

(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2004-240976
(P2004-240976A)

(43) 公開日 平成16年8月26日(2004.8.26)

(51) Int. Cl.⁷
G06F 17/21

F I
G06F 17/21 586A

テーマコード(参考)
5B009

審査請求 未請求 請求項の数 20 O L (全 23 頁)

(21) 出願番号 特願2004-27313 (P2004-27313)
(22) 出願日 平成16年2月3日(2004.2.3)
(31) 優先権主張番号 10/356,972
(32) 優先日 平成15年2月3日(2003.2.3)
(33) 優先権主張国 米国(US)

(71) 出願人 500046438
マイクロソフト コーポレーション
アメリカ合衆国 ワシントン州 9805
2-6399 レッドモンド ワン マイ
クロソフト ウェイ
(74) 代理人 100077481
弁理士 谷 義一
(74) 代理人 100088915
弁理士 阿部 和夫
(72) 発明者 ジョージ タニワキ
アメリカ合衆国 98052-6399
ワシントン州 レッドモンド ワン マイ
クロソフト ウェイ(番地なし)

最終頁に続く

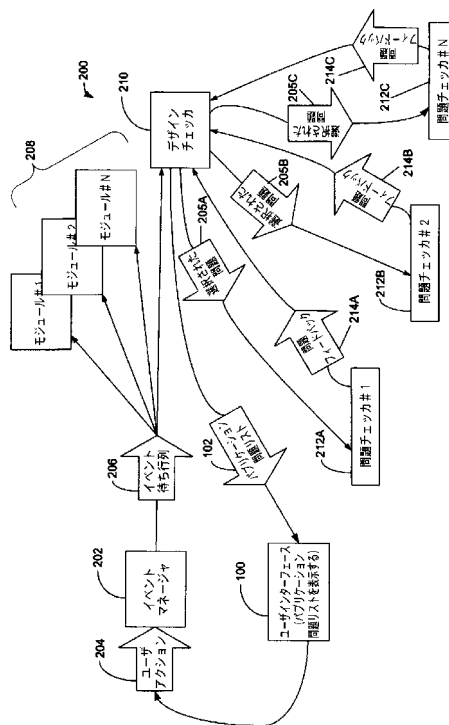
(54) 【発明の名称】 パブリケーションデザイン問題をチェックし、解決するためのシステムおよび方法

(57) 【要約】

【課題】 パブリケーションデザイン問題をチェックし、解決する。

【解決手段】 パブリッシングプログラムは、ドキュメントが編集されている、または生成されている中に起こり得るデザイン問題に関する継続的で、リアルタイムのフィードバックを、提供することができる。このフィードバックは、ドキュメントにおける現在の編集または生成の結果である可能性があるあらゆるデザイン問題のリアルタイムリストを有するユーザインターフェースを表示することを含むことが可能である。このリストは、ドキュメントが生成される、または編集されるにつれて更新される。ユーザは、所望に応じて何回でも、リストとドキュメントの間で切替えを行うことができる。このリストは、電子ドキュメントの全ての現在の潜在的なパブリケーション問題、ならびに電子ドキュメントに存在する可能性があるパブリケーション問題のタイプを明らかにすることができる。

【選択図】 図2A



【特許請求の範囲】**【請求項 1】**

ドキュメントのデザインパブリケーション問題をモニタして修正するための、コンピュータによって実施される方法であって、

ドキュメント内でオブジェクトを生成すること、編集すること、および削除することのいずれか 1 つに関する入力を受け取るステップと、

前記入力を受け取る間に、前記ドキュメントに関して起こり得るパブリケーションデザイン問題のステータスに対処するユーザインターフェースを表示するステップと、を含むことを特徴とする方法。

【請求項 2】

前記ユーザインターフェースにおいて起こり得るパブリケーションデザイン問題のリストを表示するステップを、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】

起こり得るパブリケーションデザイン問題のステータスに対処する前記ユーザインターフェースにおいて入力を受け取るステップを、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 4】

前記起こり得るパブリケーションデザイン問題の前記ステータスに関連したオプションを表示するステップを、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 5】

前記入力の構成要素を識別するステップと、イベントのリストの中に前記構成要素を記憶するステップとを、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 6】

前記イベントのリストからパブリケーションに無関係のイベントを濾過して除くステップを、さらに含むことを特徴とする請求項 5 に記載の方法。

【請求項 7】

ドキュメントの起こり得るパブリケーション問題をモニタして解決するための、コンピュータによって実施される対話型方法であって、

ドキュメント内のアクションをモニタするステップと、

前記アクションに回答して、起こり得るパブリケーション問題のステータスに関連した前記ドキュメント内でフィードバックを提供するステップと、を含むことを特徴とする方法。

【請求項 8】

前記ドキュメント内でフィードバックを提供するステップは、前記ドキュメントとは別個の表示領域内でユーザインターフェースを表示するステップを、さらに含むことを特徴とする請求項 7 に記載の方法。

【請求項 9】

前記ドキュメントに関する出力のタイプを特定するステップと、

前記出力のタイプに関連したフィードバックを提供するステップと、をさらに含むことを特徴とする請求項 7 に記載の方法。

【請求項 10】

ドキュメントに関して起こり得るパブリケーション問題のリストを表示するステップと、

前記リストからの問題の選択に回答して前記ドキュメントにおける問題を識別するステップと、をさらに含むことを特徴とする請求項 7 に記載の方法。

【請求項 11】

ドキュメントに関して起こり得るパブリケーション問題のリストを表示するステップと、

前記リストからの前記問題の選択に回答して、前記問題の 1 つを含む前記ドキュメント

10

20

30

40

50

におけるページを表示するステップとを、さらに含むことを特徴とする請求項 7 に記載の方法。

【請求項 1 2】

ドキュメントに関して起こり得るパブリケーション問題のリストを表示するステップと、

前記リストからの前記問題の選択にตอบสนองして、前記ドキュメントにおける問題をスクリーン機構を使用することにより強調表示するステップとを、さらに含むことを特徴とする請求項 7 に記載の方法。

【請求項 1 3】

ドキュメントに関して起こり得るパブリケーション問題のリストを表示するステップと

前記リストからの前記問題の選択にตอบสนองして、編集するために前記ドキュメントに戻るステップとを、さらに含むことを特徴とする請求項 7 に記載の方法。

【請求項 1 4】

パブリケーション問題を解決するために前記ドキュメントを自動的に編集するステップを、さらに含むことを特徴とする請求項 7 に記載の方法。

【請求項 1 5】

作業管理ファイルを生成するためのコンピュータシステムであって、
処理ユニットと、

メモリ記憶デバイスと、

データを表示するために前記処理ユニットに結合された表示デバイスと、

前記処理ユニットに命令を提供するために前記メモリ記憶デバイスの中に記憶されたプログラムとを含み、

前記処理ユニットは、前記プログラムの前記命令にตอบสนองして、

ドキュメントを表示する間に起こり得るデザインパブリケーション問題のリストを表示するステップと、

前記ドキュメントを編集すること、および前記問題のリストにおける問題に関するさらなる情報を要求すること、のいずれか一つのための入力を受け取るステップと、を行うように動作することを特徴とするシステム。

【請求項 1 6】

前記プロセッサは、さらに加えて、

前記ドキュメント内のアクションをモニタするステップと、

前記アクションに関連した問題を識別するステップとを行うように動作する、ことを特徴とする請求項 1 5 に記載のシステム。

【請求項 1 7】

前記プロセッサは、さらに加えて、

前記ドキュメント内のアクションをモニタするステップと、

前記アクションにตอบสนองして前記問題のリストを更新するステップとを行うように動作する、ことを特徴とする請求項 1 5 に記載のシステム。

【請求項 1 8】

前記プロセッサは、さらに加えて、前記リストから問題が選択されたことにตอบสนองして、問題を含む前記ドキュメントにおけるページを表示するステップを行うように動作することを特徴とする請求項 1 5 に記載のシステム。

【請求項 1 9】

前記プロセッサは、さらに加えて、

ユーザアクションをイベントのリスト中に記憶するステップと、

前記ユーザアクションが前記ドキュメントのパブリケーションにどのように影響するかを特定するステップとを行うように動作する、ことを特徴とする請求項 1 5 に記載のシステム。

【請求項 2 0】

10

20

30

40

50

前記プロセッサは、さらに加えて、

ユーザアクションを潜在的なデザインパブリケーション問題と共にマッピングするステップと、

前記ドキュメントに対して実行するチェックのリストを生成するステップと、

前記チェックのリストを実行するステップとを行うように動作する、ことを特徴とする請求項15に記載のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、電子ドキュメント(electronic document)をチェック(check)し、編集(edit) 10
することに関する。より具体的には、本発明は、パブリッシング(publishing)に先立って
潜在的な問題(potential problem)について電子ドキュメントを精査(reviewing)し、編集
(editing)するための対話型プロセス(interactive process)に関する。

【背景技術】

【0002】

従来のドキュメント検査プログラム(document inspection program)は、電子ドキュメ
ントがパブリケーションされることに先立ち、起こり得る問題(possible problem)に関し
て電子ドキュメントを精査し、チェックすることができる。例えば、電子ドキュメントが
商業印刷プロセスで出版される場合、ドキュメント編集プログラムによって生成された電
子ドキュメントを、起こり得る誤り(possible error)または問題(problem)に関して 20
チェックするのに、しばしば、ドキュメント検査プログラムが使用される。ドキュメント検査
プロセス(document inspection process)は、電子ドキュメントをパブリケーションする
ことに先立って、起こり得る誤りを識別することができる。そのようなプロセスにより、
商業印刷(commercial printing)、Webサイトパブリッシング(website publishing)、
デスクトップパブリッシング、およびその他の同様のパブリッシングアプリケーションな
どのアプリケーションに費やされる時間とお金を相当に節約することが可能である。

【0003】

具体的には、従来のドキュメント検査チェックプログラムは、ドキュメント編集プログ
ラムによって生成された電子ドキュメントを精査して、ページ際上部の余白が大きすぎな
いか、オブジェクトがページ上に適切に配置されているかどうか、ページ上に配置された 30
オブジェクトが見えないことはないかどうか、特定のアプリケーションに関して不適切な
色が使用されていないかどうかを判定することができる。以上は、従来のドキュメント検
査ソフトウェアによって識別される可能性がある問題のいくつかに過ぎない。

【0004】

従来のドキュメント検査ソフトウェアは、潜在的なパブリケーション問題を識別するの
に非常に役立つ可能性があるが、そのようなソフトウェアは、通常、ドキュメント編集プ
ログラムによってドキュメントが生成された後に別個のステップとして実施され、または
実行され、ドキュメントが編集されている最中には行われぬ。ドキュメント編集プログ
ラムを使用して電子ドキュメントを生成した後でドキュメント検査プログラムを使用する
ことは非効率である可能性がある。というのは、電子ドキュメントにおける多くの問題は 40
互いに関連している可能性があり、ドキュメントが生成されている最中に識別されたなら
、かなり迅速に解決することが可能なはずであるからである。

【0005】

ドキュメント検査ソフトウェアの別の欠点は、このソフトウェアが、通常ドキュメント
編集プログラムとは別個のプログラムであることである。つまり、ドキュメント検査ソフ
トウェアは、通常、ドキュメント編集プログラムの製造業者とは異なる製造業者によっ
て作成される。したがって、ドキュメント検査ソフトウェア製造業者は、ドキュメント編集
プログラムのプログラミングコードの全てには精通していない可能性があり、非互換性の
問題がもたらされる可能性がある。

【0006】

さらに、従来のドキュメント検査ソフトウェアは、通常、ドキュメント編集プログラムにおいて存在する可能性がある電子ドキュメントと対話しない可能性がある。言い換えれば、ドキュメント検査ソフトウェアは、通常、ドキュメントに存在する誤りを報告するが、ドキュメント編集プログラムにおいて存在する電子ドキュメントに変更を行うことができない。第三者のドキュメント検査ソフトウェアは、通常、ドキュメント編集プログラムとは完全に別個のアプリケーションとして存在する。

【0007】

多数の別個のドキュメント検査ソフトウェアプログラムが存在するが、一部のドキュメント編集プログラムは、そのドキュメント編集プログラムに組み込まれたいくつかのドキュメント検査機能(document inspection function)を有している。ただし、それらのドキュメント検査機能は、通常、非常に限られており、ドキュメント検査ソフトウェアの問題のいくつかをも有している。

10

【0008】

ドキュメント検査ソフトウェアのいくつかの問題、およびドキュメント検査機能を組み込んだドキュメント編集プログラムの問題の一部には、電子ドキュメントが生成された後に電子ドキュメントのチェックを実行することが含まれる可能性がある。従来技術のドキュメント検査ソフトウェアおよびドキュメント検査機能の別の問題は、チェックプロセスが開始された際に問題が順次提示されることである。言い換えれば、従来のドキュメント検査プロセスが開始されたとき、通常、電子ドキュメントの最初の問題が識別され、ユーザは、識別された現在の問題を解決した後に次の問題に順次に進むことだけができる。問題を識別することには、その問題を含むページ上の領域を拡大すること、および、場合により、その問題を色で強調表示することが含まれる可能性がある。

20

【0009】

シーケンシャルチェックプロセス(sequential checking process)の最中、ユーザは、しばしば、識別された問題をどのように修正するかを決定を行うことを余儀なくされる。従来のドキュメント検査技術では、ユーザには、ドキュメントにどれだけの数のパブリッシング問題が存在するか、ドキュメントにどのようなタイプの問題が存在するかについて、全く見当がつかない。従来のドキュメント検査技術(document inspection technique)の問題の順次提示(sequential presentation)は、非常に硬直した非対話型プロセスとして特徴付けることができる。

30

【0010】

ドキュメント検査プロセス中、通常、潜在的なパブリッシング問題が拡大されるため、現在のページ上に存在する可能性がある他の問題を見ることが困難であり、しばしば、不可能である。ユーザが従来のドキュメント検査チェック技術で識別された問題を修正した場合、ユーザによって行われた処置により、現在の問題、またはチェックされなかった残りの問題のいずれかが修正されたことが、通常、ユーザに全く示されない。従来のドキュメント検査チェックプロセスでは、ユーザは、ドキュメント検査プロセスに対して、ほとんど、または全く制御できない。

【発明の開示】

【発明が解決しようとする課題】

40

【0011】

したがって、当技術分野において、対話的であり、ユーザがドキュメント検査プロセス中にいくらか制御することを可能にする、パブリケーションデザイン問題(publication design problem)をチェック(check)し、解決(resolve)するためのシステムおよび方法が必要とされている。当技術分野において、ドキュメントが編集(edit)されている、または生成(create)されている最中に、起こり得るデザイン問題(possible design problem)に関する継続的でリアルタイムのフィードバック(continuous and real time feedback)を提供することができるドキュメント検査プロセスがさらに必要とされている。当技術分野において、電子ドキュメントの全ての問題を識別し、電子ドキュメントに存在する可能性がある問題のタイプ(type of problem)識別するドキュメント検査プロセスがさらに必要と

50

されている。当技術分野において、ある繰返し生じる問題を自動修正するためのオプションをユーザに使用可能とすることができるように、パブリッシングプログラムと一体になったドキュメント検査プロセスがさらに必要とされている。最後に、当技術分野において、電子ドキュメントが生成されている、または編集されている最中に使用することが簡単であり、管理することが容易なドキュメント検査プロセスが必要とされている。

【課題を解決するための手段】

【0012】

パブリッシングプログラム(publishing program)は、ドキュメントが編集されている、または生成されている最中に起こり得るパブリケーションデザイン問題に関する継続的でリアルタイムのフィードバックを提供することができる。デザイン問題(design problem)には、パブリケーションの困難をもたらす可能性があるドキュメントにおける任意の数の誤りを含むことが可能である。例えば、ページ最上部の余白(トップマージン)が大きすぎること、またはオブジェクトが他のオブジェクトを隠していることを、問題と考えることが可能である。

10

【0013】

本発明のフィードバックは、ドキュメントにおける現在の編集または生成の結果である可能性があるあらゆるデザイン問題のリアルタイムリストを表示するユーザインターフェースを含むことが可能である。デザイン問題のこのリストは、ドキュメントをパブリケーションするのに先立って修正すべきドキュメントに関する起こり得るデザイン問題の全体的な見直し、つまり全体像を提供することができる。

20

【0014】

ユーザインターフェースおよび継続的なフィードバック(continuous feedback)をパブリッシングプログラムに統合して、ユーザインターフェースが表示され、実行されている間に、ドキュメントの編集、削除、または生成を行うことができるようにすることが可能である。これは、フィードバックおよびユーザインターフェースを提供するソフトウェアモジュールをパブリッシングプログラムに統合することができることも意味する。

【0015】

本発明のユーザインターフェースにより、ドキュメントに存在する可能性があるデザイン問題のリスト(list of design problem)からユーザが選択するのを可能にすることができる。問題のリストからデザイン問題が選択されると、その問題を含むページ、ならびに問題自体を表示することが可能である。問題は、問題を異なる色で強調表示(highlighting)すること、問題の周囲の輪郭を点滅させること、およびデザイン問題に注意を喚起する他の同様の技術など任意の様々なスクリーン機構(screen mechanism)を使用してより目立つようにすることができる。

30

【0016】

ユーザインターフェースは、特定のデザイン問題(particular design problem)を解決する、または修正するために選択することができるオプションを表示し、処理することができる。ユーザインターフェースは、自動修正オプションを表示することができる。この自動修正オプション(autocorrect option)が選択された場合、パブリッシングプログラムは、さらなるユーザアクションなしにデザイン問題を自動的に修正することができる。

40

【0017】

本発明の1つの利点は、継続的で、リアルタイムのフィードバックが、ドキュメント編集プログラムにおけるスペルチェッカと同様に、ドキュメントを生成するため、または編集するためのワークフロー(workflow)に統合されることとすることができる。本発明の別の利点は、デザイン問題のリストをユーザがいつでもナビゲートできることとすることができる。言い換えれば、デザイン問題の順次処理(sequential processing)のために、評価されているデザイン問題に関して強制的に判断させられた従来の技術とは異なり、本発明のユーザインターフェースによってリストで表示されたデザイン問題は、ユーザによってランダムに選択することが可能である。

【0018】

50

ユーザインターフェースは、デザイン問題のランダムな選択を可能にするので、ユーザインターフェースは、デザイン問題に優先順位を付けることを可能にすることになる。ユーザは、いつデザイン問題を修正することを要望するかを決めることができる。さらに、ユーザが所望する場合、あるチェックを行わないようにパブリッシングプログラムに命令することができる。つまり、潜在的なデザイン問題を識別するためのあるチェックの設定を行って、パブリッシングプログラムによって実行されないようにすることが可能である。

【0019】

ユーザインターフェースによって表示された問題のリストから、2つまたはそれより多くの問題の間の関係を確認することができる。さらに、この継続的で、リアルタイムのフィードバックにより、現在の生成、追加、または削除に対する即時のフィードバックを見ることによってデザイン問題をどのように回避するかを学習するようにユーザを訓練することが可能である。

10

【0020】

ドキュメントにおける継続的で、リアルタイムのフィードバックとは、ドキュメントで変更が行われるにつれ、その変更に応答してデザイン問題を更新することが可能であることを意味する。したがって、ドキュメントに変更が行われ、この変更によってリストに存在するデザイン問題が修正された場合、変更が完了するとすぐにリストからその問題が取り除かれるようにリストを更新することが可能である。

【0021】

本発明は、ドキュメントがパブリケーションされた際、出力タイプを自動的に検出することができる。出力タイプは、いくつかの出力タイプだけを挙げると、コンピュータプリンタを介するデスクトップパブリッシング、インターネットを介するWebサイトのデザインおよびパブリケーション、または商業印刷のいずれか1つとすることが可能である。出力タイプのこの自動検出により、デザイン問題の効率的な検出を可能にすることができる。というのは、パブリッシングプログラムは、別の出力タイプに関して存在する可能性がある問題を回避しながら、または探索せずに、その出力タイプに特有の問題を探索することができるからである。

20

【発明を実施するための最良の形態】

【0022】

パブリッシングプログラム(publishing program)は、ドキュメントが編集(edit)されている、または生成(create)されている最中に、起こり得るデザイン問題(possible design problem)に関する継続的で、リアルタイムのフィードバック(continuous and real-time feedback)を提供することができる。このフィードバックは、ドキュメントにおける現在の編集または生成の結果である可能性があるあらゆるデザイン問題のリアルタイムリスト(real-time list)を有するユーザインターフェースを表示することを含むことが可能である。

30

【0023】

例示的な動作環境

本発明の例示的な実施形態について、以下に、いくつかの図の全てにおいて同様の符号が同様の要素を表わしている図面を参照して説明する。図1Aは、本発明の様々な実施形態を実施するための例示的な動作環境を示すブロック図である。例示的な動作環境100は、従来のパーソナルコンピュータ120の形態で汎用コンピューティングデバイスを含む。一般に、パーソナルコンピュータ120は、処理ユニット121、システムメモリ122、ならびにシステムメモリ122から処理ユニット121までを含む様々なシステム構成要素を結合するシステムバス123を含む。システムバス123は、様々なバスアーキテクチャのいずれかを使用するメモリバスまたはメモリコントローラ、周辺バス、およびローカルバスを含むいくつかのタイプのバス構造のいずれかであることが可能である。システムメモリは、読み取り専用メモリ(ROM)124、およびランダムアクセスメモリ(RAM)125を含む。始動中などにパーソナルコンピュータ120内部の要素間で

40

50

情報を転送するのに助ける基本ルーチンを含む基本入力/出力システム(BIOS)126が、ROM124の中に記憶されている。

【0024】

パーソナルコンピュータ120は、図示していないハードディスクに対して読み取りおよび書込みを行うためのハードディスクドライブ127、取外し可能な磁気ディスク129に対して読み取りおよび書込みを行うための磁気ディスクドライブ128、およびCD-ROMまたは他の光媒体などの取外し可能な光ディスク131に対して読み取りおよび書込みを行うための光ディスクドライブ130をさらに含む。ハードディスクドライブ127、磁気ディスクドライブ128、および光ディスクドライブ130は、それぞれ、ハードディスクドライブインターフェース132、磁気ディスクドライブインターフェース133、および光ディスクドライブインターフェース134でシステムバス123に接続される。

10

【0025】

本明細書で説明する例示的な環境は、ハードディスク127、取外し可能な磁気ディスク129、および取外し可能な光ディスク131を使用するが、磁気カセット、フラッシュメモリカード、デジタルビデオディスク、ベルヌーイカートリッジ、RAM、ROMなどの、コンピュータによってアクセス可能なデータを記憶することができる他のタイプのコンピュータ可読媒体も、例示的な動作環境において使用できることが、当分野の技術者には理解されよう。以上のドライブ、および関連するコンピュータ可読媒体により、コンピュータ実行可能命令、データ構造、プログラムモジュール、およびその他のデータの揮発性ストレージがパーソナルコンピュータ120に提供される。

20

【0026】

オペレーティングシステム135、スプレッドシートプログラム136、データベースプログラム137、およびドキュメント編集プログラム138Aを含むいくつかのプログラムモジュールが、ハードディスク127、磁気ディスク129、光ディスク131、ROM124、またはRAM125に記憶されることが可能である。プログラムモジュールには、特定のタスクを行う、または特定の抽象データタイプ(particular abstract data type)を実行(implement)するルーチン、サブルーチン、プログラム、オブジェクト、構成要素(component)、データ構造(data structure)等が含まれる。本発明の態様は、パブリッシングプログラム138Bを使用して電子ドキュメントを生成するように実施することが可能である。

30

【0027】

ユーザは、キーボード140やポインティングデバイス142などの入力デバイスを介してコマンドおよび情報をパーソナルコンピュータ120に入力することができる。ポインティングデバイスには、マウス、トラックボール、および電子タブレットと併せて使用することができる電子ペンが含まれることが可能である。その他の入力デバイス(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、サテライトディッシュ、スキャナ等が含まれることが可能である。以上の入力デバイス、およびその他の入力デバイスは、しばしば、システムバス123に結合されたシリアルポートインターフェース146を介して処理ユニット121に接続されるが、パラレルポート、ゲームポート、ユニバーサルシリアルバス(USB)などの他のインターフェースで接続してもよい。また、表示デバイス147も、ビデオアダプタ148などのインターフェースを介してシステムバス123に接続されることが可能である。モニタ147に加えて、パーソナルコンピュータは、通常、スピーカやプリンタなどの他の周辺出力デバイス(図示せず)を含む。

40

【0028】

パーソナルコンピュータ120は、1つまたは複数の遠隔コンピュータ149に対する論理接続を使用するネットワーク化された環境において動作することが可能である。遠隔コンピュータ149は、別のパーソナルコンピュータ、サーバ、クライアント、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であることが可能である。遠隔コンピュータ149は、通常、パーソナルコンピュータ120に関連し

50

て前述した要素の多く、または全てを含むが、メモリ記憶デバイス150だけを図1に示している。図1に描いた論理接続は、ローカルエリアネットワーク(LAN)151およびワイドエリアネットワーク(WAN)152を含む。そのようなネットワーク環境は、オフィス、企業全体のコンピュータ網、イントラネット、およびインターネットで一般的である。

【0029】

LANネットワーク環境で使用される場合、パーソナルコンピュータ120は、しばしば、ネットワークインターフェースまたはネットワークアダプタ153を介してローカルエリアネットワーク151に接続される。WANネットワーク環境で使用される場合、パーソナルコンピュータ120は、通常、インターネットなどのWAN152を介して通信を接続するためのモデム154、またはその他の手段を含む。内部にあることも、外部にあることも可能なモデム154は、シリアルポートインターフェース146を介してシステムバス123に接続される。ネットワーク化された環境では、遠隔コンピュータ149に関連して描いたプログラムモジュール155、156、またはプログラムモジュールの部分は、遠隔のメモリ記憶デバイス150の中に記憶されることが可能である。図示したネットワーク接続は、例示的であり、コンピュータ間で通信リンクを確立する他の手段も使用できることが認められよう。

【0030】

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家庭用電化製品またはプログラマブル家庭用電化製品、ネットワークパーソナルコンピュータ、ミニコンピュータ、メインフレームコンピュータ等を含む他のコンピュータシステム構成で実施してもよいことが、当分野の技術者には認められよう。また、本発明は、通信網を介してリンクされた遠隔の処理デバイスによってタスクが行われる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る。分散コンピューティング環境では、プログラムモジュールは、ローカルのメモリ記憶デバイスと遠隔のメモリ記憶デバイスの両方の中に配置されることが可能である。

【0031】

ユーザインターフェースの例示的な表示

図1Bを参照すると、電子ドキュメントにおける現在の編集または生成の結果である可能性がある潜在的なデザイン問題104A、104B、104C、および104Dのリアルタイムリスト102を含むユーザインターフェース100が示されている。ユーザインターフェース100は、コンピュータスクリーン上で表示することが可能なビジュアルオブジェクトを含むが、本発明のリアルタイムのフィードバックは、情報の視覚表示に限定されないことが、当分野の技術者には認識されよう。リアルタイムのフィードバックは、サウンド、音声合成、触覚表示、およびユーザとのその他の形態の対話型通信(interactive communication)を含むことが可能である。

【0032】

図1Bに示した例示的な実施形態では、デザイン問題のリアルタイムリスト(real-time list)102の最初のエン트리104Aが、「ページ1でテキストボックスが空である」ことを述べるメッセージを含むことが可能である。この最初のエン트리104A、および残りのエン트리104B、104C、104Dは、ユーザが選択することが可能である。精査(review)されている現在の電子ドキュメントに関して、このリスト102は、評価(evaluate)されている電子ドキュメントの起こり得るデザイン問題の完全なリスト(complete list)である。問題リスト102がウインドウまたはユーザインターフェース100のサイズを超えた場合、スクロールボタン106を使用して問題リスト102をスクロールすることができる。

【0033】

前述した通り、本発明の利点は、ユーザが電子ドキュメントを生成している、または編集している最中に、そのユーザが、起こり得る(possible)パブリケーションデザイン問題(publication design problem)104のリスト102をいつでもナビゲートできることで

ある。ユーザインターフェース100のこのナビゲーション能力により、デザイン問題104のランダムな選択も可能になる。ユーザは、リスト102に表示された特定のデザイン問題104をいつ修正することを要望するかを決めることができる。また、ユーザインターフェース100によって表示された問題104のリスト102から見て取ることができる通り、リスト102に表示された問題の間の関係を確認することもできる。

【0034】

次に、図1Cを参照すると、起こり得るデザインパブリケーション問題104の最初のエントリ104Aが選択されたことに応答してメニュー108が表示されている。メニュー108は複数のオプション110を含むことが可能である。第1のオプション110Aは、識別された特定の問題にナビゲートするためのコマンドを含むことが可能である。第2のオプション110Bは、リスト102で識別された問題104Aをシステムが自動的に修正する自動修正機能を含むことが可能である。第3のオプション110Cは、現在の電子ドキュメント内でそのタイプの問題に関して再びチェックしないようにするコマンドを含むことが可能である。第4のオプション110Dは、リスト102から選択された問題の性質およびタイプを説明するテキストを表示するコマンドを含むことが可能である。図1Cがさらに強調する通り、ユーザインターフェース100は、非常に対話的であり、電子ドキュメントを編集している、または生成している最中に同時に実行されることが可能なドキュメント検査プロセスに対してユーザが相当なコントロールを有することを可能にする。

10

【0035】

次に、図1Dを参照すると、空のテキストボックス114を含む電子ドキュメント112が、ユーザインターフェース100に隣接して表示される。電子ドキュメント112は、図1Cに示されたメニュー108の第1のオプション110Aの選択(選択された問題にナビゲートする)に応答して表示される。テキストボックス114が、円形マーカ114Aで強調表示され、ユーザの注意を潜在的なデザイン問題に集中させる。本発明は、円形マーカ114A、114Bに限定されない。潜在的なデザイン問題は、問題を異なる色で強調表示すること、問題の周囲の輪郭を点滅させること、ならびにサウンド、音声合成、およびユーザの注意を潜在的なパブリケーションデザイン問題に向けさせる他の同様な方法などの他のコミュニケーションを介してユーザに問題を伝えることなどの円形マーカ114A、114B以外の任意の様々なスクリーン機構を使用して、より目立たせること

20

30

【0036】

次に、図1Eを参照すると、この図は、ユーザが選択された問題を修正した後のユーザインターフェース100および電子ドキュメント112を示している。ユーザインターフェース100で見られる通り、アイコン116が、図1Dに示したアイコンの前の出現(previous iteration)から変化している。具体的には、図1Eに示したアイコン116は、最初の問題104Aが解決されたことをユーザに示す「チェック」マークを表示するように変化している。ただし、前述した通り、本発明は、電子ドキュメントにおける問題がどのように識別されたか、またはどのように修正されたかについての視覚表示に限定されない。アイコン116の変化に加えて、またはその代替えとして、サウンドを再生すること、音声合成を起動すること、または問題104Aが修正されたことにユーザが気付くように何らかの他の仕方でユーザに通知を行うことにより、問題が解決されたことをユーザに知らせることが可能である。

40

【0037】

次に、図1Fを参照すると、ユーザインターフェース100が、電子ドキュメントにおいて実行して、または使用して、電子ドキュメントにパブリケーションデザイン問題が存在するかどうか判定することができる例示的なチェックリスト(list of checks)116を示している。ユーザインターフェース100のこのウィンドウにより、電子ドキュメントにおいて実行される、または使用されることが可能なチェックの選択(selection)、または選択解除(deselection of checks)が可能になる。例示的なチェックには、ページ最上

50

部の余白(トップマージン)が大きすぎるかどうか、オブジェクトがページからはずれているかどうか、オブジェクトがページ上で見えるかどうか、オブジェクトが不適切な色を有していないかどうか、またはオブジェクトがページ上で不釣り合い(object is disproportional on a page)でないかどうか、あるオブジェクトがページ上の別のオブジェクトを覆っていないかが含まれることが可能であるが、以上には限定されない。図1Fは、さらなるチェックを含む。ただし、本発明は、前述したチェック、または図1Fに示したチェックに限定されない。起こり得るパブリケーション問題に関する他のチェックも、本発明の範囲および趣旨を超えるものではない。

【0038】

次に、図1Gを参照すると、この表示は、リアルタイムチェックリスト102(この図では示さない)に関する表示オプション118を含むユーザインターフェース100を示している。図1Gに示したユーザインターフェース100は、ユーザが、電子ドキュメントのどの特定のページが起こり得るデザインパブリケーション問題に関してチェックされるべきかを選択することができるようにするページ範囲オプション119をさらに含んでいる。

【0039】

例示的なアーキテクチャ

次に、図2Aを参照すると、この図は、本発明の最終結果の(end-resulting)パブリケーションデザイン問題をチェックするためのシステムおよび方法を提供するように対話するソフトウェア要素(software element)を示している。本発明は、本明細書で説明し、例示的なアーキテクチャ図および添付の流れ図で示す機能を実現する複数のコンピュータプログラム、または単一のコンピュータプログラムを含むことが可能である。

【0040】

ただし、コンピュータプログラミングにおいて本発明を実施する多数の異なるやり方が存在する可能性があり、本発明が、いずれか1組のコンピュータプログラム命令に限定されるものと解釈すべきではないことは、明白であろう。さらに、例えば、技能を有するプログラマは、例示的なアーキテクチャおよび流れ図、ならびに本出願の本文における関連する説明に基づき、開示する本発明を実施するコンピュータプログラムなどを難なく書くことができる。したがって、特定の1組のプログラムコード命令の開示は、本発明をどのように実施し、使用するかについての十分な理解に必要であるとは考えられていない。請求するコンピュータプログラムの本発明による機能を、機能およびプログラムフローを例示する残りの図面に関連する以下の説明でより詳細に説明する。

【0041】

システム200は、イベントマネージャ202と、デザインチェッカ210と、1つまたは複数の問題チェッカ212(A)、212(B)、212(C)と、デザイン問題のリスト102を表示するためのユーザインターフェース100とを含むことが可能である。イベントマネージャ202は、電子ドキュメント112内で行われるユーザアクション204をモニタするプログラムモジュールを含むことが可能である。このユーザアクション204は、イベントリストの中に別個のイベントとして記憶されることが可能である。包括的なユーザアクション204の終わり、イベントマネージャ202が、繰り返し多いイベントをイベントリストから濾過して除くことが可能である。

【0042】

例えば、ユーザが、あるオブジェクトを10回、連続して移動した場合、イベント待ち行列206の中に存在するイベントリストの中に10のイベントが記憶される。イベントマネージャ202は、イベントリストからイベントを濾過して除くことができる。具体的には、イベントマネージャ202は、イベントリストから繰り返しの多いイベントを濾過して除くことができる。次に、イベントマネージャ202は、イベント待ち行列(event queue)206の中の濾過済みのイベントリストをデザインチェッカ210に送ることができる。

【0043】

ユーザアクション 204 は、コンピュータとの多くの異なる形態の通信を含むことが可能である。ユーザアクションは、オブジェクトを移動すること、オブジェクトのサイズを変更すること、オブジェクトの色を変更すること、電子ドキュメントのテキストまたは背景の色を変更すること、特定のセットの色 (specific set of colors) から特定の他のセットの色に移行すること (RGB 色から CMYK 色に移行するなどの)、ディスプレイ上の音声コマンド起動、スタイラス起動、またはペン起動、タッチスクリーンから受け取られた入力、および電子ドキュメントを変更するようにユーザが行うことができる任意の他のタイプのアクションを含むことが可能であるが、以上には限定されない。

【0044】

イベントリストを含むイベント待ち行列 206 を、ドキュメントパブリッシングプログラムに存在する可能性がある他のプログラムモジュール 208 に伝送することが可能である。一部のプログラムモジュール 208 に関して、イベント待ち行列 206 の濾過が、イベントマネージャ 202 によって行われられない可能性がある。その他のプログラムモジュールは、グラフィックスマネージャ、および他のドキュメントパブリッシングプログラムモジュールを含むことが可能であるが、以上には限定されない。イベントマネージャ 202 は、具体的には、デザインチェッカ 210 を宛先とするイベント待ち行列 206 内のデータを濾過する。

【0045】

デザインチェッカ 210 は、イベントマネージャ 202 によって送られたイベント待ち行列 206 からの濾過済みのイベントリストを受け取るプログラムモジュールを含むことが可能である。デザインチェッカ 210 は、イベントリストをさらに処理することができる。イベントリストの処理は、通常、システムのアイドル時間中に行われる。言い換えれば、アイドル時間は、ユーザがデータを入力していない、またはパブリケーションシステムのコマンドを全く起動していない時間を含む。

【0046】

基本的に、本発明は、ユーザがワードプロセッシングシステムの電子ドキュメントと対話しない期間中に、ワードプロセッシングプログラムが、ドキュメントのスペリングを継続的にチェックする仕方と同様に、潜在的なパブリケーションデザイン問題のリアルタイムのフィードバックをユーザに提供する。アイドル時間を使用してドキュメントをチェックすることの 1 つの目的は、ユーザを中断させないことである。別の目的は、空いた CPU サイクルが存在するときにチェックを実行することである。

【0047】

デザインチェッカ 210 は、イベントリストから、電子ドキュメントのパブリケーションに無関係のイベントを濾過して除くことができる。言い換えれば、デザインチェッカは、イベントマネージャ 202 によって行われる第 1 の濾過に対して、2 回目にイベントリストを濾過することができる。

【0048】

図 2B を簡単に参照すると、この図は、イベント待ち行列 206 の中に記憶されたイベントデータのエン트리 220 を含む。イベントデータのエン트리 220 は、イベント識別子 (identifier) 220 (A)、オブジェクト識別子 220 (B)、およびイベント固有情報 (event specific information) 220 (C) を含むことが可能である。イベント固有情報 220 (C) は、イベントに関して何が生じたかについてのさらなる情報を含むことが可能である。例えば、イベント固有情報 220 (C) は、ユーザイベントが生じたページ上の変更の位置を記述することができる。

【0049】

図 2A を再び参照すると、デザインチェッカ 210 は、電子ドキュメント 112 に対して実行されるべきチェック 218 のリスト 216 (図 2C に示す) を形成するために、潜在的な問題に対してイベントリスト内の残りのイベントをマッピングすることもできる。

【0050】

図 2C を簡単に参照すると、この図は、電子ドキュメントに対して実行されるべきチェ

ック 218 のリスト 216 を示している。チェック 218 のこのリスト 216 は、デザインチェッカ 210 がイベントリスト内のイベントをチェックのリストにマップしたことに応答して生成されることが可能である。イベント - チェック - ルックアップテーブル(event-to-check-lookup table)において行われるこのマッピングプロセスのさらなる詳細は、図 5 に関連して以下により詳細に説明する。

【0051】

図 2 A を再び参照すると、デザインチェッカ 210 は、チェックのリスト 216 から繰り返しの多いチェック 218 を濾過して除くことができ、デザインチェッカ 210 は、チェックのリスト 216 から、評価されている現在のイベントによって解決されている可能性があるチェック 218 も濾過して除くことができる。例えば、ユーザが、デザインパブリケーション問題を生じさせていたオブジェクトを削除した場合、デザインチェッカ 210 は、そのオブジェクトの削除に先立ってそのオブジェクトに関連していたチェック 218 を除去することが可能である。

10

【0052】

デザインチェッカ 210 は、個々のチェック 205 (A)、205 (B)、205 (C) を 1 つまたは複数の問題チェッカ 212 (A)、212 (B)、および 212 (C) に転送することができる。問題チェッカ 212 から何らかのフィードバック 214 (A)、214 (B)、214 (C) を受け取った後、デザインチェッカ 210 は、問題チェッカ処理の結果として電子ドキュメント 112 において明らかにされたデザイン問題のリスト 102 を構成することができる。デザインチェッカは、表示、およびユーザとの対話のためにデザイン問題のこのリスト 102 をユーザインターフェース 100 に転送することができる。

20

【0053】

それぞれの問題チェッカ 212 は、デザインチェッカ 210 から受け取られたチェック照会(check inquiry)を実行することができる 1 つまたは複数のプログラムモジュールを含むことが可能である。チェックは、電子ドキュメントにおける特定のオブジェクト、または特定のページが、特定のパブリケーションデザイン問題を有するかどうか判定することができる。問題チェッカ 212 は、ドキュメントにおける特定のオブジェクト、1 組のオブジェクト、またはページがデザインパブリケーション問題を有するかどうかをデザインチェッカ 210 に報告することができる。

30

【0054】

デザインチェッカ 210 と通信しているのが、ユーザインターフェース 210 である。ユーザインターフェース 100 は、デザインチェッカ 210 からデザイン問題のリスト 102 を受け取り、表示デバイス上で問題のリスト 102 を表示することができる 1 つまたは複数のプログラムモジュールを含むことが可能である。ユーザインターフェース 100 は、表示されている特定の問題 104 に特有の入力をユーザから受け取ることができる。そのような入力は、選択された問題 104 を表示し、修正する要求を含むことが可能である。また、ユーザインターフェース 100 は、表示されている問題のリスト 102 内の選択されたデザイン問題 104 をどのように修正するかについてのメニュー 108 にリストされたようなオプションも表示することが可能である。

40

【0055】

前述した通り、ユーザインターフェース 100 は、視覚的にユーザと対話することだけに限定されない。ユーザインターフェース 100 は、サウンド、音声合成、触覚表示、および他の知覚(sensory)デバイスまたは通信デバイスを使用してユーザと対話し、潜在的なデザイン問題についてユーザに警報することができる。

【0056】

ドキュメントのデザインパブリケーション問題をモニタし、修正するための例示的な方法

次に、図 3 A および 3 B を参照すると、これらの図は、パブリケーション問題を修正するための例示的なプロセス 300 の概要の論理流れ図を示している。以下に説明するいく

50

つかのステップは、本発明が説明する通りに機能するには、当然、他のステップに先行しなければならないことに留意されたい。

【0057】

ただし、本発明は、順序またはシーケンスが本発明の機能を変えることがない場合、説明するステップの順序に限定されない。つまり、一部のステップは、本発明の範囲および趣旨を逸脱することなく、他のステップの前または後に、あるいは他のステップと並列に行うことが可能であることが認識されている。

【0058】

判定ステップ303は、例示的なプロセス300の最初のステップである。判定ステップ303で、今回が、方法の初回の実行であるかどうかの判定が行われる。判定ステップ303に対する照会が否定的である場合、「NO」分岐を辿って図3(B)に対するページマーカAに進み、プロセスがステップ341に進められる。判定ステップ303に対する照会が肯定的であった場合、「YES」分岐を辿ってステップ306に進む。ステップ306で、デザインチェッカ210は、電子ドキュメント112におけるオブジェクトの定義または指定を受け取ることが可能である。電子ドキュメントにおけるオブジェクトの指定は、パブリッシングプログラムまたはドキュメント編集プログラムによって生成されることが可能である。

10

【0059】

ステップ309において、電子ドキュメントに間する出力のタイプが、自動的に判定される。例えば、デザインチェッカ210は、電子ドキュメント112内のオプションのユーザ選択に基づき、電子ドキュメントが、商業印刷を予定しているか、Webパブリケーションを予定しているか、デスクトップパブリッシングを予定しているか、他の同様な出力形態を予定しているかを判定することができる。ただし、他の代替の例示的な実施形態では、ドキュメント出力タイプのこの自動検出は、非活動化されるか、または利用できない可能性があり、したがって、ユーザが電子ドキュメントに関する出力タイプを手動で選択する。

20

【0060】

ステップ312において、選択された出力タイプに依存する現在の電子ドキュメントに適用されるべきチェックのリストにアクセスを行うことが可能である。その後、ステップ315で、デザインチェッカ210は、図1Fに示す通り、ユーザインターフェース100のオプションのリスト116内などで選択されたオプションに基づき、ユーザが、いくつかのチェックがスキップされるようにフラグで示したかどうか判定することができる。

30

【0061】

ステップ318において、デザインチェッカ210は、電子ドキュメント112における全てのオブジェクトおよびページに対してチェックを行う。具体的には、デザインチェッカ210は、チェックの全てを1つまたは複数の問題チェッカ212に転送する。

【0062】

ステップ321において、問題チェッカ212によって行われる1つまたは複数のチェックに対する照会が肯定的である場合(現在の電子ドキュメントに問題が存在することを意味する)、デザインチェッカ210は、パブリケーション問題リスト102を生成する。次に、ステップ323で、ユーザインターフェース100は、パブリケーション問題のリスト102を表示する。

40

【0063】

判定ステップ326において、ユーザインターフェース100は、表示問題104の1つが選択されているかどうか判定する。判定ステップ326に対する照会が否定的である場合、「NO」分岐を辿ってページコネクタAに進み、プロセスは、図3(B)のステップ341に進む。

【0064】

判定ステップ326に対する照会が肯定的であった場合、「YES」分岐を辿ってステップ329に進み、選択された問題104(A)に関するオプション110が表示される

50

。次に、判定ステップ 332 で、ユーザインターフェース 100 が、問題 104 (A) に関するオプション 110 (A) の 1 つが選択されているかどうか判定する。判定ステップ 332 に対する照会が否定的である場合、「NO」分岐を辿って判定ステップ 338 に進む。判定ステップ 332 に対する照会が肯定的である場合、「YES」分岐を辿ってステップ 335 に進み、選択されたオプションが行われる。例えば、「この項目に進め」というコマンドが選択された場合、選択された問題 104 (A) を含むページが、問題 104 のリスト 102 に隣接して表示される。この時点で、プロセスは、図 3 (B) のステップ 341 に進む。

【0065】

次に、図 3 (B) を参照すると、この図は、電子ドキュメント 112 内のパブリケーション問題を修正するためのプロセス 300 の別の部分を示している。ステップ 326 ないし 338 は、図 3 (B) のステップ 341 ないし 380 と並列に動作している可能性があることに留意されたい。つまり、ユーザがユーザインターフェース 100 を選択した場合、プロセスは、ユーザがいずれの (any) 識別されたパブリケーション問題 (identified publication problems) も修正することができるように、ステップ 326 に戻る。

10

【0066】

ステップ 341 において、ユーザが電子ドキュメント 112 内で作業することを選択した場合、電子ドキュメント 112 内のユーザアクションが、イベントマネージャ 202 によってモニタされる。ステップ 344 において、イベントマネージャは、ユーザアクションの構成要素を識別し、その構成要素をイベントのリスト内の別々のイベントとしてイベント待ち行列 206 の中に記憶する。

20

【0067】

判定ステップ 347 において、イベントマネージャ 202 により、包括的なユーザアクションの終りが生じたかが判定される。判定ステップ 347 に対する照会が否定的であった場合、「NO」分岐を辿ってステップ 341 に戻る。判定ステップ 347 に対する照会が肯定的であった場合、「YES」分岐を辿ってステップ 350 に進む。

【0068】

ステップ 350 において、イベントマネージャ 202 は、イベント待ち行列 206 のイベントリスト内の重複するイベントを濾過して除く。判定ステップ 353 において、デザインチェッカ 210 は、システムがアイドル状態であるかどうか判定することができる。ステップ 353 に対する照会が否定的であった場合、「NO」分岐を辿ってステップ 341 に戻る。判定ステップ 353 に対する照会が肯定的であった場合、「YES」を辿ってステップ 356 に進み、デザインチェッカ 210 が、イベントマネージャ 202 から受け取られたイベント待ち行列 206 の中のイベントリストから、パブリケーションに無関係のイベントを濾過して除くことができる。

30

【0069】

ステップ 359 において、デザインチェッカ 210 は、削除されたオブジェクトに関わる可能性があるチェックを濾過して除くことができる。ルーチン 362 で、デザインチェッカ 210 は、イベント - チェック - ルックアップテーブル (event-to-check-lookup table) 500 (図 5 に示す) を使用することによってイベントに関連する潜在的なパブリケーション問題にマッピングすることができる。ルーチン 362 のさらなる詳細は、図 5 に関して以下に説明する。イベントに関連する潜在的なパブリケーション問題にマップした後、デザインチェッカ 210 は、ドキュメントに対して実行されるチェックのリスト 216 を構成すること、またはそのリスト 216 に追加を行うことができる。

40

【0070】

ステップ 368 において、デザインチェッカ 210 は、実行されるチェックのリスト 216 から重複するチェックをさらに濾過して除く。判定ステップ 374 で、デザインチェッカ 210 は、システムが依然としてアイドルであるかどうか判定することができる。判定ステップ 374 は、デザインチェッカ 210 によって実行される全てのステップと並列に実行することが可能であることに留意されたい。言い換えれば、ユーザが電子ドキュメ

50

ントと対話する場合、ステップ341後のあらゆるステップを停止する、または保留することができる。判定ステップ374に対する照会が否定的であった場合、「NO」分岐を辿り、プロセスは、ステップ341に戻る。判定ステップ374に対する照会が肯定的であった場合、「YES」分岐を辿ってルーチン377に進む。

【0071】

ルーチン377において、ドキュメントに対して実行されるべきチェックのリスト216内の各チェックが、それぞれ問題チェッカ212によって実行される。ルーチン377のさらなる詳細は、図6に関連して以下に説明する。

【0072】

ステップ380において、デザインチェッカ210は、問題チェッカ212によるチェックの実行に応答してパブリケーション問題のリスト102を構成すること、またはそのリスト102に問題を追加することができる。次に、プロセスは、ステップ323に戻る。

10

【0073】

次に、図4を参照すると、この図は、電子ドキュメント112においてイベントを潜在的なパブリケーション問題に対してマッピングするための例示的なサブメソッドまたはルーチン362を示している。ステップ403は、このサブメソッドにおける最初のステップであって、各イベントに関してルックアップテーブル500の中のイベントに関連する全ての起こり得るパブリケーション問題を明らかにすることを含む。

【0074】

次に、判定ステップ406において、評価されている現在のイベントに関して、評価されている現在のイベントが、既に識別されている (previously-identified)パブリケーション問題を修正するかどうか判定される。判定ステップ406に対する照会が肯定的である場合、「YES」分岐を辿ってステップ409に進み、デザインチェッカ210が、電子ドキュメントに対して実行されるべきチェックのリスト216からチェック218を除去する。判定ステップ406に対する照会が否定的であった場合、「NO」分岐を辿ってステップ412に進み、現在の電子ドキュメント112に関して実行されるべきチェック218のリスト216に問題が追加される。

20

【0075】

次に、判定ステップ415において、デザインチェッカ210により、全てのイベント、および関連する問題が精査されたかどうか判定される。判定ステップ415に対する照会が否定的であった場合、「NO」分岐を辿って判定ステップ406に戻る。判定ステップ415に対する照会が肯定的であった場合、「YES」分岐を辿ってステップ418に進み、プロセスは、ステップ362で図3(B)に戻る。

30

【0076】

次に、図5を参照すると、この図は、デザインチェッカ210によって使用される例示的なイベント-チェック-ルックアップテーブル(event-to-check-lookup table)500を示している。テーブルの1つの軸は、イベントタイプのリストを含み、他方、他の軸上で、潜在的なパブリケーション問題のリストを含む。デザインチェッカ210は、マッピングテーブル500をくまなく調べながら、個々のチェックがそれぞれの問題チェッカ212に送られるチェックのリストを生成する。テーブル500が示す通り、イベントタイプ番号1のようなイベントタイプが、そのイベントに関連する複数の問題を有することが可能である。

40

【0077】

例えば、イベントタイプが電子ドキュメント112におけるページ上のオブジェクトの動きである場合、起こり得るパブリケーション問題には、以下が含まれることが可能である。すなわち、オブジェクトが、ページから外れるように動かされている可能性があること、オブジェクトが、ページ最上部により大きな余白が存在するように動かされている可能性があること、動かされたオブジェクトが、ページ上の別のオブジェクトを隠しているか、または覆っている可能性があること、ならびにページ上のオブジェクトの動きに関連

50

する可能性があるその他の同様な問題である。本発明は、前述したイベントタイプ例に限定されない。電子ドキュメントにおける情報の編集または生成に関わる他のイベントタイプも、本発明の範囲および趣旨を超えるものではない。

【0078】

次に、図6を参照すると、この図は、電子ドキュメントに対してチェックを実行するための例示的なサブメソッド(submethod)377を示している。判定ステップ603は、システムがアイドルであるかどうか判定されるサブメソッド377の最初のステップである。ステップ374と同様に、判定ステップ603は、デザインチェッカ210によって実行される全てのステップと並列に実行することが可能であることに留意されたい。言い換えれば、ユーザが電子ドキュメントと対話する場合、サブメソッド377におけるステップ603の後のあらゆるステップを、停止するか、または保留することが可能である。判定ステップ603に対する照会が否定的であった場合、「NO」分岐を辿ってステップ615に進み、プロセスは、図3Bのステップ341に戻る。判定ステップ603に対する照会が肯定的であった場合、「YES」分岐を辿って判定ステップ606に進む。

10

【0079】

判定ステップ606において、問題チェッカ212が、問題リスト102内に現在の問題が存在するかどうか判定する。判定ステップ606に対する照会が肯定的であった場合、「YES」分岐を辿って判定ステップ616に進む。判定ステップ606に対する照会が否定的であった場合、「NO」分岐を辿って判定ステップ609に進む。

【0080】

判定ステップ609において、ボリュームチェッカ212が、イベントにより評価されているパブリケーション問題が生じさせられているかどうか判定する。判定ステップ609に対する照会が肯定的であった場合、「YES」分岐を辿って612に進み、問題が、パブリケーション問題リスト102に組み込むために問題チェッカ212によってフラグで示される。判定ステップ609に対する照会が否定的であった場合、「NO」分岐を辿って判定ステップ627に進む。

20

【0081】

判定ステップ616において、問題チェッカ212は、現在の問題に関連するイベントにより問題が修正されたかどうか判定する。判定ステップ616に対する照会が肯定的である場合、「YES」分岐を辿ってステップ621に進み、問題チェッカ212が、リスト102からその問題を除去するようにデザインチェッカ310に通知する。判定ステップ616に対する照会が否定的であった場合、「NO」分岐を辿ってステップ624に進み、問題チェッカ212が、パブリケーション問題リスト102に残る問題をフラグで示す。

30

【0082】

判定ステップ627において、デザインチェッカ210は、現在の電子ドキュメントに対してさらなるチェックを実行する必要があるかどうか判定する。判定ステップ627に対する照会が肯定的であった場合、「YES」分岐を辿って判定ステップ603に戻る。判定ステップ627に対する照会が否定的であった場合、「NO」分岐を辿ってステップ628に進み、プロセスは、図3(B)のステップ362に戻る。

40

【0083】**その他の実施の形態**

前述した通り、図に示した本発明は、視覚表示だけを描いているが、ユーザと他の形態の通信も本発明の範囲および趣旨を超えるものではないことが、当業者には理解できよう。例えば、本発明は、サウンド、音声合成、触覚表示により、さらには匂いまたは味によってさえ潜在的なデザインパブリケーション問題の警報をユーザに行うことができる。同様に、電子ドキュメントに対するユーザアクションは、マウス入力またはキーボード入力だけでないアクションを含むことが可能である。ユーザアクションは、音声コマンド、ペンとスクリーンの活動、タッチスクリーン活動、および他の同様なユーザアクションをさらに含むことが可能である。

50

【0084】

本発明は、対話的であり、ドキュメントチェックプロセス中にユーザがいくらかのコントロールを有することを可能にするパブリケーションデザイン問題をチェックし、解決するためのシステムおよび方法を提供する。また、本発明は、ドキュメントが編集されている、または生成されている最中に起こり得るデザイン問題に関する継続的で、リアルタイムのフィードバックを提供するドキュメントチェックプロセスも提供することができる。

【0085】

さらに、本発明は、電子ドキュメントの全ての問題を明らかにし、電子ドキュメントに存在する可能性がある問題のタイプを識別するドキュメント検査プロセスも提供する。本発明のこのドキュメント検査プロセスは、ある繰り返し生じる問題の自動修正のためのオプションがユーザに用意されていることが可能であるようにパブリッシングプログラムと一体になっている。本発明のドキュメント検査プロセスは、電子ドキュメントが生成されている、または編集されている最中に使用することが簡単であり、管理することが容易である。

10

【0086】

以上は、本発明の例示的な実施形態だけに関連しており、特許請求の範囲によって定義される本発明の趣旨および範囲を逸脱することなく、実施形態に多数の変更を加えることができることを理解されたい。

【図面の簡単な説明】

【0087】

20

【図1A】通常のコンピュータアーキテクチャ内部の本発明の例示的なソフトウェア要素を示す図である。

【図1B】本発明の1つの例示的な実施形態による電子ドキュメントにおける潜在的なデザイン問題のリアルタイムリストを表示するユーザインターフェースを示す例示的なディスプレイ画面図である。

【図1C】本発明の1つの例示的な実施形態による選択された図1Bのリアルタイムリストのエントリを示す例示的なディスプレイ画面図である。

【図1D】本発明の1つの例示的な実施形態による図1Cのリアルタイムリストの最初のエントリの前の選択に従って拡大された潜在的なデザイン問題を有する電子ドキュメントに隣接する例示的なユーザインターフェースを示すディスプレイ画面図である。

30

【図1E】本発明の1つの例示的な実施形態によるリアルタイムリストの最初のエントリによって識別された問題が生成された後のリアルタイムリストを示すディスプレイ画面図である。

【図1F】本発明の1つの例示的な実施形態によるドキュメントにパブリケーションデザイン問題が存在するかどうか判定するようにドキュメントに対して実行されるチェックのリストを含むユーザインターフェースを示すディスプレイ画面図である。

【図1G】本発明の1つの例示的な実施形態による表示オプションと、評価されているページ範囲を含むユーザインターフェースを示すディスプレイ画面図である。

【図2A】本発明のシステムおよび方法を提供するように対話するさらなるソフトウェア要素を示す図である。

40

【図2B】本発明の1つの例示的な実施形態によってモニタされるイベントデータの例示的な要素を示す図である。

【図2C】本発明の1つの例示的な実施形態によるマッピングプロセスの後に明らかにされることが可能なチェックの例示的なリストを示す図である。

【図3A】本発明の1つの例示的な実施形態によるパブリケーションデザイン問題をチェックし、解決するための方法を示す図である。

【図3B】本発明の1つの例示的な実施形態によるパブリケーションデザイン問題をチェックし、解決するための方法を示す図である。

【図4】本発明の1つの例示的な実施形態による潜在的なデザインパブリケーション問題に対してイベントをマッピングするためのサブメソッドを示す図である。

50

【図5】本発明の1つの例示的な実施形態による例示的なイベント - チェック - ルックアップテーブルを示す図である。

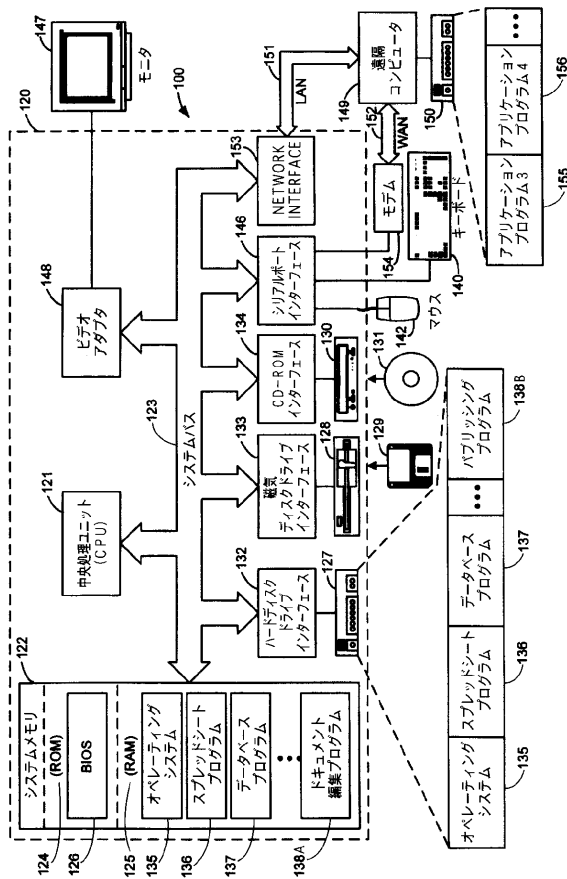
【図6】本発明の1つの例示的な実施形態による起こり得るデザインパブリケーション問題に関してドキュメントに対するチェックを実行するためのサブメソッドを示す図である。

【符号の説明】

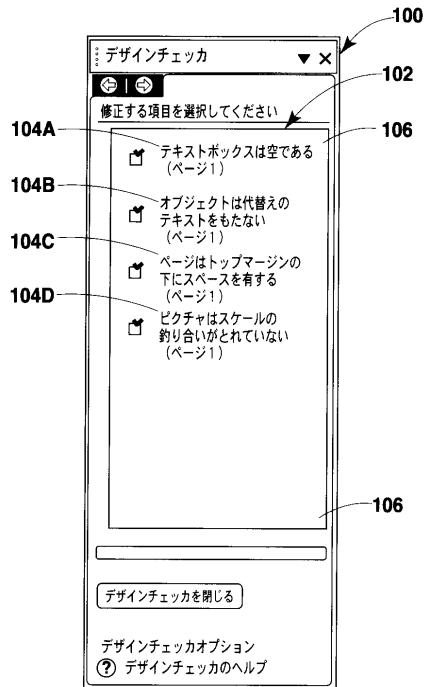
【0088】

- 100 ユーザーインターフェース
- 102 パブリケーション問題リスト
- 200 システム
- 202 イベントマネージャ
- 204 ユーザアクション
- 205 A、205 B、205 C 選択された問題
- 206 イベント待ち行列
- 208 モジュール
- 210 デザインチェッカ
- 212 A、212 B、212 C 問題チェッカ
- 214 A、214 B、214 C 問題フィードバック

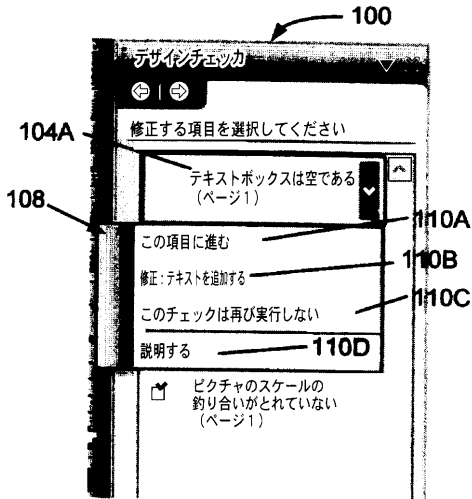
【図1A】



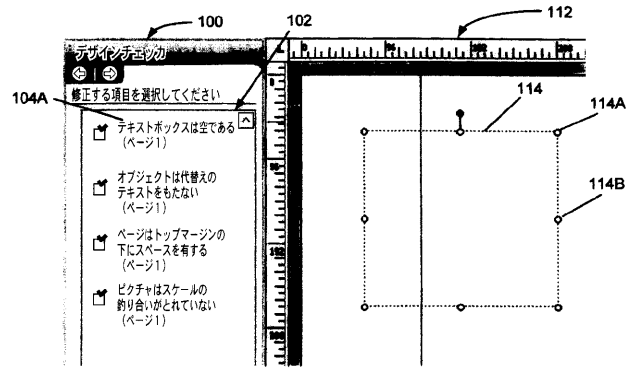
【図1B】



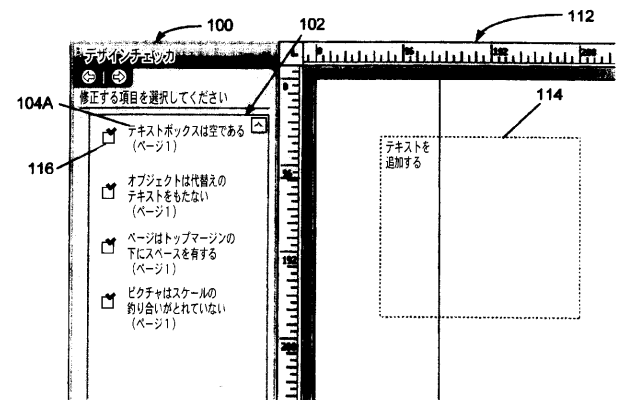
【図1C】



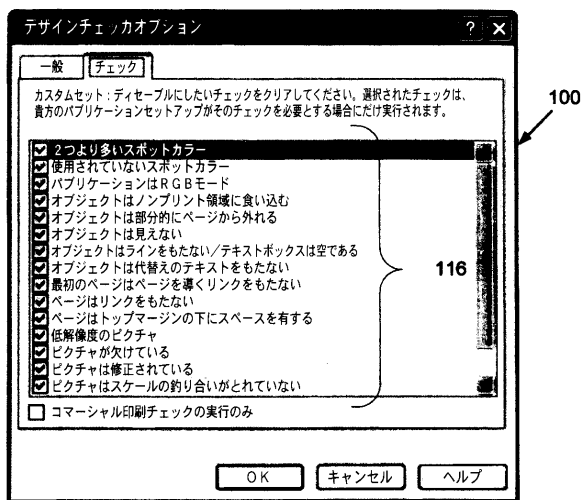
【図1D】



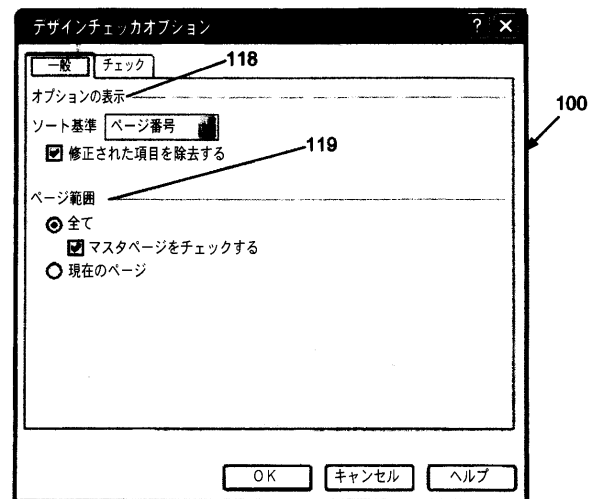
【図1E】



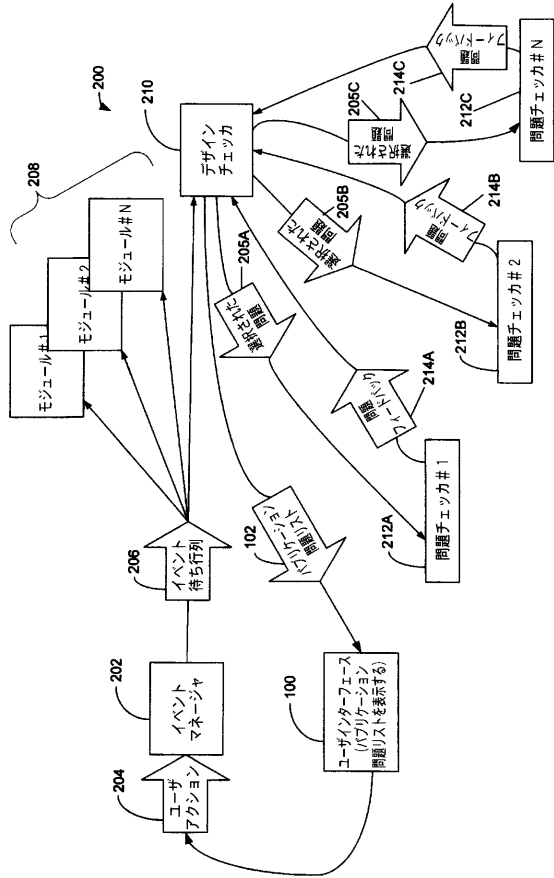
【図1F】



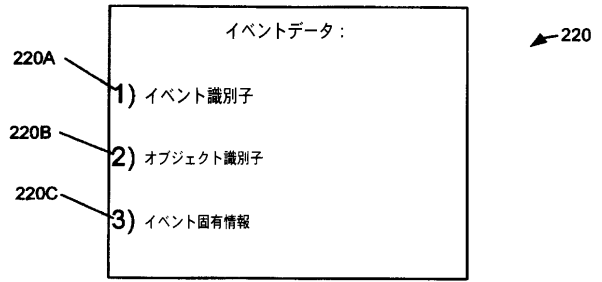
【図1G】



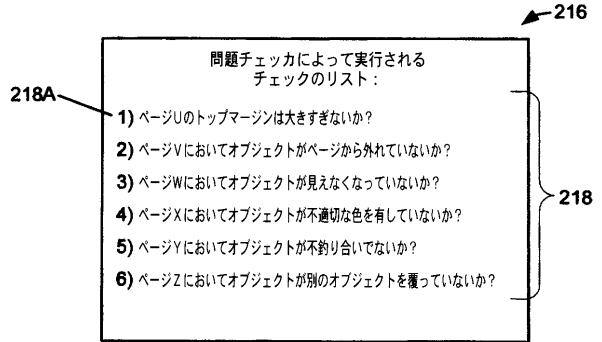
【図 2 A】



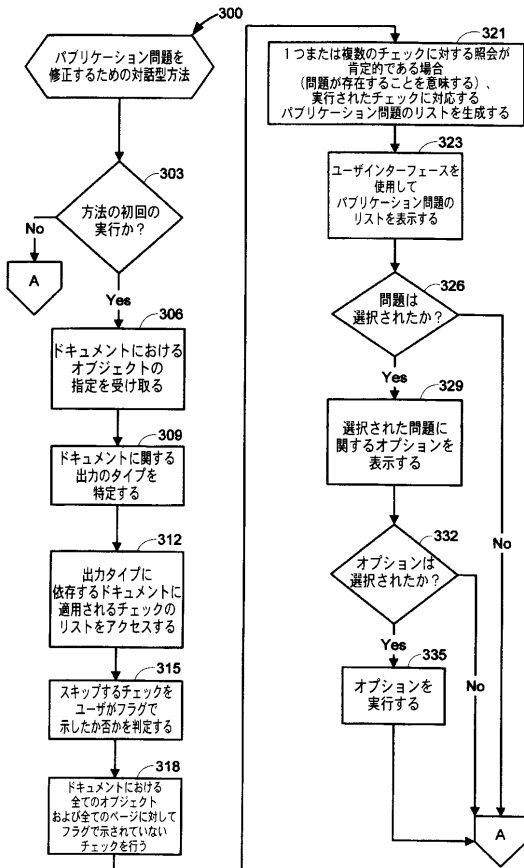
【図 2 B】



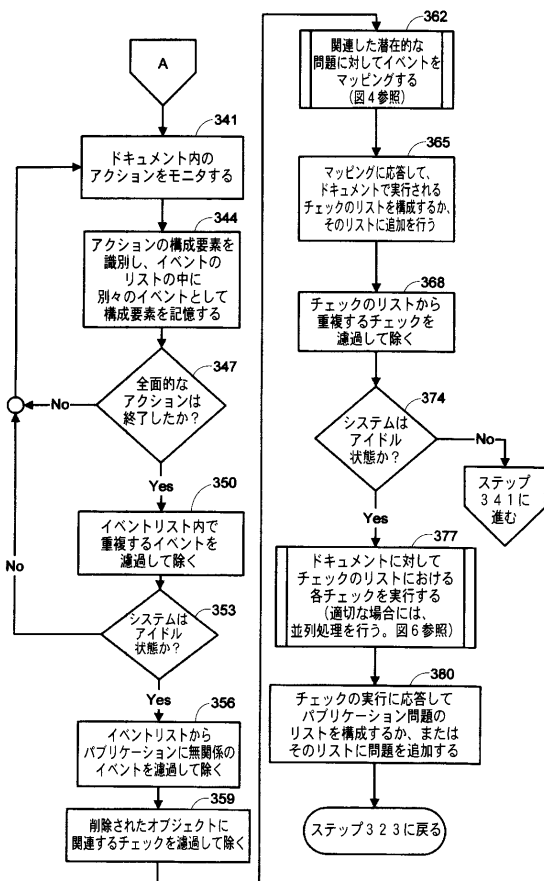
【図 2 C】



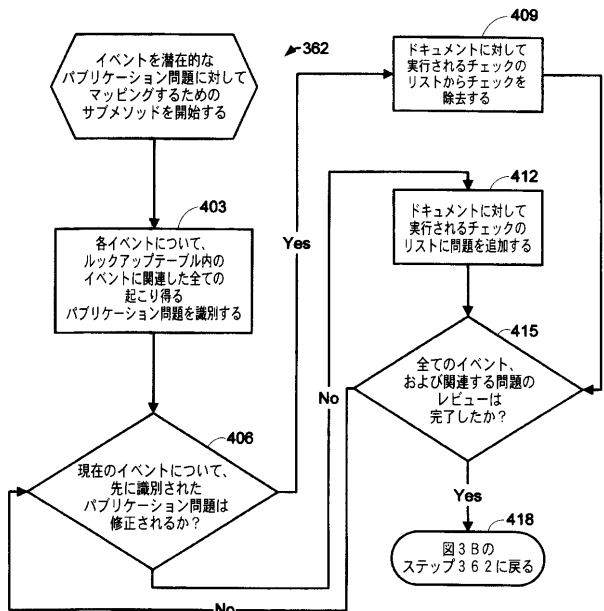
【図 3 A】



【図 3 B】



【 図 4 】

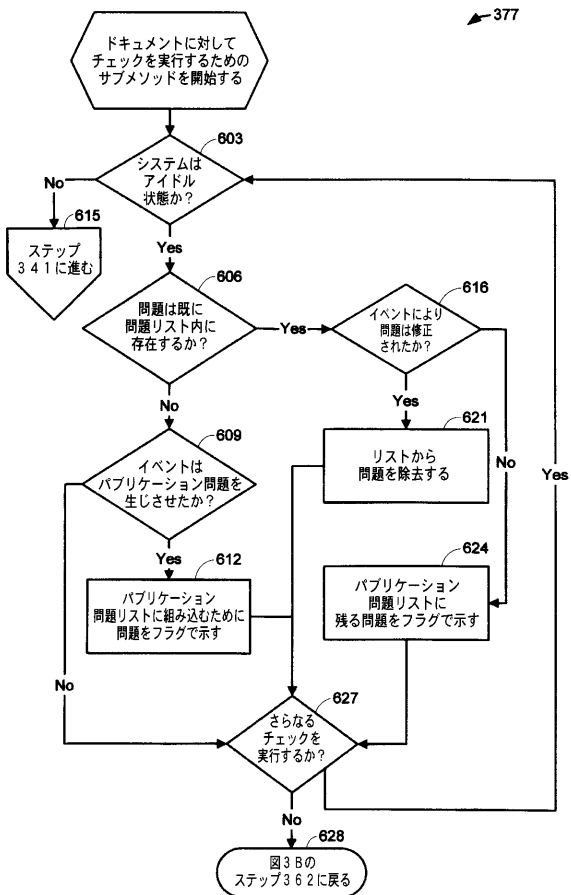


【 図 5 】

500

チェックする イベント-ルックアップテーブル	イベント タイプ # 1	イベント タイプ # 2	イベント タイプ # N
起り得るパブリケーション問題# 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
起り得るパブリケーション問題# 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
起り得るパブリケーション問題# N	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

【 図 6 】



フロントページの続き

- (72)発明者 ニティン チョブラ
アメリカ合衆国 98004 ワシントン州 ベルビュー ノースイースト 36 プレイス 1
1380 アpartment ビー236
- (72)発明者 ベンジャミン エリック ロス
アメリカ合衆国 98115 ワシントン州 シアトル 23 アベニュー ノースイースト 7
018
- (72)発明者 マシュー ウッド
アメリカ合衆国 98052 - 6399 ワシントン州 レッドモンド ワン マイクロソフト
ウェイ(番地なし)
- (72)発明者 ドナルド エルマン
アメリカ合衆国 98121 ワシントン州 シアトル ファースト アベニュー 2801 ナ
ンバー409
- (72)発明者 ジョシュ ポロック
アメリカ合衆国 98033 ワシントン州 カークランド カークランド サークル 720
ビー203
- Fターム(参考) 5B009 NA00 NC00 QA00 SA03 TB03