

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5042502号  
(P5042502)

(45) 発行日 平成24年10月3日(2012. 10. 3)

(24) 登録日 平成24年7月20日(2012. 7. 20)

(51) Int.Cl.

F I

G 0 6 F 12/00 (2006.01)

G 0 6 F 12/00 5 0 5

請求項の数 20 (全 27 頁)

(21) 出願番号	特願2006-14029 (P2006-14029)	(73) 特許権者	500046438
(22) 出願日	平成18年1月23日 (2006. 1. 23)		マイクロソフト コーポレーション
(65) 公開番号	特開2006-202297 (P2006-202297A)		アメリカ合衆国 ワシントン州 9805
(43) 公開日	平成18年8月3日 (2006. 8. 3)		2-6399 レッドモンド ワン マイ
審査請求日	平成21年1月23日 (2009. 1. 23)		クロソフト ウェイ
(31) 優先権主張番号	11/041, 400	(74) 代理人	100077481
(32) 優先日	平成17年1月21日 (2005. 1. 21)		弁理士 谷 義一
(33) 優先権主張国	米国 (US)	(74) 代理人	100088915
			弁理士 阿部 和夫
		(72) 発明者	ハイヨン ワン
			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

最終頁に続く

(54) 【発明の名称】 シリアルバイナリ形式でドキュメントを格納するシステムおよび方法

(57) 【特許請求の範囲】

【請求項 1】

インクドキュメント構造を生成するアプリケーション以外のアプリケーション上でインクドキュメントを表示するコンピュータにより実行される方法であって、前記インクドキュメント構造はシリアル化されたデータ形式に従って格納されており、前記方法は、

前記インクドキュメントのインクに関連する少なくとも1つのルートノードを含むインクドキュメント構造を含むインクドキュメントを生成するステップと、

前記インクドキュメント構造をシリアルバイナリ形式で格納するステップであって、

前記インクドキュメント構造に関連付けられたサイズデータを前記シリアルバイナリ形式の第1のデータフィールドに格納するステップと、

前記インクドキュメント構造で使用可能なデータを表す少なくとも1つのフラグを含むインクドキュメントディスクリプタデータを、前記シリアルバイナリ形式の第2のデータフィールド内に格納するステップと、

前記少なくとも1つのルートノードのルートノードデータであって、前記第2のデータフィールド内のインクドキュメントディスクリプタデータの少なくとも1つのフラグにより指示されるルートノードデータを、前記シリアルバイナリ形式の第3のデータフィールドに格納するステップと

を含む、シリアルバイナリ形式を格納するステップと、

前記インクドキュメント構造を生成するアプリケーション以外のアプリケーションから前記シリアルバイナリ形式を有する前記格納されたインクドキュメント構造にアクセスす

10

20

るステップと、

前記インドキュメント構造を有するインドキュメントを、前記インドキュメント構造を生成するアプリケーション以外のアプリケーション上で表示するステップとを備え、前記インドキュメントは、前記シリアルバイナリ形式の第1のデータフィールド内のインドキュメント構造に関連付けられたサイズデータ、前記シリアルバイナリ形式の第2のデータフィールド内のインドキュメントディスクリプタデータ、および前記シリアルバイナリ形式の第3のデータフィールド内の少なくとも1つのルートノードのルートノードデータに基づいて、前記シリアルバイナリ形式から生成されることを特徴とする方法。

【請求項2】

10

前記シリアルバイナリ形式を格納するステップは、ダーティ領域のデータを前記シリアルバイナリ形式の後続のデータフィールド内に格納するステップをさらに含み、前記シリアル化されたデータ内のダーティ領域のデータの存在は前記シリアルバイナリ形式の第2のデータフィールド内のドキュメントディスクリプタデータに関連付けられたフラグの1つにより指示され、前記インドキュメント構造を有するインドキュメントを、前記ドキュメント構造を生成するアプリケーション以外のアプリケーション上で表示するステップは、前記シリアルバイナリ形式の後続のデータフィールド内のダーティ領域のデータを処理することにより前記シリアルバイナリ形式から前記インドキュメントを生成するステップ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】

20

前記ダーティ領域のデータは、インクアナライザにより解析されていないインクデータの位置データ、およびインクアナライザにより解析されていない非インクデータの位置データを含むグループの少なくとも1つの要素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】

前記ダーティ領域のデータは、前記ダーティ領域のデータに関連付けられた矩形のカウント、上データ、左データ、幅データ、および高さデータを含むグループの少なくとも1つの要素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】

前記シリアルバイナリ形式を格納するステップは、グローバル意識別子データを前記シリアルバイナリ形式の後続のデータフィールド内に格納するステップをさらに含み、前記シリアルバイナリ形式内のグローバル意識別子データの存在は、前記シリアルバイナリ形式の第2のデータフィールド内のドキュメントディスクリプタデータに関連付けられたフラグの1つにより指示され、前記インドキュメント構造を有するインドキュメントを、前記インドキュメントを生成するアプリケーション以外のアプリケーション上で表示するステップは、前記シリアルバイナリ形式の後続のデータフィールド内のグローバル意識別子データを処理することにより前記シリアルバイナリ形式から前記インドキュメントを生成するステップをさらに含むことを特徴とする請求項1に記載の方法。

30

【請求項6】

前記グローバル意識別子データは、アプリケーション特有のノード型およびアプリケーション特有の拡張プロパティを含むグループの少なくとも1つの要素を含むことを特徴とする請求項5に記載の方法。

40

【請求項7】

前記シリアルバイナリ形式を格納するステップは、文字列テーブルデータを前記シリアルバイナリ形式の後続のデータフィールドに格納するステップをさらに含み、前記シリアルバイナリ形式内の文字列テーブルデータの存在は、前記シリアルバイナリ形式の第2のデータフィールド内のドキュメントディスクリプタデータに関連付けられたフラグの1つにより指示され、前記インドキュメント構造を有するインドキュメントを、前記インドキュメントを生成するアプリケーション以外のアプリケーション上で表示するステップは、前記シリアルバイナリ形式の後続のデータフィールド内の文字列テーブルデータを

50

処理することにより前記シリアルバイナリ形式から前記インクドキュメントを生成するステップ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 8】

前記文字列テーブルデータは、解析ヒントサフィックスデータ、プレフィックステキストデータ、擬似事実データ、ヒントノードデータ、単語リストデータ、カスタムリンクノードデータ、および認識された文字列データを含むグループの少なくとも 1 つの要素に関連付けられていることを特徴とする請求項 7 に記載の方法。

【請求項 9】

前記シリアルバイナリ形式を格納するステップは、リンクデータを前記シリアルバイナリ形式の後続のデータフィールドに格納するステップをさらに含み、前記シリアル化されたデータ内のリンクデータの存在は、前記シリアルバイナリ形式の第 2 のデータフィールドのドキュメントディスクリプタデータに関連付けられたフラグの 1 つにより指示され、前記インクドキュメント構造を有するインクドキュメントを、前記インクドキュメントを生成するアプリケーション以外のアプリケーション上で表示するステップは、前記シリアルバイナリ形式の後続のデータフィールド内のリンクデータを処理することにより前記シリアルバイナリ形式から前記インクドキュメントを生成するステップをさらに含むことを特徴とする請求項 1 に記載の方法。

10

【請求項 10】

前記リンクデータは、前記ドキュメント構造に関連付けられたリンクのカウント、リンクデータサイズ、リンクディスクリプタ、リンク元ノードインデックスデータ、およびリンク先ノードインデックスデータを含むグループの少なくとも 1 つの要素を含むことを特徴とする請求項 9 に記載の方法。

20

【請求項 11】

前記シリアルバイナリ形式を格納するステップは、カスタムプロパティデータを前記シリアルバイナリ形式の後続のデータフィールドに格納するステップをさらに含み、前記シリアル化されたデータ内のカスタムプロパティデータの存在は、前記シリアルバイナリ形式の第 2 のデータフィールド内のドキュメントディスクリプタデータに関連付けられたフラグの 1 つにより指示され、前記インクドキュメント構造を有するインクドキュメントを、前記インクドキュメントを生成するアプリケーション以外のアプリケーション上で表示するステップは、前記シリアルバイナリ形式の後続のデータフィールド内のカスタムプロパティデータを処理することにより前記シリアルバイナリ形式から前記インクドキュメントを生成するステップをさらに含むことを特徴とする請求項 1 に記載の方法。

30

【請求項 12】

前記カスタムプロパティデータは、サイズデータ、および前記カスタムプロパティデータを表すバイトの配列を含むグループの少なくとも 1 つの要素を含むことを特徴とする請求項 11 に記載の方法。

【請求項 13】

前記シリアルバイナリ形式の第 3 のデータフィールド内の少なくとも 1 つのルートノードのルートノードデータはディスクリプタデータを含み、前記シリアルバイナリ形式内のディスクリプタデータの存在は、前記ルートノードデータに関連付けられたデータを指示する 1 つまたは複数のフラグを含み、前記インクドキュメント構造を有するインクドキュメントを、前記インクドキュメントを生成するアプリケーション以外のアプリケーション上で表示するステップは、前記シリアルバイナリ形式のルートノードデータ内のディスクリプタデータを処理することにより前記シリアルバイナリ形式から前記インクドキュメントを生成するステップをさらに含むことを特徴とする請求項 1 に記載の方法。

40

【請求項 14】

前記ルートノードデータは前記ルートノードデータのサイズを指示するサイズデータを含むことを特徴とする請求項 13 に記載の方法。

【請求項 15】

前記ルートノードデータは、ノード位置データ、ストロークデータ、子ノードデータ、

50

ノード既知プロパティデータ、およびノードカスタムプロパティデータを含むグループの少なくとも1つの要素を含むことを特徴とする請求項13に記載の方法。

【請求項16】

インクドキュメント構造を生成するアプリケーション以外のアプリケーション上でインクドキュメントを表示するコンピュータ実行可能命令を有するコンピュータ読み取り可能格納媒体であって、前記インクドキュメント構造はシリアル化されたデータ形式に従って格納されており、前記命令は、

前記インクドキュメントのインクに関連する少なくとも1つのルートノードを含むインクドキュメント構造を含むインクドキュメントを生成すること、

前記インクドキュメント構造をシリアルバイナリ形式で格納することであって、

前記インクドキュメント構造に関連付けられたサイズデータを前記シリアルバイナリ形式の第1のデータフィールドに格納すること、

前記インクドキュメント構造で使用可能なデータを表す少なくとも1つのフラグを含むインクドキュメントディスクリプタデータを、前記シリアルバイナリ形式の第2のデータフィールドに格納すること、および

前記少なくとも1つのルートノードのルートノードデータであって、前記第2のデータフィールド内のインクドキュメントディスクリプタデータの少なくとも1つのフラグにより指示されるルートノードデータを、前記シリアルバイナリ形式の第3のデータフィールドに格納すること

を含む、シリアルバイナリ形式を格納すること、

前記インクドキュメント構造を生成するアプリケーション以外のアプリケーションから前記シリアルバイナリ形式を有する前記格納されたインクドキュメント構造にアクセスすること、および

前記インクドキュメント構造を有するインクドキュメントを、前記インクドキュメント構造を生成するアプリケーション以外のアプリケーション上で表示することを含み、前記インクドキュメントは、前記シリアルバイナリ形式の第1のデータフィールド内のインクドキュメント構造に関連付けられたサイズデータ、前記シリアルバイナリ形式の第2のデータフィールド内のインクドキュメントディスクリプタデータ、および前記シリアルバイナリ形式の第3のデータフィールド内の少なくとも1つのルートノードのルートノードデータに基づいて、前記シリアルバイナリ形式から生成されることを特徴とするコンピュータ読み取り可能格納媒体。

【請求項17】

前記ルートノードデータは、予想データ、サイズデータ、ノード位置データ、ストロークデータ、子ノードデータ、ノード既知プロパティデータ、およびノードカスタムプロパティデータを含むグループの少なくとも1つの要素を含むことを特徴とする請求項16に記載のコンピュータ読み取り可能格納媒体。

【請求項18】

インクドキュメント構造を生成するアプリケーション以外のアプリケーション上でインクドキュメントを表示するシステムであって、前記インクドキュメントはシリアル化されたデータ形式に従って格納されており、前記システムは、

プロセッサと、

コンピュータ実行可能命令を格納しているメモリとを備え、前記コンピュータ実行可能命令は、

前記インクドキュメントのインクに関連する少なくとも1つのルートノードを含むインクドキュメント構造を含むインクドキュメントを生成すること、

前記インクドキュメント構造をシリアルバイナリ形式で格納することであって、

前記インクドキュメント構造に関連付けられたサイズデータを前記シリアルバイナリ形式の第1のデータフィールドに格納すること、

前記インクドキュメント構造で使用可能なデータを表す少なくとも1つのフラグを含むインクドキュメントディスクリプタデータを、前記シリアルバイナリ形式の第2のデ

10

20

30

40

50

ータフィールド内に格納すること、および

前記少なくとも1つのルートノードのルートノードデータであって、前記第2のデータフィールド内のインクドキュメントディスクリプタデータの少なくとも1つのフラグにより指示されるルートノードデータを、前記シリアルバイナリ形式の第3のデータフィールドに格納すること

を含む、シリアルバイナリ形式を格納すること、

前記インクドキュメント構造を生成するアプリケーション以外のアプリケーションから前記シリアルバイナリ形式を有する前記格納されたインクドキュメント構造にアクセスすること、および

前記インクドキュメント構造を有するインクドキュメントを、前記インクドキュメント構造を生成するアプリケーション以外のアプリケーション上で表示することのために構成され、前記インクドキュメントは、前記シリアルバイナリ形式の第1のデータフィールド内のインクドキュメント構造に関連付けられたサイズデータ、前記シリアルバイナリ形式の第2のデータフィールド内のインクドキュメントディスクリプタデータ、および前記シリアルバイナリ形式の第3のデータフィールド内の少なくとも1つのルートノードのルートノードデータに基づいて、前記シリアルバイナリ形式から生成されることを特徴とするシステム。

#### 【請求項19】

前記シリアルバイナリ形式を格納することは、前記シリアルバイナリ形式の後続のデータフィールド内に、ダーティ領域のデータ、グローバル意識別子データ、文字列テーブルデータ、リンクデータ、およびカスタムプロパティデータを含むグループの少なくとも1つの要素を格納することをさらに含むことを特徴とする請求項18に記載のシステム。

#### 【請求項20】

前記ルートノードデータは、予想データ、サイズデータ、ノード位置データ、ストロークデータ、子ノードデータ、ノード既知プロパティデータ、およびノードカスタムプロパティデータを含むグループの少なくとも1つの要素を含むことを特徴とする請求項18に記載のシステム。

#### 【発明の詳細な説明】

#### 【技術分野】

#### 【0001】

タブレットPCを使用することで、ユーザはふつう、画面に直接描画または書くことができる。この描画または手書きは、一般に「インキング」と呼ばれる。インキングは、一種のユーザ入力であり、タッチスクリーン、およびユーザが従来のペンと紙で書くのと同様にコンピューティングペン(computing pen)を使用し画面上に書くことを含むことができる。インキングは、さまざまなアプリケーションで使用される。例えば、インキングは、ドローイングアプリケーション、ペインティングアプリケーション、ワードプロセッシングアプリケーション、クレジットカード署名アプリケーションなどで使用することができる。

#### 【背景技術】

#### 【0002】

インキングは、ただ単なるペンストロークの視覚的表現を含むだけではなく、データ型をも含むことができる。データ構造は知られているが、情報を格納するために使用されるデータ構造のサイズはひどく大きくなり、扱いにくいものとなる可能性がある。また、プログラム間にドキュメント構造の互換性があれば、コンピュータの効率および一般的な使い勝手が向上する。しかし、互換性は、ユーザが一方のアプリケーションから他方のインクアプリケーションへインクデータを転送することを望んでいる場合に問題になりうる。

#### 【発明の開示】

#### 【発明が解決しようとする課題】

#### 【0003】

一般に、本発明のいくつかの態様は、ドキュメント構造を生成し、シリアルバイナリ形

10

20

30

40

50

式でそのドキュメント構造を格納するシステムおよび方法に関する。本発明はまた、インクドキュメント構造を生成し、他のアプリケーションからアクセス可能なようにインクドキュメント構造を格納するシステムおよび方法にも関係する。本発明は、さらに、インキング全体の再解析を必要とすることなくインキングの一部を修正または変更するシステムおよび方法に関する。

【課題を解決するための手段】

【0004】

本発明のいくつかの態様は、ドキュメントデータが他のアプリケーションからアクセス可能なようにシリアル化されたバイナリ形式でドキュメントデータを格納するためのデータ構造を備えるコンピュータ可読媒体に関する。ルートノードデータを含むドキュメント構造が生成される。ドキュメントは、シリアルバイナリデータ形式で格納される。シリアルバイナリ形式は、第1のデータフィールド内のドキュメント構造に関連付けられたサイズデータを格納することを含む。シリアルバイナリ形式はまた、第2のデータフィールドにドキュメントディスクリプタデータを格納することを含み、ドキュメント構造ディスクリプタデータは、データ予想を示すための少なくとも1つのフラグを含む。シリアルバイナリ形式は、さらに、第3のデータフィールドにルートノードデータを格納することを含み、ルートノードデータは、フラグの少なくとも1つにより示される。

10

【0005】

本発明の他の態様は、2分木構造をシリアル化形式で格納するコンピュータ実装方法に関する。コンピュータ実装方法は、第1のデータフィールド内にドキュメント構造のサイズデータを格納することを含む。このコンピュータ実装方法はまた、第2のデータフィールドにドキュメント構造ディスクリプタデータを格納することを含み、ドキュメント構造ディスクリプタデータは、データ予想を示すための少なくとも1つのフラグを含む。このコンピュータ実装方法は、さらに、第3のデータフィールドにルートノードデータを格納することを含み、ルートノードデータは、フラグの少なくとも1つにより示される。

20

【0006】

本発明のさらに他の態様は、コンピュータ実行可能命令が格納されているコンピュータ可読媒体に関する。これらの命令は、インクドキュメント構造を生成することを含み、インクドキュメント構造は、少なくとも1つのルートノードを含む。これらの命令はまた、シリアルバイナリ形式でインクドキュメント構造を格納することを含む。シリアルバイナリ形式は、第1のデータフィールド内のインクドキュメント構造に関連付けられたサイズデータを格納することを含む。シリアルバイナリ形式はまた、第2のデータフィールドにインクドキュメントディスクリプタデータを格納することを含み、インクドキュメント構造ディスクリプタデータは、データ予想を示すための少なくとも1つのフラグを含む。シリアルバイナリ形式は、さらに、第3のデータフィールドにルートノードデータを格納することを含むことができ、ルートノードデータは、フラグの少なくとも1つにより示される。

30

【発明を実施するための最良の形態】

【0007】

本発明のいくつかの実施形態について、これ以降、本発明の一部をなし、実例として、本発明を実施する特定の例示的な実施形態を示す、添付の図面を参照しつつさらに詳しく説明する。しかし、本発明は、多くの異なる形態で具現化することができ、したがって、本明細書で述べたいくつかの実施形態に限定されるものと解説すべきではなく、むしろ、これらの実施形態は、本開示が網羅的で完全であり、本発明の範囲を完全に当業者に伝えるように実現される。とりわけ、本発明は、方法またはデバイスとして実現することができる。したがって、本発明は、全体としてハードウェア実施形態、全体としてソフトウェア実施形態、またはソフトウェアおよびハードウェアの態様を組み合わせた実施形態の形を取りうる。したがって、以下の詳細な説明は、限定的な意味で解釈すべきではない。

40

【0008】

シリアルバイナリ形式でデータを格納する方法およびシステムの例示的な実施形態

50

一般に、本発明は、インクドキュメント構造を生成し、他のアプリケーションからアクセス可能のようにインクドキュメント構造を格納するシステムおよび方法に関係する。より具体的には、本発明は、格納効率を高めるためデータをシリアルバイナリ形式で格納するシステムおよび方法に関係する。本発明はまた、インキング全体の再解析を必要とすることなくインクの一部を修正または変更するシステムおよび方法に関係する。本明細書で述べている説明ではインクドキュメントの格納およびロードを参照しているとしても、本明細書で参照されているシリアルバイナリ形式は、他の型のデータを格納するために使用することができる。例えば、本発明は、ワードプロセッシングアプリケーション、スプレッドシートアプリケーション、ドローイングアプリケーション、グラフィックスアプリケーション、ノートアプリケーション、画像アプリケーションなどに関連するデータを含むことができる。簡潔に述べると、シリアルバイナリ形式は、2分木構造に関連付けられた任意の型のデータを格納するために使用することができる。

10

#### 【0009】

##### 1. インクアプリケーションの例示的な実施形態

本発明の一態様の一般的な文脈として、インクアプリケーションは、ユーザがペンを使用してデータを入力するときに視覚的フィードバックをリアルタイムで返すことができる。しかし、インキングは、ペンストロークの視覚化よりもさらに多くのものを含み、データ型を含むことができる。ユーザは、ペン、インク、インク解析、およびインク認識のさまざまなレベルの機能をサポートするデジタイザ用のアプリケーションを構築することができる。このようなアプリケーションは、単純なテキスト入力の認識から複雑なインクドキュメントの作成および編集に至るまでさまざまである。

20

#### 【0010】

インクアプリケーションはまた、インク-テキスト間変換も含むことができる。状況によっては、直接的インク入力を受け付けないアプリケーションもある。このような状況では、インクアプリケーションは、手書き認識を実装し、インクをテキストに変換することにより、直接的インク入力を受け付けないアプリケーションにカット&ペーストすることができる。アプリケーションはまた、インクオブジェクトおよび他のドキュメントオブジェクトに関するその文脈も認識することができる。他の実施形態では、ユーザは、インクを操作し、インクを使用して、テキスト、グラフィックス、ベクトル形状、マルチメディアオブジェクトなどを含むリッチドキュメントをオーサリングすることができる。このような実施形態では、インクを、インクオブジェクトをリフローし重ねることができるデータ型として取り扱う。

30

#### 【0011】

インク入力は、未処理 (raw) インクデータの形態でアプリケーションに関連付けることができる。一実施形態では、未処理インクデータをインクアナライザに送り、未処理インクデータを処理し、未処理インクデータと別にすることができるインクドキュメント構造を生成することができる。インクアナライザは、未処理インクデータを管理可能な複数のストローク成分に分割するため解析および認識プロセスを実装することができる。以下でさらに詳しく説明するように、一実施形態では、インクアナライザは、木のそれぞれのノードが未処理インクデータの一部との関係を定義する2分木を有するインクドキュメント構造を生成することができる。インクドキュメント構造を使用すると、プラットフォームに関連付けられたインクアプリケーションは、未処理インクデータとインクドキュメント構造を関係付けて、オリジナルのインクおよび関連付けられたインクドキュメント構造をロードすることができる。また、インクドキュメント構造を使用することで、ユーザは、インクドキュメント全体を再解析することなくインクをロードし修正することもできる。また、本発明では、1つのプラットフォーム上の複数のアプリケーション間でインクを共有することができる。

40

#### 【0012】

図3は、インクドキュメントデータをシリアルバイナリ形式で格納するためのシステムの例示的な一態様の一般的な概要を表す。図に示されているように、システム300は、

50

デジタイザ302、アプリケーション304、およびインクアナライザ306を含む。デジタイザ302は、図1に関して説明されているコンピューティングデバイスを含むことができる。デジタイザ302は、図2に関して説明されているモバイルコンピューティングデバイスも含むことができる。一実施形態では、デジタイザは、ワシントン州レッドモンドの本件特許出願人のWINDOWS（登録商標）XP TABLET PC EDIT ION上で稼働するTABLET PCデジタイザを含む。しかし、デジタイザ302は、インクアプリケーションの動作をしやすくするデバイスを含むことができる。

#### 【0013】

デジタイザ302は、ユーザ入力ストローク（例えば、手書きおよび/または描画ストローク）を2値化し、一実施形態では、データを未処理データ記憶装置（storage）308に格納する。未処理データ記憶装置308は、デジタイザ302からのデータを維持することができる種類の記憶装置である。未処理データ記憶装置308も、1つまたは複数のアプリケーションおよび/または1つまたは複数のプラットフォームに関連付けることができる。他の実施形態では、デジタイザ302は、未処理データ記憶装置308をバイパスし、2値化されたデータをアプリケーション304に送信する。

#### 【0014】

アプリケーション304は、プラットフォームに関連付けられた任意のアプリケーションを含むことができる。一実施形態では、アプリケーション304は、インクを使いやすくするアプリケーションである。アプリケーション304は、ワードプロセッシングアプリケーション、ペイントアプリケーション、製図アプリケーション、ドローイングアプリケーション、クレジットカード署名アプリケーションなどを含むことができる。一実施形態では、アプリケーション304は、本件特許出願人のInkEdit、InkPicture、および/またはOneNoteを含む。他の実施形態では、アプリケーション304は、保存オペレーションおよびロードオペレーションを実行することができる。保存オペレーションは、インクデータおよび非インクデータを保存することを含むことができる。アプリケーション304は、未処理インクデータを未処理データ記憶装置308に保存し、アプリケーション304は、インクドキュメント構造をプラットフォームに関連付けられたインクアナライザに保存することができる。ロードオペレーションでは、アプリケーション304は、以下でさらに詳しく説明するようにインクドキュメント構造と未処理インクデータをロードし、統合することができる。

#### 【0015】

インクアナライザ306は、アプリケーション304から未処理インクデータを受信するように構成することができる。インクドキュメント構造を生成するため、インクアナライザ306は未処理インクデータに対し構造解析を実行するように構成される。構造解析は、未処理データの解析および未処理データの認識を含むことができる。

#### 【0016】

一実施形態では、構造解析を行うことで、テキスト認識、手書き、および描画分類、およびレイアウト解析を容易にすることができる。インクアナライザ306は、連携動作してテキスト認識機能を高める解析コンポーネントおよびレコグナイザコンポーネントを含むことができる。例えば、パーサー（parser）は、インクがレコグナイザに送られる前に前処理ステップとしてオペレーションを実行することができる。前処理では、パーサーは、マルチラインインク（multi-lined ink）を解析して「きれい（clean）に」し、それを一度に1パーセルずつレコグナイザに送ることができる。パーセルは、インクドキュメントの一部を含むことができる。パーサーは、さらに、不正な入力ストローク順序情報を是正して、入力順序にかかわらずすべてのストロークが確実に認識されるように構成することができる。また、パーサーは、隣接行に関する情報を生成することができる。例えば、2つの隣接行が1つの行頭文字で始まるという事実は、現在行が行頭文字で始まることを示す強い指標となりうる。

#### 【0017】

他の実施形態では、インクアナライザ306の解析オペレーションはまた、インクを描

10

20

30

40

50



画または手書きとして分類することを含むこともできる。手書きは、単語を認識しやすくするインクストロークを含むことができる。描画ストロークは、手書きストロークでないものを含むことができる。例えば、図4を参照すると、ストローク「H」は、手書きストロークを含み、「下線」は、描画ストロークを含むことができる。この方法では、一実施形態において、手書きストロークは、レコグナイザに送られる唯一のストロークとすることができる。

#### 【0018】

インクアナライザ306のさらに他の実施形態では、レイアウト解析は、お互いおよび非インクデータに関する手書きおよび描画ストロークの細目(break down)を含む。インクアナライザ306がインキングのストロークを解析した後、これらのストロークの木表現(つまり、インクドキュメント構造)を生成することができる。簡潔にいうと、インクアナライザ306は、ドキュメントを2分木で格納し、シリアルバイナリ形式を介してその2分木を他のアプリケーションからアクセスできるようにすることが可能な任意の種類のアナライザを含むことができる。シリアルバイナリ形式がインクドキュメント構造を参照しつつ本明細書で説明されているとしても、シリアルバイナリ形式を使用して、ドキュメント木構造に関連付けられた任意の種類の情報を格納することができる。

#### 【0019】

インクアナライザ306が未処理データに基づいてインクドキュメント構造を生成した後、インクドキュメント構造はアプリケーション304から利用することができるようになる。インクドキュメント構造は、ライブインクドキュメント構造を含むことができる。格納オペレーションが引き起こされる(instigated)と、アプリケーション側は、インクアナライザ306がインクドキュメント構造を格納することを要求する。インクアナライザ306がプラットフォームコンポーネントであるという点において、インクドキュメント構造は他のインクアプリケーションから利用可能である。例えば、ユーザがワードプロセッシングドキュメントでインクを生成する場合、このインクを、再度解析する必要なく、ドローイングアプリケーションにカット&ペーストすることができる。この実施例では、ドローイングアプリケーションは、インクドキュメント構造からオリジナルのインクを生成する方法を認識する。また、インクは、解析されシリアルバイナリ形式で保存されるため(後述)、インク全体の再解析を必要とすることなく、修正し、効率よく格納することができる。修正された部分は、インクドキュメント構造の単一のパーセル(parcel)に対応し、したがって、変更されたパーセルの再解析のみをすればよい。

#### 【0020】

一般に、ロードオペレーションの際に、アプリケーション304は、未処理インクデータ、非インクデータ(non-ink data)、およびインクドキュメント構造をロードすることができる。未処理インクデータは、未処理データ記憶装置308からロードすることができる。非インクデータも、未処理データ記憶装置308からロードすることができる。しかし、非インクデータは、アプリケーション304に関連付けられた記憶装置からロードされると考えられる。インクドキュメント構造は、プラットフォームコンポーネントであってよい、インクアナライザ306からロードすることができる。一実施形態では、アプリケーション304は、再解析を行わなくてもインクがロードされるように未処理データおよびインクドキュメント構造に関連付ける。

#### 【0021】

図4は、本発明の一態様による例示的なインキング400を表している。インキング400は、テキスト、図面、テーブル、チャートなどに関連付けることができるか、またはそれらとの関係を持つことができる。また、インキング400は、さまざまな種類の手書き、描画、形状、言語、記号、および歪みを含むこともできる。以下でさらに詳しく説明するが、インキング400は、インクドキュメント構造の複数のノードに相関する複数の入力を含むことができる。例えば、参照番号402は、手書き領域を示している。

#### 【0022】

他の実施例として、参照番号404は、位置揃え(alignment)レベルを示し

10

20

30

40

50

ている。図 4 に例示されているインキング 4 0 0 の最初と最後の行は、同じレベルにインデントされ、したがって、位置揃えレベル 4 0 4 を示している。インキング 4 0 0 の真ん中の行は、内側にインデントされ、したがって、他の位置揃えレベルを示している。

#### 【 0 0 2 3 】

さらに他の実施例では、参照番号 4 0 6 は、段落を示し、参照番号 4 0 8 は、1 行を示す。インキング 4 0 0 はまた、単語 4 1 0 も含み、図には示されていないが、単語 4 1 0 はまた、ストロークも含むことができる。ストロークは、単語の一部を含むことができる。

#### 【 0 0 2 4 】

図 5 は、例示的なインクドキュメント構造 5 0 0 を表す。例示的なインクドキュメント構造 5 0 0 は、例示的なインキング 4 0 0 に関係する。インクドキュメント構造 5 0 0 は、インクドキュメント構造の一例にすぎない。データ構造の表現を実現しやすくする任意の種類の木構造を実装することができる。インクドキュメント構造 5 0 0 は、ルートノード 5 0 1、手書き領域ノード 5 0 2、位置揃えレベルノード 5 0 4、段落ノード 5 0 6、行ノード 5 0 8、単語ノード 5 1 0、および/またはストロークノード(図に示されていない)などの複数のノードを含む。インクドキュメント構造 5 0 0 はまた、描画ノード 5 1 2、ヒントノード 5 1 4、および1つまたは複数のリンクも含むことができる。

#### 【 0 0 2 5 】

図 4 において、線描画 4 1 2 は、「Mr . B h a t t a c h a r y a y」という名前の下線である。線描画 4 1 2 は、図 5 では描画ノード 5 1 2 により表される。線描画 4 1 2 は単語「Mr .」および「B h a t t a c h a r y a y」に関連付けられるという点において、描画ノード 5 1 2 および単語ノード 5 1 0 および 5 1 1 は、図 5 に示されているようなリンクを通じて関連付けられる。同様に、参照番号 4 1 4 は、ある種類のヒントを表す。一実施形態では、ヒント 4 1 4 はヒントボックスを含む。ヒント 4 1 4 は、入力が数、英字、記号、構造、コード、順序などであることを示す。例えば、図 4 では、ヒントは、入力が3桁以下の数であることを示唆するヒントを含むことができる。したがって、インクアナライザは、「5」を「S」と取り違えることがない。ヒント 4 1 4 は手書き「35」に関連付けられているという点において、ヒントノード 5 1 4 は、図 5 に示されているようにリンクを通じて単語ノード 5 1 5 に関連付けることができる。上述の実施例は、例示および説明のみを目的としている。

#### 【 0 0 2 6 】

この方法では、インキング 4 0 0 は、ノードを通じてインクドキュメント構造 5 0 0 として表すことができる。例えば、ストロークノード(図に示されていない)は、単語ノード 5 1 0 の子であってよい。単語ノード 5 1 0 は、行ノード 5 0 8 の子であってよく、行ノード 5 0 8 は、段落ノード 5 0 6 の子であってよい。同様に、段落ノード 5 0 6 は、位置揃えレベルノード 5 0 4 の子であってよく、位置揃えレベル点 5 0 4 は、手書き領域ノード 5 0 2 の子であってよい。この方法で、ルートノード 5 0 1 は、その子ノードのすべての情報含むことができる。一実施形態では、インキング全体 4 0 0 は、ルートノード 5 0 1 を参照して表すことができる。ドキュメント木構造内のドキュメントの表現がしやすくなる限り任意の個数のノードを任意の種類のドキュメントに関連付けることができる。

#### 【 0 0 2 7 】

##### 2 . インクドキュメントのシリアル化

図 6 は、シリアルバイナリ形式 6 0 0 でドキュメント構造を内部的に格納するための例示的な一実施形態を表す。インクドキュメント構造が本明細書で参照されているとしても、シリアルバイナリ形式 6 0 0 を使用して任意の種類の木ドキュメント構造を格納することができる。インクドキュメント構造が生成される場合、ドキュメント構造に関係する1つまたは複数の文字列が存在する。一実施形態では、圧縮は、これらの文字列の L e m p e l - Z i v W e l c h 形式(「L Z W 形式」)を含む。しかし、文字列は、文字列のサイズを縮小する圧縮形式により圧縮できると考えられる。図 6 は、データの記憶装置 6 0 4 ~ 6 1 8 の拡大図を含む(一部のデータは状況に応じて格納される)。一

10

20

30

40

50

実施形態では、記憶装置は、記憶領域を節約するため符号なし整数を格納しやすくする多バイト符号化（「MBE」）値を含む。

【0028】

シリアルバイナリデータブロック602は、インドキュメントのシリアル化されたバイナリデータを含み、データブロック604～618により表される。データブロック604～618は、シリアルバイナリデータブロック602全体の拡大図を表す。サイズデータ604は、シリアルバイナリデータブロック602に格納される第1の情報とすることができる。サイズデータ604は、インドキュメント構造のサイズに関連付けられたデータを含む。

【0029】

インドキュメントディスクリプタデータ606は、サイズデータ604に従うことができる。インドキュメントディスクリプタデータ606は、シリアルバイナリデータブロック602に含まれるデータの型に関して予想に関連付ける任意の型のデータを含むことができる。この予想は、インドキュメント構造で使用可能な関連付けられたデータを表すフラグの集合により示すことができる。これらのフラグは、シリアルバイナリデータブロック602内で使用可能なデータを示すことができる。データブロック604～618は、インドキュメント構造に関連付けることができるデータの少数の例にすぎない。本発明の一実施形態では、ルートノードデータ614（以下でさらに説明する）は、常に、インドキュメントディスクリプタ606内のフラグに関連付けられる。

【0030】

ダーティ領域（Dirty region）データ608は、すべてのインドキュメント構造に関連付けられるわけではないオプションデータである。ダーティ領域データ608は、保存前に完全には解析されていないインドキュメント構造内のデータを指す。ダーティ領域データ608は、TextWord、Imageなどのインクデータと非インクデータの両方を指すことができる。ダーティ領域データ608は、インドキュメントディスクリプタデータ606に関連付けられたフラグにより示すことができる。インドキュメントディスクリプタデータ606がダーティ領域を示すフラグを含む場合、このフラグは、インドキュメント構造が有限の、空でないダーティ領域を持つことを示す。ダーティ領域データ608が存在する場合、このデータは、ダーティ領域の再作成をしやすくするためバイナリ形式で格納される一連の矩形として表すことができる。インドキュメントが完全に解析される状況では、ダーティ領域データ608は、存在しない場合があり、インドキュメントディスクリプタ606内にフラグを必要としない。一実施形態では、ダーティ領域データ608（もし存在すれば）は、インドキュメントディスクリプタ606の直後に来る。

【0031】

一実施形態では、ダーティ領域データ608は、領域データとしてシリアルバイナリデータブロック602に格納される。領域データ形式は、ダーティ領域データ608、非インク葉文脈ノードに対する位置データ、またはヒントノードに対する位置を格納するために使用することができる。領域データは、領域データのエリア全体を定義する個別の矩形の配列を含むことができる。ストリームから領域データオブジェクト（例えば、ダーティ領域データ608）を適切に再構築するために、領域データは矩形のカウントを含むことができる。すべての矩形について、領域データは、上データ、左データ、幅データ、および高さデータに関する情報を含むことができる。矩形データを記述する個々の値は、MBEまたは符号付き多バイト符号化（「SMBE」）を使用して格納することができる。永続的領域データの表現の一例は以下のとおりである。

【0032】

10

20

30

40

## 【表 1】

```

MBE [Count of Rectangles]
[Rectangle data]      }
...                    } Count of Rectangles
[Rectangle Data]      }

```

矩形データは、以下のように表すことができる。

## 【 0 0 3 3 】

## 【表 2】

10

```

SMBE [Rectangle.Left]
SMBE [Rectangle.Top]
SMBE [Rectangle.Width]
SMBE [Rectangle.Height]

```

## 【 0 0 3 4 】

グローバル一意識別子（「GUID」）テーブルデータ 6 1 0 は、すべてのインドキュメント構造に関連付けられるわけではないオプションデータである。GUID テーブルデータ 6 1 0 は、GUID テーブルおよび / または GUID 値のリストに関連付けられている MBE GUID の個数のカウントを含むことができる。GUID 値のリストは、それぞれの GUID に対する 16 バイト符号なしリテラル値（unsigned literal value）を含むことができる。インドキュメント構造またはドキュメント木構造内の個々のノードは、GUID により識別される任意のデータを含むことができる。この任意のデータは、周知のデータ型および特定のアプリケーションに関連付けられているデータ型を含むことができる。特定のアプリケーションに関連付けられているデータ（つまり、カスタムプロパティデータ）では、データは特定の GUID と突き合わせて格納される。GUID テーブルデータ 6 1 0 は、演繹的に知られることのないインドキュメント構造に関して使用される GUID の値を指定する。GUID テーブルデータ 6 1 0 は、インドキュメントレベルまたは文脈ノードレベルで任意のカスタムプロパティデータに対応し、その後、MBE を介して、GUID テーブルデータ 6 1 0 に関するゼロを基点とするインデックスと呼ばれる。例えば、事前定義されていない GUID は、アプリケーション特有の拡張ノード型およびノードに関するアプリケーション特有の拡張プロパティを含むことができる。GUID テーブルデータ 6 1 0 がシリアルバイナリデータブロック 6 0 2 に関して存在している状況では、その存在は、ドキュメントディスクリプタデータ 6 0 6 に関係するフラグにより識別される。同様に、GUID テーブルデータ 6 1 0 が存在しない場合、インドキュメントディスクリプタデータ 6 0 6 内でフラグはセットされない。永続的 GUID テーブルデータの表現の一例は以下のとおりである。

## 【 0 0 3 5 】

## 【表 3】

40

```

MBE [Count of Guids]
[GUID]      }
...         } Count of Guids
[GUID]      }

```

## 【 0 0 3 6 】

文字列テーブルデータ 6 1 2 は、すべてのインドキュメント構造に関連付けられるわけではないオプションデータである。文字列テーブルデータ 6 1 2 は、文字列テーブル内の MBE 文字列の個数のカウント、圧縮された文字列データのサイズ、および / または圧

50

縮された文字列データを含むことができる。文字列テーブルデータ 6 1 2 は、解析ヒント、サフィックスデータ、プレフィックステキストデータ、擬似事実データ、ヒント名データ、単語リストデータ、カスタムノードリンクデータ、および認識された文字列データに関連付けることができる。本発明の一態様に関して、文字列テーブルデータ 6 1 2 は、重複を含むことができる。インドキュメント構造が特定の順序でロードされる限り、文字列テーブルデータ 6 1 2 へのインデックスを保持することで、文字列テーブルデータ 6 1 2 から適切な文字列データをロードすることができる。

【 0 0 3 7 】

インデックスは、文字列が文字列テーブルデータ 6 1 2 に関連付けられるたび毎に書き込まれるわけではない。このような状況では、1 インスタンス当たり少なくとも 1 バイトが節約される。さらに、文字列テーブルデータ 6 1 2 内の文字列は L Z W 圧縮とすることができる。L Z W 圧縮と組み合わせてすべての文字列についてインデックスを書かないことにより、文字列のサイズを実質的に縮小することができる。文字列テーブルデータ 6 1 2 がシリアルバイナリデータブロック 6 0 2 に関して存在している状況では、その存在は、インドキュメントディスクリプタデータ 6 0 6 に関係するフラグにより識別される。同様に、文字列テーブルデータ 6 1 2 が存在しない場合、インドキュメントディスクリプタデータ 6 0 6 内でフラグはセットされない。永続的文字列テーブルデータの表現の一例は以下のとおりである。

【 0 0 3 8 】

【表 4】

// StringTable Data

MBE [Count of strings]

MBE [Size of Compressed string data]

[Compressed string data bytes]

【 0 0 3 9 】

ルートノードデータ 6 1 4 は、ルートノードのサイズに関係するデータおよび / またはルートノードに関連付けられたデータを含む。ルートノードデータ 6 1 4 は、図 8 に関して説明されているように格納することができる（以下でさらに詳しく説明する）。一態様では、ルートノードデータは、ルートノードデータ 6 1 4 が空であってもすべてのインドキュメント構造と関連する必須データである。インドキュメントディスクリプタデータ 6 0 6 に関連付けられたフラグは、ルートノードデータ 6 1 4 の存在を示すことができる。

【 0 0 4 0 】

リンクデータ 6 1 6 は、すべてのインドキュメント構造に関連付けられるわけではないオプションデータである。リンクデータ 6 1 6 は、インドキュメント構造のノードが同じインドキュメント構造内の他のノードにリンクされているかどうかを示すデータを含む。リンクデータ 6 1 6 は、インドキュメント構造に関連して大域的に保持されるようにできる。リンクデータ 6 1 6 を格納するときに、リンクデータ 6 1 6 は、インドキュメント構造に関連付けられたリンクの個数のカウントを含むことができる。個別のリンクデータ 6 1 6 はまた、データの M B E サイズも含むことができる。一態様では、M B E サイズデータの後にリンクディスクリプタが続き、これにより、リンクの型およびリンク元情報 ( o r i g i n i n f o r m a t i o n ) を識別する。他の態様では、リンクディスクリプタデータの後に、リンク元ノード ( s o u r c e n o d e ) インデックスの S M B E 値とリンク先ノード ( d e s t i n a t i o n n o d e ) インデックスの S M B E 値が続く。リンク元ノードインデックスおよびリンク先ノードインデックスにより、リンク元ノードおよびリンク先ノードをそれぞれ識別する。さらに他の態様では、リンクデータ 6 1 6 がカスタムリンクを含むことをリンクディスクリプタデータが示している場合、グローバル文字列テーブル内のインデックスにより示されたグローバル文字列テ

10

20

30

40

50

ルからカスタムリンクデータが読み込まれる。リンクデータ 6 1 6 がシリアルバイナリデータブロック 6 0 2 に関して存在している状況では、その存在は、インドキュメントディスクリプタデータ 6 0 6 に関係するフラグにより識別される。同様に、リンクデータ 6 1 6 が存在しない場合、インドキュメントディスクリプタデータ 6 0 6 内でフラグはセットされない。永続的リンクデータの表現の一例は以下のとおりである。

【 0 0 4 1 】

【表 5】

[ContextLink Descriptor] //1byte

SMBE [Source Node Index]

10

SMBE [Destination Node Index]

【 0 0 4 2 】

カスタムプロパティデータ 6 1 8 は、すべてのインドキュメント構造に関連付けられるわけではないオプションデータである。カスタムプロパティデータ 6 1 8 は、インドキュメント構造に関連付けることができ、一態様では、ノードに関連付けられたカスタムプロパティデータとして格納される。カスタムプロパティデータは、アプリケーションがノードと関連付ける任意のデータを含むことができる。カスタムプロパティデータは G U I D により識別され、既知または未知の G U I D を含むことができる。G U I D が未知の状況では、G U I D は G U I D テーブルデータ 6 1 0 として格納することができる。カスタムプロパティデータ 6 1 8 を格納する際に、フラグによりカスタムプロパティデータ 6 1 8 を既知の値として識別することができる。他の態様では、カスタムプロパティデータ 6 1 8 の格納は、G U I D テーブルデータ 6 1 0 へのインデックスを含む。カスタムプロパティデータ 6 1 8 の記憶装置はまた、データのサイズの M B E 値およびそのデータを表すバイトの配列を含むこともできる。カスタムプロパティデータ 6 1 8 がシリアルバイナリデータブロック 6 0 2 に関して存在している状況では、その存在は、インドキュメントディスクリプタデータ 6 0 6 に関係するフラグにより識別される。同様に、カスタムプロパティデータ 6 2 0 が存在しない場合、インドキュメントディスクリプタデータ 6 0 6 内でフラグはセットされない。永続的インドキュメント構造の表現の一例は以下のとおりである。

20

30

【 0 0 4 3 】

## 【表 6】

MBE [Size]	
<InkStructureDescriptor-1byte>	
// Dirty Region Data	
[AnalysisRegion Data]	
//GuidTable Data	
MBE [Count of Guids]	10
[GUID]        }	
...        } Count of Guids	
[GUID]        }	
// StringTable Data	
MBE [Count of strings]	
MBE [Size of LZ Compressed string data]	
[LZ Compressed string data]	20
//Root Node data	
MBE [Size]	
[Data]	
//Global Context Link Data	
MBE [Size of the Link Table]	
[Individual Link Data]	

## 【 0 0 4 4 】 30

図 7 は、文脈ノードデータ 7 0 0 を内部に格納するための例示的な一実施形態を表している。一実施形態では、ルートノードデータ 6 1 4 は、文脈ノードであり、文脈ノードデータ 7 0 0 として格納される。文脈ノードデータ 7 0 0 は、インクドキュメントのシリアル化されたバイナリデータに含まれることができ、データブロック 7 0 4 ~ 7 1 6 により表される。データブロック 7 0 4 ~ 7 1 6 は、文脈ノードデータ 7 0 2 の拡大図を表す。

## 【 0 0 4 5 】

ノードディスクリプタデータ 7 0 4 は、インクドキュメント構造のそれぞれのノードに関連付けられているデータを含むことができる。ノードディスクリプタデータ 7 0 4 は、ノードデータの構成とともにインクドキュメント構造に関連付けられたノードの種類をも定義するフラグの集合体により示すことができる。

40

## 【 0 0 4 6 】

ノードサイズデータ 7 0 6 は、特定のノード上に格納される可能な既知のプロパティを含むことができる（例えば、格子データ、境界ボックスデータ、およびノまたはピンニングフラグデータ）。ノードサイズデータ 7 0 6 はまた、位置データ、子下位ノードデータ、およびストロークデータとともに未知のプロパティ（拡張ノカスタムプロパティ）を含むこともできる。一態様では、ノードサイズデータ 7 0 6 は、ノードディスクリプタデータ 7 0 4 の直後に置くことができる。簡潔に述べると、ノードディスクリプタデータ 7 0 4 は、文脈ノード木全体のサイズを示すことができる。

## 【 0 0 4 7 】

ノード位置データ 7 0 8 は、すべてのノード型に関連付けられるわけではないオブショ

50

ンデータである。ノード位置データ708が存在している状況では、その存在は、ノードディスクリプタデータ704に関係するフラグにより識別される。同様に、ノード位置データ708が存在しない場合、ノードディスクリプタデータ704内でフラグはセットされない。一態様では、ノードディスクリプタデータ704が非インク葉ノードを示している場合、ノード位置データ708が続くことができる。非インク葉ノードは、子ノードを持たない、またストロークデータを含まないノードを含むことができる。例えば、非インク葉ノードは、イメージノード、テキストノード、またはヒントノードを含むことができる。一実施形態では、ノード位置データ708は領域データとして格納される。領域データは、領域データのエリア全体を定義する個別の矩形の配列を含むことができる。ストリームから領域データオブジェクト（例えば、ノード位置データ708）を適切に再構築するために、領域データは矩形のカウントを含むことができる。すべての矩形について、領域データは、上データ、左データ、幅データ、および高さデータに関する情報を含むことができる。矩形データを記述するこれら個々の値は、MBEまたはSMBEを使用して格納することができる。永続的領域データの表現の一例は以下のとおりである。

【0048】

【表7】

MBE [Count of Rectangles]

[Rectangle data] }

... } Count of Rectangles

[Rectangle Data] }

矩形データは、以下のように表すことができる。

SMBE [Rectangle.Left]

SMBE [Rectangle.Top]

SMBE [Rectangle.Width]

SMBE [Rectangle.Height]

【0049】

ストロークデータ710は、すべてのノード型に関連付けられるわけではないオプションデータである。ストロークデータ710は、ストロークデータを含む任意のノードに関連付けられたデータを含むことができる。例えば、ストロークデータ710は、非分類インクノード、単語ノード、または描画ノードに関連付けることができる。ストロークデータ710が存在している状況では、その存在は、ノードディスクリプタデータ704に関係するフラグにより識別することができる。同様に、ストロークデータ710が存在しない場合、ノードディスクリプタデータ704内でフラグはセットされない。

【0050】

ストロークデータ710が存在する場合、記憶装置には、ノードに関連付けられたストロークの数のMBE値を入れることができる。一態様では、それぞれのストロークは、ストロークディスクリプタフラグの集合体を含む1バイトのストロークディスクリプタに関連付けられる。これらのフラグは、MBEストローク識別データを示すストローク識別を示すことができる。一態様では、ストローク識別フラグがセットされていない場合、ストローク識別は、取り出された最後のストローク識別を含むことができる。これらのフラグはまた、手書きに関連付けられているストロークの型を識別する手書きストロークディスクリプタフラグも含むことができる。他の態様では、これらのフラグは、描画に関連付けられているストロークの型を識別する描画ストロークディスクリプタフラグを含むことができる。これらのフラグはまた、ハイライトに関連付けられているストロークの型を識別するハイライターディスクリプタフラグも含むことができる。さらに他の実施形態では、これらのフラグは、ストロークの有効な確認済みの祖先を識別する確認済み祖先ディスクリプタフラグを含むことができる。確認済み祖先シリアル化インデックスのMBE値をス

10

20

30

40

50



トリーム内に格納することができる。さらに他の実施形態では、これらのフラグは、ストロークに関連付けられた言語を識別するストローク言語の識別フラグを含むことができる。言語に対応する符号付き符号化値 (signed encoded value) は、ストリーム内に格納できる。さらに他の実施形態では、フラグに関連付けられた値は、ストロークディスクリプタデータに応じて上述の順序で格納される。ストロークの識別を容易にする任意の型のフラグをセットすることもさらに考えられる。

【0051】

子ノードデータ712は、すべてのノード型に関連付けられるわけではないオプションデータである。子ノードデータ712が存在している状況では、その存在は、ノードディスクリプタデータ704に関係するフラグにより識別することができる。同様に、子ノードデータ712が存在しない場合、ノードディスクリプタデータ704内でフラグはセットされない。子ノードデータ712は、コンテナ型ノードを含むことができる（つまり、段落ノード、行ノード、位置揃えノード、手書き領域ノード、および/またはルートノードなど）。コンテナノードは、子ノードを含む任意のノード型を含むことができる。子ノードデータ712の記憶装置は、子ノードの個数およびそれぞれの子ノードからのデータを格納することを含む。それぞれの子からのデータは、文脈ノードデータ702と同じようにして格納される。

【0052】

ノード既知プロパティデータ714は、すべてのノード型に関連付けられるわけではないオプションデータである。ノード既知プロパティデータ714が存在している状況では、その存在は、ノードディスクリプタデータ704に関係するフラグにより識別することができる。同様に、ノード既知プロパティデータ714が存在しない場合、ノードディスクリプタデータ704内でフラグはセットされない。

【0053】

ノード既知プロパティデータ714は、データ型および形式が知られており、データサイズを縮小するために最適化を容易にするプロパティを含む。ノード既知プロパティデータ714は、回転された境界ボックスデータ（8個の整数の配列）、認識格子データ（可変長を含むバイトの配列）、注釈データ確認データ（整数型）、および/またはヒントデータ（ヒントノード用）を含むことができる。一実施形態では、このデータは、データのバイナリ表現を最適化するために定義済みの方法により格納される。例えば、整数の配列は、符号付き符号化形式で保存することができる。また、構造体またはクラスデータなどの複合データ型は、効率よくデータを定義するバイナリ形式で格納することができる。永続的既知プロパティデータの表現の一例は以下のとおりである。

【0054】

10

20

30

## 【表 8】

[KnownProperty Descriptor] // 1 byte

[RotatedBounding BoxData]

8\*SMBE[integer representing coordinates]

[RecognitionLattice]

MBE[Size of the Lattice Data]

[Lattice Data]

[Confirmation]

SMBE [Confirmation]

[Annotation]

SMBE [Annotation]

[AnalysisHintProperties]

[AnalysisHintData]

10

## 【 0 0 5 5 】

ノードカスタムプロパティデータ 7 1 6 は、すべてのノード型に関連付けられるわけではないオプションデータである。ノードカスタムプロパティデータ 7 1 6 が存在している状況では、その存在は、ノードディスクリプタデータ 7 0 4 に関するフラグにより識別することができる。同様に、ノードカスタムプロパティデータ 7 1 6 が存在しない場合、ノードディスクリプタデータ 7 0 4 内でフラグはセットされない。ノードカスタムプロパティデータは、アプリケーションがノードと関連付ける任意のデータを含むことができる。ノードカスタムプロパティデータ 7 1 6 は G U I D により識別され、既知または未知の G U I D を含むことができる。G U I D が未知の状況では、G U I D は G U I D テーブルデータ 6 1 0 として格納することができる。ノードカスタムプロパティデータ 7 1 6 を格納する際に、フラグによりノードカスタムプロパティデータ 7 1 6 を既知の値として識別することができる。他の態様では、ノードカスタムプロパティデータ 7 1 6 を格納することは、G U I D テーブルデータ 6 1 0 へのインデックスを含む。ノードカスタムプロパティデータ 7 1 6 の記憶装置はまた、データのサイズの M B E 値およびそのデータを表すバイトの配列を含むこともできる。永続的文脈ノードデータの表現の一例は以下のとおりである。

20

30

## 【 0 0 5 6 】

## 【表 9】

MBE [Size]

// Location Data - For Non-Ink Leaf nodes

[AnalysisRegion Data]

40

// Stroke Data - For Ink Leaf nodes

MBE [Count of Strokes]

[StrokeData] }

... } Count of Strokes

[StrokeData] }

## 【 0 0 5 7 】

50

それぞれのストロークデータプロブ ( b l o b ) は、以下のようにストリームで表される。

【 0 0 5 8 】

【 表 1 0 】

<1 byte of StrokeDescriptor Flags>	
MBE [StrokeId]	
MBE [Index for the Confirmed Ancestor Node]	
MBE [LanguageId of the stroke]	10
// Children data - For Container Node	
<1 byte Node Descriptor>	
MBE [Size of the subnode data]	
[SubNode data]	
<1 byte Node Descriptor>	
MBE [Size of the subnode data]	
[SubNode data]	
[Data for Known Properties]	20
[TagMaxKnownPropertyCount + Index into Global Guid Table]	
MBE [Size Custom Properties]	
[Custom Property Data]	

【 0 0 5 9 】

3. シリアル化形式でドキュメント格納するための例示的なプロセス

図 8 は、インクドキュメントデータをシリアル化バイナリ形式で格納するためのシステムの一般的な実施形態を表している。システム 8 0 0 は、開始ブロック 8 0 2 から始まり、インクドキュメントが生成されるブロック 8 0 4 へ流れる。ブロック 8 0 4 は、ユーザからのストローク入力を 2 値化するデジタイザを備えることができる。デジタイザは、コンピューティングデバイス（例えば、図 1）、モバイルコンピューティングデバイス（例えば、図 2）、本件特許出願人の W I N D O W S（登録商標） X P T A B L E T E D I T I O N 上で稼働する T A B L E T P C、またはインクアプリケーションのオペレーションを容易にする任意のデバイスを含むことができる。さらに、本明細書の説明でインクドキュメントを参照しているとしても、システム 8 0 0 は、任意の型のデータに関して実装することができる。例えば、本発明は、ワードプロセッシングアプリケーション、スプレッドシートアプリケーション、ドローイングアプリケーション、グラフィックスアプリケーション、ノートアプリケーション、画像アプリケーションなどに関連して使用することもできる。一実施形態では、アプリケーションは、本件特許出願人の I n k E d i t、I n k P i c t u r e、および/または O n e N o t e を含む。ブロック 8 0 4 から、プロセス 8 0 0 はブロック 8 0 6 に流れる。

【 0 0 6 0 】

ブロック 8 0 6 は、未処理インクデータを生成するステップを示す。一実施形態では、未処理インクデータを生成することは、デジタイザでストローク入力を未処理インクデータに変換し、未処理インクデータを未処理データ記憶装置に格納することを含む。他の実施形態では、未処理インクデータを生成することは、デジタイザでストローク入力を未処理インクデータに変換し、未処理インクデータをアプリケーションに送信することを含む。次いでプロセス 8 0 0 は、ブロック 8 0 8 に流れる。

## 【 0 0 6 1 】

ブロック 8 0 8 は、インクドキュメント構造を生成するステップを示す。ブロック 8 0 8 は、アプリケーションから未処理インクデータを受信するインクアナライザを含むことができる。インクアナライザは、図 3 に関して上で詳しく説明されているように解析および認識オペレーション用に構成することができる。ブロック 8 0 8 は、さらに、図 4 および 5 に従って上述のようにインクドキュメント構造を生成することを含むことができる。他の実施形態では、インクドキュメント構造は、インキングに相関する複数のノードを含む。例えば、インクドキュメント構造は、手書き領域ノード、位置揃えノード、段落ノード、行ノード、単語ノード、またはストロークノードを含むことができる。インクドキュメント構造は、さらに、描画ノードおよび/またはヒントノードを含むこともできる。インクドキュメント構造は、インキングの 2 分木表現を実現しやすくする任意の型のノードを含むことができることが考えられる。他の実施形態では、これらのノードは、関係するノードと相関するリンクに関連付けることができる。

10

## 【 0 0 6 2 】

ブロック 8 1 0 に流れるときに、インクドキュメント構造は、大域的にアクセス可能なように格納できる。一実施形態では、インクドキュメント構造は、圧縮され、図 6 および 7 に関して上で説明されているように格納される。このような場合、インキングは、アプリケーションと連携して生成することができ、その後、他のアプリケーションからアクセス可能なように格納することができる。言い方を変えると、プラットフォームに関連付けられている他のアプリケーションは、インクドキュメント構造および未処理インクを使用して解析されたインクを再生成することができる。このようにアクセス可能であるため、アプリケーション間でカット & ペーストのオペレーションを容易に実行できる。また、インクは、インクが解析され、認識され、シリアル化形式で保存されている限り、インクドキュメント全体の再解析を行わなくても修正することができる。

20

## 【 0 0 6 3 】

## 4 . 例示の動作環境

図 1 を参照すると、本発明を実装する例示的なシステムは、コンピューティングデバイス 1 0 0 などのコンピューティングデバイスを含む。基本的な構成では、コンピューティングデバイス 1 0 0 は、少なくとも 1 つの処理ユニット 1 0 2 およびシステムメモリ 1 0 4 を備えるのがふつうである。コンピューティングデバイスの正確な構成と種類に応じて、システムメモリ 1 0 4 は揮発性 ( R A M など )、不揮発性 ( R O M、フラッシュメモリ など )、またはこれら 2 つの何らかの組み合わせとすることができる。システムメモリ 1 0 4 は、通常、オペレーティングシステム 1 0 5、1 つまたは複数のアプリケーション 1 0 6 を含み、またプログラムデータ 1 0 7 を含むこともある。一実施形態では、アプリケーション 1 0 6 は、さらに、インキングオペレーション用のアプリケーション 1 2 0 を含む。基本構成は、図 1 において点線 1 0 8 内のコンポーネントにより示されている。

30

## 【 0 0 6 4 】

コンピューティングデバイス 1 0 0 は、さらに特徴または機能を追加することもできる。例えば、コンピューティングデバイス 1 0 0 は、磁気ディスク、光ディスク、またはテープなどの追加データ記憶装置デバイス ( 取り外し可能および/または取り外し不可能 ) を備えることもできる。このような追加記憶装置は、図 1 では、取り外し可能記憶装置 1 0 9 および取り外し不可能記憶装置 1 1 0 により例示されている。コンピュータ記憶媒体は、コンピュータ可読命令、データ構造体、プログラムモジュール、またはその他のデータなどの情報を格納する方法または技術で実装される揮発性および不揮発性、取り外し可能および取り外し不可能媒体を含むことができる。システムメモリ 1 0 4、取り外し可能記憶装置 1 0 9、および取り外し不可能記憶装置 1 1 0 は、すべてコンピュータ記憶媒体の例である。コンピュータ記憶媒体としては、限定はしないが、R A M、R O M、E E P R O M、フラッシュメモリまたはその他のメモリ技術、C D - R O M、デジタル多目的ディスク ( D V D ) またはその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その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

40

50

使用することができコンピューティングデバイス 100 によりアクセスできるその他の媒体がある。このような任意のコンピュータ記憶媒体を装置 100 の一部とすることができる。さらにコンピューティングデバイス 100 は、キーボード、マウス、ペン、音声入力デバイス、タッチ入力デバイスなどの（複数の）入力デバイス 112 を有することもできる。ディスプレイ、スピーカ、プリンタなどの（複数の）出力デバイス 114 を備えることもできる。これらの装置はすべて、当業に周知であるため、本明細書でさらに詳しい説明をする必要はない。

#### 【0065】

また、コンピューティング装置 100 は、デバイスがネットワークまたはワイヤレスメッシュネットワークなどを経由して他のコンピューティングデバイス 118 と通信するために使用する通信接続（群）116 も含むことができる。通信接続（群）116 は、通信媒体の一実施例である。通信媒体は、通常、コンピュータ可読命令、データ構造体、プログラムモジュール、または搬送波もしくはその他のトランスポートメカニズムなどの被変調データ信号によるその他のデータを具現するものであり、任意の情報配信媒体を含む。「被変調データ信号」という用語は、信号内に情報を符号化するような方法で特性のうちの 1 つまたは複数の設定または変更された信号を意味する。例えば、限定はしないが、通信媒体としては、有線ネットワークまたは直接配線接続などの有線媒体、および、音響、RF、赤外線、およびその他の無線媒体などの無線媒体がある。本明細書で使用されているコンピュータ可読媒体という用語は、記憶媒体と通信媒体の両方を含む。

#### 【0066】

図 2 は、本発明の例示的な一実施形態で使用するモバイルコンピューティングデバイスを示している。図 2 を参照すると、本発明を実装する例示的なシステムは、モバイルコンピューティングデバイス 200 などのモバイルコンピューティングデバイスを含む。モバイルコンピューティングデバイス 200 は、プロセッサ 260、メモリ 262、ディスプレイ 228、およびキーパッド 232 を備える。メモリ 262 は、一般に、揮発性メモリ（例えば、RAM）と不揮発性メモリ（例えば、ROM、フラッシュメモリなど）の両方を含む。モバイルコンピューティングデバイス 200 は、メモリ 262 内に常駐し、プロセッサ 260 上で実行される、オペレーティングシステム 264 を格納する。キーパッド 232 は、プッシュボタン式数値ダイヤリングパッド（典型的な電話機などの）、またはマルチキーキーボード（従来のキーボードなど）としてよい。ディスプレイ 228 は、液晶ディスプレイ、またはモバイルコンピューティングデバイスで一般的に使用される他の種類のディスプレイとすることができる。ディスプレイ 228 は、タッチセンシティブとすることができ、その場合、入力デバイスとしても機能する。

#### 【0067】

1 つまたは複数のアプリケーションプログラム 266 がメモリ 262 内にロードされ、オペレーティングシステム 264 上で実行される。アプリケーションプログラムのいくつかの実施例としては、電話ダイヤラープログラム、電子メールプログラム、スケジューリングプログラム、PIM（個人情報管理）プログラム、ワードプロセッシングプログラム、スプレッドシートプログラム、インターネットブラウザプログラムなどがある。モバイルコンピューティングデバイス 200 はまた、メモリ 262 内に不揮発性記憶装置 268 を含む。不揮発性記憶装置 268 は、モバイルコンピューティングデバイス 200 の電源が切れた場合に失われてはならない永続的情報を格納するために使用することができる。アプリケーション 266 では、電子メールまたは電子メールアプリケーションにより使用されるその他のメッセージ、PIM で使用する連絡先情報、スケジューリングプログラムにより使用されるアポイントメント情報、ワードプロセッシングアプリケーションにより使用されるドキュメントなどの情報を使用し、記憶装置 268 内に格納することができる。一実施形態では、アプリケーション 266 は、さらに、インキングオペレーション用のアプリケーション 280 を含む。

#### 【0068】

モバイルコンピューティングデバイス 200 は、1 つまたは複数の電池として実装する

ことができる電源 270 を備える。電源 270 は、さらに、AC アダプタまたは電池を補助または充電する電力供給用ドッキングクレードルなどの外部電源を備えることも可能である。

#### 【0069】

モバイルコンピューティングデバイス 200 は、2 種類の外部通知メカニズム、LED 240 とオーディオインターフェイス 274 とともに示されている。これらのデバイスは、作動させたときに、プロセッサ 260 およびその他のコンポーネントが節電のためシャットダウンする可能性があっても通知メカニズムにより指示される継続期間の間オンのままになるように電源 270 に直接結合することができる。LED 240 は、ユーザがデバイスの電源投入ステータスを示す処置を講じるまでいつまでもオンのままになるようにプログラムすることができる。オーディオインターフェイス 274 は、可聴信号を供給し、ユーザから可聴信号を受信するために使用される。例えば、オーディオインターフェイス 274 をスピーカに結合することで可聴出力を供給し、マイクに結合することで、電話での会話を容易にするためなど、可聴入力を受け取ることができる。

10

#### 【0070】

モバイルコンピューティングデバイス 200 はまた、無線周波通信など通信を送受信する機能を実行する無線インターフェイス層 272 も含む。無線インターフェイス層 272 を使用すると、通信事業者またはサービスプロバイダを介して、モバイルコンピューティングデバイス 200 と外部世界との無線接続性を容易に実現できる。無線インターフェイス層 272 との間の送信は、オペレーティングシステム 264 の制御の下で実行される。つまり、無線インターフェイス層 272 により受信される通信は、オペレーティングシステム 264 を介してアプリケーションプログラム 266 に、またその逆に広めることができる。

20

#### 【0071】

上記の詳説、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関する完全な説明となっている。本発明の多くの実施形態は、本発明の精神と範囲を逸脱することなく実装できるため、本発明は付属の請求項によって定められる。

#### 【図面の簡単な説明】

#### 【0072】

【図 1】本発明の一態様で利用できる例示的なコンピューティングデバイスを示す図である。

30

【図 2】本発明の一態様で利用できる例示的なモバイルデバイスを示す図である。

【図 3】インクドキュメントデータをシリアルバイナリ形式で格納するためのシステムの例示的な一態様を示す図である。

【図 4】本発明の一態様による例示的なインキングを示す図である。

【図 5】図 4 に表されているインキングの一部を表す例示的なインクドキュメント構造を示す図である。

【図 6】本発明のいくつかの態様によるシリアルバイナリデータブロックを格納するためのデータ構造を示す図である。

【図 7】本発明のいくつかの態様による文脈ノードデータを格納するためのデータ構造を示す図である。

40

【図 8】インクドキュメント構造をシリアルバイナリ形式で格納するための一般的な態様を示す流れ図である。

#### 【符号の説明】

#### 【0073】

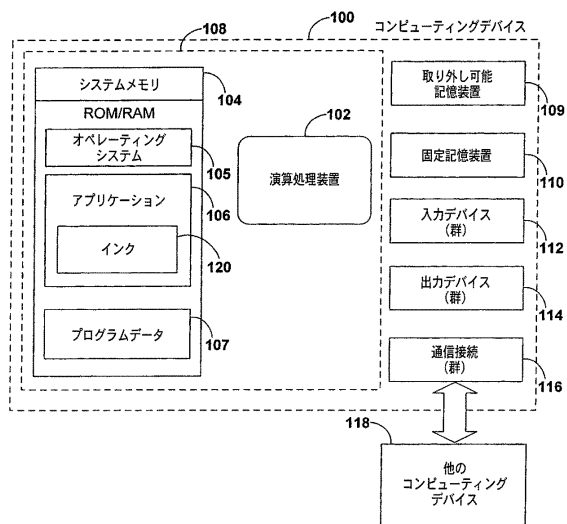
- 100 コンピューティングデバイス
- 104 システムメモリ
- 105 オペレーティングシステム
- 106 アプリケーション
- 120 インク

50

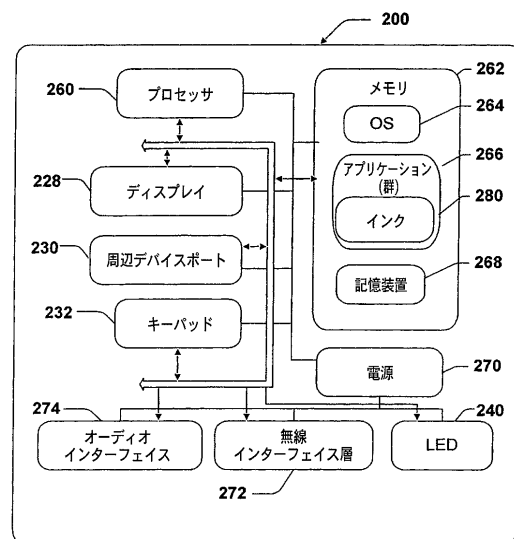
1 0 7	プログラムデータ	
1 0 2	演算処理装置	
1 0 9	取り外し可能記憶装置	
1 1 0	固定記憶装置	
1 1 2	入力デバイス（群）	
1 1 4	出力デバイス（群）	
1 1 6	通信接続（群）	
1 1 8	他のコンピューティングデバイス	
2 2 8	ディスプレイ	
2 3 0	周辺デバイスポート	10
2 3 2	キーパッド	
2 4 0	L E D	
2 6 0	プロセッサ	
2 6 2	メモリ	
2 6 4	O S	
2 6 6	アプリケーション（群）	
2 6 8	記憶装置	
2 7 0	電源	
2 7 2	無線インターフェイス層	
2 7 4	オーディオインターフェイス	20
2 8 0	インク	
3 0 2	デジタイザ	
3 0 4	アプリケーション	
3 0 6	インクアナライザ	
3 0 8	未処理データ記憶装置	
5 0 1	ルート	
5 0 2	手書き領域	
5 1 2	描画	
5 0 4	位置揃えレベル	
5 0 6	段落	30
5 0 8	行	
5 1 0	単語	
5 1 4	ヒント	
5 1 5	単語	
6 0 2	シリアルバイナリデータブロック	
6 0 4	サイズデータ（インクドキュメント全体のサイズ）	
6 0 6	インクドキュメントディスクリプタデータ	
6 0 8	ダーティ領域データ（オプション）	
6 1 0	G U I Dテーブルデータ（オプション）	
6 1 2	文字列テーブルデータ（オプション）	40
6 1 4	ルートノードデータ	
6 1 6	リンクデータ（オプション）	
6 1 8	カスタムプロパティデータ（オプション）	
7 0 2	文脈ノードデータ	
7 0 4	ノードディスクリプタデータ	
7 0 6	ノードサイズデータ	
7 0 8	ノード位置データ（オプション）	
7 1 0	ストロークデータ（オプション）	
7 1 2	子ノードデータ（オプション）	
7 1 4	ノード既知プロパティデータ（オプション）	50

## 7 1 6 ノードカスタムプロパティデータ ( オプション )

【 図 1 】

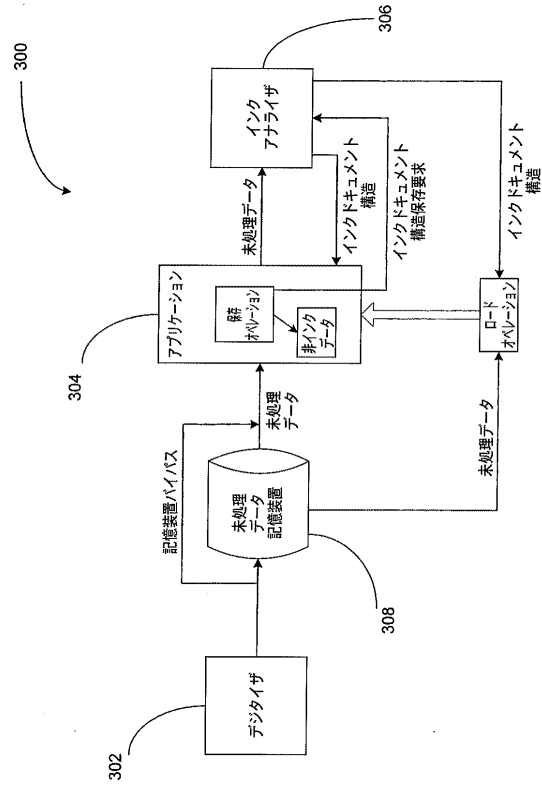


【 図 2 】

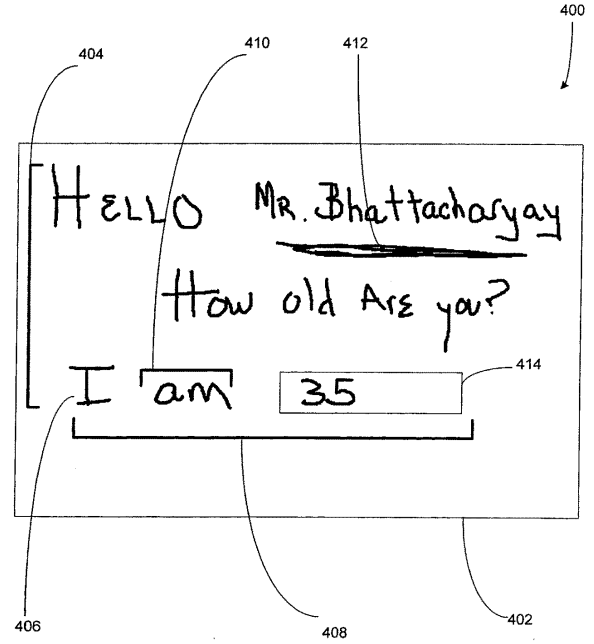




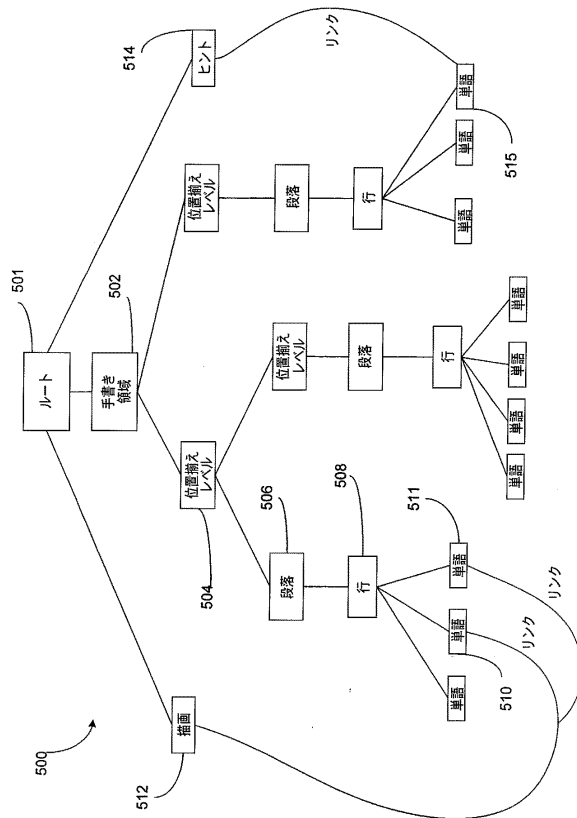
【図 3】



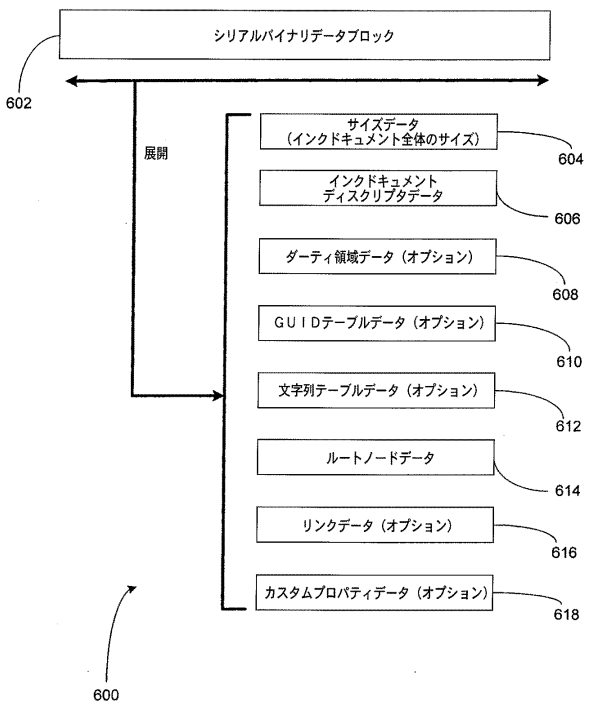
【図 4】



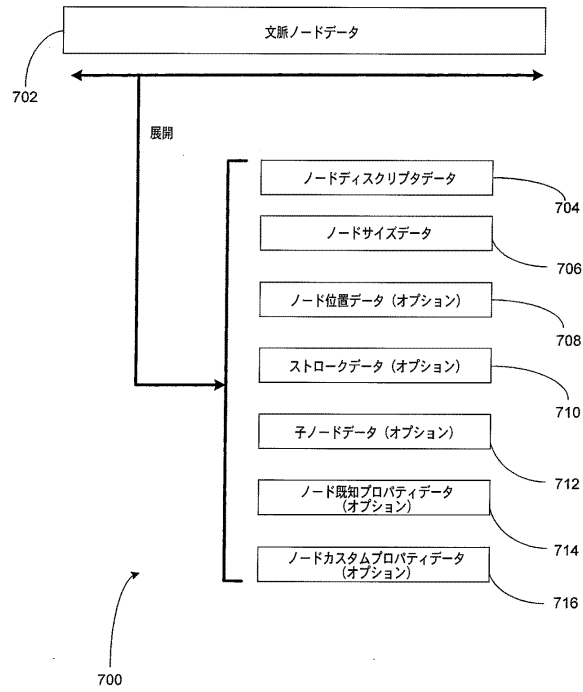
【図 5】



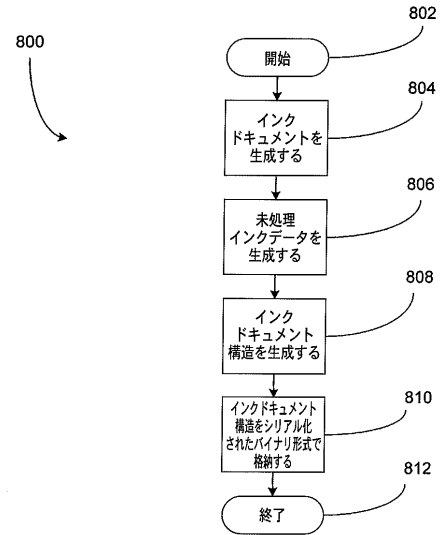
【図 6】



【図 7】



【図 8】



## フロントページの続き

- (72)発明者 ジェイミー エヌ・ウェイカム  
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 ジェローム ジョセフ ターナー  
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
- (72)発明者 セバスチャン ポウローズ  
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 スッパ バッタチャライ  
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

審査官 池田 聡史

- (56)参考文献 特開2002-082937(JP,A)  
特開2000-029909(JP,A)  
特開平08-235221(JP,A)  
Charles Alfred, オブジェクト指向データベースがシステム開発を変革する(下), 日経エ  
レクトロニクス, 日本, 日経BP社, 1994年12月 5日, 第623号, pp.13~14  
鈴木信夫, オブジェクト指向性を強めるOLEの新バージョン2.0, 日経バイト, 日本, 日  
B P社, 1993年12月 1日, 第120号, pp.178~185

- (58)調査した分野(Int.Cl., DB名)  
G06F 12/00  
G06F 17/21 ~ 17/30  
JSTPlus(JDreamII)