

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2006-107439

(P2006-107439A)

(43) 公開日 平成18年4月20日(2006.4.20)

(51) Int. Cl.	F I	テーマコード (参考)
<b>G06F 17/21 (2006.01)</b>	G06F 17/21 530K	5B009
<b>G06F 3/048 (2006.01)</b>	G06F 17/21 580J	5E501
	G06F 3/00 651B	

審査請求 未請求 請求項の数 24 O L 外国語出願 (全 20 頁)

(21) 出願番号 特願2005-183574 (P2005-183574)  
 (22) 出願日 平成17年6月23日 (2005. 6. 23)  
 (31) 優先権主張番号 10/954, 328  
 (32) 優先日 平成16年9月30日 (2004. 9. 30)  
 (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 ジョナサン ピー. ローゼンバーグ  
 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

最終頁に続く

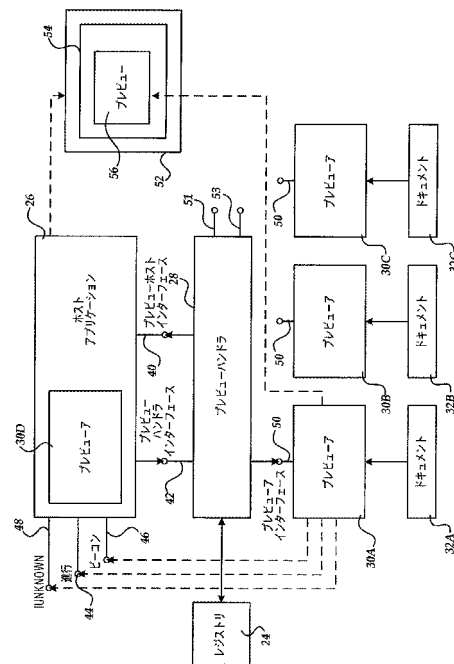
(54) 【発明の名称】 ドキュメントプレビューを提供する方法、システムおよび装置

(57) 【要約】

【課題】 ドキュメントプレビューを生成するための方法および装置を提供すること。

【解決手段】 本方法に従って、プレビューハンドラは、1つまたは複数のプレビューアに対するインターフェースを提供する。これらのプレビューアは、1つまたは複数のドキュメントタイプについてのビジュアルプレビューを生成することが可能なソフトウェアコンポーネントである。ドキュメントについてのプレビューを生成する、ホストアプリケーションからの要求にตอบสนองして、プレビューハンドラは、ドキュメントについてのプレビューを生成するための適切なプレビューアを識別するように動作可能である。次いで、プレビューハンドラは、プレビューアを呼び出し、プレビューアに指示して、ドキュメントをロードし、呼び出しているホストアプリケーションが提供するウィンドウのサブウィンドウに、プレビューを示すことができる。

【選択図】 図2



**【特許請求の範囲】****【請求項 1】**

ドキュメントプレビューを提供する方法であって、

1つまたは複数のホストアプリケーションによって呼び出し可能であり、前記ホストアプリケーションによって識別されるドキュメントのビジュアルプレビューを生成するためのプレビューアを識別しロードするように動作可能なプレビューハンドラを提供するステップと、

前記プレビューハンドラにおいてホストアプリケーションからドキュメントのビジュアルプレビューを生成する要求を受け取るステップと、

前記要求を受け取るステップに回答して、前記ドキュメントの前記プレビューを生成することが可能なプレビューアを識別するステップと、 10

前記ドキュメントをロードし、前記ドキュメントの前記ビジュアルプレビューを生成するように前記プレビューアに指示する要求を前記プレビューハンドラから前記プレビューアへと送信するステップと

を備えたことを特徴とする方法。

**【請求項 2】**

前記ホストアプリケーションからの要求は、前記プレビューを生成すべきウィンドウに対するハンドルをさらに含み、および前記プレビューアは、前記ホストアプリケーションによって識別される前記ウィンドウ内のサブウィンドウ内に前記プレビューを表示するようさらに動作可能であることを特徴とする請求項 1 に記載の方法。 20

**【請求項 3】**

前記ドキュメントの前記プレビューを生成することが可能なプレビューアを識別するステップは、前記ホストアプリケーションに前記プレビューアを識別するように要求するステップを含むことを特徴とする請求項 2 に記載の方法。

**【請求項 4】**

前記ドキュメントの前記プレビューを生成することが可能なプレビューアを識別するステップは、前記ホストアプリケーションに前記プレビューアに対応するクラス識別子を識別するように要求するステップを含むことを特徴とする請求項 2 に記載の方法。

**【請求項 5】**

前記ドキュメントの前記プレビューを生成することが可能なプレビューアを識別するステップは、レジストリの内容を検索して前記プレビューアを識別するステップを含むことを特徴とする請求項 2 に記載の方法。 30

**【請求項 6】**

前記プレビューアが、前記ドキュメントを正常にロードしたかどうかを決定するステップと、

前記プレビューアが、前記ドキュメントを正常にロードしなかったと決定するステップに回答して、前記プレビューハンドラから前記ホストアプリケーションにエラーメッセージを戻すステップと

をさらに備えたことを特徴とする請求項 2 に記載の方法。

**【請求項 7】**

前記プレビューハンドラにおいて前記ホストアプリケーションから前記エラーメッセージに対応するメッセージを表示する要求を受け取るステップと、 40

前記要求に回答して前記メッセージを表示するステップと

をさらに備えたことを特徴とする請求項 6 に記載の方法。

**【請求項 8】**

前記ホストアプリケーションから前記プレビューハンドラにおいて進行インターフェースに対するポインタを受け取るステップと、

前記進行インターフェースに対する前記ポインタを前記プレビューハンドラから前記プレビューアに渡すステップと、

前記プレビューアから前記ホストアプリケーションに進行メッセージを送信して前記進 50

行インターフェースを閲覧するステップと

をさらに備えたことを特徴とする請求項 7 に記載の方法。

【請求項 9】

前記ドキュメントの前記プレビューを生成するために必要な追加のファイルのロケーションを求めて前記ホストアプリケーションに照会するステップをさらに備えたことを特徴とする請求項 7 に記載の方法。

【請求項 10】

前記ホストアプリケーションから前記プレビューハンドラにおいてビーコンインターフェースに対するポインタを受け取るステップと、

前記プレビューハンドラから前記プレビューアへと前記ビーコンインターフェースに対するポインタを渡すステップと、

前記ビーコンインターフェース上で前記ドキュメントの前記プレビューを生成するために必要なサポートファイルをロードする許可を求めて前記ホストアプリケーションに照会するステップと

をさらに備えたことを特徴とする請求項 7 に記載の方法。

【請求項 11】

コンピュータによって実行されると、ドキュメントのプレビューを生成するためのプレビューアを識別しロードするための 1 つまたは複数のメソッドを示すよう動作可能なプレビューハンドラインターフェースを前記コンピュータが 1 つまたは複数のホストアプリケーションに対して示すようにすることになるコンピュータ実行可能命令を有することを特徴とするコンピュータ読取可能な媒体。

【請求項 12】

前記プレビューハンドラインターフェースは、プレビューハンドラを初期化するための `initialize` メソッドを含むことを特徴とする請求項 11 に記載のコンピュータ読取可能な媒体。

【請求項 13】

前記プレビューハンドラインターフェースは、前記プレビューアを識別しロードするための `load` メソッドを含むことを特徴とする請求項 12 に記載のコンピュータ読取可能な媒体。

【請求項 14】

前記プレビューアを識別するステップは、前記プレビューアの識別子について前記ホストアプリケーションによって示されるインターフェースを呼び出すステップを含むことを特徴とする請求項 13 に記載のコンピュータ読取可能な媒体。

【請求項 15】

前記プレビューアを識別するステップは、前記プレビューアのクラス識別子について前記ホストアプリケーションによって示されるインターフェースを呼び出すステップを含むことを特徴とする請求項 14 に記載のコンピュータ読取可能な媒体。

【請求項 16】

前記プレビューアを識別するステップは、前記ドキュメントタイプについてのプレビューアとして登録されたプレビューアを求めてレジストリを検索するステップを含むことを特徴とする請求項 14 に記載のコンピュータ読取可能な媒体。

【請求項 17】

前記プレビューハンドラインターフェースは、前記プレビューを表示するための `show` メソッドを含むことを特徴とする請求項 14 に記載のコンピュータ読取可能な媒体。

【請求項 18】

前記プレビューアは、前記プレビューを表示するための `show` メソッドを含むプレビューアインターフェースを示すよう動作可能であり、前記プレビューハンドラは、前記プレビューハンドラインターフェース上で前記 `show` メソッドに対する呼出しを受け取るステップにตอบสนองして前記プレビューアインターフェース上で前記 `show` メソッドを呼び出すように動作可能であることを特徴とする請求項 14 に記載のコンピュータ読取可能な

10

20

30

40

50

媒体。

【請求項 19】

ドキュメントプレビューを提供するための方法であって、  
ホストアプリケーションにおいてプレビューホストのインターフェースを示すステップと、

プレビューハンドラがドキュメントタイプについてのプレビューアの識別子を取得できるようにするための `get previewer` メソッドと、

前記プレビューハンドラがドキュメントタイプについてのプレビューアのためのクラス識別子を取得できるようにするための `get previewer class identifier` メソッドと、

前記プレビューハンドラが前記ホストアプリケーションによって提供される他のインターフェースに対するポインタを取得できるようにするための `get interface` メソッドと、

前記プレビューハンドラがプレビューすべきドキュメントがロードされていることを前記ホストアプリケーションに通知できるようにするための `done loading` メソッドと

を備えたことを特徴とする方法。

【請求項 20】

プレビューハンドラにおいて、前記プレビューハンドラのインターフェースを示すステップと、

前記プレビューハンドラを初期化するための `initialize` メソッドと、

前記プレビューハンドラにプレビューアを識別するように指示し、およびプレビューすべき前記ドキュメントを前記プレビューアにロードするための `load` メソッドと、

前記プレビューを表示するための `show` メソッドと、前記プレビューが表示されたウィンドウが変化していることを前記プレビューハンドラに指示するための `window changed` メソッドと、

前記プレビューハンドラの前記動作を終了処理するための `uninitialize` メソッドと

をさらに備えたことを特徴とする請求項 19 に記載の方法。

【請求項 21】

プレビューアにおいて、前記プレビューアのインターフェースを示すステップと、

プレビューすべき前記ドキュメントをロードする命令を受け取るための `load` メソッドと、

前記プレビューを表示するための命令を表示するための `show` メソッドと、

前記プレビューが表示される前記ウィンドウが変化していることを示す指示を受け取るための `window changed` メソッドと

をさらに備えたことを特徴とする請求項 20 に記載の方法。

【請求項 22】

前記プレビューアインターフェース上の前記 `load` メソッドは、前記プレビューハンドラインターフェース上の前記 `load` メソッドに対する呼出しを受け取るステップにตอบสนองして前記プレビューハンドラによって呼び出されることを特徴とする請求項 21 に記載の方法。

【請求項 23】

前記プレビューアインターフェース上の前記 `show` メソッドは、前記プレビューハンドラインターフェース上の前記 `show` メソッドに対する呼出しを受け取るステップにตอบสนองして前記プレビューハンドラによって呼び出されることを特徴とする請求項 21 に記載の方法。

【請求項 24】

前記プレビューアインターフェース上の前記 `window changed` メソッドは、前記プレビューハンドラインターフェース上の前記 `window changed` メソ

10

20

30

40

50

ッドに対する呼出しを受け取るステップに回答して前記プレビューハンドラによって呼び出されることを特徴とする請求項 21 に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、一般には、コンピュータシステムに関し、より詳細には、ドキュメントプレビューを提供する方法、システムおよび装置に関する。

【背景技術】

【0002】

現代のグラフィックオペレーティングシステムを有するコンピュータシステムを使用する際には、ドキュメントのプレビューを閲覧することがしばしば望ましい。ドキュメントプレビューとは、ドキュメントに関連するアプリケーションプログラムを用いてドキュメントがオープンされた場合に、ユーザが何を見るはずであるかについての模擬を行うドキュメントの読取り専用のレンダリングのことである。一部のアプリケーションプログラムにより、ユーザは、ドキュメントの静的イメージを閲覧できるようになるが、静的イメージは、ユーザがページを移動し、またはナビゲートすることができる真のプレビューではない。他のプログラムでは、ユーザが、ドキュメントに関連するアプリケーションプログラムを起動して、ドキュメントの内容を閲覧する必要がある。例えば、電子メール（「Eメール（e-mail）」）のクライアントアプリケーションプログラム内においては、電子メールのメッセージに添付された1つまたは複数のドキュメントのプレビューを閲覧することが時として望ましいことがある。しかし、現行の電子メールのアプリケーションプログラムでは、ユーザは、添付されたドキュメントに関連するアプリケーションを起動してドキュメントを閲覧する必要がある。このプロセスは、ユーザにとって時間がかかり、また苛立たしい場合がある。

10

20

【0003】

様々なコンテキストにおいて、ドキュメントプレビューを閲覧することが望ましいこともある。例えば、前述の電子メールのクライアントアプリケーションプログラムなどのアプリケーションプログラムのコンテキストの中からドキュメントプレビューを閲覧することが望ましいこともある。また、グラフィックデスクトップシェル（`graphical desktop shell`）などのオペレーティングシステムプログラムのコンテキストの中からドキュメントプレビューを閲覧することが望ましいこともある。例えば、ユーザが、シェルを利用して、ドキュメントを含むフォルダへとナビゲートするとき、ユーザは、関連するアプリケーションプログラムを起動する必要なしに、ドキュメントのプレビューを速やかに閲覧することが望ましいこともある。同様に、ファイルを開くために利用されるダイアログボックスのコンテキストにおいてドキュメントプレビューを速やかに閲覧することが望ましいこともある。このようにして、ユーザは、ドキュメントに関連するアプリケーションプログラムを開く必要なしに、ドキュメントプレビューを閲覧して、ドキュメントの内容を速やかに調べることができる。しかし、現行のシステムを利用してドキュメントをプレビューするためには、各アプリケーションプログラムおよびオペレーティングシステムは、ドキュメントをプレビューするためにそれら自体のコードを実装する必要がある。

30

40

【発明の開示】

【発明が解決しようとする課題】

【0004】

コンピュータシステムの応答性に悪影響を及ぼすことなく様々なコンテキストの中でドキュメントを速やかにプレビューするための機構を提供することが望ましいこともある。さらに、ドキュメントをプレビューする際に出合うようなエラーも、ホストオペレーティングシステムまたはアプリケーションの性能または安定性に影響を与えるべきではない。本発明の様々な実施形態が構成されてきたのは、これらおよび他の考察に関連している。

50

## 【課題を解決するための手段】

## 【0005】

本発明の実施形態によれば、以上および他の問題は、ドキュメントプレビューを提供するための方法、システム、および装置によって解決される。本発明の実施形態では、オペレーティングシステムプログラムおよびアプリケーションプログラムが利用して、ドキュメントに関連するアプリケーションプログラムを起動することなしにドキュメントの速やかなプレビューを提供することができる共通の機構が提供される。本発明の様々な実施形態が提供する機構により、プレビューは、ページを移動し、ナビゲートすることができるようになる。これらの実施形態では、コンピュータシステムの性能または安定性に厳しい影響を与えることがないプレビュー機構も提供される。

10

## 【0006】

本発明の一態様によれば、ドキュメントプレビューを生成するための方法が提供される。この方法によれば、ドキュメントのプレビューを表示することを望むホストアプリケーションが呼び出すことができるプレビューハンドライントラフェース ( `preview handler interface` ) を示すプレビューハンドラが提供される。例えば、電子メールクライアントアプリケーションは、ホストアプリケーションを含んでおり、プレビューハンドライントラフェースを呼び出して、電子メールメッセージに添付されたドキュメントのプレビューを表示することができる。プレビューハンドライントラフェースを介して、プレビューハンドラは、プレビューア ( `previewer` ) を識別して、ドキュメントのプレビューを表示し、プレビューアをロードし、プレビューアに指示してプレビューを構成し表示するように動作可能である。

20

## 【0007】

プレビューを構成し表示するために、ホストアプリケーションは、要求を用いてプレビューハンドライントラフェースを呼び出して、プレビューすべきドキュメントをロードすることができる。要求は、ドキュメントへのパスと、プレビューを表示すべきウィンドウへのハンドルを含むことができる。要求を受け取ることに応答して、プレビューハンドラは、その適切なプレビューアを識別して、ドキュメントについてのプレビューを生成することができる。プレビューアは、ホストアプリケーションが提供するオプションのインターフェースに照会することにより、登録済みのプレビューアを求めてレジストリを検索することにより、または他の手段によって、識別することができる。

30

## 【0008】

プレビューアがそのドキュメントタイプについて識別された後に、プレビューハンドラは、プレビューアが示すインターフェースを呼び出して、プレビューアに指示してドキュメントをロードすることができる。プレビューハンドラはまた、プレビューアインターフェースを呼び出して、ホストアプリケーションが識別するウィンドウのサブウィンドウ内のドキュメントのプレビューを示すことができる。プレビューアを別のプロセス空間で実行して、ホストアプリケーションの性能に影響を与えないようにすることもできる。同様な理由から、プレビューハンドラとプレビューア間の通信は、バックグラウンドスレッド ( `background thread` ) 上で行うこともできる。

40

## 【0009】

プレビューアが使用するためのホストアプリケーションにより、いくつかのオプションのインターフェースを実装することができる。特に、ホストアプリケーションにより、ビーコンインターフェース ( `beaconing interface` ) を提供することができる。インターフェースを介して、プレビューアは、ドキュメントのプレビューを生成するために必要なサポートファイルをロードする許可を求めてホストアプリケーションに照会することができる。ホストアプリケーションにより、進行インターフェース ( `progress interface` ) を実装することができる。インターフェースを介してプレビューアは、ドキュメントのローディングおよびプレビューの生成の進行などの進行インジケータを提供することができる。またプレビューアが、ドキュメントのプレビューを生成するために必要な追加のファイルの位置を見出すためのインターフェースを取得するこ

50

とができるようにするインターフェースも提供することができる。

【0010】

本発明の実施形態は、コンピュータプロセス、コンピューティングシステムとして実装することもでき、またコンピュータプログラム製品やコンピュータ読取可能な媒体などの製品として実装することもできる。コンピュータプログラム製品は、コンピュータシステムによって読取可能な、コンピュータプロセスを実行するための命令のコンピュータプログラムを符号化したコンピュータ記憶媒体とすることができる。コンピュータプログラム製品はまた、コンピューティングシステムによって読取可能な、コンピュータプロセスを実行するための命令のコンピュータプログラムを符号化した、搬送波上の伝搬信号とすることもできる。

10

【0011】

本発明を特徴づける、これらおよび他の様々な特徴、ならびに利点については、以下の詳細な説明を読み、その関連する図面を検討することから明らかになる。

【発明を実施するための最良の形態】

【0012】

次に図面を参照して、本発明の様々な態様について説明する。図面において、同一の参照符号は、同一の要素を示す。特に、図1およびその対応する説明では、本発明の実施形態を実装することができる適切なコンピューティング環境の簡単な全般的な説明を提供することを意図している。本発明では、パーソナルコンピュータ上のオペレーティングシステム上で実行されるプログラムモジュールに関連して実行されるプログラムモジュールの一般的な文脈で説明しているが、本発明は、他のタイプのコンピュータシステムおよびプログラムモジュールと組み合わせて実装することもできることが、当業者には理解されよう。

20

【0013】

一般的に、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、コンポーネント、データ構造および他のタイプの構造体を含んでいる。さらに、本発明は、ハンドヘルド装置、マルチプロセッサシステム、マイクロプロセッサベースまたはプログラム可能な家庭用電化製品、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コンフィギュレーションを用いて実行することもできることが当業者には理解されよう。本発明はまた、タスクが、通信ネットワークを介してリンクされたリモート処理装置によって実施される分散コンピューティング環境で実行することもできる。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、ローカルとリモートとの両方のメモリ記憶装置において配置することができる。

30

【0014】

次に図1を参照して、本発明の様々な実施形態において利用されるコンピュータ2の例示のコンピュータアーキテクチャについて説明することにする。図1に示すコンピュータアーキテクチャは、中央演算処理装置5(「CPU」)、ランダムアクセスメモリ9(「RAM」)および読取専用メモリ11(「ROM」)を含むシステムメモリ7、ならびにメモリをCPU5に結合するシステムバス12を含む従来のデスクトップコンピュータまたはラップトップコンピュータを示している。起動時などコンピュータ内のエレメント間で情報を転送する助けをする基本ルーチンを含む基本入出力システムは、ROM11に記憶される。コンピュータ2は、以下でさらに詳細に説明しているオペレーティングシステム16、アプリケーションプログラム、および他のプログラムモジュールを記憶するための主記憶装置14をさらに含んでいる。

40

【0015】

主記憶装置14は、バス12に接続された主記憶装置コントローラ(図示せず)を介してCPU5に接続される。主記憶装置14およびその関連するコンピュータ読取可能な媒体は、コンピュータ2のための不揮発性記憶装置を提供する。本明細書に含まれるコンピュータ読取可能な媒体の説明は、ハードディスクやCD-ROMドライブなどの主記憶装

50

置のことを意味しているが、コンピュータ読取可能な媒体は、コンピュータ2からアクセスすることができるどのような使用可能な媒体とすることもできることを当業者には理解されたい。

**【0016】**

実施例として、限定するものではないが、コンピュータ読取可能な媒体は、コンピュータ記憶媒体および通信媒体を含むことができる。コンピュータ記憶媒体は、コンピュータ読取可能な命令、データ構造、プログラムモジュール、他のデータなどの情報の記憶のためのあらゆる方法または技術で実装される揮発性媒体および不揮発性媒体、着脱可能媒体および着脱不能媒体を含んでいる。コンピュータ記憶媒体は、それだけには限定されないが、RAM、ROM、EPROM、EEPROM、フラッシュメモリまたは他のソリッドステートメモリ技術、CD-ROM、デジタル多用途ディスク(「DVD」)または他の光記憶装置、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装置、あるいは所望の情報を記憶するために使用することができ、コンピュータ2によってアクセスすることができる他のあらゆる媒体を含んでいる。

10

**【0017】**

本発明の様々な実施形態によれば、コンピュータ2は、インターネットなどのネットワーク18を介した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。コンピュータ2は、バス12に接続されるネットワークインターフェースユニット20を介してネットワーク18に接続することができる。ネットワークインターフェースユニット20を利用して、他のタイプのネットワークおよびリモートコンピュータシステムに接続することもできることを理解されたい。コンピュータ2は、(図1には示していない)キーボード、マウス、または電子スタイラスを含めて、いくつかの他の装置からの入力を受け取り、処理するための入出力コントローラ22も含むことができる。同様に、入出力コントローラ22は、ディスプレイスクリーン、プリンタ、または他のタイプの出力装置に対して出力を提供することもできる。

20

**【0018】**

以上で簡単に述べたように、ワシントン州レッドモンド市在の本願発明の特許出願人からのWINDOWS(登録商標)XPオペレーティングシステムなど、ネットワーク化されたパーソナルコンピュータの動作を制御するための適切なオペレーティングシステム16を含めて、いくつかのプログラムモジュールおよびデータファイルをコンピュータ2の主記憶装置14およびRAM9に記憶することができる。主記憶装置14およびRAM9は、1つまたは複数のプログラムモジュールを記憶することもできる。特に、主記憶装置14およびRAM9は、ホストアプリケーション26を記憶することもできる。ホストアプリケーション26は、ユーザに対してドキュメント32のプレビューを提供することを望む、あらゆるタイプのプログラムを含んでいる。本発明の実施形態によれば、ホストアプリケーション26は、マイクロソフトコーポレーションからのOUTLOOKパーソナル情報マネージャアプリケーションプログラムを含んでいる。本発明の様々な実施形態を介して、電子メールメッセージに添付されたドキュメントについてプレビューを生成し、表示することができる。しかし、ホストアプリケーション26は、ワードプロセッサアプリケーションプログラム、スプレッドシートアプリケーションプログラム、プレゼンテーションアプリケーションプログラム、オペレーティングシステム16、または他のあらゆるタイプのプログラムを含むことができることを理解されたい。これらのプログラムでは、例えば、ファイルオープンダイアログボックスにおいて、オペレーティングシステムシェル内に、または他のあらゆるロケーションにおけるドキュメントプレビューを提供するために、本明細書で説明している機構を利用することができる。

30

40

**【0019】**

主記憶装置14およびRAM9は、プレビューハンドラプログラム28を記憶するようにも動作可能である。プレビューハンドラ28は、ドキュメントのプレビューを提供するための機構である。以下でより詳細に説明しているように、プレビューハンドラ28は、ホストアプリケーション26などのアプリケーションがドキュメントプレビューを生成す

50



るために呼び出すことができるインターフェースを実装する。ドキュメントプレビューの用意については、プレビューハンドラ 28 が提供するインターフェースによってホストアプリケーション 26 から抽出される。

#### 【0020】

本発明の実施形態によれば、プレビューハンドラ 28 は、1つまたは複数のプレビュープログラム 30 を利用して、ドキュメント 32 の実際のプレビューを生成することができる。プレビューアは、特定のタイプのドキュメントについてのプレビューアとして、プレビューハンドラ 28、ホストアプリケーション 26、またはオペレーティングシステム 16 に登録することができる。登録済みのプレビューアに関する情報は、登録済みのプレビューアの識別子およびロケーションに関して、プレビューハンドラ 28、ホストアプリケーション 26、またはオペレーティングシステム 16 が保持することができる。オペレーティングシステム 16 は、情報をレジストリ 24 に記憶することができる。ホストアプリケーション 26 は、様々なタイプのドキュメントについてのプレビューアとしてそれ自体を登録することもできること理解されたい。ホストアプリケーション 26、プレビューハンドラ 28、およびプレビューア 30 の動作に関するさらなる詳細については、図 2 ~ 5 に関して以下で説明することにする。

10

#### 【0021】

次に図 2 を参照して、本発明の実施形態が利用する様々なソフトウェアコンポーネントに関して、追加の詳細を提供することにする。図 2 に示すように、ホストアプリケーション 26 は、プレビューハンドラ 28 が実装するプレビューハンドラインターフェース 42 の使用を介してプレビューハンドラ 28 と情報をやりとりする。次にプレビューハンドラ 28 は、ホストアプリケーションが実装するプレビューホストインターフェース 40 を経由してホストアプリケーションと情報をやりとりすることができる。プレビューハンドラ 28 は、プレビューアインターフェース 50 の使用を介してこれらのプレビューア 30 A ~ 30 D と情報をやりとりすることもできる。ホストアプリケーション 26 は、プレビューア 30 A ~ 30 C に関して本明細書で説明している方法と同様な方法で、プレビューア 30 D を実装し、プレビューハンドラ 28 と情報をやりとりすることができることを理解されたい。プレビューハンドラ 28 は、レジストリ 24 から値を検索して、適切なプレビューアおよびそのロケーションを識別することができることも理解されたい。

20

#### 【0022】

以上で簡単に説明したように、プレビューア 30 A ~ 30 C のそれぞれは、1つまたは複数のドキュメントタイプについてのプレビューを生成するように動作可能である。ドキュメント 32 A ~ 32 C は、一般的に主記憶装置 14 上のプレビューア 30 A ~ 30 C からアクセス可能なロケーションに記憶される。ロードする要求がプレビューハンドラ 28 によって発行されるときに、ロケーションは、プレビューア 30 A ~ 30 C に渡される。プロセスについては、以下でさらに詳細に説明することにする。

30

#### 【0023】

プレビューア 30 A ~ 30 C は、1つまたは複数のオプションのインターフェースの使用を介して、ホストアプリケーション 26 と直接に情報をやりとりすることができる。特に、進行インターフェース 44 を利用して、ホストアプリケーション 26 に対して進行インジケータを提供することができる。次いで進行インジケータをユーザに対して表示して、例えば、プレビュー表示の進行のグラフィック表示を提供することができる。ピーコンインターフェース 46 をプレビューア 30 A ~ 30 C が利用して、ホストアプリケーション 26 からドキュメントに関連するサポートファイルをロードする許可を取得することができる。例えば、プレビューすべきドキュメントが、ハイパーテキストマークアップ言語 (「HTML」) ドキュメントである場合、グラフィックスおよび他の参照ファイルをロードして、プレビューを提供する必要があることもある。参照ファイルをロードするのに先立って、プレビューアは、ホストアプリケーション 26 からの許可を要求して、ピーコンインターフェース 46 を介してそれを行うことができる。I U n k n o w n インターフェース 48 をプレビューア 30 A ~ 30 C が利用して、ドキュメントのプレビューを生成

40

50

するために必要なサポートファイルの位置を見出すなど、様々なファンクションを実施するためのインターフェースを提供することができる。当業者には知られているように、IUnknownインターフェース48は、所与のオブジェクト上で他のインターフェースに対するポインタをクライアントに取得させる。

#### 【0024】

プレビューを提供するために、ホストアプリケーション26は、ウィンドウ52に対するハンドルをプレビューハンドラ28に渡す。次いでプレビューハンドラは、ホストアプリケーションが作成するウィンドウの内側に子ウィンドウ（またはサブウィンドウ）を作成する。次いで、プレビューアが識別されインスタンス化されるときに、子ウィンドウに対するハンドルは、適切なプレビューア30A~30Cに渡される。ウィンドウ内のプレビュー56をレンダリングするために、プレビューアは、ウィンドウ52内にサブウィンドウ54を作成する。プレビューを指示するユーザ入力は、プレビューアが受け取ることになるので、プレビューアは、あるタイプの入力を処理し、他のタイプの入力をプレビューハンドラ28へと渡し、その後、ホストアプリケーション26に渡すように動作可能とすることができる。このようにして、実際にはホストアプリケーション26のために意図される、プレビューアが受け取る入力を、ホストアプリケーション26が処理することもできる。

#### 【0025】

依然として図2を参照して、プレビューハンドラインターフェース42、プレビューホストインターフェース40、およびプレビューアインターフェース50に関する追加の詳細を提供することにする。プレビューハンドラインターフェース42は、ホストアプリケーション26がプレビューハンドラ28のインスタンスを作成した後に呼び出される「initialize（初期化）」メソッドを提供する。initializeメソッドにより、プレビューハンドラ28は、ウィンドウ52上でそれ自体を初期化できるようになる。initializeメソッドに対して渡されるパラメータは、ホストアプリケーション26のインスタンス、プレビューホストインターフェース40に対するポインタ、およびプレビューハンドラ28の動作を定義するために利用することができる1つまたは複数のフラグである。インターフェース42上で、プレビューをレンダリングすべきウィンドウ52に対するハンドルを獲得するsetwindow（ウィンドウ設定）メソッドを提供することもできる。

#### 【0026】

プレビューハンドラインターフェース42は、「load（ロード）」メソッドも提供する。ユーザが、プレビューしたいと思うファイルまたは1組のファイルを選択するときなど、ドキュメントプレビューを生成する要求を受け取る際には、loadメソッドが、ホストアプリケーション26によって呼び出される。メソッドが呼び出されるときに、プレビューハンドラ28は、プレビューする1つまたは複数のファイル名を保存し、ファイルをプレビューするバックグラウンドスレッドを呼び出す。次いで、バックグラウンドスレッドは、プレビューするファイルが実際に存在することを確認し、ファイルについての拡張子を取得し、拡張子についてプレビューア30A~30Cの位置を見出そうと試みる。プレビューアの位置が見出されると、プレビューアを利用してファイルをプレビューする。loadメソッドは、プレビューするファイルに至るフルパスをパラメータとして獲得する。loadメソッドの動作に関するさらなる詳細については、以下で提供することにする。

#### 【0027】

プレビューハンドラインターフェース42は、「show（表示）」メソッドも含んでいる。loadメソッドに対する呼出しにより、ファイルがプレビューアに正常にロードされた後に、showメソッドは、ホストアプリケーション26によって呼び出される。ホストアプリケーション26がプレビューへとファイルをレンダリングしたいと思う場合に、ホストアプリケーションは、showメソッドを呼び出す。プレビューハンドラインターフェース42は、「showerrormessage（エラーメッセージ表示

10

20

30

40

50

)」メソッドも含んでいる。loadメソッドまたはshowメソッドが、エラーコードを返しており、ホストアプリケーション26は、プレビューハンドラ28が、コードについてのエラーメッセージを表示することを望む場合に、show error messageメソッドは、呼び出される。show error messageメソッドは、ロードし、またはパラメータとして示すことができなかつたエラーコードおよびファイルのファイル名を獲得する。

**【0028】**

プレビューハンドラインターフェース42は、「window changed (ウィンドウ変化)」メソッドも含んでいる。set windowメソッドに渡されたウィンドウ52が何らかの方法で変化するときメソッドは呼び出される。例えば、ウィンドウ52のサイズが変化する場合に、ホストアプリケーション26は、メソッドを呼び出して、プレビューハンドラ28に変化について通知することができる。次いで、プレビューハンドラ28は、適切なプレビューアを呼び出して、サブウィンドウ52およびプレビュー56のサイズをアップデートすることができる。プレビューハンドラインターフェース42はまた、プレビューすべき追加のファイルが残っていないときに呼び出すことができる「uninitialize (初期化しない)」メソッドも含んでいる。uninitializeメソッドは、プレビューハンドラ28を初期化しない(uninitialize)。

10

**【0029】**

プレビューホストインターフェース40は、プレビューハンドラ28がホストアプリケーション26にとって使用可能となり得るリソースを使用できるようにするためにホストアプリケーション26が実装するオプションのコールバックインターフェースである。例えば、ホストアプリケーション26が、特定のドキュメントタイプについてのプレビューアをすでにロードしている場合、ホストアプリケーション26が、プレビューハンドラ28にプレビューアそれ自体をロードするように試みさせる代わりに、インターフェースをロード済みのプレビューアに渡すことが有用となり得る。

20

**【0030】**

プレビューホストインターフェース40は、「get previewer (プレビューア獲得)」メソッドを提供する。プレビューハンドラ28が、あるファイルについてのプレビューアの位置を見出そうと試みているときに、メソッドは、プレビューハンドラ28によって呼び出すことができる。get previewerメソッドを介して、プレビューハンドラ28は、ファイルタイプについてのプレビューアの識別子についてホストアプリケーション26に照会することができる。get previewerメソッドは、プレビューすべきファイルの拡張子を入力として獲得し、ファイルタイプについてのプレビューアの識別子を示す出力パラメータ、またはNULLを提供する。メソッドは、例えばホストアプリケーション26がすでにプレビューアをロードし終わっているとき、またはホストアプリケーション26がプレビューア30Dそれ自体を提供するとき有用となり得る。

30

**【0031】**

get previewerメソッドがプレビューアを返さない場合には、プレビューハンドラ28は、プレビューホストインターフェース40が提供する「get previewer class ID (プレビューアクラスID獲得)」メソッドを利用することができる。get previewer class IDメソッドを介して、ホストアプリケーション26は、ホストアプリケーションが所与の拡張子の特定のファイルについて使用したいと望むあらゆるプレビューアについてのクラスIDをプレビューハンドラ28に対して渡すことができる。例えば、ホストアプリケーション26は、特定の拡張マークアップ言語(「XML」)ファイルでは特定のプレビューアを利用することを指定したいと思うこともある。get previewer class IDメソッドは、プレビューすべきファイルの拡張子を入力として獲得し、ファイルタイプについてのプレビューアのクラスIDを示す出力パラメータ、またはNULLを提供する。

40

50

## 【0032】

プレビューホストインターフェース40はまた、「`get interface` (インターフェース獲得)」メソッドも提供する。プレビューハンドラ28が、ファイルをロードするように指示されるとき、プレビューハンドラは、ロードおよびショーに使用するために、プレビューアに渡すために、ビーコンインターフェース46、進行インターフェース44、および`IUnknown`インターフェース48に対するポインタを獲得しようと試みる。`get interface`メソッドは、目的のために利用される。`get interface`メソッドは、要求されているインターフェースの識別子をパラメータとして獲得し、それが返される場合には、要求されたインターフェースに対するポインタを出力パラメータとして返す。

10

## 【0033】

プレビューホストインターフェース40はまた、「`done loading` (ローディング完了)」メソッドも提供する。プレビューアにファイルのローディングが行われるときに、メソッドを呼び出して、ホストアプリケーション26に次のステップを取り込むことができることを知らせることができる。例えば、`done loading`メソッドに対する呼出しを受け取ることに応答して、ホストアプリケーション26は、プレビューハンドラインターフェース42上で`show`メソッドを呼び出すことができる。代わりに、ドキュメントをロードすることができなかつた場合、`done loading`メソッドは、エラーメッセージと共に呼び出すことができる。この場合には、ホストアプリケーション26は、プレビューハンドラインターフェース42上で「`show error message`」メソッドを呼び出すことができる。

20

## 【0034】

プレビューアインターフェース50は、プレビューア30A~30Dが実装するベースインターフェースである。インターフェースを実装することにより、プレビューアを呼び出して、ファイルをロードし、ファイルをサブウィンドウにレンダリングし、必要に応じてドキュメント内ナビゲーションを提供し、プレビューが完了したときにファイルをロードしない(`unload`)ことができる。プレビューアは、バックグラウンドスレッド上で呼び出すことができ、ユーザがプレビュー上で待っていることがあるので、できるだけ速くロードレンダリングするようにも試みるべきである。

## 【0035】

プレビューアインターフェース50は、特定のファイル上でプレビューアを初期化するために呼び出される「`load`」メソッドを見せている。`load`メソッドは、プレビューすべきドキュメントへのパス、ならびにビーコンインターフェース46、進行インターフェース44、および`IUnknown`インターフェース48に対するオプションのポインタをパラメータとして獲得する。これらのインターフェースは、前述の方法で利用することができる。

30

## 【0036】

`load`メソッドに対する呼出しに応答してドキュメントをロードする際に、プレビューアは、進行ダイアログなどのどのようなユーザインターフェースも呼び出すべきではない。ドキュメントは、書込みについてロックされるべきではなく、プレビューアは、決してドキュメントを変更またはアップデートすべきではない。プレビューアはまた、ドキュメントがパスワード保護されている場合には、静かに機能しなくなるべきである。プレビューアはまた、ドキュメントに含まれるどのようなマクロも実行すべきではない。プレビューアはまた、`load`メソッドを一度呼び出すことができ、異なる可能性のあるウィンドウを用いて(以下で説明する)「`show`」メソッドを、何度も呼び出すことができるように設計すべきである。`load`メソッドは、レンダリングすべきファイルを準備するに際して、含まれるどのような時間のかかる作業でも実施すべきであり、したがって、「`show`」メソッドは、できる限り高速にすることが可能である。ユーザがプレビューをキャンセルしようとする場合には、`load`メソッドは、ファイルのロードを停止し、できる限り速やかに戻るべきである。

40

50

## 【0037】

以上で簡潔に説明したように、プレビューインターフェース50もまた、「show」メソッドを見せることができる。メソッドを呼び出して、プレビューに、loadメソッドに渡されたドキュメントを指定されたウィンドウ内へとレンダリングさせる。これは、またファイルをウィンドウに対してレンダリングする必要なしに、ビューアがファイルをロードできるようにするloadメソッドとは異なるメソッドである。showメソッドは、プレビューをレンダリングすべきウィンドウに対するハンドルをパラメータとして獲得する。

## 【0038】

プレビューインターフェース50は、「window changed」メソッドも見せている。メソッドを呼び出して、プレビューが現在使用しているウィンドウが、何らかの形で変化していることをプレビューに知らせる。メソッドは、一般的にプレビューハンドライナーフェース42上におけるwindow changedメソッドに対する呼出しに回答して呼び出される。プレビューインターフェース50はまた、「unload (ロードしない)」メソッドも示す。メソッドを呼び出して、プレビューがloadメソッドを介してロードされたファイルをロードしないべきであることをプレビューに知らせる。プレビューハンドライナーフェース42、プレビューホストインターフェース、およびプレビューインターフェース50が提供するこれらの様々なメソッドの用途および動作に関するさらなる詳細については、図3~5を参照して以下でさらに詳細に説明することにする。

## 【0039】

本発明の実施形態によれば、プレビューハンドラ28はまた、IService Providerインターフェース51またはIOle CommandTargetインターフェース53もサポートすることができる。インターフェース51は、コンポーネントが無関係の機能を見せることができるようにする既存のインターフェースである。インターフェース51を介して、ホストアプリケーション26は、現行のプレビューに、あるタイプの機能をプレビューがサポートするかどうかについて照会することができる。インターフェース53は、呼出し元(ホスト26)が、クライアント(プレビュー)に対してコマンドを発行できるようにする別の既存のインターフェースである。プレビューハンドラ28は、現行のプレビューに対するパススルー(pass-through)としてインターフェースを実装する。

## 【0040】

次に図3を参照して、ドキュメントプレビューを提供するためのプロセスを示す例示のルーチン300について説明することにする。本明細書で提示されるこれらのルーチンの説明を読むに際して、本発明の様々な実施形態の論理的な動作は、(1)コンピューティングシステム上で実行されるコンピュータ実装された動作シーケンスもしくはプログラムモジュール、および/または(2)コンピューティングシステム内の相互接続されたマシン論理回路もしくは回路モジュールとして実装されることを理解されたい。実装については、本発明を実装するコンピューティングシステムの性能要件に依存した選択の問題である。したがって、図3~5に示す論理的な動作、および本明細書で説明している本発明の実施形態を作成することについては、動作、構造装置、動作またはモジュールと様々に呼ばれる。これらの動作、構造装置、動作およびモジュールは、本明細書に添付される特許請求の範囲内に列挙されるように、本発明の趣旨および範囲を逸脱することなく、ソフトウェア、ファームウェア、専用デジタルロジック、およびこれらのあらゆる組合せの形で実装することができることが、当業者には理解されよう。

## 【0041】

ルーチン300は、動作302から開始され、ここでは、1つまたは複数のドキュメントのプレビューを生成する要求を受け取っているかどうかについての決定が行われる。かかる要求は、例えば、ユーザが添付された1つまたは複数のドキュメントを有する電子メールメッセージを選択することによって行うことができる。代わりに、かかる要求につい

10

20

30

40

50

ては、オペレーティングシステムまたは他のタイプのプログラムによって、プログラムの  
に行うこともできる。プレビュー要求を受け取っていない場合、ルーチン300は、動作  
302に戻り、ここでかかる別の決定が行われる。要求を受け取っている場合には、ルー  
チン300は、動作302から動作304へと継続される。

#### 【0042】

動作304において、ホストアプリケーション26は、プレビューハンドラ28のイン  
スタンスを作成する。次いで、ルーチン300は、動作306へと継続され、ここで、ホ  
ストアプリケーション26は、プレビューハンドラインターフェース42上で、`initialize`  
メソッドを呼び出す。`initialize`メソッドに対する呼出しを受け  
取ることに応答して、プレビューハンドラ28は、前述の方法でそれ自体を初期化する。  
`set window`メソッドを、プレビューを生成すべきウィンドウに対するハンドル  
を用いて呼び出すこともできる。

10

#### 【0043】

動作306から、ルーチン300は、動作308へと継続され、ここでホストアプリケ  
ーション26は、プレビューハンドラインターフェース42上で`load`メソッドを呼び  
出す。`load`メソッドに対する呼出しを受け取ることに応答して、プレビューハンドラ  
28は、プレビューすべき特定のタイプのドキュメントについてのプレビューアを識別し  
、プレビューアをロードし、プレビューアに指示してドキュメントをロードするように試  
みる。このプロセスは、動作310において行われる。プレビューアの位置を見出し、ド  
キュメントをロードするための例示のルーチン400について、図4を参照して以下で説  
明することにする。

20

#### 【0044】

動作310から、ルーチン300は、動作312へと継続され、ここでは、プレビュー  
アが識別され、指定済みのドキュメントをロードすることができたかどうかについての決  
定が行われる。プレビューアが、ドキュメントをロードすることができなかつた場合、ル  
ーチン300は、動作316へと分岐し、ここでは、エラーメッセージを用いて、プレビ  
ューホストインターフェース40上で`done loading`メソッドに対する呼出し  
によって、エラーメッセージがホストアプリケーション26に対して戻される。かかるエ  
ラーに応答して、ホストアプリケーション26は、プレビューハンドラインターフェース  
42上で`show error message`メソッドを呼び出すことができる。動作  
316から、ルーチン300は、動作318へと継続され、ここでルーチン300は、終  
了する。

30

#### 【0045】

動作312において、プレビューアがドキュメントをロードすることができたと決定さ  
れる場合には、ルーチン300は、動作314へと継続され、ここで、成功メッセージを  
用いてプレビューホストインターフェース40上で`done loading`メソッドを  
呼び出すことができる。かかるメッセージを受け取ることに応答して、ホストアプリケ  
ーション26は、プレビューハンドラインターフェース42上で`show`メソッドを呼び出  
すことができる。それに応じて、プレビューハンドラ28は、プレビューアインターフェ  
ース50上で`show`メソッドを呼び出して、プレビューアにプレビューをサブウィンド  
ウ54で表示させることができる。プレビューがもはや必要なくなると、様々な`unin  
itialize`メソッドを呼び出してプレビューアおよびプレビューハンドラ28を初  
期化しないことができる。これらのコンポーネントを初期化しないための例示のルーチ  
ン500について、図5に関連して以下で説明する。動作314から、ルーチン300は、  
動作318へと継続され、ここでルーチン300は終了する。

40

#### 【0046】

次に図4を参照して、ドキュメントについての適切なプレビューアを識別し、ロードす  
るための例示のルーチン400について説明することにする。ルーチン400は、動作4  
02から開始され、ここでは、バックグラウンドスレッドが、プレビューアと情報をやり  
とりするために作成される。バックグラウンドスレッドを利用し、その結果、プレビュー

50

アが応答しなくなるイベントにおいて新しいバックグラウンドスレッドを作成し利用することができるようになる。これにより、ホストアプリケーション 26 は応答可能な状態にとどまることもできるが、一方、プレビューハンドラ 28 は、プレビューアがファイルをロードしレンダリングするのを待つ。これにより、プレビューハンドラ 28 は、ユーザがプレビューするよう要求しているファイルをプレビューしたいとは思わないとユーザが決定する場合について取り扱うことが可能になる。例えば、ユーザがあるスプレッドシートファイルを選択し、プレビューアがファイルをロードしている間に、ユーザがあるイメージを本当にプレビューしたいと思うと決定する場合、ユーザは、ホストアプリケーション 26 でイメージを選択することができる。プレビューハンドラ 28 は、既存のバックグラウンドスレッドが、イメージについてのプレビューアをロードできる状態にあるかどうかを速やかに決定することができる。そうでない場合には、プレビューハンドラ 28 は、イメージプレビューアについてのバックグラウンドスレッドをロードし、スプレッドシートファイルがローディングを完了するのを待たずに、イメージのプレビューをロードすることができる。

#### 【0047】

動作 402 から、ルーチン 400 は、動作 404 へと継続され、ここでは、必要とされるプレビューアがすでにロードされているかどうかについての決定が行われる。性能を改善するために、プレビューハンドラ 28 は、プレビューアが必要とされた後のある期間にわたって、メモリにおいてプレビューアから離れることができる。このようにして、プレビューアは、効率的にキャッシュすることができ、プレビューアについての後続要求においてロード時間を節約することができる。動作 406 において、プレビューアがすでにロードされている場合、ルーチン 400 は、動作 408 へと継続され、ここでは、現行のロード済みのプレビューアが、プレビューすべきドキュメントについてのプレビューアとして指定される。必要とされるプレビューアがロードされていない場合には、ルーチン 400 は、動作 406 から動作 422 へと分岐する。

#### 【0048】

動作 422 において、プレビューアが、それ自体を特定のドキュメントファイルタイプのプレビューアとして登録されているかどうかを決定する検索がレジストリ 24 について行われる。プレビューアがレジストリにおいて識別される場合、ルーチン 400 は、動作 426 へと分岐し、ここで、レジストリで識別されたプレビューアは、ドキュメントについてのプレビューアとして指定される。プレビューアがレジストリで見出されない場合には、ルーチン 400 は、動作 410 へと継続される。

#### 【0049】

動作 410 において、プレビューハンドラ 28 は、プレビューホストインターフェース 40 上で `get previewer` メソッドを呼び出す。前述のように、`get previewer` メソッドにより、プレビューハンドラ 28 は、ホストアプリケーション 26 からプレビューアの識別子を要求することができるようになる。動作 412 において、ホストが、`get previewer` メソッドに対する呼出しに回答してプレビューアを識別していることが決定される場合、ルーチン 400 は、動作 414 へと分岐し、ここで識別済みのプレビューアが、ドキュメントについてのプレビューアとして指定される。ホストアプリケーション 26 がプレビューアを指定しない場合には、ルーチン 400 は、動作 416 へと継続される。

#### 【0050】

動作 416 において、プレビューホストインターフェース 40 上で `get previewer class ID` メソッドに対して呼出しが行われる。ホストアプリケーション 26 が、呼出しに回答してプレビューアのクラス ID を戻す場合、ルーチン 400 は、動作 418 から動作 420 に分岐し、ここでクラス ID によって識別されるプレビューアが、現行のドキュメントについてのプレビューアとして指定される。そうでない場合には、ルーチン 400 は、動作 418 から動作 428 へと継続され、ここでエラーメッセージが戻される。

10

20

30

40

50

## 【 0 0 5 1 】

動作 4 0 8、4 1 4、4 2 0、および 4 2 6 から、ルーチン 4 0 0 は、動作 4 3 0 へと継続される。動作 4 3 0 において、指定されたプレビューアは、バックグラウンドスレッド上でプロセス外オブジェクトとしてロードされる。次いでルーチン 4 0 0 は、動作 4 3 2 へと継続され、ここで、ドキュメントをロードし、プレビューを構築する試みがプレビューアによって行われる。次いで、ルーチン 4 0 0 は、動作 4 3 4 へと継続され、ここで、ドキュメントが正常にロードされたかどうかについての決定が行われる。ドキュメントが正常にロードされなかった場合、ルーチン 4 0 0 は、動作 4 3 4 から動作 4 3 8 へと分岐し、ここでエラーメッセージが戻される。ドキュメントが正常にロードされる場合には、ルーチン 4 0 0 は、動作 4 3 6 へと継続され、ここで成功メッセージが戻される。動作 4 3 6 および 4 3 8 から、ルーチン 4 0 0 は、動作 4 4 0 へと継続され、ここでルーチン 4 0 0 は戻る。

10

## 【 0 0 5 2 】

次に図 5 を参照して、プレビューアおよびプレビューハンドラ 2 8 をロードしないための例示のルーチン 5 0 0 について説明することにする。ルーチン 5 0 0 は、動作 5 0 2 から開始され、ここでプレビューハンドラ 2 8 は、プレビューア上で `unload` メソッドを呼び出す。かかる呼出しを受け取ることに応答して、プレビューアは、ドキュメントをロードせず、それ自体の初期化を行わない。次いでルーチン 5 0 0 は、動作 5 0 4 へと継続され、ここでは、`uninitialize` メソッドがプレビューハンドラインターフェース 4 2 上で呼び出される。この呼出しは、プレビューハンドラにそれ自体の初期化を行わない。動作 5 0 4 から、ルーチン 5 0 0 は、動作 5 0 6 へと継続され、ここでルーチン 5 0 0 は終了する。

20

## 【 0 0 5 3 】

上述に基づいて、本発明の様々な実施形態は、ドキュメントプレビューを提供するための方法、システム、装置、およびコンピュータ読取可能な媒体を含むことを理解されたい。以上の明細書、実施例およびデータは、本発明の構成の製造および使用についての完全な説明を提供する。本発明の多数の実施形態は、本発明の趣旨および範囲を逸脱することなく行うことができるので、本発明は、添付の特許請求の範囲に属する。

## 【 図面の簡単な説明 】

## 【 0 0 5 4 】

30

【 図 1 】 本発明の様々な実施形態に関して、利用されおよび提供されるコンピュータシステムを例示するコンピュータシステムのアーキテクチャの図である。

【 図 2 】 本発明の実施形態によって提供される様々なソフトウェアコンポーネントおよびインターフェースの態様を例示するソフトウェアのアーキテクチャの図である。

【 図 3 】 本発明の実施形態に従って提供される、ドキュメントプレビューを提供するための様々なソフトウェアコンポーネントの動作を示すフロー図である。

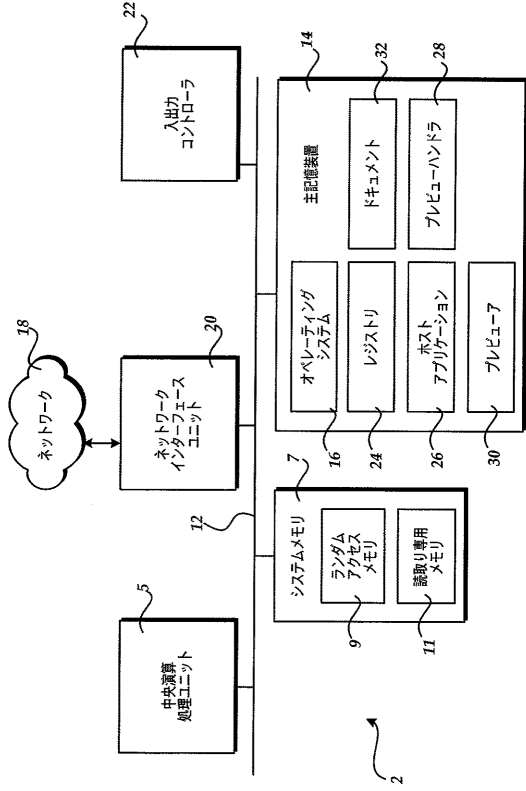
【 図 4 】 本発明の実施形態に従って提供される、ドキュメントプレビューを提供するための様々なソフトウェアコンポーネントの動作を示すフロー図である。

【 図 5 】 本発明の実施形態に従って提供される、ドキュメントプレビューを提供するための様々なソフトウェアコンポーネントの動作を示すフロー図である。

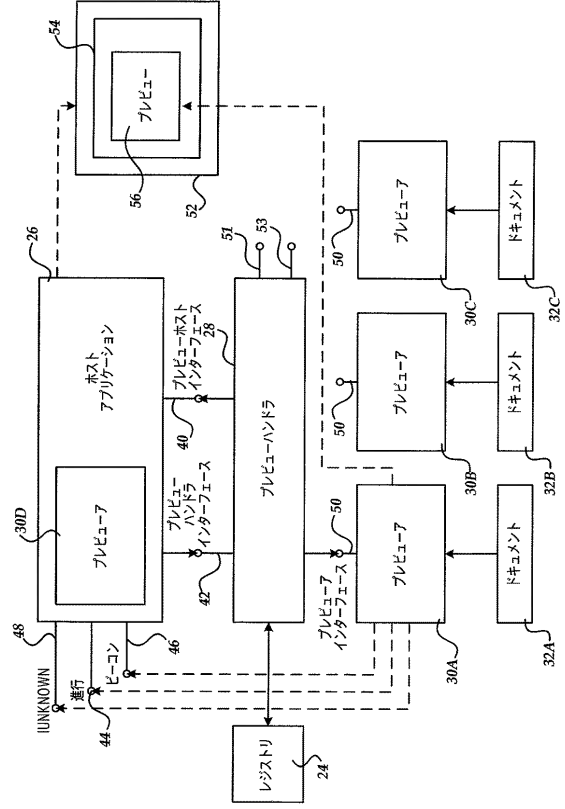
40



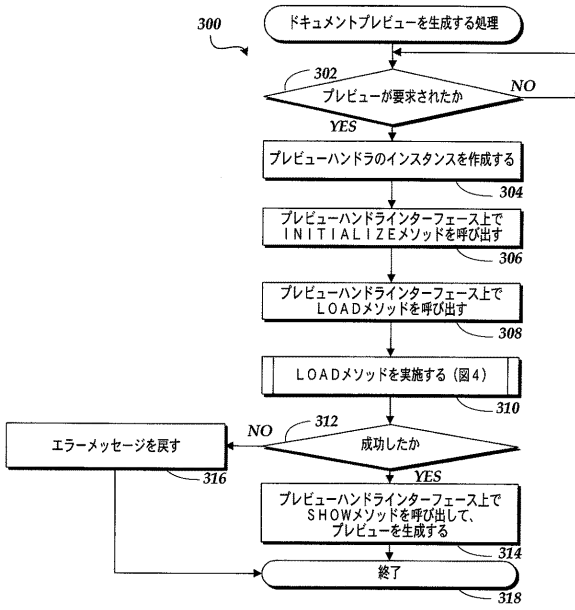
【図1】



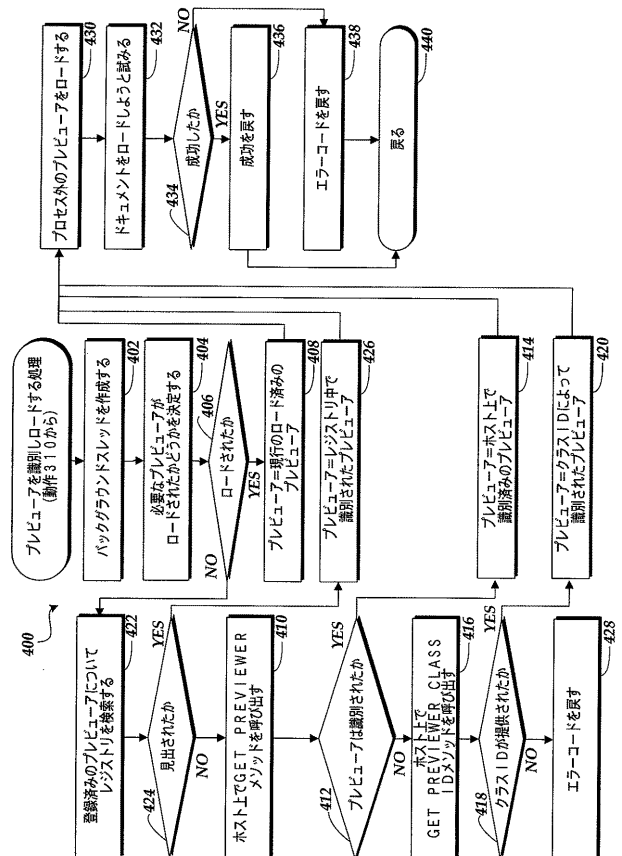
【図2】



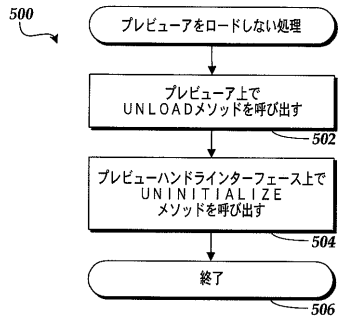
【図3】



【図4】



【 図 5 】



---

フロントページの続き

(72)発明者 ラエ - クリスティ ショー

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) 5B009 NA14 RB21 RB34 TA06

5E501 AC15 CA02 FA14 FB04

【外国語明細書】

2006107439000001.pdf