(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5060018号
(P5060018)
(45)発行日　平成24年10月31日(2012.10.31)　(24)登録日　平成24年8月10日(2012.8.10)

(51)Int.Cl.　F I
G06F 13/00 (2006.01)　G06F 13/00　353C

請求項の数 4　(全 36 頁)

(21)出願番号	特願2004-275873 (P2004-275873)
(22)出願日	平成16年9月22日 (2004.9.22)
(65)公開番号	特開2005-129029 (P2005-129029A)
(43)公開日	平成17年5月19日 (2005.5.19)
審査請求日	平成19年9月25日 (2007.9.25)
審判番号	不服2011-7048 (P2011-7048/J1)
審判請求日	平成23年4月4日 (2011.4.4)
(31)優先権主張番号	10/692,384
(32)優先日	平成15年10月23日 (2003.10.23)
(33)優先権主張国	米国 (US)

(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　９８０５２－６３９９　レッドモンド　ワン　マイクロソフト　ウェイ
(74)代理人　110001243
特許業務法人　谷・阿部特許事務所
(74)復代理人　100115624
弁理士　濱中　淳宏
(74)復代理人　100129171
弁理士　柿沼　健一

最終頁に続く

(54)【発明の名称】一対多のデータ投影のシステムおよび方法

(57)【特許請求の範囲】
【請求項１】
プレゼンタのマシンである投影デバイスから複数の参加者のマシンに対してプレゼンテーションを提供する方法であって、
前記投影デバイスが、前記プレゼンテーションについてのプレゼンテーションセッションを開始するステップであって、前記プレゼンテーションセッションは前記複数の参加者のマシンに各々対応する複数の投影ターゲットデバイスを接続するためのセッションである、ステップと、
前記投影デバイスが、前記プレゼンテーションについてのプレゼンテーションデバイスを生成して当該投影デバイスに登録するステップであって、前記プレゼンテーションデバイスは前記複数の投影ターゲットデバイスが見出すことが可能なＵＰｎＰデバイスであり、前記プレゼンテーションデバイスは一意的なデバイス名を有し、前記プレゼンテーションデバイスは招待状を要求するアクションを実行するメソッドを含む、ステップと、
前記投影デバイスが、前記投影ターゲットデバイスから、前記投影ターゲットデバイスの一意的なデバイス名を伴う許可要求を受け取るステップと、
前記投影デバイスが、前記許可要求に応答して前記プレゼンテーションに対する招待状を生成するステップであって、
前記投影デバイスが、前記許可要求に応答して、前記招待状を要求するアクションを実行するメソッドを実行して、招待状を生成するＡＰＩのポインタを得て、
前記投影デバイスが、前記プレゼンテーションデバイスの一意的なデバイス名を入力

として前記ポインタで示された前記招待状を生成するＡＰＩを使用して招待状を生成し、ここで前記投影ターゲットデバイスの一意的なデバイス名によって識別されるプロパティを含む招待状が生成される、ステップと、

前記投影デバイスが、前記投影ターゲットデバイスに対して前記招待状を送信するステップと、

前記投影デバイスが、前記プレゼンテーションデバイスと前記複数の投影ターゲットデバイスとの間を接続する端末サービスセッションを生成して前記複数の参加者が前記プレゼンテーションを閲覧できるようにするステップと

を含むことを特徴とする方法。

【請求項２】

前記投影デバイスが、プロジェクタ装置に対応するＵＰｎＰデバイスである少なくとも１つのプロジェクタデバイスを求めてＳＳＤＰ探索を実施して、前記少なくとも１つのプロジェクタデバイス見出すステップと、

前記投影デバイスが、前記少なくとも１つのプロジェクタデバイスのうちの少なくとも１つを選択するステップと、

前記投影デバイスが、前記選択されたプロジェクタデバイスに対応するプロジェクタ装置との接続を確立するステップと、

前記投影デバイスが、前記選択されたプロジェクタデバイスについて招待状を生成するステップと、

前記投影デバイスが、前記選択されたプロジェクタデバイスに対して招待状を送付するステップと、

前記投影デバイスが、前記レゼンテーションセッションと前記選択されたプロジェクタデバイスに対応するプロジェクタ装置との間を接続する接続メソッドを開始するステップであって、それによってその前記選択されたプロジェクタデバイスに対応するプロジェクタ装置上の端末サービスクライアントは、前記端末サービスセッションとの接続を確立する、ステップと、

前記投影デバイスが、前記選択されたプロジェクタデバイスに対応するプロジェクタ装置のディスプレイ設定を設定するステップであって、

前記投影デバイスが、前記選択されたプロジェクタデバイスによって提供される当該プロジェクタ装置のディスプレイ設定を取得するためのメソッドを実行し、

前記投影デバイスが、取得した前記プロジェクタのディスプレイ設定に基づいて前記選択されたプロジェクタデバイスのディスプレイ設定を設定する、ステップと

をさらに含むことを特徴とする請求項１に記載の方法。

【請求項３】

プレゼンタのコンピュータである投影デバイスから複数の参加者のマシンに対してプレゼンテーションを提供するためのプログラムを記録したコンピュータ読取可能な記録媒体であって、前記プログラムは、前記投影デバイスに、

前記プレゼンテーションについてのプレゼンテーションセッションを開始するステップであって、前記プレゼンテーションセッションは前記複数の参加者のマシンに各々対応する複数の投影ターゲットデバイスを接続するためのセッションである、ステップと、

前記プレゼンテーションについてのプレゼンテーションデバイスを生成して当該投影デバイスに登録するステップであって、前記プレゼンテーションデバイスは前記複数の投影ターゲットデバイスが見出すことが可能なＵＰｎＰデバイスであり、前記プレゼンテーションデバイスは一意的なデバイス名を有し、前記プレゼンテーションデバイスは招待状を要求するアクションを実行するメソッドを含む、ステップと、

前記投影ターゲットデバイスから、当該投影ターゲットデバイスの一意的なデバイス名を伴う許可要求を受け取るステップと、

前記許可要求に応答して前記プレゼンテーションに対する招待状を生成するステップであって、

前記許可要求に応答して、前記招待状を要求するアクションを実行するメソッドを実

行して、招待状を生成するＡＰＩのポインタを得て、
　前記プレゼンテーションデバイスの一意的なデバイス名を入力として前記ポインタで示された前記招待状を生成するＡＰＩを使用して招待状を生成し、ここで前記投影ターゲットデバイスの一意的なデバイス名によって識別されるプロパティを含む招待状が生成される、ステップと、
　前記投影デバイスが、前記投影ターゲットデバイスに対して前記招待状を送信するステップと、
　前記投影デバイスが、前記プレゼンテーションデバイスと前記複数の投影ターゲットデバイスとの間を接続する端末サービスセッションを生成して前記複数の参加者が前記プレゼンテーションを閲覧できるようにするステップと
　を実行させることを特徴とする記録媒体。
【請求項４】
　前記プログラムは、前記投影デバイスに、
　プロジェクタ装置に対応するＵＰｎＰデバイスである少なくとも１つのプロジェクタデバイスを求めてＳＳＤＰ探索を実施して、前記少なくとも１つのプロジェクタデバイス見出すステップと、
　前記少なくとも１つのプロジェクタデバイスのうちの少なくとも１つを選択するステップと、
　前記選択されたプロジェクタデバイスに対応するプロジェクタ装置との接続を確立するステップと、
　前記選択されたプロジェクタデバイスについて招待状を生成するステップと、
　前記選択されたプロジェクタデバイスに対して招待状を送付するステップと、
　前記レゼンテーションセッションと前記選択されたプロジェクタデバイスに対応するプロジェクタ装置との間を接続する接続メソッドを開始するステップであって、それによってその前記選択されたプロジェクタデバイスに対応するプロジェクタ装置上の端末サービスクライアントは、前記端末サービスセッションとの接続を確立する、ステップと、
　前記選択されたプロジェクタデバイスに対応するプロジェクタ装置のディスプレイ設定を設定するステップであって、
　前記選択されたプロジェクタデバイスによって提供される当該プロジェクタ装置のディスプレイ設定を取得するためのメソッドを実行し、
　取得した前記プロジェクタのディスプレイ設定に基づいて前記選択されたプロジェクタデバイスのディスプレイ設定を設定する、ステップと
　をさらに実行させることを特徴とする請求項３に記載の記録媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
　本発明は、一般に情報プレゼンテーションに関し、より詳細には１人のプレゼンタから多数の参加者への情報投影のためのシステムおよび方法、ならびに投影デバイスに関する。
【背景技術】
【０００２】
　近年実現された技術的進歩にもかかわらず、今日の職場においては、従来の多数のビジネス活動が依然として顕著な位置を占めている。かかる１つの活動が、ミーティングまたは会議である。しばしば、かかる共同活動には、１人の参加者による他の幾人かの参加者への資料のプレゼンテーションが必要とされることになる。情報を提示する従来の手段には、スクリーン上への投影、モニタまたは他の光能動デバイス上のディスプレイ、あるいは白板、黒板またはイーゼル上などハードコピー形式のプレゼンテーションが含まれる。これらの各方法には、コスト、複雑さ、効果の観点から利点と、欠点があるが、投影およびディスプレイは一般に、スライドまたはビデオを介して提示できるような高速に変化するデータにとって最も適している。さらに、投影およびディスプレイはまた、マイクロソ

フト（登録商標）ブランドのＰｏｗｅｒＰｏｉｎｔスライドプレゼンテーション中などコンピュータまたは他のコンピューティングデバイスの画面からの情報を共有するためにも理想的である。

【０００３】

投影技術のユーザおよび参加者にとっては残念なことに、コンピューティングデバイスからの情報の投影およびディスプレイについての従来の機構は、開始された後には非常に効果的であるが、ＶＧＡケーブルなどのコードの接続、および様々なパラメータの設定を要し、しばしば設定することが複雑であった。この複雑さは、プレゼンテーション前およびプレゼンテーション中に面倒な問題をもたらし、しばしばプレゼンテーションを開始する際に遅延をもたらす可能性がある。さらに、かかる機構によって、１人の話者から、別の話者へのプレゼンテーションの簡単で迅速な制御移転が可能ではなくなっている。したがって、例えば、第１の話者が、ＰＣを使用してＰＯＷＥＲ　ＰＯＩＮＴのスライドショウを提示しており、第２の話者にその場を譲ろうと思う場合には、第２の話者は、一般に自分の椅子を離れて第１の話者のＰＣの隣の場所へと歩いていく必要があり、その時間の間、他の参加者にとっては、時間が途切れてしまい、遅延が生じ、また気が散ることになる。

【０００４】

別の問題は、かかるミーティングの間に提示される情報のセキュリティに関係している。特に、ある出席者が物理的に部屋に存在し、またはプレゼンテーションが催される部屋の中を見ることができるとすれば、提示された情報を閲覧するその人の能力を制限しようがない。物理的な方法を使用してプレゼンテーションの物理的な場所への初期のアクセスを防止することができるが、かかる方法は、前述のように複数のプレゼンテーションが同じ場所で催されるときにはセキュリティを補償しない。すなわち、ある人がプレゼンテーションの場所にいるとすれば、その場所で催されるすべてのプレゼンテーションを目にすることができるのである。

【０００５】

この問題に対処するために、多数のセミナーの場所では、いくつかの異なるより小さなプレゼンテーションの場所または部屋を利用している。このようにして、各プレゼンテーションに対して許可された個人を、より注意深く選別することができる。しかし、これによってプレゼンタのセキュリティ問題には対処できるが、これによってしばしば参加者がこのプレゼンテーションに出席することが困難になってしまう。これらの参加者には、今や所望のプレゼンテーションを閲覧するために場所から場所へと物理的に移動し、各現場で彼らの資料を包みに入れ、包みから出す必要がある。さらに、身体障害をもつ参加者は、プレゼンテーションの開始に間に合わない可能性があり、またはあるプレゼンテーションを早めに離れて物理的な場所の移動をしなければならないこともある。参加者が、間違った部屋に入り、特定のプレゼンテーションの場所を見つけることができないこともあるので追加の資料が入手できなくなってしまうこともある。

【発明の開示】

【発明が解決しようとする課題】

【０００６】

したがって、当技術分野においては、プレゼンタと参加者の両者の要件および要望に対処し、プレゼンテーションについてのセキュリティを提供し、プレゼンテーション中に共有される情報の制御を可能にする一対多（１：Ｍ）の情報投影のシステムおよび方法の必要性が存在する。

【０００７】

本発明は、多数の参加者への情報のディスプレイのための新しい改善されたシステムおよび方法を提供するものである。より詳細には、本発明は、多数の参加者へのプレゼンテーションの送達のための新しい改善されたシステムおよび方法を提供する。このプレゼンテーションは、プロジェクタおよび／または多数の参加者のディスプレイデバイス上に表示できることが好ましい。このプレゼンテーションは、公開されたものでもよく、またあ

る参加者がこのプレゼンテーションを閲覧することを「許可する」のに先立って実施すべきセキュリティ認証を必要とするものでもよい。

【課題を解決するための手段】

【０００８】

本発明の好ましい一実施形態では、プレゼンテーションも参加者も、ユニバーサルプラグアンドプレイ規格（Ｕｎｉｖｅｒｓａｌ　Ｐｌｕｇ　ａｎｄ　Ｐｌａｙ　ｓｔａｎｄａｒｄ）に準拠しており、その結果、ネットワーク上でこれらを見出すことができる。端末サービスセッションを介して接続が行われる。プレゼンタの観点からは、プロジェクタおよび参加者を見出すことにより、誰に対して、また何がこのプレゼンテーションに対して許されるかについての制御が可能になる。これは、このプレゼンテーションへの許可を得るために使用される、プレゼンタが生成する招待状を要求することによって実現することができる。セキュリティも、同様にパスワードを要求することによって追加することができる。このプレゼンタはまた、あるプレゼンテーションを公開されたものとして指定することもでき、このプレゼンテーションに対する許可を要求するどの参加者に対しても招待状を自動的に生成することもできる。参加者の観点からは、ネットワーク上のプレゼンテーションデバイスを求めて探索を実施することによって入手可能なプレゼンテーションを見出すことができる。次いで、自分が閲覧したいと思うプレゼンテーションを選択し、それに対する許可を要求することができる。プレゼンタが、そのプレゼンテーションを停止したいと思うときには、プレゼンタは、そのプレゼンテーションを停止するようにプレゼンテーションマネージャにただ通知することになる。次いで、このプレゼンテーションマネージャは、すべての接続を切り離してどの状態もクリーンアップすることになる。

【０００９】

本発明の一実施形態では、ＡＰＩ（アプリケーションプログラミングインターフェース）が提供される。これらのＡＰＩは、プレゼンテーションを閲覧するメソッド、およびプレゼンテーションを提供するメソッドを含んでいる。この閲覧ＡＰＩは、ディスプレイデバイスを登録し登録抹消するメソッド、およびあるプレゼンテーションに参加するメソッドを含んでいる。プレゼンテーション提供ＡＰＩは、プレゼンテーションを開始し停止するメソッド、参加者を招待し切り離すメソッド、プレゼンタの画面にフィルタをかけるメソッド、ユーザおよびプロジェクタのリストまたは個々のユーザおよびプロジェクタを検索するメソッド、プロジェクタの機能、状態、および接続リストを検索するメソッド、ならびにプロジェクタディスプレイの設定およびモードを検索し設定するメソッドを含んでいる。

【００１０】

本明細書中に組み込まれ、本明細書の一部を形成する添付図面は、本発明のいくつかの態様を示し、この説明と一緒に本発明の原理を説明する役割を果たしている。

【００１１】

本発明をある種の好ましい実施形態に関連して説明するが、これらの実施形態に限定する意図はない。これとは逆に、添付の特許請求の範囲によって定義される本発明の趣旨と範囲に含まれるすべての代替形態、変更形態、および均等物が包含されるものとする。

【発明を実施するための最良の形態】

【００１２】

図面を参照すると、本発明は、適切なコンピューティング環境中に実装されるものとして示されている。図面中、同様な参照番号は、同様な要素について言及している。必ずしも必要ではないが、本発明は、パーソナルコンピュータが実行する、プログラムモジュールなどコンピュータ実行可能な命令の一般的な状況で説明されることになる。一般に、プログラムモジュールには、特定のタスクを実施し、または特定の抽象データ型を実装する、ルーチン、プログラム、オブジェクト、コンポーネント、データ構造などが含まれる。さらに、本発明は、ハンドヘルドデバイス、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またはプログラム可能な家電、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などを含めて、他のコンピュータシステム構成を用いて実施する

ことができることが、当業者には理解されよう。本発明は、分散コンピューティング環境においても実施することができ、そこでは、タスクは、通信ネットワークを介してリンクされるリモート処理デバイスによって実施される。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をローカルメモリ記憶デバイスにもリモートメモリ記憶デバイスにも配置することができる。

【００１３】

図１は、本発明を実装することができる適切なコンピューティングシステム環境１００の一例を示すものである。このコンピューティングシステム環境１００は、適切なコンピューティング環境の一例にすぎず、本発明の使用または機能の範囲について何らかの限定を示唆することを意図したものではない。また、このコンピューティング環境１００は、例示の動作環境１００に示されるコンポーネントの任意の１つまたは組合せに関連して何らかの依存性または必要性を有するものと解釈すべきではない。

【００１４】

本発明は、他の多数の汎用または専用のコンピューティングシステム環境またはコンピューティングシステム構成を用いて動作可能である。本発明とともに使用するのに適する可能性があるよく知られているコンピューティングシステム、コンピューティング環境、および／またはコンピューティング構成の例には、それだけには限定されないが、パーソナルコンピュータ、サーバコンピュータ、ハンドヘルドデバイス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、プログラム可能な家電、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、前述のシステムまたはデバイスなどのうちのどれかを含む分散コンピューティング環境などが含まれる。

【００１５】

本発明は、コンピュータが実行する、プログラムモジュールなどのコンピュータ実行可能命令の一般的な状況で説明することができる。一般に、プログラムモジュールは、特定のタスクを実施し特定の抽象データ型を実装するルーチン、プログラム、オブジェクト、コンポーネント、データ構造などを含んでいる。本発明はまた、分散コンピューティング環境において実施することもでき、その場合、タスクは、通信ネットワークを介してリンクされるリモート処理デバイスによって実施される。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めてローカルコンピュータ記憶媒体にもリモートコンピュータ記憶媒体にも配置することができる。

【００１６】

図１を参照すると、本発明を実装するための例示のシステムは、コンピュータ１１０の形態の汎用コンピューティングデバイスを含んでいる。コンピュータ１１０のコンポーネントには、それだけには限定されないが、処理装置１２０、システムメモリ１３０、およびシステムメモリを含めて様々なシステムコンポーネントを処理装置１２０に結合するシステムバス１２１が含まれ得る。システムバス１２１は、メモリバスまたはメモリコントローラ、周辺バス、および様々なバスアーキテクチャのうちのどれかを使用したローカルバスを含めていくつかのタイプのバス構造のうちのどれであってもよい。一例として、限定するものではないが、かかるアーキテクチャには、ＩＳＡ（Ｉｎｄｕｓｔｒｙ Ｓｔａｎｄａｒｄ Ａｒｃｈｉｔｅｃｔｕｒｅ）バス、ＭＣＡ（Ｍｉｃｒｏ Ｃｈａｎｎｅｌ Ａｒｃｈｉｔｅｃｔｕｒｅ）バス、ＥＩＳＡ（Ｅｎｈａｎｃｅｄ ＩＳＡ）バス、ＶＥＳＡ（Ｖｉｄｅｏ Ｅｌｅｃｔｒｏｎｉｃｓ Ｓｔａｎｄａｒｄ Ａｓｓｏｃｉａｔｅ）ローカルバス、およびメザニン（Ｍｅｚｚａｎｉｎｅ）バスとしても知られているＰＣＩ（Ｐｅｒｉｐｈｅｒａｌ Ｃｏｍｐｏｎｅｎｔ Ｉｎｔｅｒｃｏｎｎｅｃｔ）バスが含まれる。

【００１７】

コンピュータ１１０は、一般に様々なコンピュータ可読媒体を含んでいる。コンピュータ可読媒体は、コンピュータ１１０がアクセスすることができる任意の使用可能な媒体とすることができ、揮発性媒体も不揮発性媒体も着脱可能媒体も着脱不能媒体も含んでいる

。一例として、限定するものではないが、コンピュータ可読媒体は、コンピュータ記憶媒体と通信媒体を含むことができる。コンピュータ記憶媒体は、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または他のデータなどの情報を記憶するための任意の方法または技術で実装される揮発性媒体も不揮発性媒体も着脱可能媒体も着脱不能媒体も含んでいる。コンピュータ記憶媒体には、それだけには限定されないが、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、フラッシュメモリまたは他のメモリ技術、ＣＤ－ＲＯＭ、ＤＶＤ（デジタル多用途ディスク）または他の光ディスクストレージ、磁気カセット、磁気テープ、磁気ディスクストレージまたは他の磁気記憶デバイス、または所望の情報を記憶するために使用でき、コンピュータ１１０がアクセスすることができる他の任意の媒体が含まれる。通信媒体は一般に、搬送波や他の搬送機構など変調されたデータ信号の形で、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を具現化する、任意の情報配信媒体を含んでいる。用語「変調されたデータ信号」は、信号に情報を符号化するようにその特性のうちの１つまたは複数の特性が設定または変更された信号を意味する。一例として、限定するものではないが、通信媒体には、有線ネットワークや直接配線接続などの有線媒体と、音響、ＲＦ、赤外線、他の無線媒体などの無線媒体が含まれる。前述のどの組合せもまた、コンピュータ可読媒体の範囲内に含まれる。

【００１８】

システムメモリ１３０は、ＲＯＭ（読取り専用メモリ）１３１やＲＡＭ（ランダムアクセスメモリ）１３２など揮発性メモリおよび／または不揮発性メモリの形態のコンピュータ記憶媒体を含む。起動時などにコンピュータ１１０内の要素間の情報を転送するのを助ける基本ルーチンを含めてＢＩＯＳ（基本入出力システム）１３３は、一般にＲＯＭ１３１に記憶される。ＲＡＭ１３２は一般に、処理装置１２０が直接アクセス可能な、または現在処理装置１２０によって動作中の、あるいはその両方のデータおよび／またはプログラムモジュールを含んでいる。一例として、限定するものではないが、図１にオペレーティングシステム１３４、アプリケーションプログラム１３５、他のプログラムモジュール１３６、およびプログラムデータ１３７を示す。

【００１９】

コンピュータ１１０はまた、他の着脱可能／着脱不能な揮発性／不揮発性のコンピュータ記憶媒体を含むこともできる。一例として、図１は、着脱不能な不揮発性磁気媒体から情報を読み取りまたはそれに情報を書き込むハードディスクドライブ１４１、着脱可能な不揮発性磁気ディスク１５２から情報を読み取りまたはそれに情報を書き込む磁気ディスクドライブ１５１、およびＣＤ－ＲＯＭや他の光媒体など着脱可能な不揮発性の光ディスク１５６から情報を読み取りまたはそれに情報を書き込む光ディスクドライブ１５５を示している。例示の動作環境中で使用することができる他の着脱可能／着脱不能な揮発性／不揮発性のコンピュータ記憶媒体には、それだけには限定されないが、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、ソリッドステートＲＡＭ、ソリッドステートＲＯＭなどが含まれる。このハードディスクドライブ１４１は、一般にインターフェース１４０などの着脱不能なメモリインターフェースを介してシステムバス１２１に接続され、磁気ディスクドライブ１５１および光ディスクドライブ１５５は、一般にインターフェース１５０などの着脱可能なメモリインターフェースによってシステムバス１２１に接続される。

【００２０】

以上で論じ、図１に示したドライブおよびそれらに関連するコンピュータ記憶媒体は、コンピュータ１１０用の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および他のデータの記憶領域を提供する。図１において、例えばハードディスクドライブ１４１は、オペレーティングシステム１４４、アプリケーションプログラム１４５、他のプログラムモジュール１４６、およびプログラムデータ１４７を記憶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１３４、アプリケーションプログラム１３５、他のプログラムモジュール１３６、およびプログラムデータ１３７と同じ、または異なるものとすることができることに留意されたい。オペレーティングシス

テム１４４、アプリケーションプログラム１４５、他のプログラムモジュール１４６、およびプログラムデータ１４７は、この文書では異なる番号が付与され、少なくともこれらが異なる複製であることを示している。ユーザは、キーボード１６２や、マウス、トラックボールまたはタッチパッドとも一般に呼ばれるポインティングデバイス１６１などの入力デバイスを介してコンピュータ１１０にコマンドおよび情報を入力することができる。他の入力デバイス（図示せず）には、マイクロホン、ジョイスティック、ゲームパッド、衛星パラボラアンテナ、スキャナなどが含まれる。これらおよび他の入力デバイスは、しばしばシステムバスに結合されたユーザ入力インターフェース１６０を介して処理装置１２０に接続されるが、パラレルポート、ゲームポート、ＵＳＢ（ユニバーサルシリアルバス）など他のインターフェースおよびバス構造によって接続することもできる。モニタ１９１または他のタイプのディスプレイデバイスもまた、ビデオインターフェース１９０などのインターフェースを介してシステムバス１２１に接続される。モニタに加えて、コンピュータはまた、スピーカ１９７やプリンタ１９６などの他の周辺出力デバイスを含むこともでき、出力周辺インターフェース１９５を介してこれらを接続することができる。

【００２１】

コンピュータ１１０は、リモートコンピュータ１８０など１つ以上のリモートコンピュータへの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１８０は、別の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、または他の一般的なネットワークノードとすることができ、一般にパーソナルコンピュータ１１０に関連して先に説明した要素のうちの多くまたはすべてを含んでいるが、図１には、メモリ記憶デバイス１８１だけしか示していない。図１に示す論理接続は、ＬＡＮ（ローカルエリアネットワーク）１７１およびＷＡＮ（ワイドエリアネットワーク）１７３を含んでいるが、他のネットワークを含むこともできる。かかるネットワーキング環境は、オフィス、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インターネットでは一般的なものになっている。

【００２２】

ＬＡＮネットワーク環境で使用するとき、パーソナルコンピュータ１１０は、ネットワークインターフェースまたはアダプタ１７０を介してＬＡＮ１７１に接続される。ＷＡＮネットワーク環境で使用するとき、コンピュータ１１０は、一般にインターネットなどＷＡＮ１７３上の通信を確立するためのモデム１７２または他の手段を含んでいる。モデム１７２は、内蔵または外付けでもよいが、ユーザ入力インターフェース１６０または他の適切な機構を介してシステムバス１２１に接続することができる。ネットワーク環境では、パーソナルコンピュータ１１０に関連して示されるプログラムモジュール、またはその一部分をリモートメモリ記憶デバイスに記憶することができる。一例として、限定するものではないが、図１は、リモートアプリケーションプログラム１８５がメモリデバイス１８１上に存在するものとして示している。ここに示したネットワーク接続は、例示的なものであり、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することもできることが理解されよう。

【００２３】

以下の説明では、他に指定がない限り、本発明は、１台以上のコンピュータによって実施される動作およびオペレーションの記号表現に関して説明することにする。したがって、かかる動作およびオペレーションは、時にコンピュータ実行されるものとも呼ばれるが、構造化形式でデータを表現する電気信号によるコンピュータの処理装置による操作を含んでいることが理解されよう。この操作は、データを変換し、またはコンピュータのメモリシステム中のロケーションにデータを保持し、当業者にはよく理解されるような方法でコンピュータのオペレーションを再構成し、または変更する。データが保持されるデータ構造は、特定のプロパティがデータのフォーマットによって定義される、メモリの物理ロケーションである。しかし、本発明は前述の状況で説明されているが、以下で説明される様々な動作およびオペレーションがハードウェアでも実装できることが当業者には理解されるように、これは限定を意味するものではない。

【００２４】
実際に、プログラミングインターフェース（またはさらに簡単には、インターフェース）は、コードの１つ以上のセグメントが、コードの他の１つ以上のセグメントによって提供される機能と通信し、または機能にアクセスすることができるようにするための任意の機構、プロセス、プロトコルと見なすことができる。あるいは、プログラミングインターフェースは、他のコンポーネントの１つ以上の機構、メソッド、ファンクションコール、モジュールなどと通信して結合することができる、システムのコンポーネントの１つ以上の機構、メソッド、ファンクションコール、モジュール、オブジェクトなどと見なすこともできる。前述の文章中の用語「コードのセグメント」には、適用される専門用語にかかわらず、コードセグメントが別々にコンパイルされるかどうか、コードセグメントがソースコード、中間コードまたはオブジェクトコードのいずれとして提供されるか、コードセグメントが実行時のシステムまたはプロセスで利用されるかどうか、コードセグメントが同じマシン上もしくは異なるマシン上に配置されまたは複数のマシン上に分散されているかどうか、あるいはコードセグメントが表す機能が、全部ソフトウェアで、全部ハードウェアで、またはハードウェアとソフトウェアの組合せで実装されるかにかかわらず、１つ以上の命令またはコード行が含まれることが意図されており、例えばコードモジュール、オブジェクト、サブルーチン、ファンクションなどが含まれる。
【００２５】
概念的には、プログラミングインターフェースは、一般的に図２または図３に示すように見なすことができる。図２は、第１および第２のコードセグメントが通信する経路（ｃｏｎｄｕｉｔ）としてのインターフェースＩｎｔｅｒｆａｃｅ１を示している。図３は、（第１および第２のコードセグメントの一部でもよくそうでなくてもよい）インターフェースオブジェクトＩ１およびＩ２を含むものとして、インターフェースを示しており、これらＩ１およびＩ２により、システムの第１および第２のコードセグメントは、媒体Ｍを介して情報をやりとりすることができる。図３を考慮すれば、インターフェースオブジェクトＩ１およびＩ２を同じシステムの別々のインターフェースとして考えることができ、またこのオブジェクトＩ１およびＩ２に媒体Ｍを加えたものがこのインターフェースを構成すると考えることもできる。図２および図３は、双方向の流れ（flow）およびこのフローの各側にインターフェースを示すが、一部の実装は、一方向の情報フローしかもたず（または、以下で説明するように情報フローをもたず）、あるいは一方の側にインターフェースオブジェクトをもつだけの場合もある。一例として、限定するものではないが、ＡＰＩ（アプリケーションプログラミングインターフェース）、エントリポイント、メソッド、ファンクション、サブルーチン、リモートプロシージャコール、ＣＯＭ（コンポーネントオブジェクトモデル）インターフェースなどの用語は、プログラミングインターフェースの定義内に包含されている。
【００２６】
かかるプログラミングインターフェースの態様は、それによって第１のコードセグメントが情報（この場合、「情報」は、その最も広い意味で使用され、データ、コマンド、要求などを含んでいる）を第２のコードセグメントに伝送するメソッド、それによって第２のコードセグメントが情報を受け取るメソッド、およびこの情報の構造、シーケンス、シンタックス、構成、スキーマ、タイミングおよび内容を含むことができる。この点で、情報がインターフェースによって定義される方法で搬送される限りは、媒体が有線または無線、あるいは両方の組合せであろうがなかろうが、潜在的な伝送媒体それ自体はインターフェースのオペレーションにとって重要ではないこともある。一部の状況では、１つのコードセグメントが第２のコードセグメントが実施する機能に単にアクセスする場合、情報転送は、別の機構（例えば、コードセグメント間の情報フローとは別のバッファ、ファイルなどに配置される情報）を介するかまたは存在しないので、情報は、従来の意味で一方向または両方向に渡されないこともある。例えばこれらのコードセグメントが柔軟結合または緊密結合の構成中のシステムの一部であるかどうかに応じてこれらの態様のうちのどれかまたはすべてが所与の状況では重要となることもあり、したがって、このリストは例

示的であり非限定的であると考えるべきである。

【００２７】

プログラミングインターフェースのこの概念は、当業者には知られており、本発明の前述の詳細な説明から明らかである。しかし、プログラミングインターフェースを実装する他の方法があり、特別に除外されない限り、これらも本明細書に添付される特許請求の範囲によって包含されるものとする。かかる他の方法は、図２および図３の単純化した図に比べてより高度または複雑なものに見えるが、これらはそれにもかかわらず同様なファンクションを実施して全体的に同じ結果を実現する。次に、プログラミングインターフェースのいくつかの例証としての代替実装形態について簡単に説明することにする。

【００２８】

Ａ．ファクタ分解

１つのコードセグメントから別のコードセグメントへの通信は、この通信を複数の個別の通信に分解することによって間接的に達成することができる。これを図４および図５に概略的に示している。図に示すように、いくつかのインターフェースを分解可能な機能セットの観点から説明することができる。したがって、ちょうど２４、すなわち２×２×３×２を数式的に提供できるように、図２および図３のインターフェース機能をファクタ分解して同じ結果を実現することができる。それによって、図４に示すように、同じ結果を実現しながら、インターフェースＩｎｔｅｒｆａｃｅ１が提供するファンクションを細かく分割してこのインターフェースの通信を複数のインターフェースＩｎｔｅｒｆａｃｅ１Ａ、Ｉｎｔｅｒｆａｃｅ１Ｂ、Ｉｎｔｅｒｆａｃｅ１Ｃなどに変換することができる。図５に示すように、同じ結果を実現しながら、インターフェースＩ１が提供するファンクションを細かく分割して複数のインターフェースＩ１ａ、Ｉ１ｂ、Ｉ１ｃなどにすることができる。同様に、第１のコードセグメントから情報を受け取る第２のコードセグメントのインターフェースＩ２をファクタ分解して複数のインターフェースＩ２ａ、Ｉ２ｂ、Ｉ２ｃなどにすることができる。ファクタ分解する際に、第１のコードセグメントと共に含まれるインターフェースの数は、第２のコードセグメントと共に含まれるインターフェースの数に一致する必要はない。図４および図５の場合のいずれにおいても、インターフェースＩｎｔｅｒｆａｃｅ１およびＩ１の機能的な趣旨は、それぞれ図２および図３と同じある。インターフェースのファクタ分解ではまた、ファクタ分解を認識するのを困難にすることができるように、関連（associative）プロパティ、通信（communicative）プロパティ、および他の数学的プロパティを引き継ぐこともできる。例えば、オペレーションの順序は、重要でないこともあり、したがって、インターフェースが実行するファンクションが、このインターフェースに到達する十分前に、別のコードまたはインターフェースによって実行され、またはこのシステムの別のコンポーネントによって実行されることができる。さらに、プログラミング技術分野の当業者なら、同じ結果を実現する異なるファンクションコールを行う様々な方法があることが理解できよう。

【００２９】

Ｂ．再定義

一部のケースでは、意図した結果を依然として達成しながら、プログラミングインターフェースの一部の態様（例えば、パラメータ）を無視し、追加し、または再定義することを可能にすることができる。例えば、図２のインターフェースＩｎｔｅｒｆａｃｅ１が、３つのパラメータｉｎｐｕｔ、ｐｒｅｃｉｓｉｏｎ、およびｏｕｔｐｕｔを含み、第１のコードセグメントから第２のコードセグメントへ出されるファンクションコールＳｑｕａｒｅ（ｉｎｐｕｔ，ｐｒｅｃｉｓｉｏｎ，ｏｕｔｐｕｔ）を含んでいると想定する。中央のパラメータｐｒｅｃｉｓｉｏｎが、図６に示すように所与のシナリオにおいて関係がない場合には、このパラメータを好都合にも無視し、または（この状況では）ｍｅａｎｉｎｇｌｅｓｓパラメータで置き換えることさえできる。また、関係のないａｄｄｉｔｉｏｎａｌパラメータを追加することもできる。どちらの場合にも、入力が第２のコードセグメントによって２乗された後、出力が返される限り、２乗の機能を実現することができる。ｐｒｅｃｉｓｉｏｎは、あるダウンストリームまたはそのコンピューティングシステムの

他の部分に対して相当重要なパラメータになることもある。しかし、ｐｒｅｃｉｓｉｏｎが２乗を計算するという狭い目的では必ずしも必要でないことが認識された後には、置き換え、または無視することができる。例えば、有効なｐｒｅｃｉｓｉｏｎ値を渡す代わりに、誕生日などの無意味な値を、結果に悪影響を与えることなく渡すことができる。同様に、図７に示すように、インターフェースＩ１は、無視しまたはこのインターフェースにパラメータを追加するために再定義されたインターフェースＩ１’によって置き換えられる。インターフェースＩ２も同様に、不必要なパラメータ、または他のどこかで処理することができるパラメータを無視するために再定義されるインターフェースＩ２’として再定義することができる。ここでの要点は、一部のケースでは、プログラミングインターフェースは、ある目的では必要とされないパラメータなどの態様を含むこともあり、したがって、それらを無視し、または再定義し、あるいは他の目的のために他のどこかで処理することができることである。

【００３０】

Ｃ．インラインコーディング

２つの別々のコードモジュールの機能のうちの一部または全部を、これらモジュールの間の「インターフェース」が形式を変更するようにマージすることも実現可能である。例えば、図２および図３の機能をそれぞれ図８および図９の機能に変更することもできる。図８において、前述の図２の第１および第２のコードセグメントを、これらの両方を含むモジュールにマージする。このケースでは、これらのコードセグメントは、依然として互いに情報をやりとりすることができるが、そのインターフェースを、単一のモジュールにより適した形式に合うようにすることができる。したがって、例えば、正式なコールステートメント（Ｃａｌｌ　ｓｔａｔｅｍｅｎｔ）およびリターンステートメント（Ｒｅｔｕｒｎ　ｓｔａｔｅｍｅｎｔ）は、もはや必要ではないが、インターフェースＩｎｔｅｒｆａｃｅ１に従う同様な処理または応答が依然として効力があるようにすることができる。同様に、図９に示すように、図３のインターフェースＩ２の一部（または全部）をインターフェースＩ１にインラインに書き込んでインターフェースＩ１”を形成することができる。図に示すように、インターフェースＩ２をＩ２ａとＩ２ｂに分割し、インターフェース部分Ｉ２ａをインターフェースＩ１とインラインにコード化してインターフェースＩ１”を形成している。具体的な例として、図３のインターフェースＩ１が、インターフェースＩ２で受け取り、第２のコードセグメントによって（２乗すべき）ｉｎｐｕｔを用いて渡された値を処理した後に２乗された結果を、ｏｕｔｐｕｔを用いて戻すファンクションコールＳｑｕａｒｅ（ｉｎｐｕｔ，ｏｕｔｐｕｔ）を実施することを考えてみる。かかるケースでは、（ｉｎｐｕｔを２乗する）第２のコードセグメントで実施する処理を、このインターフェースへのコールなしに第１のコードセグメントで実施することができる。

【００３１】

Ｄ．分離（ｄｉｖｏｒｃｅ）

１つのコードセグメントから別のコードセグメントへのコミュニケーションは、このコミュニケーションを複数の個別のコミュニケーションへと分解することによって間接的に達成することができる。これを図１０および図１１に概略的に示す。図１０に示すように、１つ以上のミドルウェア（これらは元のインターフェースからの機能および／またはインターフェースファンクションを分離することから、分離インターフェース（Ｄｉｖｏｒｃｅ　Ｉｎｔｅｒｆａｃｅ））を提供して、第１のインターフェースＩｎｔｅｒｆａｃｅ１上のコミュニケーションを変換して、これらを異なるインターフェースに、このケースではインターフェースＩｎｔｅｒｆａｃｅ２Ａ、Ｉｎｔｅｒｆａｃｅ２Ｂ、およびＩｎｔｅｒｆａｃｅ２Ｃに適合するようにする。例えば、Ｉｎｔｅｒｆａｃｅ１プロトコルによるオペレーティングシステムと情報をやりとりするように設計されたインストールベースのアプリケーションがあるが、次いでオペレーティングシステムが、異なるインターフェースを、このケースではインターフェースＩｎｔｅｒｆａｃｅ２Ａ、Ｉｎｔｅｒｆａｃｅ２Ｂ、およびＩｎｔｅｒｆａｃｅ２Ｃを使用するように変更する場合にこれが行われ得る。ここでの要点は、第２のコードセグメントが使用する元のインターフェースは、第１の

コードセグメントが使用するインターフェースとはもはや互換性がないように変更され、したがって中間手段を使用して元のインターフェースと新しいインターフェースの間で互換性があるようにすることである。同様に、図１１に示すように、第３のコードセグメントを導入して、分離インターフェースＤＩ１を用いてインターフェースＩ１からのコミュニケーションを受け取り、分離インターフェースＤＩ２を用いて例えば、ＤＩ２と共に動作するが同じ機能結果を提供するように再設計されたインターフェースＩ２ａおよびＩ２ｂへとインターフェース機能を伝送することができる。同様にＤＩ１およびＤＩ２は一緒に動作して、同じまたは同様な機能結果を提供しながら図３のインターフェースＩ１およびＩ２の機能を新しいオペレーティングシステムに変換することができる。

【００３２】

Ｅ．書換え

さらに別の可能な変形は、コードを動的に書き換えてこのインターフェース機能を何か別のもので置き換えるが全体的には同じ結果を実現するものである。例えば、中間言語（例えばＭｉｃｒｏｓｏｆｔ　ＩＬ、Ｊａｖａ（登録商標）ＢｙｔｅＣｏｄｅなど）で表現されるコードセグメントが、（．Ｎｅｔフレームワーク、Ｊａｖａ（登録商標）ランタイム環境、他の同様なランタイムタイプの環境によって提供されるものなど）実行環境中のＪＩＴ（Ｊｕｓｔ－ｉｎ－Ｔｉｍｅジャストインタイム）コンパイラまたはインタプリタに提供されるシステムがあり得る。第１のコードセグメントから第２のコードセグメントへ動的に変換して、すなわちこれらを第２のコードセグメント（元の第２のコードセグメントまたは異なる第２のコードセグメント）で必要とされる場合がある異なるインターフェースに適合するようにこのＪＩＴコンパイラを記述することができる。これを図１２および図１３に示す。図１２から分かるように、この手法は、前述の分離シナリオと同様である。例えば、インストールベースのアプリケーションが、Ｉｎｔｅｒｆａｃｅ１プロトコルによるオペレーティングシステムと情報をやりとりするように設計されているが、次いでこのオペレーティングシステムが、異なるインターフェースを使用するように変更される場合に、この手法を行うことができる。このＪＩＴコンパイラを使用してインストールベースのアプリケーションからの進行中のコミュニケーションをこのオペレーティングシステムの新しいインターフェースに適合するようにすることができる。図１３に示すように、これらのインターフェースを動的に書き換えるこの手法を適用して、インターフェースをも動的にファクタ分解し、または別の方法で変更することができる。

【００３３】

同じまたは同様な結果を、代替実施形態を介したインターフェースとして実現する前述のシナリオは、直列および／または並列に、または他の介入するコードを用いて様々な方法で組み合わせることもできることにも留意されたい。したがって、以上で提示された代替実施形態は、互いに排他的ではなく、混合させマッチさせ組み合わせて、図２および図３に提示した一般的なシナリオと同じまたは同等なシナリオを作成することができる。ほとんどのプログラミング構成体と同様に、本明細書中では説明されていないこともあるがそれにもかかわらず本発明の趣旨と範囲によって表されるインターフェースの同じまたは同様な機能を達成する他の同様な方法があることにも留意されたい。すなわち、インターフェースの価値の基礎をなすものは、少なくとも部分的にインターフェースによって表される機能であり、インターフェースが可能とする有利な結果であることに留意されたい。

【００３４】

これを念頭に置いて、次に図１４ａ着目する。この図は、本発明を実装できるネットワーク環境を概略的に示すものである。詳細には、図に示すネットワーク環境は、投影デバイスまたはプレゼンタ２０１を含んでおり、これは、１台以上の投影ターゲットデバイスまたは参加者２０３、２０５、２０７、および２０９へと情報の投影を行う、例えば図１を参照して前述したデバイスなど任意のコンピューティングデバイスとすることができる。投影ターゲットデバイスまたは参加者２０３～２０７はまた、ユーザとしても知られており、プレゼンタ２０１と同様に非専用のコンピューティングデバイスとして示されている。このケースでは、例えば、参加者（ユーザ）２０１～２０７は、一例として限定する

ものではないが、ラップトップコンピュータ、デスクトップコンピュータ、ハンドヘルドコンピューティングデバイス、他の任意の多目的のコンピューティングデバイス、またはこれらのタイプのデバイスのどのような組合せでもよく、またこれらを使用することができる。ユーザは、従来のコンピューティングデバイスを利用する必要も、従来のコンピューティングデバイスである必要もなく、例えばテレビジョンシステムであってもよい。ネットワーク２１１は、参加者２０３～２０７とプレゼンタ２０１の間で情報を転送するために使用可能である。

【００３５】

同様に、参加者は、追加してまたは代わりに電子会議室のプロジェクタなどの専用の投影デバイス、すなわちディスプレイデバイス２０９を含んでおり、プロジェクタと呼ぶこともある。他の参加者（ユーザ）２０３～２０７と同様に、参加者（プロジェクタ）２０９は、ネットワーク２１１を介して（やはりユーザである）プレゼンタ２０１と情報をやりとりすることが好ましい。ネットワーク２１１は、任意のタイプのネットワークでもよいが、一般にプレゼンタ２０１とネットワーク２１１との間の、またネットワーク２１１と参加者（ユーザ）２０３～２０７との間の無線インターフェースを含むことになる。さらに、ネットワーク２１１と参加者（プロジェクタ）２０９との間のインターフェースは、有線または無線であることが好ましい。例えば、プロジェクタ２０９は、会議室などの特定の場所に長期間とどまることができるので、プロジェクタ２０９からネットワーク２１１への有線インターフェースをもつことによりデバイスの実用性はあまり低下しない。ネットワーク２１１それ自体は一般に、必ずしも必要ではないが、企業ＬＡＮ、ＷＡＮ、他の従来技術による全体または部分的に有線化されたネットワークなどの有線インフラストラクチャであることになる。

【００３６】

代替的なネットワーク環境を図１４ｂに概略的に示す。詳細には、プレゼンタ２０１および参加者２０３～２０９は、無線リンク２１３～２２５から構成されるアドホックな無線ネットワークを介して相互接続される。アドホックなネットワークは、あらゆるノードをあらゆる他のノードに直接接続する必要がないので、必ずしもすべての接続２１３～２２５が必要ではないことに留意されたい。例えば、１つのノードは、アドホックなネットワークのすべてのノードに対して、別のノードへの１つの接続を介して間接的に接続することができる。したがって、アドホックなトポロジには、リング、ライン、ウェブ、ハブアンドスポーク（ｈｕｂ－ａｎｄ－ｓｐｏｋｅ）および／または他のトポロジが必要に応じて含まれる。しばしば、特定のデバイスの他のデバイスからの物理的距離は、特定のデバイスが直接に接続するアドホックネットワークの、もしあれば１つまたは複数のデバイスによって決まる。

【００３７】

前述のコンポーネントの使用および対話のシナリオについて、対話の仕組みの詳細な説明に進む前に以下に簡単に説明する。本発明のシステムは、家庭環境におけるなど非ビジネス設定でも同様に使用可能であるが、プレゼンタ２０１は、一般に会議室またはミーティング室の環境において、個人の望むものを提示する立場にあり、受け取る側の個人の関心を引く資料を投影する。関心を引く資料はイメージやビデオなどのグラフィック形式、またはドキュメントやチャートの形などのテキスト形式とすることができ、またオーディオ要素を含むこともできる。本発明の一実施形態では、対象の資料は専らオーディオ情報である。対象の資料は、コンピュータ生成されたものである必要はないが、コンピュータ可読なフォーマットでローカルにまたはリモートにプレゼンタ２０１からアクセス可能であることが好ましい。参加者（ユーザ）２０３～２０７は、受け取る側の個人のラップトップコンピュータでもよいが、参加者（プロジェクタ）２０９は、モバイルラップトップやハンドヘルドデバイスと違って、会議室のプロジェクタもしくは大型画面モニタや一般に任意の１ユーザに物理的に関連づけられない他のディスプレイなど専用の投影システムとすることができる。

【００３８】

参加者２０３～２０９間のネットワーク接続は、一般に各デバイスが提示側のユーザに関連するプレゼンタ２０１の通信範囲内に入るときに開始される。したがって、例えば、プロジェクタ２０９は、会議室に恒久的に存在するものと想定する。提示側のユーザがプレゼンタ２０１と共に会議室に入るとき、アドホックまたはネットワークインフラストラクチャを介した無線接続がプロジェクタ２０９とプレゼンタ２０１の間に形成される。次いで、提示側ユーザは、彼のデバイス２０１からプロジェクタ２０９の画面上に、受け取る側の個々人が見られるように資料の投影を行うことができる。このようにして、提示側のユーザは、どのようなケーブルまたはコードも物理的に接続せずにプレゼンテーションを行っており、同様にプレゼンテーションを終了し、あるいはどのようなケーブルまたはコードも切り離さずに別の投影デバイスを使用する別の提示側の個人へとその制御を転送することができる。

【００３９】

ほぼ同じ方法で、提示側ユーザは、対象の資料を参加者（ユーザ）２０３～２０７などいくつかのターゲットデバイスに提示することができる。例えば、本発明の一実施形態では、プレゼンタ２０１とユーザ２０３～２０７の間のネットワーク接続は、ユーザが物理的な接続を見出し、操作する必要なしに自動的に見出した後に自動的に実行することができる。このケースでは、資料のプレゼンテーションは、プレゼンタ２０１から受け取る側の個々人の所有するラップトップコンピュータとすることができる参加者の画面に対して行われる。この特定の仕組みについては、以下でさらに十分に論じることにするが、それには様々なセキュリティ態様が関連することになる。

【００４０】

図１５は、本発明のシステムおよび方法において存在する異なるコンポーネントおよびバイナリ（ｂｉｎａｒｙ）を示している。プレゼンテーションデバイス５００は、プレゼンタのマシン上に作成され登録される。このデバイスは、ＵＰｎＰに登録され、登録された後にＳＳＤＰ（Ｓｉｍｐｌｅ Ｓｅｒｖｉｃｅ Ｄｉｓｃｏｖｅｒｙ Ｐｒｏｔｏｃｏｌ）を介して見出すことが可能である。このデバイスは、登録プロセス中にＵＰｎＰが作成するその一意的なデバイス名（ＵＤＮ）で識別されることが好ましい。このデバイスは、プレゼンテーションサービス（ｐｒｅｓｅｎｔａｔｉｏｎｓｅｒｖｉｃｅ）として知られているＵＰｎＰサービスを含んでいる。ユーザディスプレイデバイス５０２は、参加者のマシン上に作成され登録される。このデバイスはまた、ＵＰｎＰに登録され、登録された後にＳＳＤＰを使用して見出すことができる。このデバイスもまた、そのＵＤＮによって識別され、ユーザディスプレイサービス（ｕｓｅｒｄｉｓｐｌａｙｓｅｒｖｉｃｅ）として知られているＵＰｎＰサービスを含んでいる。この図１５に存在する３つのバイナリは、ＣＲＰ ｐｒｅｓｅｎｔａｔｉｏｎ．ｄｌｌ５０４、ＣＲＰ ａｔｔｅｎｄｅｅ．ｄｌｌ５０６、およびＭｉｃｒｏｓｏｆｔ．ＣＲＰ．ｄｌｌ５０８を含んでいる。このＣＲＰ ｐｒｅｓｅｎｔａｔｉｏｎ．ｄｌｌは、ＵｐｎＰのプレゼンテーションデバイス（ｐｒｅｓｅｎｔａｔｉｏｎｄｅｖｉｃｅ）であるプレゼンテーションデバイスを含んでいる。さらに、このバイナリは、プレゼンテーションデバイスの登録と登録抹消のための機能を提供する。これらの方法を使用してこのプレゼンテーションデバイスの登録を行い、登録を抹消する。さらに、プロジェクタ制御機能も含まれる。これらの方法を使用してプロジェクタを制御し、例えばプロジェクタ機能、ディスプレイモードなどを獲得する。最後に、このＣＲＰ ｐｒｅｓｅｎｔａｔｉｏｎ．ｄｌｌは、参加者の制御機能を含んでいる。これらの方法を使用して参加者への招待状を送る。

【００４１】

このバイナリＣＲＰ ａｔｔｅｎｄｅｅ．ｄｌｌは、ユーザディスプレイについてのＵＰｎＰデバイスであるユーザディスプレイデバイスのための機能を提供する。ユーザディスプレイデバイスの登録および登録抹消の機能も提供される。これらの方法を使用してユーザディスプレイデバイスの登録を行い、登録を抹消する。最後に、このＣＲＰ ａｔｔｅｎｄｅｅ．ｄｌｌバイナリは、プレゼンテーション制御のための機能を含んでいる。これらのメソッドを使用してプレゼンタからの招待状を要求する。

【００４２】
このＭｉｃｒｏｓｏｆｔ．ＣＲＰ．ｄｌｌバイナリは、ＣＲＰ ａｔｔｅｎｄｅｅ．ｄｌｌバイナリおよびＣＲＰ ｐｒｅｓｅｎｔａｔｉｏｎ．ｄｌｌバイナリ上に提供されるラッパ（ｗｒａｐｐｅｒ）である。これは、ユーザフレンドリに管理されたインターフェースをアプリケーションに対して公開する。このＭｉｃｒｏｓｏｆｔ．ＣＲＰ．ｄｌｌはまた、端末サービスセッション５１０制御し作成する機能を実装する。
【００４３】
本発明のＣＲＰ（ｃｏｎｆｅｒｅｎｃｅ ｒｏｏｍ ｐｒｏｊｅｃｔｏｒ会議室プロジェクタ）のシステムおよび方法は、３つの部分に依存する。第１はアプリケーションに提供されるユーザインターフェースである。第２はＴＳ（ｔｅｒｍｉｎａｌ ｓｅｒｖｉｃｅターミナルサービス）である。別のマシン（プロジェクタまたは参加者）に対してプレゼンテーションの投影を行うために端末サービス協調ＡＰＩが使用される。端末サービスは１つまたは複数の端末サービスクライアントが端末サービスサーバに接続し、ＴＳサーバ（一般にデスクトップ）によって画面表示されるセッションを閲覧することができるようにする。このプレゼンタは端末サービスサーバとしての役割を果たし、参加者およびプロジェクタは端末サービスクライアントとしての役割を果たす。第３に、ＣＲＰは、ＵＰｎＰに依存する。ＵＰｎＰは、任意のアプリケーションがローカルマシンまたはリモートマシン上に呼び出すことができる１組のアクションを公開するデバイスの概念を有する。ＵＰｎＰはまた、ＵＰｎＰデバイスを見出すためのＳＳＤＰも提供する。
【００４４】
典型的な設定では、プレゼンテーション、参加者、およびプロジェクタは、すべてＵＰｎＰデバイスである。ＳＳＤＰを使用することによって、ユーザは、ローカルネットワーク上に存在するプレゼンテーションデバイス、参加者デバイス、およびプロジェクタデバイスを見出すことができる。ＴＳクライアントがＴＳサーバに接続するためには、このＴＳクライアントは、ＴＳサーバが生成する招待状を必要とする。サーバからクライアントに招待状を転送するためには、ＵＰｎＰデバイスが提供するアクションが使用される。ＵＰｎＰデバイス上で呼び出すことができるアクションが存在し、例えば、プレゼンテーションデバイスは招待状を要求するアクションを有することになるが、参加者デバイスは参加者を招待するために使用することができるアクションを有することになる。
【００４５】
プレゼンタがプレゼンテーションを提供したいと思うとき、プレゼンタは、端末サービスセッション５１０を作成することになる。プレゼンタが招待したいと思う参加者およびプロジェクタが存在する場合、プレゼンタは、端末サービスチケットを作成し、これらの参加者およびプロジェクタに招待状を送付する。ある参加者を招待するためには、その参加者デバイス上で招待アクションが呼び出される。あるプロジェクタに接続するためには、そのプロジェクタＵＰｎＰデバイス上で接続アクションが呼び出される。プロジェクタおよび参加者は、招待状をもった後に、端末サービスクライアントにその招待状を提供することになる。次いで、この端末サービスクライアントは、この端末サービスセッションと連絡を取ることになる。
【００４６】
当技術分野においてよく知られているように、ＴＳサーバは、ＣＯＭコンポーネントである。ＴＳサーバは、前述のＭｉｃｒｏｓｏｆｔ．ＣＲＰ．ｄｌｌバイナリによって使用される。最新のＴＳ協調ＡＰＩは、このセッションを作成するために使用される。他の任意のＣＯＭコンポーネントと同様に、ＴＳサーバも、登録して使用する必要がある。当技術分野においてまたよく知られているように、ＴＳクライアントは、ＡｃｔｉｖｅＸコントロールである。このＡｃｔｉｖｅＸコントロールは、招待状を使用することが要求される場合には、このコントロールの幅、高さ、招待状、およびパスワードのようなそれに関連する一部のプロパティを有している。ＡｃｔｉｖｅＸコントロールは、登録して使用する必要がある。その発見機構は一般に、ＣＲＰシステム中でこのデバイスを見出すためのＳＳＤＰになる。次いで、これらのデバイスのＵＤＮを入力として提供することになる。

【００４７】
本発明の実施形態の一例におけるこれらのデバイスについての状態変数は、特定の値または値のリストを表すために使用される一般的な文字列Ａ＿Ｓｔｒｉｎｇ、およびそのディスプレイデバイスの名前文字列または識別文字列であるＡ＿Ｎａｍｅを含んでいる。
【００４８】
両方の状態を紹介したので、次にアクションについて簡単に論じることにする。これらのアクションは、招待メソッド（ｉｎｖｉｔｅ　ｍｅｔｈｏｄ）、プレゼンタフレンドリな命名メソッド（ｎａｍｅ　ｍｅｔｈｏｄ）、およびプレゼンテーションフレンドリな命名メソッドを含んでいる。ＧｅｔＮａｍｅメソッドは、参加者の識別文字列を戻す。指定された値が無効の場合には、このメソッドは、ＩｎｖａｌｉｄＶａｌｕｅエラー記述を戻す。
【００４９】
本発明の一実施形態では、ユーザは、２つのオペレーション、すなわちプレゼンテーションを提供し、またはプレゼンテーションを閲覧するというオペレーションを実施することができる。プレゼンテーションを提供する際には、プレゼンタは、プロジェクタおよび／または別のユーザのディスプレイデバイス上にこのプレゼンテーションを表示したいと思うかもしれない。プレゼンテーションを提供しようと望む際には、プレゼンタは、以下のステップを実施することになる。最初に、このプレゼンタは、プレゼンテーションセッションを開始する。これにより、他のユーザが見出すことが可能となるプレゼンテーションデバイスが作成されることになる。これによりまた、端末サービスセッションが作成されることになる。
【００５０】
プレゼンタが、近くのプロジェクタを見出し、これらのプロジェクタに接続したいと思うこともある。これを行うために、プレゼンタは、これらのプロジェクタを求めてＳＳＤＰ探索を実施する。このＳＳＤＰ探索は、プロジェクタのリストを戻すことになる。次いでプレゼンタは、このリストから１つまたは複数のプロジェクタを選択することになる。次いで、プロジェクタのＵＤＮ、および必要ならパスワードが、接続を確立するために提供される。ＴＳセッションの招待マネージャを使用してパスワードに基づいてこのプロジェクタについての招待状が作成される。次いで、この招待状は、プロジェクタのディスプレイデバイス上の接続アクションを呼び出すことによって、そのプロジェクタに送付されることになる。
【００５１】
ユーザは、他のユーザがこのプレゼンテーションを閲覧するように招待したいと思うこともある。これをするためには、プレゼンタは、他のユーザを求めて探索することができる。この探索では、ユーザのディスプレイデバイスのＵＤＮが提供されることになる。ＵＤＮが見出された後に、端末サービス招待マネージャはその被招待者に対する招待状を作成することになる（招待だけを行うためには、別の方法で一般的な招待状を使用する）。次いで、招待状が、ユーザのディスプレイデバイス上の招待アクションを呼び出すことによってその被招待者に送られることになる。
【００５２】
プレゼンタはまた、別のユーザがそのプレゼンテーションに参加したいと思うときに通知してほしいと思うこともある。前述のプレゼンテーションデバイス上のアクションＲｅｑｕｅｓｔＩｎｖｉｔａｔｉｏｎが、そのプレゼンテーションに参加したいと思っているユーザによって呼び出されることになる。このケースでは、このユーザについての情報と共にコールバックがそのプレゼンタに与えられる。次いで、プレゼンタは、この情報を使用して他のどのユーザとも全く同様にそのユーザを招待することができる。あるいは、プレゼンタは、招待で頭を悩ませたくないと思うこともあり、どのような認証またはパスワードもなしに誰でもそのプレゼンテーションに簡単に参加できるようにしたいと思うこともある。両ケースで、プレゼンタのプレゼンテーションデバイス上で、ＲｅｑｕｅｓｔＩｎｖｉｔａｔｉｏｎアクションが呼び出されるときはいつでも、この招待状は直ちにその

ユーザに送付されることになる。このプレゼンテーションが、セキュリティ保護されたものである場合には、その招待状は、パスワードで保護されることになる。

【００５３】

プレゼンタがプレゼンテーションを停止したいと思うときは、プレゼンタは、ただこのプレゼンテーションデバイスの登録を抹消しこのＴＳセッションをクローズすることになる。

【００５４】

ユーザが実施する他のオペレーションは、プレゼンテーションを閲覧することである。このケースでは、参加者は、まず利用可能なプレゼンテーションを見出したいと思うことがある。当技術分野においてよく知られているように、参加者は、ＳＳＤＰを使用してこれを簡単に行うことができる。参加者は、プレゼンテーションのＵＤＮを受け取り、次いで、そのプレゼンテーションデバイス上で関連情報を伴うアクションＲｅｑｕｅｓｔＩｎｖｉｔａｔｉｏｎを呼び出すことになる。これは、プレゼンテーションがかかる要求を介してアクセスを可能とするケースだけで有効である。プレゼンタが、参加者がそのプレゼンテーションを閲覧することができるようにする場合、参加者は、その招待状を用いてこのディスプレイ上のアクションＩｎｖｉｔｅを呼び出すことになる。次いで、招待状がそのＴＳクライアントに提供されて接続を確立することになる。次いで、ＴＳクライアントは、このプレゼンタ上のＴＳセッションに接続されることになる。もしもプレゼンタが招待状を求められることなく参加者を招待したいと思う場合には、プレゼンタは、前述のように、ディスプレイデバイス上のアクションＩｎｖｉｔｅを直接に呼び出すことになる。

【００５５】

接続が確立された後に、このプレゼンテーションは、本出願の譲受人に譲渡され、その教示および開示が参照によってその全体が本明細書中に組み込まれている２００２年６月２５日に出願された「IMPROVED DATA PROJECTION SYSTEM AND METHOD」という名称の同時係属出願第１０／１７９，４３１号に記載の方法に従って、その参加者に、ユーザにもプロジェクタにも、提供することができる。この同時係属の出願に記載のように、プレゼンテーションを提供することができるネットワークは、インフラストラクチャモードで、またはアドホックネットワークの一部として動作させられる８０２．１１に準拠の無線リンクを含む。もちろん、他の無線ネットワーク、ならびに有線ネットワークも同様に使用することができることが当業者には理解されよう。また、この同時係属出願に記載のように、ユーザの発表および位置は、どちらにしても標準的なユニバーサルプラグアンドプレイのアナウンスメント機構を介して実行することができ、それによって参加者のユニバーサルプラグアンドプレイコンポーネントは、プレゼンタのユニバーサルプラグアンドプレイコンポーネントに対してその存在と機能を知らせる。あるいは、プレゼンタのユニバーサルプラグアンドプレイコンポーネントは、参加者のユニバーサルプラグアンドプレイコンポーネントを求めて積極的に探索を行い位置決めすることができる。さらにまた、プレゼンタがその存在とそのプレゼンテーションについて知らせるようにこのプロセスを逆にすることもでき、その結果、参加者は、そのプレゼンテーションを見出すことができ、前述のようにそれへの許可を求めることができる。

【００５６】

以前に紹介したように、Ｍｉｃｒｏｓｏｆｔ．ＣＲＰ．ｄｌｌバイナリは、管理されたインターフェースを公開する。このＣＲＰシステムにおいて提供される主要なクラスは、プレゼンテーションクラス、参加者デバイスクラス、プロジェクタクラス、および参加者クラスを含んでいる。このプレゼンテーションクラスは、プレゼンタが利用する主要なクラスである。プレゼンテーションを開始するために、プレゼンタは、このクラスをインスタンス生成する必要があることになる。参加者デバイスクラスを使用してプレゼンテーションを閲覧する。すなわち、プレゼンテーションを閲覧するためには、このクラスをインスタンス生成する必要がある。参加者デバイスクラスは、参加者デバイスを作成し、プレゼンテーションについての招待状を提供する。プロジェクタクラスは、プロジェクタを表す。プロジェクタクラスは、プロジェクタデバイスと通信を行い、プロジェクタについて

の情報を獲得する。最後に、参加者クラスは、参加者を表す。参加者クラスは、参加者に制御を与え、または参加者を切り離すメソッドおよびプロパティを提供する。これらの各クラスについてのより詳細な説明を次に提供する。

【００５７】

Ｍｉｃｒｏｓｏｆｔ．ＣＲＰ．Ｐｒｅｓｅｎｔａｔｉｏｎクラスの概要を以下に表１に示している。

【００５８】

【表１－１】

```
Microsoft.CRP.Presentation

Constructors
public Presentation(String presentationFriendlyName, String
presenterFriendlyName)

public Presentation(String presentationFriendlyName, String
presenterFriendlyName, String password)

Properties
public Boolean InvitationOnly { get; set; }

Methods
public void Close()
public void Dispose()
public void ConnectProjector(Projector projector)
public void ConnectProjector(Projector projector, String password)
public void DisconnectProjector(Projector projector)
public void Invite(String attendeeUdn)
public void Invite(String attendeeUdn, String password)
public void Filter()

Events
public AttendeeListChanged(Object sender,
AttendeeListChangedEventArgs)
public InvitationRequested(Object Sender,
InvitationRequestedEventArgs)
```

【００５９】

【表１－２】

```
public ProjectorListChanged(Object Sender,
ProjectorListChangedEventArgs)
```

表１

【００６０】

この表１から分かるように、プレゼンテーションクラスは、２つのコンストラクタ（ｃｏｎｓｔｒｕｃｔｏｒ）を含んでいる。第１のプレゼンテーションコンストラクタは、新しいプレゼンテーションを作成する。内部的には、このコンストラクタは、まず端末サービスセッションをインスタンス生成する。この端末サービスセッションは、タイプＲＤＰのセッションである。第２に、端末サービスセッションは、このセッションをオープンする。第３に、このクラスは、そのＴＳセッションオブジェクトから招待マネージャを検索する。この招待マネージャは、その招待状を作成する機能を提供する。第４に、このコンストラクタは、一般的な招待状を作成する。次いで、このコンストラクタは、参加者を接続し切り離すイベントへとフックする。最後に、このメソッドは、ＣＲＰ　ｐｒｅｓｅｎ

ｔａｔｉｏｎ．ｄｌｌバイナリ中の登録メソッドを呼び出すことによってプレゼンテーションデバイスを登録する。この登録メソッドは、このプレゼンテーションに対してハンドルを戻すことになる。第２のプレゼンテーションコンストラクタは、一般的な招待状の生成中にパスワードが入力として提供されるという点が異なるだけで、第１のコンストラクタとまさしく同様に動作する。

【００６１】

このプレゼンテーションクラスのプロパティは、プレゼンテーションが招待専用プレゼンテーションであるか否かを制御する、招待専用プロパティを含んでいる。そのプレゼンテーションが、招待専用である場合には、招待状を求める要求はどれも拒否されることになる。そのプレゼンテーションが招待専用だけではない場合、招待状要求のイベントに対して代理人が提供されない場合には、招待状要求の呼出し側には、一般的な招待状作成構成が提供され、あるいは招待状要求のイベントに対して代理人が提供される場合には、コールバックがこのアプリケーションに与えられる。

【００６２】

このプレゼンテーションクラスが公開するメソッドは、参加者デバイスのＵＤＮによって識別される参加者を招待するために使用される招待メソッドを含んでいる。このメソッドは単に、ＣＲＰ　ｐｒｅｓｅｎｔａｔｉｏｎ．ｄｌｌバイナリが提供する参加者招待ＡＰＩを呼び出す。プレゼンテーションオブジェクトの構築中に作成される一般的な招待状が使用される。第２の招待メソッドも、参加者デバイスのＵＤＮによって識別される参加者を招待するために提供される。このメソッドを介して参加者に提供される招待状は、パスワード保護されている。このメソッドで呼び出されるステップは、ＴＳ招待マネージャを使用した招待状の作成である。この招待状は、一度だけ使用することができ、招待状が参加者ＵＤＮによって識別されるプロパティを有する。２つの招待状は、同じ名前をもつことができないので、招待状がこの参加者ＵＤＮについて作成された後には、新しい招待状を、この同じ参加者について作成することはできない。このメソッドの第２のステップは、第１のステップで作成された招待状をもつＣＲＰ　ｐｒｅｓｅｎｔａｔｉｏｎ．ｄｌｌバイナリが提供される参加者招待ＡＰＩに対する呼出しである。

【００６３】

このプレゼンテーションクラスはまた、プレゼンテーションをプロジェクタに接続するプロジェクタ接続メソッドを含んでいる。関与するステップは、このプロジェクタについての招待状の作成を含んでいる。このようにするのは、ＴＳが、参加者とプロジェクタの区別をせず、ＴＳ参加者オブジェクトを伴う非接続のイベントだけを生成するからである。このＴＳ参加者オブジェクトは、３つのプロパティ、ＩＤ、名前、および招待状を有する。ＩＤは、ＴＳセッションによって生成され、ローカルには一意的である。名前は、まさにフレンドリな名前であり、招待状は、最初に参加者に送付された招待状である。接続されているユーザから招待されたユーザをリンクするものはただ、招待状である。したがって、プロジェクタを参加者から分離するためには、このメソッドは、プロジェクタごとに別々の招待状を生成する。接続イベントが始動されるとき、ちょうど接続されたＴＳ参加者の招待状がプロジェクタに与えられたものかどうかが検証される。このケースの場合には、このプロジェクタ接続イベントが始動され、そうでない場合には、参加者接続イベントが始動される。このメソッドの第２ステップでは、一度だけ使用することができるプロパティを有する招待状が作成され、招待状の名前は、プロジェクタのＵＤＮである。これを行うことにより、所与のセッションのための１つのプロジェクタについてただ１つの招待状しか作成できない。この招待状が作成された後に、第３のステップである、ＣＲＰ　ｐｒｅｓｅｎｔａｔｉｏｎ．ｄｌｌバイナリが公開するＣｏｎｎｅｃｔＰｒｏｊｅｃｔｏｒＡＰＩが呼び出される。このＡＰＩは、このプロジェクタに対してセッショントークンを戻す。次いで、第４に、このプロジェクタについてのセッショントークンおよび招待状が、このプロジェクタオブジェクトに関連づけられる。次いで、最後に、このプロジェクタオブジェクトが、プロジェクタリストに追加される。そのプロジェクタリストは、招待されているすべてのプロジェクタを含むアレイリストである。このプロジェクタが切り

離される場合には、このプロジェクタはそのリストから取り除かれる。このシステムは、コールバックからのプロジェクタリスト上で動作することができるので、ロックによって保護される。第２のプロジェクタ接続メソッドも提供され、このメソッドと前述のプロジェクタ接続メソッドの間のただ１つの違いは、チケットが入力として提供されるパスワードを用いて生成されることである。

【００６４】

このプレゼンテーションクラスはまた、プロジェクタを切り離すプロジェクタ切離しメソッドも含んでいる。プロジェクタの切離しに関与するステップは、プロジェクタおよびプロジェクタリストを求めての探索、プロジェクタを切り離すためのＣＲＰ　ｐｒｅｓｅｎｔａｔｉｏｎ．ｄｌｌバイナリが提供するプロジェクタ切離しＡＰＩに対する呼出し、およびプロジェクタに関連するＴＳ参加者オブジェクト上の切離しメソッドに対する呼出しを含んでいる。このクラスはまた、プレゼンテーションオブジェクトを処分（ｄｉｓｐｏｓｅ）するために使用されるクローズ／処分メソッドも含んでいる。このプレゼンテーションオブジェクトを処分するのに関与するステップは、ＣＲＰ　ｐｒｅｓｅｎｔａｔｉｏｎ．ｄｌｌバイナリが公開する登録抹消メソッドを呼び出すことによってこのＰＮＰプレゼンテーションデバイスの登録を抹消することを含んでいる。最後に、このＴＳセッションがクローズされる。

【００６５】

このプレゼンテーションクラスはまた、様々なイベントも提供する。最初に、参加者リストが変更されるとき、参加者リスト変更イベントが信号によって伝えられる。参加者が接続され、または切り離されるときに、この参加者リストを変更することができ、このイベントは、ＴＳセッションによって生成される接続および切離しイベントに関連づけられる。このイベント引き数は、２つのもの、すなわち参加者オブジェクト、および生起（接続／切離し）された変更タイプを含んでいる。このイベントは、プレゼンテーションオブジェクトの構築中に、アプリケーションがＴＳセッションからの接続イベントおよび切離しイベントについて登録するときに生成される。次いで、このＴＳセッションは、任意のユーザ（参加者またはプロジェクタ）が接続を確立するときにこれらのイベントを生成する。このＴＳセッション生成イベントは、パラメータとしてＴＳ参加者オブジェクトを有する（前述のように、ＴＳ参加者は、３つのプロパティ、すなわちＩＤ、名前、および招待状を有する）。プロジェクタと参加者の間の区別をする必要があるので、前述のように、別々の招待状がプロジェクタについて作成される。このＴＳイベントがプロジェクタのために生成されるかどうかを解明するために、接続されているユーザが使用する招待状に基づいてプロジェクタリストを探索する。プロジェクタがそのリスト中に存在する場合、プロジェクタリスト変更イベントが生成され、そうでない場合には、参加者リスト変更イベントが生成される。プロジェクタが接続され、または切り離されたときに、このプロジェクタリスト変更イベントが、信号で伝えられる。以上で論じたように、このイベントは、ＴＳ接続／切離しイベントに基づいている。ユーザが、プロジェクタとして接続される場合、このＴＳ参加者オブジェクトは、プロジェクタリスト中に存在するプロジェクタオブジェクトに関連づけられる。イベント引き数プロジェクタリスト変更イベント引き数（event argument projector list changed event argument）は、２つのプロパティ、すなわちプロジェクタオブジェクト（プロジェクタリスト中に存在するオブジェクトが使用される）および生起（接続／切離し）された変更タイプを有している。

【００６６】

このプレゼンテーションクラスはまた、参加者が招待状を求めて要求したときに信号で伝えられる招待状要求イベントも有する。このイベントは、そのプレゼンテーションが招待専用である場合には信号で伝えられないはずである。この招待状は、ＣＲＰ　ｐｒｅｓｅｎｔａｔｉｏｎ．ｄｌｌバイナリが提供するコールバックに基づいている。このコールバックは、参加者が招待状を要求するためプレゼンテーションデバイスに連絡を取るときに生成される。招待状要求イベント引き数は、以下のプロパティを有する。第１が招待状を要求する参加者のフレンドリな名前であり、第２が参加者のデバイスのＵＤＮである。

アプリケーションが、このイベントについて受信しており、招待状を要求した参加者への招待状を提供しようと決定する場合には、このアプリケーションは、この招待状要求イベント引き数中で提供されるＵＤＮを有するｐｒｅｓｅｎｔａｔｉｏｎ．ｉｎｖｉｔｅを呼び出すべきである。ここでは、セキュリティの含意を理解することが重要である。参加者のＵＤＮは信用されない。したがって、明示的にユーザによって認可されない限り、ＵＤＮによって識別される参加者デバイス上にアクションを呼び出すことは望ましくない。したがって、誰もこのイベントを受信していない場合には、このＵＤＮで識別される参加者デバイス上で招待アクションを呼び出す代わりに、一般的な招待状が直ちに戻される。しかし、アプリケーションがこのイベントについて受信している場合には、このイベントが生成され、空の招待状が戻される。

【００６７】

Ｍｉｃｒｏｓｏｆｔ．ＣＲＰ．ｐｒｏｊｅｃｔｏｒクラスの概要が、以下の表２に示されている。

【００６８】

【表２】

```
Microsoft.CRP.Projector

Constructors
public Projector(String projectorUdn);

Properties
public String Capabilities{ get; }
public String DisplaySettings {get; set;}
public int State {get;}


Methods
public void Close();
public void Dispose();
public String ToString();
```

表２

【００６９】

先の表２から分かるように、このプロジェクタクラスは、１つのコンストラクタを含んでいる。このプロジェクタコンストラクタは、ＵＤＮセットだけをもつプロジェクタオブジェクトを作成する。このクラスのプロパティは、機能、ディスプレイ設定、および状態を含んでいる。この機能メソッドは、プロジェクタの機能を検索する。このメソッドは、基本的には、ＣＲＰ ｐｒｅｓｅｎｔａｔｉｏｎ．ｄｌｌバイナリが公開するＧｅｔＣａｐａｂｉｌｉｔｉｅｓＡＰＩの周囲のラッパである。ＤｉｓｐｌａｙＳｅｔｔｉｎｇｓメソッドは、そのプロジェクタのＤｉｓｐｌａｙＳｅｔｔｉｎｇｓを獲得し、または設定する。この獲得および設定は、基本的には、ＣＲＰ ｐｒｅｓｅｎｔａｔｉｏｎ．ｄｌｌバイナリが公開するＧｅｔＤｉｓｐｌａｙ設定メソッドおよびＳｅｔＤｉｓｐｌａｙ設定メソッドの周囲のラッパである。このディスプレイ設定は、プレゼンテーションがそのプロジェクタに接続される場合にのみ、獲得／設定することができる。この状態メソッドは、そのプロジェクタの状態を獲得する。このメソッドは、基本的には、ＣＲＰ ｐｒｅｓｅｎｔａｔｉｏｎ．ｄｌｌバイナリが公開するＧｅｔＳｔａｔｅＡＰＩの周囲のラッパである。このクラスが公開するメソッドは、クローズ／処分メソッドおよびＴｏＳｔｒｉｎｇメソッドである。このクローズ／処分メソッドを使用してプロジェクタオブジェクトを処分する。このオブジェクトが実施する唯一の主要なオペレーションは、ＴＳ参加者オブジェクト上で切離しを呼び出してそのプロジェクタが確実に切り離されるようにすることである。ＴｏＳｔｒｉｎｇメソッドは、そのプロジェクタのフレンドリな名前を戻す。

【００７０】
Ｍｉｃｒｏｓｏｆｔ．ＣＲＰ．ａｔｔｅｎｄｅｅ　ｄｅｖｉｃｅクラスを以下の表３に示す。
【００７１】
【表３】

```
Microsoft.CRP.AttendeeDevice

Constructors
public AttendeeDevice(String attendeeUdn);

Methods
public void Close();
public void Dispose();
public String RequestInvitation(String presentationUdn);

Events
public InvitationAvailable(Object sender, InvitationAvailableEventArgs
args);
```

表３

【００７２】
表３から分かるように、このクラスは１つのコンストラクタを含んでいる。この参加者デバイスコンストラクタは、参加者デバイスオブジェクトを作成する。プレゼンテーションを閲覧するためには、このオブジェクトを作成する必要がある。このコンストラクタは、ＣＲＰ　ａｔｔｅｎｄｅｅ．ｄｌｌバイナリが提供する登録ＡＰＩを呼び出す。この登録ＡＰＩは、参加者についてのＵＰｎＰデバイスを登録する。このクラスが公開するメソッドは、クローズ／処分メソッドおよびＲｅｑｕｅｓｔＩｎｖｉｔａｔｉｏｎメソッドを含んでいる。このクローズ／処分メソッドを使用して参加者オブジェクトを処分する。このオブジェクトが実施する唯一の主要なオペレーションは、ＣＲＰ　ａｔｔｅｎｄｅｅ．ｄｌｌバイナリが公開する登録抹消メソッドを呼び出してＵＰｎＰデバイスの登録を抹消することである。このＲｅｑｕｅｓｔＩｎｖｉｔａｔｉｏｎメソッドを使用してプレゼンタ提供のプレゼンテーションＵＤＮからの招待状を要求する。このメソッドは、基本的に、ＣＲＰ　ａｔｔｅｎｄｅｅ．ｄｌｌバイナリが公開するＲｅｑｕｅｓｔＩｎｖｉｔａｔｉｏｎＡＰＩを呼び出す。プレゼンタは、直ちに招待状を提供することができ、また参加者デバイス上で招待ＡＰＩを呼び出すこともできる。プレゼンタが招待状を直ちに提供する場合には、このメソッドは、招待状を戻し、そうでない場合には、このメソッドは、ｓｔｒｉｎｇ．ｅｍｐｔｙを戻す。このクラスについて提供される１つのイベントは、招待状利用可能イベントである。プレゼンタが参加者を招待するときにこのイベントが信号で伝えられる。この招待状利用可能イベント引き数クラス中で利用可能な情報は、招待状それ自体、プレゼンテーションフレンドリな名前、およびプレゼンタフレンドリな名前を含んでいる。参加者がこの招待状を使用したいと思う場合には、参加者は、このＴＳＡｃｔｉｖｅＸコントロールに招待状を提供し、このＡｃｔｉｖｅＸコントロール上で接続を呼び出す必要がある。次いで、このＡｃｔｉｖｅＸコントロールは、サーバに対してＴＳセッションを確立することになる。
【００７３】
Ｍｉｃｒｏｓｏｆｔ．ＣＲＰ．Ａｔｔｅｎｄｅｅクラスの概要を以下の表４に示す。
【００７４】

【表４】

```
Microsoft.CRP.Attendee

Properties
public String Name {get;}
public ControlLevel ControlLevel { get; set; }

Methods
public void Disconnect();
public String ToString();
```

表４

【００７５】

このクラスのプロパティは、名前および制御レベルを含んでいる。この命名メソッドは、この参加者のフレンドリな名前を戻す。この制御レベルメソッドは、この参加者の制御レベルを獲得／設定する。参加者が有することができる制御レベルは、対話型（すなわち、参加者はプレゼンタのマウスポインタを制御することができる）およびビュー（参加者はプレゼンテーションを閲覧することしかできない）を含んでいる。このクラスはまた、ＤｉｓｃｏｎｎｅｃｔメソッドおよびＴｏＳｔｒｉｎｇメソッドを公開する。このＤｉｓｃｏｎｎｅｃｔメソッドは参加者を切り離すが、ＴｏＳｔｒｉｎｇメソッドは、参加者のフレンドリな名前を戻す。

【００７６】

本発明の実施形態の一例では、ＡＰＩ（ａｐｐｌｉｃａｔｉｏｎ ｐｒｏｇｒａｍｍｉｎｇ ｉｎｔｅｒｆａｃｅアプリケーションプログラミングインターフェース）を使用してプレゼンテーションを閲覧しプレゼンテーションを提供するというシナリオを可能にしている。これらのＡＰＩは、一般にユーザにとっては使用可能であり、一般にプロジェクタにとっては使用可能ではない。プロジェクタ端末上での実装は、普通はプロジェクタの製造業者の責任であるが、これは本発明の範囲に対する限定とはならない。本発明に従って構築されるかかる例示の実施形態について、以下で説明する。好ましい一実施形態においては、これらのＡＰＩは、管理されたコードを含む。しかし、ＡＰＩが例示の実施形態として表されているが、本発明の範囲を限定しないことを理解すべきである。

【００７７】

これから説明する第１の組のＡＰＩは、プレゼンテーションの閲覧を可能にするＡＰＩである。これらのＡＰＩを利用する典型的なシナリオは、ユーザがプレゼンテーションをはっきりと見出し、適切な招待状を介してそのプレゼンテーションに参加することによってプレゼンテーションに参加することを望むとき、またはユーザが自分を発見可能にし、次いでそのプレゼンテーションに参加するように招待されることを望むときである。この機能をサポートするＡＰＩおよびコールバックメソッドは、ＲｅｇｉｓｔｅｒＡＰＩ、ＵｎｒｅｇｉｓｔｅｒＡＰＩ、およびＲｅｑｕｅｓｔＩｎｖｉｔａｔｉｏｎ ＡＰＩである。

【００７８】

説明すべき第１のＡＰＩの実施形態の一例、すなわちＲｅｇｉｓｔｅｒＡＰＩが、以下の表５に提供されている。

【００７９】

【表５】

```
HRESULT
Register (
      IN    PCWSTR                          pwzFriendlyName,
      IN    PFN_INVITE                pfnInvite,
      OUT   HANDLE                          * phAttendee);
```

表５

【００８０】

このＡＰＩは、ＵｓｅｒＤｉｓｐｌａｙＤｅｖｉｃｅを登録する。このデバイスが登録された後には、これをプレゼンタが見出すことができる。次いで、これらのプレゼンタは、そのプレゼンテーションに参加するための招待状をユーザに対して送付することができる。このＡＰＩ用のパラメータは、このディスプレイデバイスのフレンドリな名前とプレゼンタが参加者を招待するときに呼び出されるコールバックファンクションとである。出力パラメータは、後で登録を抹消するときに使用すべきハンドルである。このＡＰＩについてのリターン値は、正常な場合にはＳ＿ＯＫ、またはこのオペレーションを完了するだけの十分なメモリがない場合にはＥ＿ＯＵＴＯＦＭＥＭＯＲＹである。

【００８１】

ＵｎｒｅｇｉｓｔｅｒＡＰＩの実施形態の一例が、以下の表６に提示されている。

【００８２】

【表６】

```
VOID Unregister (IN HANDLE hAttendee);
```

表６

【００８３】

このＡＰＩは、ディスプレイデバイスの登録を抹消する。このＡＰＩが実行された後は、すべてのアクティブセッションは、終了させられ、ユーザは、他の参加者またはプレゼンタが見出すことができなくなる。このＡＰＩは、Ｒｅｇｉｓｔｅｒファンクションから戻されたハンドルをパラメータとして必要とし、この実施形態中ではどのようなリターン値も提供しない。

【００８４】

ＲｅｑｕｅｓｔＩｎｖｉｔａｔｉｏｎＡＰＩの実施形態の一例が、以下の表７に提示されている。

【００８５】

【表７】

```
HRESULT
RequestInvitation (
      IN    HANDLE    hAttendee,
      IN    BSTR      bstrPresentationUdn,
      OUT   BSTR      * pbstrInvitation);
```

表７

【００８６】

以上で説明したように、本発明の一実施形態では、プレゼンテーションはＵＰｎＰデバイスである。ＲｅｑｕｅｓｔＩｎｖｉｔａｔｉｏｎＡＰＩが呼び出されるとき、このデバイスのＵＤＮが入力として提供される。このＵＤＮに基づいて、プレゼンテーションデバ

イスが取得される。このプレゼンテーションデバイスにおいて、次いでアクションＲｅｑｕｅｓｔＩｎｖｉｔａｔｉｏｎがユーザについての情報を用いて呼び出される。プレゼンタが、ユーザがプレゼンテーションに参加できるようにする場合、そのプレゼンタは招待状を返信する。このＡＰＩ用のパラメータは、Ｒｅｇｉｓｔｅｒメソッドが戻すハンドル、そのプレゼンテーションについてのＵＤＮ、および出力としての招待状である。プレゼンタは、直ちに招待状を提供しようと決定することもできる。プレゼンタが招待状を提供しない場合には、これは空の文字列を含むことになる。このＡＰＩは、成功の表示、または不十分なメモリまたは無効な引き数がある場合には失敗の表示を提供する。

【００８７】

Ｒｅｇｉｓｔｅｒメソッドについて提供されるコールバックメソッドの実施形態の一例、すなわちＩｎｖｉｔｅコールバックが以下の表８に提示されている。

【００８８】

【表８】

```
HRESULT
Invite (
      IN    BSTR      bstrInvitation,
      IN    BSTR      bstrPresentationName,
      IN    BSTR      bstrPresenterName);
```

表８

【００８９】

このメソッドは、あるプレゼンタが、プレゼンテーションを閲覧するようにユーザを招待したときに呼び出される。次いで、コールバックにおいて提供される招待状を端末サービスＡＰＩと共に使用して最終的にそのプレゼンテーションを実際に閲覧するのに必要な接続を確立する。このコールバックメソッドについてのパラメータは、そのプレゼンタが提供する招待状を含んでいる。この招待状を使用してそのプレゼンテーションに参加することができる。他のパラメータは、そのプレゼンテーションのフレンドリな名前、およびそのプレゼンタのフレンドリな名前を含んでいる。このコールバックメソッドは、成功の表示を提供する。

【００９０】

以下に説明する次の組のＡＰＩは、ユーザがプレゼンテーションを提供したいと思うときに使用されるＡＰＩである。本発明の一実施形態では、これらのＡＰＩは、Ｒｅｇｉｓｔｅｒ　ＡＰＩ、ＵｎｒｅｇｉｓｔｅｒＡＰＩ、ＩｎｖｉｔｅＡｔｔｅｎｄｅｅＡＰＩ、ＤｉｓｃｏｎｎｅｃｔＰｒｏｊｅｃｔｏｒＡＰＩ、ＧｅｔＳｔａｔｅＡＰＩ、ＧｅｔＣａｐａｂｉｌｉｔｉｅｓＡＰＩ、ＧｅｔＤｉｓｐｌａｙＳｅｔｔｉｎｇｓＡＰＩ、およびＳｅｔＤｉｓｐｌａｙＳｅｔｔｉｎｇｓＡＰＩを含んでいる。

【００９１】

次にこのプレゼンテーションＡＰＩの説明に入るが、以下にＲｅｇｉｓｔｅｒＡＰＩの実施形態の一例が、表９に提示されている。

【００９２】

【表９】

```
HRESULT
Register (
    IN          BSTR
    bstrPresentationFriendlyName,
        IN      BSTR
    bstrPresenterFriendlyName,
        IN      PFN_REQUEST_INVITATAION         pfnRequestInvitation,
        OUT     PHANDLE                         phPresentation);
```

表９

【００９３】

プレゼンタが、プレゼンテーションを開始したいと思うときにこのメソッドが呼び出される。かかるセッションがこのメソッドを使用して作成されるとき、以下のステップが実施される。すなわち、プレゼンテーションデバイスが、ＵＰｎＰを用いて作成され登録される。前述のように、これによって誰でもそのプレゼンテーションを見出すことができるようになる。このメソッドと共に使用するためのパラメータは、プレゼンテーションのフレンドリな名前、プレゼンタのフレンドリな名前、およびユーザがそのプレゼンテーションを閲覧するのを要求する場合に関与するコールバックファンクションを含んでいる。このＡＰＩはさらに、作成されるプレゼンテーションセッションについてのハンドルを提供する。このＡＰＩは、不十分なメモリ、またはパラメータが無効である場合に起因する成功または失敗の表示を戻す。

【００９４】

ＵｎｒｅｇｉｓｔｅｒＡＰＩの実施形態の一例が、以下の表１０に提示されている。

【００９５】

【表１０】

```
VOID
Unregister (
      IN     HANDLE              hPresentation);
```

表１０

【００９６】

このＡＰＩは、プレゼンタがプレゼンテーションの登録を抹消したいと思うときに呼び出される。プレゼンテーションの登録を抹消するときは、そのデバイスは、ＵＰｎＰ中で登録を抹消される。この実施形態におけるこのＡＰＩについての唯一のパラメータは、停止すべきそのプレゼンテーションである。

【００９７】

ＩｎｖｉｔｅＡｔｔｅｎｄｅｅＡＰＩの実施形態の一例が、以下の表１１に提示されている。

【００９８】

【表１１】

```
HRESULT
InviteAttendee (
      IN     HANDLE                          hPresentation,
      IN     BSTR                            bstrUdn,
      IN     BSTR                            bstrInvitation);
```

表１１

【００９９】
このＡＰＩは、プレゼンタが、そのプレゼンテーションを受け取るように参加者を招待したいと思うときに呼び出される。このメソッドが呼び出されるとき、プレゼンタはＵＤＮを使用してそのデバイスを検索する。次に、この招待状を伴う、所与のＵＤＮについてのディスプレイデバイス上のＩｎｖｉｔｅメソッドに対する呼出しを行う。このメソッドについてのパラメータは、このプレゼンテーションに対するハンドル、招待すべき参加者のディスプレイデバイスのＵＤＮ、および参加者に提供すべき招待状を含んでいる。このＡＰＩは、成功または失敗の表示を戻す。この失敗は、そのオペレーションを完了するには不十分なメモリ、ハンドルが無効な場合の無効な引き数に起因することもある。
【０１００】
ＣｏｎｎｅｃｔＰｒｏｊｅｃｔｏｒＡＰＩの実施形態の一例が、以下の表１２に提供されている。
【０１０１】
【表１２－１】

```
HRESULT
ConnectProjector (
      IN    HANDLE                          hPresentation,
```

【０１０２】
【表１２－２】

```
      IN    BSTR                            bstrUdn,
      IN    BSTR                            bstrInvitation,
      OUT   BSTR                            * pbstrSessionToken);
```

表１２

【０１０３】
このＡＰＩが呼び出されるとき、この投影のＵＤＮが検索される。次いで、Ｉｎｖｉｔｅメソッドが、この招待状を伴う所与のＵＤＮについての投影デバイス上で呼び出される。このファンクションが正常に完了される場合には、セッショントークンが検索される。そのパラメータは、そのプレゼンテーションへのハンドル、そのプロジェクタに提供すべき招待状、および出力としてのセッショントークンである。このＡＰＩは、成功または失敗の表示を戻す。この失敗は、そのオペレーションを完了するには不十分なメモリ、ハンドルが無効な場合の無効な引き数に起因することもある。
【０１０４】
ＤｉｓｃｏｎｎｅｃｔＰｒｏｊｅｃｔｏｒＡＰＩの実施形態の一例が、以下の表１３に提供されている。
【０１０５】
【表１３】

```
HRESULT
DisconnectProjector (
      IN    HANDLE                          hPresentation,
      IN    BSTR                            bstrUdn,
      IN    BSTR                            bstrSessionToken);
```

表１３

【０１０６】
このＡＰＩを呼び出して指定されたデバイスへの接続を終了する。このＡＰＩについてのパラメータは、そのプレゼンテーションへのハンドルを含んでいる。切り離すべきプロ

ジェクタのＵＤＮおよび切り離すべきセッションである。このＡＰＩは、成功または失敗の表示を戻す。この失敗は、そのオペレーションを完了するには不十分なメモリ、ハンドルが無効な場合の無効な引き数に起因することもある。

【０１０７】

ＧｅｔＰｒｏｊｅｃｔｏｒＣａｐａｂｉｌｉｔｉｅｓＡＰＩの実施形態の一例が、以下の表１４に提示されている。

【０１０８】

【表１４】

```
HRESULT
GetProjectorCapabilities (
      IN    BSTR                          bstrUdn,
      OUT   PBSTR                         * pbstrCapabilities);
```

表１４

【０１０９】

このメソッドは、プレゼンタがプロジェクタの機能を取得したいと思うときに呼び出される。この投影の機能は、その投影デバイス上でＧｅｔＣａｐａｂｉｌｉｔｉｅｓメソッドを呼び出すことによって取得される。プレゼンタは、そのプロジェクタのＵＤＮを入力し、このメソッドが、そのプロジェクタに関連する機能を出力する。このＡＰＩは、成功または失敗の表示を戻す。この失敗は、そのオペレーションを完了するには不十分なメモリ、ハンドルが無効な場合の無効な引き数に起因することもある。

【０１１０】

ＧｅｔＰｒｏｊｅｃｔｏｒＳｔａｔｅＡＰＩの実施形態の一例が、以下の表１５に提示されている。

【０１１１】

【表１５】

```
HRESULT
GetProjectorState (
      IN    BSTR                          bstrUdn,
      OUT   ULONG                         * pState);
```

表１５

【０１１２】

このメソッドは、プレゼンタが、プロジェクタの現在の状態を検索したいと思うときに呼び出される。このプロジェクタ状態は、投影デバイス上でＧｅｔＳｔａｔｅメソッドを呼び出すことによって取得される。プレゼンタは、プロジェクタのＵＤＮを入力し、このメソッドは、このプロジェクタの状態を出力する。このＡＰＩは、成功または失敗の表示を戻す。この失敗は、そのオペレーションを完了するには不十分なメモリ、ハンドルが無効な場合の無効な引き数に起因することもある。

【０１１３】

ＧｅｔＰｒｏｊｅｃｔｏｒＤｉｓｐｌａｙＳｅｔｔｉｎｇｓＡＰＩの実施形態の一例が、以下の表１６に提示されている。

【０１１４】

【表１６】

```
HRESULT
GetProjectorDisplaySettings (
      IN    BSTR                          bstrUdn,
      IN    BSTR                          bstrSessionToken,
      OUT   PBSTR                         * pbstrDisplaySettings);
```

表１６

【０１１５】

このメソッドは、プレゼンタが、プロジェクタについてのディスプレイ設定を検索したいと思うときに呼び出される。このプロジェクタのディスプレイ設定は、このプロジェクタ上のＧｅｔＰｒｏｊｅｃｔｉｏｎＳｅｔｔｉｎｇｓメソッドを呼び出すことによって取得される。プレゼンタは、このプロジェクタのＵＤＮ、ＣｏｎｎｅｃｔＰｒｏｊｅｃｔｏｒが成功したときに取得されるセッショントークンを入力し、このメソッドは、このプロジェクタについてのディスプレイ設定を出力する。このＡＰＩは、成功または失敗の表示を戻す。この失敗は、そのオペレーションを完了するには不十分なメモリ、ハンドルが無効な場合の無効な引き数に起因することもある。

【０１１６】

ＳｅｔＰｒｏｊｅｃｔｏｒＤｉｓｐｌａｙＳｅｔｔｉｎｇｓＡＰＩの実施形態の一例が、以下の表１７に提示されている。

【０１１７】

【表１７】

```
HRESULT
SetProjectorDisplaySettings (
      IN    BSTR                          bstrUdn,
      IN    BSTR                          bstrSessionToken,
      IN    BSTR                          bstrDisplaySettings);
```

表１７

【０１１８】

このメソッドは、プレゼンタが、プロジェクタについてのディスプレイ設定を設定したいと思うときに呼び出される。このディスプレイ設定は、このプロジェクタデバイス上でＳｅｔＤｉｓｐｌａｙＳｅｔｔｉｎｇｓを行うことによって設定される。プレゼンタは、このプロジェクタのＵＤＮ、ＣｏｎｎｅｃｔＰｒｏｊｅｃｔｏｒが成功したときに取得されるセッショントークン、およびこのプロジェクタについて設定すべきディスプレイ設定を入力する。このＡＰＩは、成功または失敗の表示を戻す。この失敗は、そのオペレーションを完了するには不十分なメモリ、ハンドルが無効な場合の無効な引き数に起因することもある。

【０１１９】

以上で紹介したように、ＳｔａｒｔＰｒｅｓｅｎｔａｔｉｏｎＡＰＩは、コールバックファンクションを含んでいる。ＰＦＮ＿ＲＥＱＵＥＳＴ＿ＩＮＶＩＴＡＴＩＯＮコールバックファンクションの一実施形態が以下の表１８に提示されている。

【０１２０】

【表１８】

```
HRESULT
RequestInvitation (
    IN      BSTR            bstrName,
        IN  BSTR            bstrUdn,
        OUT BSTR            * pbstrInvitation);
```

表１８

【０１２１】

このコールバックファンクションは、ユーザがそのプレゼンテーションデバイス上でＲｅｑｕｅｓｔＩｎｖｉｔａｔｉｏｎメソッドを呼び出したときに呼び出される。このコールバックは、その招待状それ自体を出力パラメータとして戻すことができ、また空の文字列をただ戻すこともできる。このメソッドのパラメータは、その招待状を要求する参加者のフレンドリな名前、その招待状を要求する参加者のＵＤＮ、および出力としてのその参加者に提供すべき招待状である。このメソッドは、成功の表示を戻す。

【０１２２】

特許、特許出願、および出版を含めて、本明細書中で引用されたすべての参照は、ここにその全体が参照によって組み込まれている。すなわち、かかるあらゆる参照文献の各部分および全部が、本開示の一部をなすものと考えられ、したがって、かかるどの参照文献のどの部分もこの記述または本開示における他のどの記述によっても本開示の一部をなすことから除外されない。

【０１２３】

本発明の様々な実施形態についての以上の説明を、例証と説明の目的で提示してきた。本発明を網羅的に説明することも、また本発明を開示された厳密な実施形態に限定することも意図していない。多数の変更形態または変形形態が、前述の教示を考慮すれば可能である。説明したこれらの実施形態は、本発明の原理と、それによって当業者が本発明を様々な実施形態において、また企図される特定の使用に適した様々な変更形態を用いて利用できるようにするその実際的な応用との最良の例証を提供するために選択し説明したものである。かかる変更形態および変形形態はすべて、公平に、法律的に、公正に権利が与えられる広さに従って解釈されるとき、添付の特許請求の範囲によって決定される本発明の範囲内に含まれる。

【図面の簡単な説明】

【０１２４】

【図１】本発明が常駐するコンピュータシステムの一例を全般的に示すブロック図である。

【図２】２つのコードセグメントの間のプログラミングインターフェースを示す簡略化されたブロック図である。

【図３】２つのコードセグメントの間のプログラミングインターフェースの代替実施形態を示す簡略化されたブロック図である。

【図４】ファクタ分解の概念を示す、複数の個別のコミュニケーションに分割されるコミュニケーションを有する、２つのコードセグメント間のプログラミングインターフェースを示す簡略化されたブロック図である。

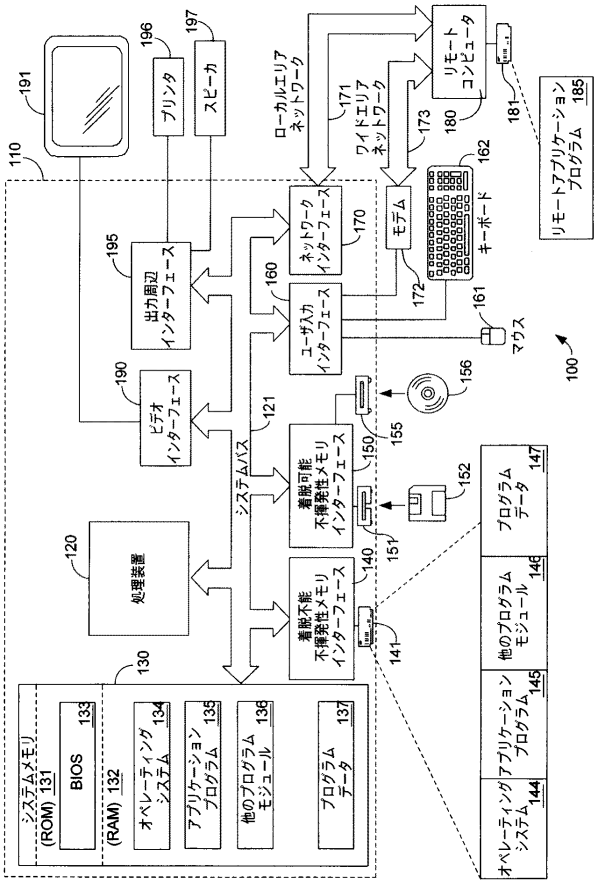
【図５】ファクタ分解の概念を示す、複数の個別のコミュニケーションに分割されるコミュニケーションを有する、２つのコードセグメント間のプログラミングインターフェースの代替実施形態を示す簡略化されたブロック図である。

【図６】再定義の概念を示す、無視され、追加され、または再定義されるある種の態様を有する、２つのコードセグメント間のプログラミングインターフェースを示す簡略化されたブロック図である。

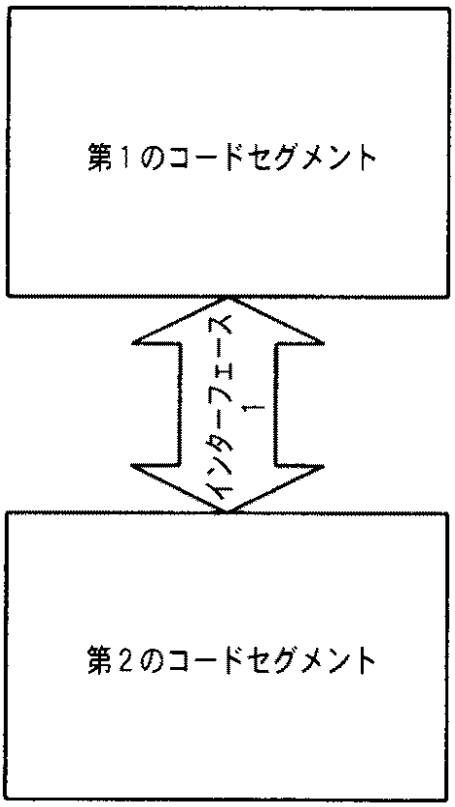
【図７】再定義の概念を示す、無視され、追加され、または再定義されるある種の態様を

有する、２つのコードセグメント間のプログラミングインターフェースの代替実施形態を示す簡略化されたブロック図である。
【図８】インラインコーディングの概念を示す、それらの間のインターフェースが形式を変更するようにマージされる２つのコードモジュールの機能の一部を有する、２つのコードセグメント間のプログラミングインターフェースを示す簡略化されたブロック図である。
【図９】インラインコーディングの概念を示す、それらの間のインターフェースが形式を変更するようにマージされる２つのコードモジュールの機能の一部を有する、２つのコードセグメント間のプログラミングインターフェースの代替実施形態を示す簡略化されたブロック図である。
【図１０】分離の概念を示す、コミュニケーションが複数の別個のコミュニケーションにコミュニケーションを分割することによって間接的に実現される、２つのコードモジュール間のプログラミングインターフェースを示す簡略化されたブロック図である。
【図１１】分離の概念を示す、コミュニケーションが複数の別個のコミュニケーションにコミュニケーションを分割することによって間接的に実現される、２つのコードモジュール間のプログラミングインターフェースの代替実施形態を示す簡略化されたブロック図である。
【図１２】書換えの概念を示す、同じ結果を実現する何か他のものでプログラミングインターフェースを置き換える動的に書き換えられたコードを示す簡略化されたブロック図である。
【図１３】書換えの概念を示す、同じ結果を実現する何か他のものでプログラミングインターフェースを置き換える動的に書き換えられたコードの代替実施形態を示す簡略化されたブロック図である。
【図１４ａ】投影を行うコンピュータと複数の投影ターゲットコンピュータとを備える複数のコンピュータ、ならびに電子会議室のディスプレイ画面またはプロジェクタを含めて本発明の一実施形態を実装することができるネットワークシステムの代替アーキテクチャを示す概略図である。
【図１４ｂ】投影を行うコンピュータと複数の投影ターゲットコンピュータとを備える複数のコンピュータ、ならびに電子会議室のディスプレイ画面またはプロジェクタを含めて本発明の一実施形態を実装することができるネットワークシステムの代替アーキテクチャを示す概略図である。
【図１５】本発明の一実施形態における、投影アプリケーションプログラムコンポーネントと関連するコンポーネントとの配置および相互接続をより詳細に示す概略図である。
【符号の説明】
【０１２５】
２０１　プレゼンタのコンピューティングデバイス
２０３，２０５，２０７　参加者（コンピューティングデバイス）
２０９　参加者（ディスプレイデバイス）
２１１　ネットワーク
２１３，２１５，２１７，２１９，２２５　無線リンク
５００　ＵＰｎＰプレゼンテーションデバイス
５０２　ＵＰｎＰユーザディスプレイデバイス
５０４，５０６，５０８　バイナリ

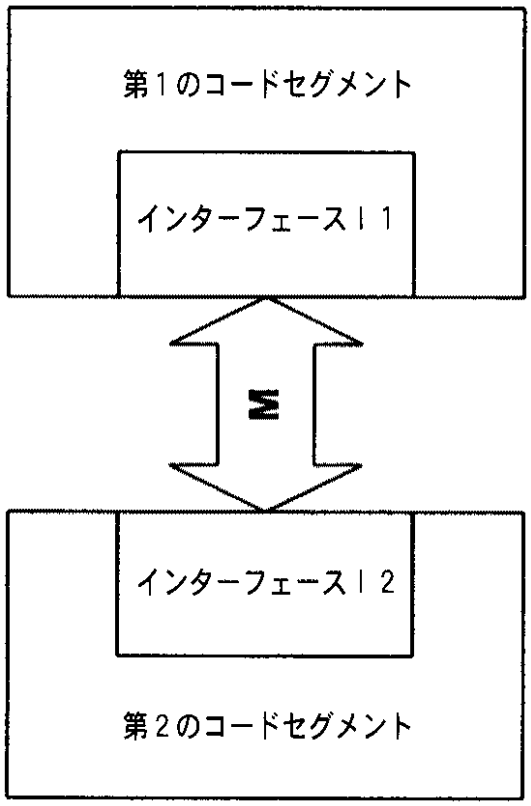
【図１】


110
191
196
プリンタ
197
スピーカ
ローカルエリアネットワーク
171
ワイドエリアネットワーク
リモートコンピュータ
180
181
リモートアプリケーションプログラム 185
162
キーボード
モデム
ネットワークインターフェース
170
195
出力周辺インターフェース
160
ユーザ入力インターフェース
172
161
マウス
100
190
ビデオインターフェース
121
システムバス
156
155
150
着脱可能不揮発性メモリインターフェース
152
151
140
着脱不能不揮発性メモリインターフェース
141
120
処理装置
130
システムメモリ
(ROM) 131
BIOS 133
(RAM) 132
オペレーティングシステム 134
アプリケーションプログラム 135
他のプログラムモジュール 136
プログラムデータ 137
プログラムデータ 147
他のプログラムモジュール 146
アプリケーションプログラム 145
オペレーティングシステム 144

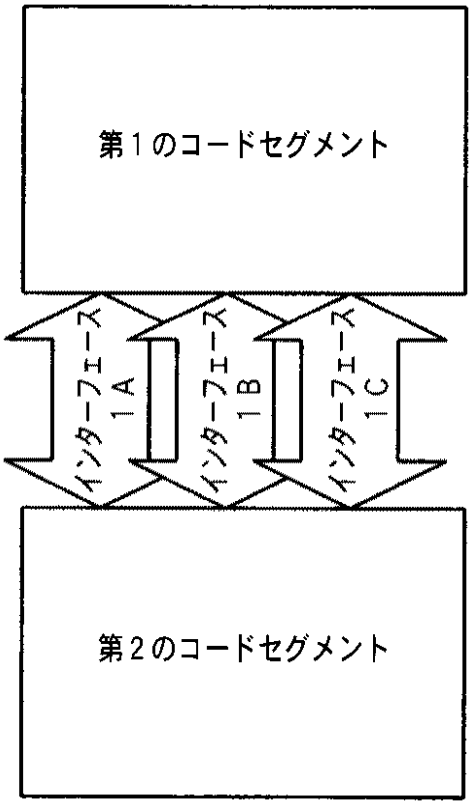
【図２】


第１のコードセグメント
インターフェース１
第２のコードセグメント

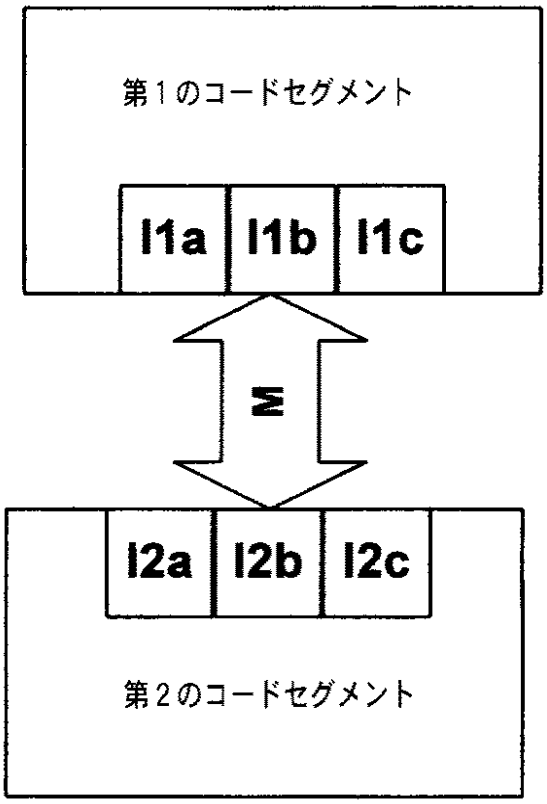
【図３】


第１のコードセグメント
インターフェース１１
M
インターフェース１２
第２のコードセグメント

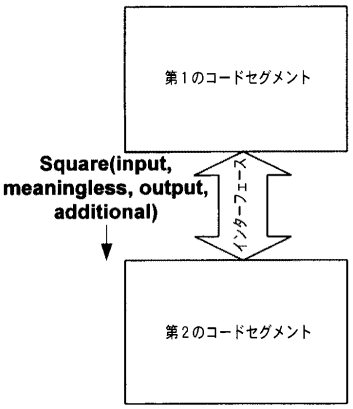
【図４】


第１のコードセグメント
インターフェース１Ａ
インターフェース１Ｂ
インターフェース１Ｃ
第２のコードセグメント

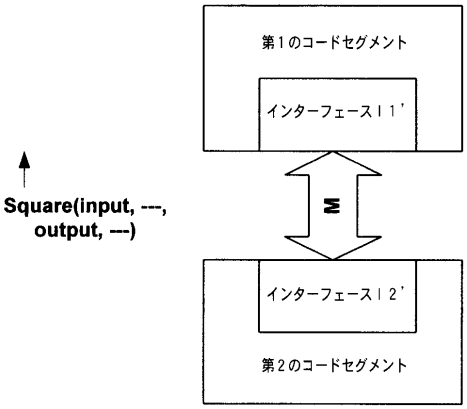
【図５】


第１のコードセグメント
I1a
I1b
I1c
M
I2a
I2b
I2c
第２のコードセグメント

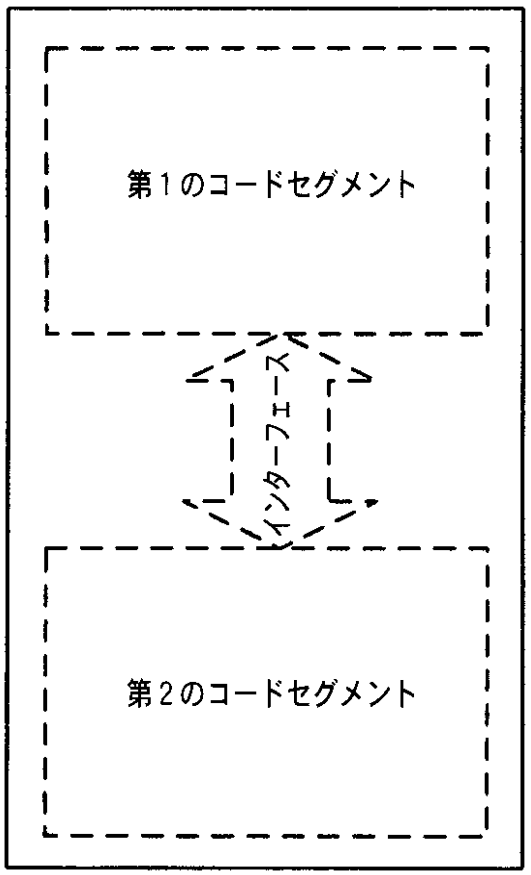
【図６】


第１のコードセグメント
Square(input, meaningless, output, additional)
インターフェース
第２のコードセグメント

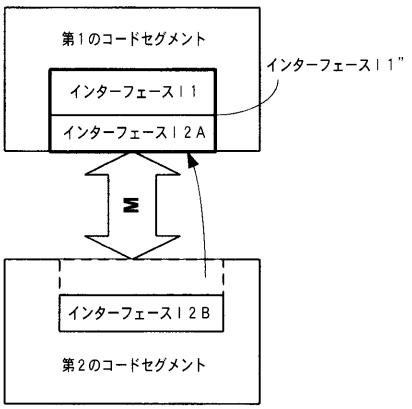
【図７】


第１のコードセグメント
インターフェースＩ１'
Square(input, ---, output, ---)
M
インターフェースＩ２'
第２のコードセグメント

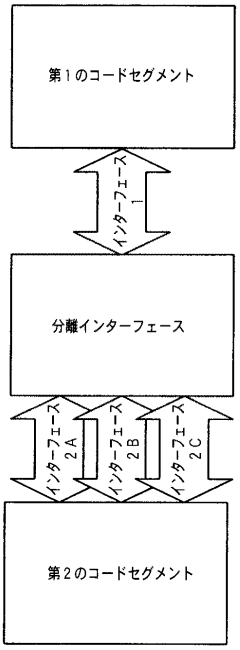
【図８】


第１のコードセグメント
インターフェース
第２のコードセグメント

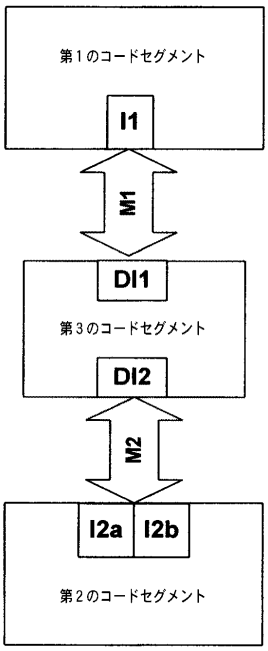
【図９】


第１のコードセグメント
インターフェースＩ１"
インターフェースＩ１
インターフェースＩ２Ａ
M
インターフェースＩ２Ｂ
第２のコードセグメント

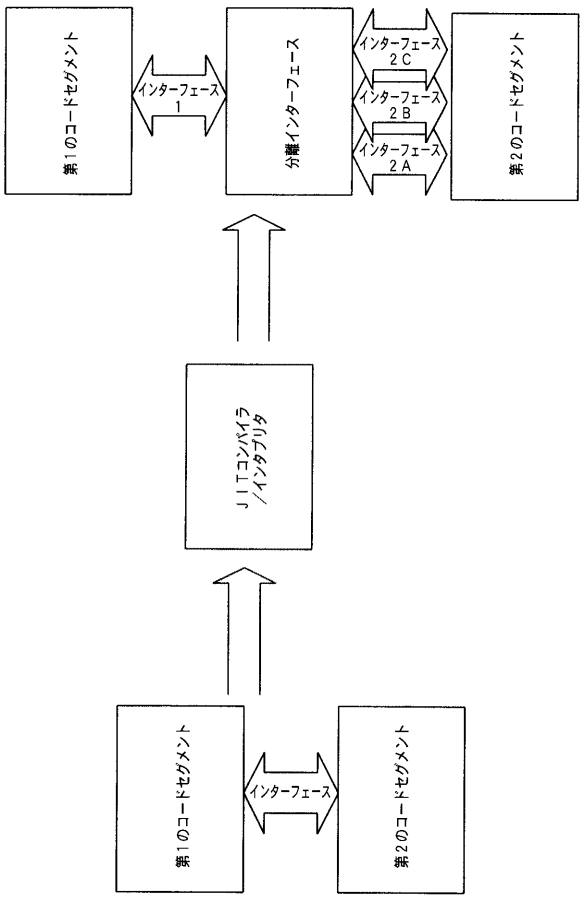
【図１０】



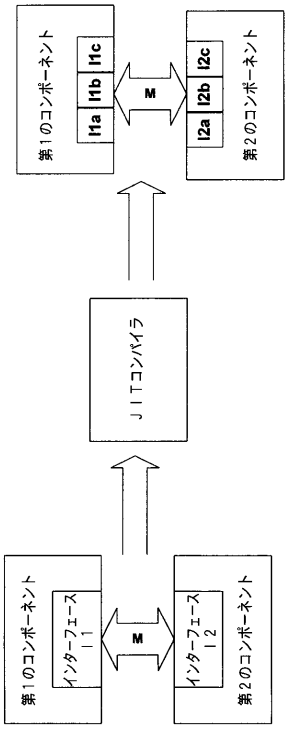
【図１１】



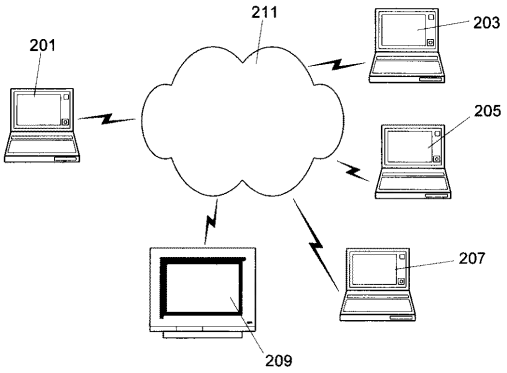
【図１２】



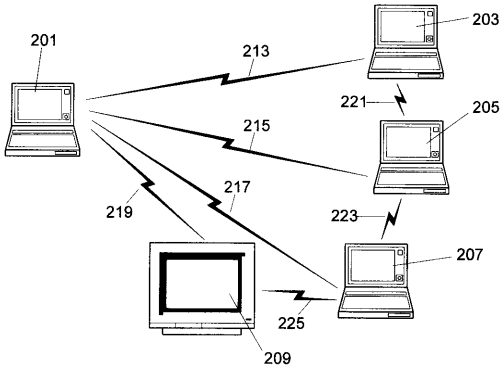
【図１３】



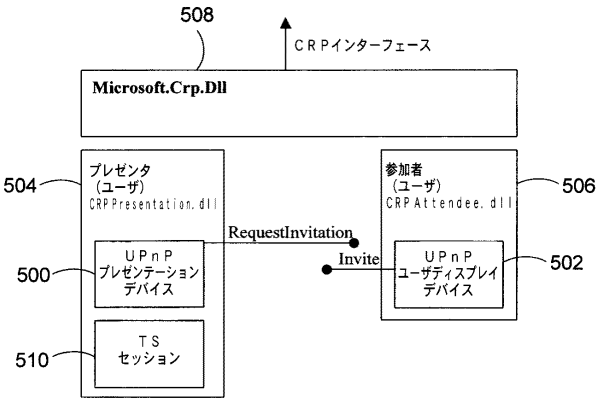
【図１４ａ】



【図１４ｂ】



【図１５】



フロントページの続き

(72)発明者 ローイット グプタ
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
(72)発明者 トッド アール. マニオン
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

合議体
審判長 清水 稔
審判官 甲斐 哲雄
審判官 山田 正文

(56)参考文献 特開２００２－６４４１８（ＪＰ，Ａ）
国際公開第０１／００８３５３（WO，Ａ１）
米国特許第６５６０６３７（ＵＳ，Ｂ１）
特開２００２－９４５３１（ＪＰ，Ａ）
特開２００３－２１８８７７（ＪＰ，Ａ）
国際公開第０３／０６２９９９（WO，Ａ２）
特開平１１－１９４９８１（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 13/00, G06F 15/00