバイスがタスクを実行する分散コンピューティング環境で実施してもよい。分散コンピューティング環境においては、プログラムモジュールは、メモリ記憶デバイスを含むローカルとリモートとの両方のコンピュータ記憶媒体に位置することができる。

20

【 0 0 5 3 】

ソフトウェアアーキテクチャのコンテキスト内のインタフェースは、ソフトウェアモジュール、コンポーネント、コード部、または他のコンピュータ実行可能命令のシーケンスを含む。前記インタフェースは、例えば、第1のモジュールの代わりにコンピューティングタスクを実施するために、第1のモジュールが第2のモジュールにアクセスすることを含む。第1および第2のモジュールは、1例において、オペレーティングシステムにより提供されるようなアプリケーションプログラミングインタフェース（API）、（例えば、ピアツーピアアプリケーション通信に対する）コンポーネントオブジェクトモデル（COM）インタフェース、および（例えば、ウェブサービス間で通信するための）拡張可能マークアップ言語メタデータ交換フォーマット（XMI）インタフェースを含む。

30

【 0 0 5 4 】

インタフェースは、Java 2プラットフォームエンタープライズエディション（J2EE）、COM、または分散COM（DCOM）の例にあるような密結合の同期実装が可能である。その代替か追加で、インタフェースは、（例えば、シンプルオブジェクトアクセスプロトコルを用いた）ウェブサービスにあるような疎結合の非同期実装が可能である。一般に、インタフェースは以下の特徴の任意の組合せを含む。すなわち、密結合、疎結合、同期、および非同期である。さらに、インタフェースは、標準プロトコル、専用プロトコル、または標準および専用プロトコルの任意の組合せに適合可能である。

40

【 0 0 5 5 】

本明細書で説明したインタフェースは全て、単一インタフェースの一部か、別個のインタフェースまたはそれらの任意の組合せで実行することができる。インタフェースをローカルまたはリモートに実行して機能を提供することができる。さらに、インタフェースは、本明細書で図示または説明したものに比べて、付加した機能性または低減した機能性を含めることができる。

【 0 0 5 6 】

動作中、コンピュータ130は、図2に示すようなコンピュータ実行可能命令を実行して、ネットワークアプリケーションプログラム間で要求されたデータを生成および通信する。

50

【0057】

付録Aは、本発明による例示的な名前空間、イベントハンドラ、およびパラメータを列挙する。付録Aにおいて、DatePickerEventHandlerハンドラ型は日付ピッカーに関するイベントを含む。日付ピッカーはユーザインタフェースコンポーネントで、これによりユーザはアプリケーションのいくつかのコンテキストで日付を選択することができる。RenderMonthイベントは、月に対応するHTMLフラグメントをレンダリングする。ユーザが2005年1月に移動するようクリックするとき、本発明のクライアントコンポーネントはRenderMonthイベントをサーバに送信し、サーバは入力パラメータ（例えば、2005年1月の日付である月を意味する‘m’）を取得して、その月に対するHTMLフラグメントをレンダリングする。レンダリングされたHTMLフラグメントはクライアントに送り返される。

10

【0058】

MessageListViewEventHandlerハンドラ型は、例えば受信箱の中のメッセージを示すメッセージリストビューに関するイベントを含む。削除イベントは、削除する2、3個のメッセージをユーザが選択するときに実行される。フォルダ識別子は属性‘fld’で表される。削除するためのメッセージ識別子の配列は、属性‘id’で表される。

【0059】

RecipientWellEventHandlerは、受信者のためのイベントハンドラに適している。このハンドラ型におけるイベントの例は‘resolve（解決）’である。‘resolve’は電子メールメッセージ内のto（宛先）、cc、およびbcc領域を文字配列であるパラメータとして取る。Resolveイベントはこれらの文字列を解析して解決する。例えば、ユーザが“BobS”とタイプすると、‘to’パラメータに対応する配列内の単一項目“BobS”のデータ要求がサーバに送信される。応答はHTML内にBob Smithの電子メールアドレス付きの“Bob Smith”を有するHTML部分フラグメントであり、それに他のデータが添付されてもよい。

20

【0060】

次にカレンダービューイベントハンドラ内のGetViewPayloadメソッドの詳細な説明を行う。このメソッドに対するパラメータは、属性‘nvs’、‘fld’、‘days’、‘vt’および‘dir’を含む。このイベントハンドラは、カレンダービューがリフレッシュされる必要があるとき、クライアントから呼び出される。カレンダービューは、30分刻みでディスプレイの左にタイムストリップを表示し、それぞれの日付は列内に表示する。

30

【0061】

カレンダービューをリフレッシュする必要があるとき、GetViewPayloadイベントが起動される。このイベントはデータをバックエンドからロードし（例えば、メールボックス内のユーザのカレンダー内で問い合わせ）、そのデータを記述するコード（例えば、JavaScript）をレンダリングする。レンダリングしたコードはペイロードと呼ばれる。ペイロードはクライアントに送信され、クライアントはペイロードデータを用いてビューを再描画する。ペイロードの例は、それぞれの予約の開始/終了時間、題目、位置、空き/ビジネステータス等を含む。また、それぞれの長方形の座標はサーバで計算され、このペイロードの一部として送信されて、クライアントサイドスクリプトを簡略化する。

40

【0062】

GetViewPayloadメソッドに関するパラメータは属性‘nvs’、‘fld’、‘days’、‘vt’および‘dir’を含む。‘nvs’属性は‘ビュー状態なし’属性であり、真の場合、ビュー状態を維持する必要が全くないことを示す。ビュー状態はビューを定義するパラメータ集合（例えば、閲覧をした日数およびビューのタイプ）である。ビュータイプは週次または日次が可能である。‘nvs’パラメータが偽のとき、ビュー状態はユーザが次にログオンするときのために保存される。‘fld’属性は

50

、参照するカレンダーフォルダの識別子に対応する“フォルダ識別子”属性である。‘day’属性は参照日付の配列を含む。‘vt’属性は日次または週次の“ビュータイプ”に対応する。‘dir’属性は随意に指定されるナビゲーション矢印の“方向”である。この属性を設定することは、ユーザがナビゲーション矢印をクリックし、サーバが‘day’属性に基づいて閲覧すべき次の日付集合を計算しなければならないことを意味する。例えば、‘day’属性が3月の2日、3日、および4日であり、‘dir’属性が+1に設定される場合には、サーバはそれらの各々に7日を足す（例えば、複数日ビューは7日進む）。‘day’属性が3月2日のみ（例えば、単一日付ビュー）の場合、右矢印をクリックすると3月3日に移動し、従ってサーバは1日を加えるのみである。

【0063】

10

GetViewPayloadメソッドに対する要求の例を以下に示す。

```
POST/mailclient/ev.owa?oeh=1&ns=CalendarView&ev=GetViewPayload
```

```
HTTP/1.1
```

```
BODY:
```

```
<param><fld>LgAAAAABjcZgVWhhRQKW7OM2Ok4GD  
DAQD4If+Jk6V9Saz9+rfNa9e8AAAAABe
```

```
dAAAB</fld><days><item>2004-12-15T00:00  
:00</item><item>2004-12-16T00:00:00</ite  
m><item>2004-12-17T00:00:00</item></days  
><vt>1</vt><nvs>1</nvs></params>
```

20

【0064】

上記要求に対する応答の例の一部を以下に示す。

```
a__cERow=1;a__sFTtl="Calendar:December200  
4";a__iNRP=1;a__rgCal=new
```

```
Array(nc("RgAAAAABjctheWhhRQKW7OM2Ok4GDBw  
D4If+Jk6V9Saz9+rfNa9e8AAAAABedAAD4If+Jk6  
V9Saz9+rfNa9e8AAAAADTfAAAI","12-15-2004
```

```
17:30:00U","12-15-200418:00:00U","Strate  
gy Meeting","Conf Room 1",2,1,"Bob Smith  
",0,1,0,0,""),nc("CAjGsFAnk8AARgAAAAABjcZ
```

30

```
gthehRQKW7OM2Ok4GDBwD4If+Jk6V9Saz9+rfNa9  
e8AAAAABedAAD4If+Jk6V9Saz9+rfNa9e8AAAAAC  
NqAAAJ","12-15-2004 21:00:00U","12-15-20
```

```
04 21:30:00U","Out of Office","",2,0,"",  
1,3,1,0,""),nc("RgAAAAABjcZgVWhhRQKW7OM2O  
k4GDBwD4If+Jk6V9Saz9+rfNa9e8AAAAABedAAD4
```

```
If+Jk6V9Saz9+rfNa9e8AAAAADThAAAI","12-16  
-2004 00:00:00U","12-16-2004 01:00:00U",  
"Portfolio Development Meeting","Conf Ro
```

40

```
om 2",2,1,"Joe A.",1,1,0,0,""));a__rgDay=  
new Array();var
```

```
t;t=newCalDay("15","Wednesday",15,12,200  
4);av(t,0,19,1,1,0);av(t,0,24,0.5,4,1);  
av(t,0.5,26,0.5,1,2);av(t,0,32,1,2,3);av
```

```
(t,0,34,1,2,4);a__rgDay[0]=t;t=newCalDay(  
"16","Thursday",16,12,2004);av(t,0,20,1,  
1,5);av(t,0,21,1,2,6);av(t,0,23,1,2,7);a
```

```
v(t,0,25,1,1,8);av(t,0,28,0.5,6,10);av(t  
,0.5,28,0.5,4,9);av(t,0.5,33,0.5,3,11);a
```

50

```

__rgDay[1] = t; t = new CalDay( "17", "Friday", 17
, 12, 2004); av(t, 0, 20, 0.5, 4, 13); av(t, 0.5, 2
0, 0.5, 2, 12); av(t, 0.5, 23, 0.5, 3, 14); av(t, 0
, 25, 0.5, 1, 15); av(t, 0, 28, 0.5, 2, 17); av(t, 0
.5, 29, 0.5, 3, 18); avf(t, 26, 22, 3); a__rgDay[2
] = t; a__rgEvt = new Array(ne(2, 0, 1, 16, 2, 0))

```

【0065】

本明細書で図示および説明した方法の実行順序または性能は、明記しない限り本質的ではない。すなわち、前記方法の要素は明記しない限り任意の順序で実施することができ、前記方法は本明細書で開示したものよりも多い、あるいは少ない要素を含むことができる。例えば、別の要素の前に、別の要素と同時にまたは別の要素の後に、特定要素を実行または実施することは、本発明の範囲内であると意図している。

【0066】

本発明または本発明の実施形態の要素を導入するとき、冠詞“a”、“an”、“the”、および“said”は1つまたは複数の要素があることを意味するよう意図される。用語“comprising”、“including”、および“having”は包括的であることを意図し、列挙した要素以外にさらなる要素がありうることを意味するよう意図している。

【0067】

上記の観点から、本発明のいくつかの目的が達成され他の有利な結果が得られることが分かるであろう。

【0068】

上記の構築、製品、および方法に対して本発明の範囲から逸脱することなく様々な変更を加えることができるが、上記の説明に含まれる全ての事柄および添付図面に示した全ての事柄は限定の意味はなく、例として解釈されるべきであることを意図している。

【0069】

付録A

サーバ配置電子メールアプリケーションプログラムに対する電子メール名前空間（例えば、Mail.Client）におけるイベントおよびパラメータの例を、以下の表に示す。

【0070】

【表1】

ハンドラタイプ	‘Mail.Clients.DatePickerEventHandler’
イベントハンドラ名	‘RenderMonth’.Verb: ‘Get’
パラメータ名	‘uF’, type: ‘System.Int32’, isArray: ‘False’, isOptional: ‘False’, isStruct: ‘False’
パラメータ名	‘m’, type: ‘System.DateTime’, isArray: ‘False’, isOptional: ‘False’, isStruct: ‘False’
イベントハンドラ名	‘GetFreeBusy’.Verb: ‘Get’
パラメータ名	‘Month’, type: ‘System.DateTime’, isArray: ‘False’, isOptional: ‘False’, isStruct: ‘False’

表A1 日付ピッカーイベントに対するイベントハンドラおよびパラメータの例

【0071】

【表 2 - 1】

ハンドラタイプ	'Mail.Clients.MessageListViewEventHandler'
イベントハンドラ名	'Delete'.Verb: 'Post'
パラメータ名	'fld', type: 'Mail.Objects.IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'id', type: 'Mail.Objects.IUniqueItemId', isArray: 'True', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'PermanentDelete'.Verb: 'Post',
パラメータ名	'id', type: 'Mail.Objects.IUniqueItemId', isArray: 'True', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'MarkAsRead'.Verb: 'Post'
パラメータ名	'id', type: 'Mail.Objects.IUniqueItemId', isArray: 'True', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'MarkAsUnread'.Verb: 'Post'
パラメータ名	'id', type: 'Mail.Objects.IUniqueItemId', isArray: 'True', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'PersistWidth', Verb: 'Post'
パラメータ名	'fld', type: 'Mail.Objects.IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'w', type: 'System.Int32', isArray: 'False', isOptional: 'False', isStruct: 'False'

10

20

30

40

【 0 0 7 2 】

【表 2 - 2】

イベントハンドラ名	'PersistHeight'. Verb: 'Post'
パラメータ名	'fld', type: 'Mail.Objects.UniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'h', type: 'System.Int32', isArray: 'False', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'PersistReadingPane'. Verb: 'Post'
パラメータ名	's', type: 'Mail.Clients.Common.ReadingPanePosition', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'fld', type: 'Mail.Objects.UniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'Refresh'. Verb: 'Get'
パラメータ名	'sId', type: 'Mail.Objects.UniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sO', type: 'Mail.Objects.SortOrder', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sC', type: 'Mail.Clients.Controls.ColumnId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'mL', type: 'System.Boolean', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sR', type: 'System.Int32', isArray: 'False', isOptional: 'False', isStruct: 'False'

10

20

30

40

【0073】

【表 2 - 3】

イベントハンドラ名	‘SeekNextItem’. Verb: ‘Get’
パラメータ名	‘sR’, type: ‘System. Int32’, isArray: ‘False’, isOptional: ‘False’, isStruct: ‘False’
パラメータ名	‘mL’, type: ‘System. Boolean’, isArray: ‘False’, isOptional: ‘False’, isStruct: ‘False’
パラメータ名	‘sId’, type: ‘Mail. Objects. IUniqueItemId’, isArray: ‘False’, isOptional: ‘False’, isStruct: ‘False’
パラメータ名	‘mId’, type: ‘Mail. Objects. IUniqueItemId’, isArray: ‘False’, isOptional: ‘False’, isStruct: ‘False’
パラメータ名	‘sC’, type: ‘Mail. Clients. Controls. ColumnId’, isArray: ‘False’, isOptional: ‘False’, isStruct: ‘False’
パラメータ名	‘sO’, type: ‘Mail. Objects. SortOrder’, isArray: ‘False’, isOptional: ‘False’, isStruct: ‘False’
イベントハンドラ名	‘SeekPreviousItem’. Verb: ‘Get’,
パラメータ名	‘mL’, type: ‘System. Boolean’, isArray: ‘False’, isOptional: ‘False’, isStruct: ‘False’
パラメータ名	‘sId’, type: ‘Mail. Objects. IUniqueItemId’, isArray: ‘False’, isOptional: ‘False’, isStruct: ‘False’
パラメータ名	‘mId’, type: ‘Mail. Objects. IUniqueItemId’, isArray: ‘False’, isOptional: ‘False’, isStruct: ‘False’
パラメータ名	‘sR’, type: ‘System. Int32’, isArray: ‘False’, isOptional: ‘False’, isStruct: ‘False’

10

20

30

40

【 0 0 7 4 】

【表 2 - 4】

パラメータ名	'sC', type: 'Mail.Clients.Controls.ColumnId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sO', type: 'Mail.Objects.SortOrder', isArray: 'False', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'ToggleMultiVsSingleLine'. Verb: 'Get'
パラメータ名	'sId', type: 'Mail.Objects.IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'mL', type: 'System.Boolean', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sR', type: 'System.Int32', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sC', type: 'Mail.Clients.Controls.ColumnId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sO', type: 'Mail.Objects.SortOrder', isArray: 'False', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'FirstPage'. Verb: 'Get'
パラメータ名	'sId', type: 'Mail.Objects.IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sR', type: 'System.Int32', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sC', type: 'Mail.Clients.Controls.ColumnId', isArray: 'False', isOptional: 'False', isStruct: 'False'

10

20

30

40

【 0 0 7 5 】

【表 2 - 5】

パラメータ名	'sO', type: 'Mail. Objects. Sort Order', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'mL', type: 'System. Boolean', isArray: 'False', isOptional: 'True', isStruct: 'False'
イベントハンドラ名	'PreviousPage'. Verb: 'Get'
パラメータ名	'sR', type: 'System. Int32', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sId', type: 'Mail. Objects. IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sC', type: 'Mail. Clients. Controls. ColumnId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sO', type: 'Mail. Objects. Sort Order', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'mL', type: 'System. Boolean', isArray: 'False', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'NextPage'. Verb: 'Get'
パラメータ名	'sC', type: 'Mail. Clients. Controls. ColumnId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sO', type: 'Mail. Objects. Sort Order', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'mL', type: 'System. Boolean', isArray: 'False', isOptional: 'False', isStruct: 'False'

10

20

30

40

【 0 0 7 6 】

【表 2 - 6】

パラメータ名	'sId', type: 'Mail.Objects.IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sR', type: 'System.Int32', isArray: 'False', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'LastPage'. Verb: 'Get'
パラメータ名	'sId', type: 'Mail.Objects.IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'mL', type: 'System.Boolean', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'tC', type: 'System.Int32', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sC', type: 'Mail.Clients.Controls.ColumnId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sO', type: 'Mail.Objects.SortOrder', isArray: 'False', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'Sort'. Verb: 'Get'
パラメータ名	'sC', type: 'Mail.Clients.Controls.ColumnId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'sO', type: 'Mail.Objects.SortOrder', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'mL', type: 'System.Boolean', isArray: 'False', isOptional: 'False', isStruct: 'False'

10

20

30

40

【 0 0 7 7 】

【表 2 - 7】

パラメータ名	'sR', type: 'System. Int32', isArray: 'False', isOptional: 'True', isStruct: 'False'
パラメータ名	'sId', type: 'Mail. Objects. IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'MsgToPeople'. Verb: 'Post'
パラメータ名	'id', type: 'Mail. Objects. IUniqueItemId', isArray: 'True', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'Move'. Verb: 'Post'
パラメータ名	'destId', type: 'Mail. Objects. IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'id', type: 'Mail. Objects. IUniqueItemId', isArray: 'True', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'Copy'. Verb: 'Post'
パラメータ名	'destId', type: 'Mail. Objects. IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'id', type: 'Mail. Objects. IUniqueItemId', isArray: 'True', isOptional: 'False', isStruct: 'False'

10

20

30

表A2 メッセージリストビューイベントハンドラに対するイベントハンドラおよびパラメータの例

【0078】

【表 3】

ハンドラタイプ	'Mail.Clients.RecipientWellEventHandler'
イベントハンドラ名	'Resolve'.Verb:'Post'
パラメータ名	'Bcc', type:'System.String', isArray:'True', isOptional:'True', isStruct:'False'
パラメータ名	'Cc', type:'System.String', isArray:'True', isOptional:'True', isStruct:'False'
パラメータ名	'To', type:'System.String', isArray:'True', isOptional:'True', isStruct:'False'
パラメータ名	'Recips', type:'Mail.Clients.Common.RecipientInfoAC', isArray:'True', isOptional:'True', isStruct:'True'
イベントハンドラ名	'ResolveOneRecipient'.Verb:'Post'
パラメータ名	'n', type:'System.String', isArray:'False', isOptional:'False', isStruct:'False'
イベントハンドラ名	'GetRecipProps'.Verb:'Post'
パラメータ名	'id', type:'System.String', isArray:'False', isOptional:'False', isStruct:'False'
パラメータ名	'rs', type:'Mail.Clients.Common.RecipientSource', isArray:'False', isOptional:'False', isStruct:'False'
イベントハンドラ名	'GetCache'.Verb:'Get'
イベントハンドラ名	'DeleteFromCache'.Verb:'Post'
パラメータ名	'Em', type:'System.String', isArray:'False', isOptional:'False', isStruct:'False'

10

20

30

表3 受信者のためのイベントハンドラに対するイベントハンドラおよびパラメータの例

40

【0079】

【表 4 - 1】

ハンドラタイプ	'Mail.Clients.CalendarViewEventHandler'
イベントハンドラ名	'PersistWidth'.Verb: 'Post'
パラメータ名	'w', type: 'System.Int32', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'fId', type: 'Mail.Objects.IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'PersistHeight'.Verb: 'Post'
パラメータ名	'fId', type: 'Mail.Objects.IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'h', type: 'System.Int32', isArray: 'False', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'PersistReadingPane'.Verb: 'Post'
パラメータ名	'isMD', type: 'System.Boolean', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	's', type: 'Mail.Clients.Common.ReadingPanePosition', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'fId', type: 'Mail.Objects.IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
イベントハンドラ名	'GetViewPayload'.Verb: 'Post'
パラメータ名	'nvs', type: 'System.Boolean', isArray: 'False', isOptional: 'True', isStruct: 'False'

10

20

30

【 0 0 8 0 】

40

【表 4 - 2】

パラメータ名	'fld', type: 'Mail. Objects. IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'days', type: 'System. DateTime', isArray: 'True', isOptional: 'True', isStruct: 'False'
パラメータ名	'vt', type: 'Mail. Clients. Controls. CalendarViewType', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'dir', type: 'System. Int32', isArray: 'False', isOptional: 'True', isStruct: 'False'
イベントハンドラ名	'Move'. Verb: 'Post'
パラメータ名	'et', type: 'System. DateTime', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'fld', type: 'Mail. Objects. IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'days', type: 'System. DateTime', isArray: 'True', isOptional: 'True', isStruct: 'False'
パラメータ名	'nvs', type: 'System. Boolean', isArray: 'False', isOptional: 'True', isStruct: 'False'
パラメータ名	'id', type: 'Mail. Objects. IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'
パラメータ名	'st', type: 'System. DateTime', isArray: 'False', isOptional: 'False', isStruct: 'False'

10

20

30

【 0 0 8 1 】

40

【表 4 - 3】

パラメータ名	<code>'vt', type: 'Mail.Clients.Controls.CalendarViewType', isArray: 'False', isOptional: 'False', isStruct: 'False'</code>
イベントハンドラ名	<code>'Delete'. Verb: 'Post'</code>
パラメータ名	<code>'nvs', type: 'System.Boolean', isArray: 'False', isOptional: 'True', isStruct: 'False'</code>
パラメータ名	<code>'vt', type: 'Mail.Clients.Controls.CalendarViewType', isArray: 'False', isOptional: 'False', isStruct: 'False'</code>
パラメータ名	<code>'id', type: 'Mail.Objects.IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'</code>
パラメータ名	<code>'fld', type: 'Mail.Objects.IUniqueItemId', isArray: 'False', isOptional: 'False', isStruct: 'False'</code>
パラメータ名	<code>'days', type: 'System.DateTime', isArray: 'True', isOptional: 'True', isStruct: 'False'</code>

10

20

表4 カレンダービューイベントハンドラに対するイベントハンドラおよびパラメータの例

【図面の簡単な説明】

30

【0082】

【図1】1つまたは複数のクライアントコンポーネントとネットワーク経由で通信するサーバコンポーネントを例示するブロック図である。

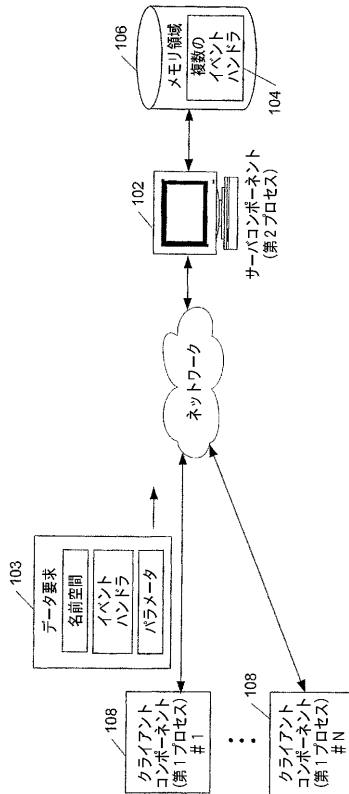
【図2】要求されたデータを通信するためのクライアントおよびサーバプロセスの動作を例示する流れ図である。

【図3】サーバコンポーネント内のモジュールを説明、例示するブロック図である。

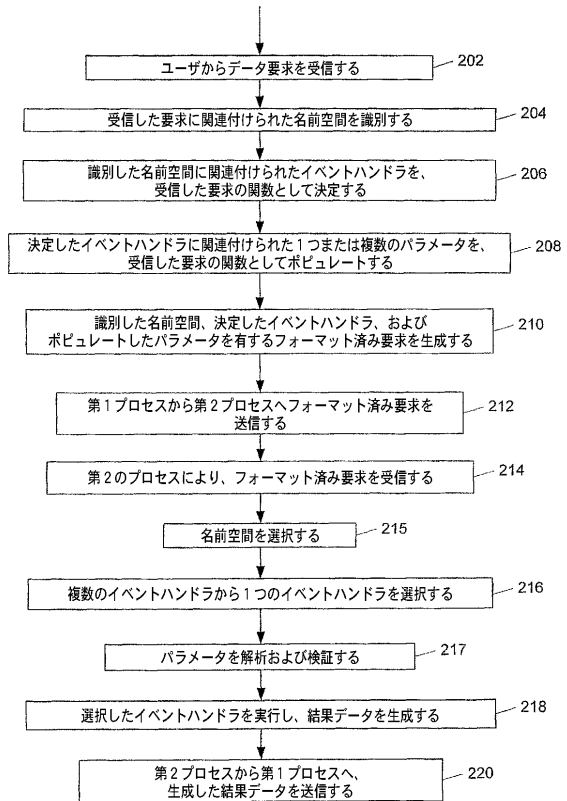
【図4】イベントハンドラクラスの集合を説明、例示するブロック図である。

【図5】本発明を実装可能な適切なコンピューティングシステム環境の1例を説明するブロック図である。

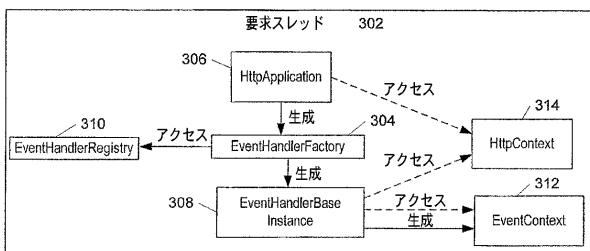
【 図 1 】



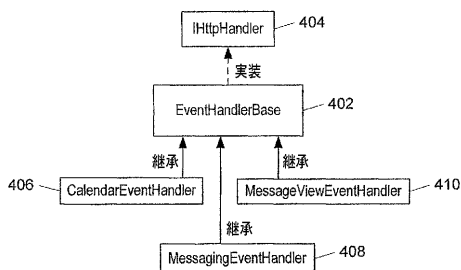
【 図 2 】



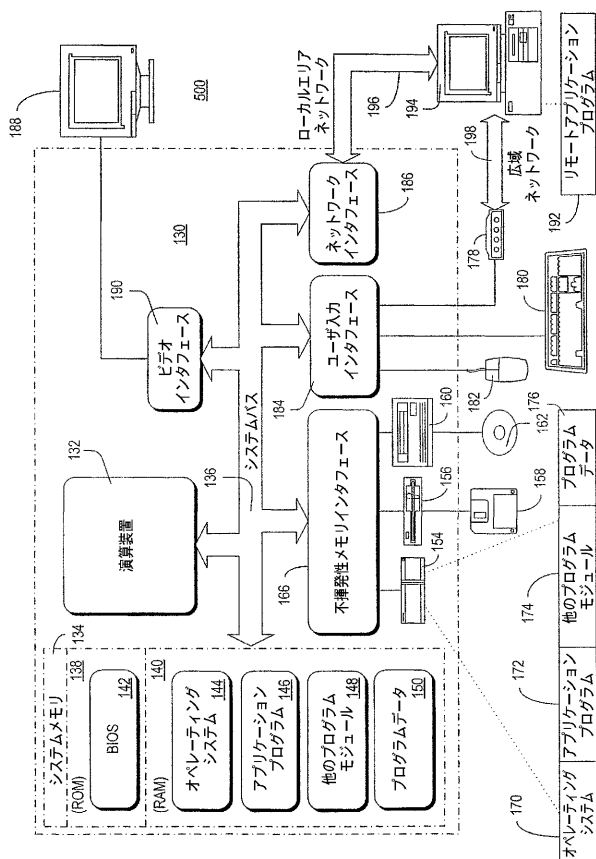
【 図 3 】



【 図 4 】



【 図 5 】



フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LS, MW, MZ, NA, 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, IS, IT, LT, LU, LV, MC, NL, PL, 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, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW

(特許庁注：以下のものは登録商標)

- 1 . フロッピー
- 2 . J A V A