(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5508953号
(P5508953)

(45)発行日　平成26年6月4日(2014.6.4)　(24)登録日　平成26年3月28日(2014.3.28)

(51)Int.Cl.		F I		
G06F 17/27	(2006.01)	G06F	17/27	E
G06F 17/21	(2006.01)	G06F	17/21	550A
		G06F	17/21	592A

請求項の数 8　(全 17 頁)

(21)出願番号　特願2010-146739 (P2010-146739)
(22)出願日　平成22年6月28日 (2010.6.28)
(65)公開番号　特開2012-8965 (P2012-8965A)
(43)公開日　平成24年1月12日 (2012.1.12)
審査請求日　平成25年1月29日 (2013.1.29)

(73)特許権者　000233055
株式会社日立ソリューションズ
東京都品川区東品川四丁目１２番７号
(74)代理人　100091096
弁理士　平木　祐輔
(74)代理人　100102576
弁理士　渡辺　敏章
(74)代理人　100101063
弁理士　松丸　秀和
(72)発明者　松本　俊子
東京都品川区東品川四丁目１２番７号　日立ソフトウエアエンジニアリング株式会社内

審査官　長　由紀子

最終頁に続く

(54)【発明の名称】文書処理装置及びプログラム

(57)【特許請求の範囲】

【請求項１】

単語の区切りに空白文字が存在する言語で作成された文章を含む文書に所定の処理を実行し、処理結果を管理する文書処理装置であって、

前記処理結果を登録するための登録用データベースと、

前記文章を構成する単語と空白文字の有無の情報を含む元文書を読み込んで得られる元文書情報を格納する文書データベースと、

前記言語における文字或いは単語と記号との間の表記ルールに基づいて、前記文章に含まれる隣接する二つの文字が同一の単語に含まれるか否か判定する単語区切り判定処理部と、

前記単語区切り判定処理部による処理結果を表示装置に表示する表示処理部と、

入力指示に応答して、前記単語区切り判定処理部による処理結果を前記登録用データベースに登録する登録処理部と、

前記文章から前記空白文字を無視してメタデータを抽出する処理を行うメタデータ抽出処理部と、を有し、

前記単語区切り判定処理部は、前記隣接する二つの文字が同一の単語に含まれるか否かについての判定結果を用いて、前記抽出したメタデータに前記空白文字を再挿入し、

前記登録処理部は、前記空白文字が再挿入されたメタデータを、前記単語区切り判定処理部による処理結果として前記登録用データベースに登録することを特徴とする文書処理装置。

【請求項２】
請求項１において、
前記単語区切り判定処理部は、前記表記ルールに基づいた判定処理によって前記隣接する二つの文字が同一の単語に含まれると断定できないときに、前記元文書情報を参照して、前記隣接する二つの文字間に空白文字を挟むか否か確認し、当該確認結果に基づいて、前記文章中の隣接する二つの文字が同じ単語に含まれるかどうか判定することを特徴とする文書処理装置。
【請求項３】
請求項２において、
さらに、文字列の識別子と文字列オブジェクトの特徴を含む文字列オブジェクト情報と、前記文字列の各文字がどの文字列に含まれるかを示す文字情報と、を格納するメモリを含み、
前記単語区切り判定処理部は、前記元文書情報を参照しても前記隣接する二つの文字が同一の単語に含まれると断定できないときに、前記文字列オブジェクト情報及び前記文字情報を参照して、前記隣接する二つの文字が同じ文字列オブジェクトに含まれるかどうか確認し、当該確認結果に基づいて、前記文章中の隣接する二つの文字が同じ単語に含まれるかどうか判定することを特徴とする文書処理装置。
【請求項４】
請求項３において、
前記文字列オブジェクト情報は、さらに、それぞれの文字列の位置情報を含み、
前記単語区切り判定処理部は、前記文字列オブジェクト情報及び前記文字情報を参照しても前記隣接する二つの文字が同一の単語に含まれると断定できないときに、前記それぞれの文字列の位置情報を参照して、前記文字列オブジェクトの間隔が空いているか近接しているか確認し、当該確認結果に基づいて、前記文章中の隣接する二つの文字が同じ単語に含まれるかどうか判定することを特徴とする文書処理装置。
【請求項５】
コンピュータを、単語の区切りに空白文字が存在する言語で作成された文章を含む文書に所定の処理を実行し、処理結果を管理する文書処理装置として機能させるためのプログラムであって、
前記コンピュータには、前記文章を構成する単語と空白文字の有無の情報を含む元文書を読み込んで得られる元文書情報を格納する文書データベースが接続されており、
前記コンピュータに、前記言語における文字或いは単語と記号との間の表記ルールに基づいて、前記文章に含まれる隣接する二つの文字が同一の単語に含まれるか否か判定する単語区切り判定処理を実行させるためのプログラムコードと、
前記単語区切り判定処理の結果を表示装置に表示させるためのプログラムコードと、
前記コンピュータに、入力指示に応答して、前記単語区切り判定処理の結果を登録用データベースに登録させるためのプログラムコードと、
前記コンピュータに、前記文章から前記空白文字を無視してメタデータを抽出するメタデータ抽出処理を実行させるプログラムコードと、を有し、
前記単語区切り判定処理を実行するためのプログラムコードは、前記コンピュータに、前記隣接する二つの文字が同一の単語に含まれるか否かについての判定結果を用いて、前記抽出したメタデータに前記空白文字を再挿入する処理を実行させるためのプログラムコードを含み、
前記登録用データベースに登録させるためのプログラムコードは、前記コンピュータに、前記空白文字が再挿入されたメタデータを、前記単語区切り判定処理部による処理結果として前記登録用データベースに登録させるためのプログラムコードを含むことを特徴とするプログラム。
【請求項６】
請求項５において、
前記単語区切り判定処理を実行するためのプログラムコードは、前記表記ルールに基づ

いた判定処理によって前記隣接する二つの文字が同一の単語に含まれると断定できないときに、前記元文書情報を参照して、前記隣接する二つの文字間に空白文字を挟むか否か確認し、当該確認結果に基づいて、前記文章中の隣接する二つの文字が同じ単語に含まれるかどうか判定する処理を、前記コンピュータに実行させるためのプログラムコードを含むことを特徴とするプログラム。

【請求項７】

請求項６において、

前記コンピュータは、さらに、文字列の識別子と文字列オブジェクトの特徴を含む文字列オブジェクト情報と、前記文字列の各文字がどの文字列に含まれるかを示す文字情報と、を格納するメモリを含み、

前記単語区切り判定処理を実行するためのプログラムコードは、前記元文書情報を参照しても前記隣接する二つの文字が同一の単語に含まれると断定できないときに、前記文字列オブジェクト情報及び前記文字情報を参照して、前記隣接する二つの文字が同じ文字列オブジェクトに含まれるかどうか確認し、当該確認結果に基づいて、前記文章中の隣接する二つの文字が同じ単語に含まれるかどうか判定する処理を、前記コンピュータに実行させるためのプログラムコードを含むことを特徴とするプログラム。

【請求項８】

請求項７において、

前記文字列オブジェクト情報は、さらに、それぞれの文字列の位置情報を含み、

前記単語区切り判定処理を実行するためのプログラムコードは、前記文字列オブジェクト情報及び前記文字情報を参照しても前記隣接する二つの文字が同一の単語に含まれると断定できないときに、前記それぞれの文字列の位置情報を参照して、前記文字列オブジェクトの間隔が空いているか近接しているか確認し、当該確認結果に基づいて、前記文章中の隣接する二つの文字が同じ単語に含まれるかどうか判定する処理を、前記コンピュータに実行させることを特徴とするプログラム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、文書処理装置及びプログラムに関し、例えば、大量に存在する業務文書のファイルデータを効率的に管理するための技術に関する。

【背景技術】

【０００２】

近年、組織内の文書を効率的に取扱うための技術に対する要求が高まっている。例えば、日本版ＳＯＸ法（金融商品取引法）の施行に伴い、企業の営業活動における証憑の管理ニーズが高まっている。また例えば、企業内の情報、その中でも特にリレーショナルデータベースに格納されない（定型でない）文書データが急激に増大しており、情報爆発と呼ばれる現象が起きている。このような状況のもとで、文書をタイトル・作成日・作成者などのメタデータで管理・検索したいというニーズも高まっている。例えば営業文書であれば、文書名・顧客名・作成日・注文番号などの業務ＩＤで検索を行うことができれば、内部統制の監査において必要な文書を迅速に探し出すことができる。また、設計文書であれば、文書名・作成元部署・作成日・製品コードなどで検索を行うことができれば、技術情報の有効活用に効果がある。さらに、クレーム・不具合情報の記録文書であれば、発生日・対策日・製品名・被害額・部品名などで検索を行うことができれば、類似の不具合の発生時における迅速な対応に効果がある。また、業務規定・通達などの文書であれば、文書の種別・作成日・実施期間などで検索を行うことができれば、ルールに沿った効率的な業務遂行に効果がある。

【０００３】

定型でない文書を解析してメタデータを自動的に取得する技術は多く提案されている（下記特許文献及び非特許文献参照）。これらの技術では、文書中に記載されている内容を読み込む際、空白文字を無視した処理を行うことが効果的である。なぜなら、文字の配置

を整えるための空白文字の影響を受けずにメタデータを抽出できるためである。例えば図１Ａに示すように、センタリングを実現するために空白文字を挿入したり、図１Ｂに示すように、空白文字やタブ文字を挿入することで配置を整えたりといったことが行われる。図１Ａおよび図１Ｂにおいて、１００に示すような「□（四角）」は全角空白文字を、１０１に示すような「・（ドット）」は半角空白文字を、１０２に示すような「→（矢印）」はタブ文字を示す。このような空白文字の影響を受けずにメタデータを抽出するためには、文字データの読み込みの時点で空白文字を読み飛ばすことが有効である。

【先行技術文献】

【特許文献】

【０００４】

【特許文献１】特開平１１－１８４８９４号公報

【特許文献２】特許第３４２５８３４号公報

【特許文献３】特許第３４２５４０８号公報

【非特許文献】

【０００５】

【非特許文献１】勝山・直井・武部，ビジネス文書を対象としたキーワード自動抽出技術，FUJITSU，49，5，pp.404-409 (1998-09)

【非特許文献２】Ishitani，Y.，Document Transformation System from Papers to XML Data Based on Pivot XML Document Method，Proceedings of the Seventh International Conference on Document Analysis and Recognition (2003)

【発明の概要】

【発明が解決しようとする課題】

【０００６】

既存技術を用いて、英語等の言語による文書であって、各単語の区切りに空白文字が存在する文書からメタデータを抽出する場合、単語ごとに空白文字が挿入された形での出力を行う必要がある。

【０００７】

しかしながら、上述したように、文字データの読み込みの時点では空白文字を読み飛ばしているため、図２に示すように、全ての単語がつながった形での出力となってしまう。日本語の場合と異なり、英語等の場合では、全ての単語がつながった形での出力は著しく可読性を欠く。このことの解決策としては、まず、メタデータの単語がつながった状態での抽出を行い、その後で、メタデータ抽出元の文書（以下、「元文書」と呼ぶことがある）を参照して「単語の区切り」を調べ、抽出されたメタデータに空白文字を再挿入することが考えられる。

【０００８】

ところが、英語等の文書の場合に、各単語の区切りを確実に検出し、空白文字を確実に再挿入することは困難である。

【０００９】

本発明はこのような状況に鑑みてなされたものであり、文章を構成する単語の区切りに空白文字が存在する言語によって作成された文書において、各単語の区切りを確実に検出し、空白文字を区切りに確実に再挿入することができる技術を提供するものである。

【課題を解決するための手段】

【００１０】

上記課題を解決するために、本発明は、単語の区切りに空白文字が存在する言語で作成された文章を含む文書に所定の処理を実行し、処理結果を管理する文書処理装置を提供する。当該文書処理装置では、言語における文字或いは単語と記号との間の表記ルールに基づいて、文章に含まれる隣接する二つの文字が同一の単語に含まれるか否か判定（単語区切り判定処理）する。そして、単語区切り判定処理による処理結果を表示装置に表示すると共に、入力指示に応答して、単語区切り判定処理部による処理結果を登録用データベースに登録する。

【００１１】
さらに、単語区切り判定処理の前に、文章を構成する単語と空白文字の有無の情報を含む元文書を読み込んで得られる元文書情報を格納する文書データベースの文章から空白文字を無視してメタデータを抽出する処理を行う。そして、単語区切り判定処理では、隣接する二つの文字が同一の単語に含まれるか否かについての判定結果を用いて、抽出したメタデータに前記空白文字を再挿入し、空白文字が再挿入されたメタデータを、単語区切り判定処理の結果として登録用データベースに登録する。
【００１２】
単語区切り判定処理では、表記ルールに基づいた判定処理によって隣接する二つの文字が同一の単語に含まれると断定できないときに、元文書情報を参照して、隣接する二つの文字間に空白文字を挟むか否か確認し、当該確認結果に基づいて、文章中の隣接する二つの文字が同じ単語に含まれるかどうか判定する。
【００１３】
本発明による文書処理装置は、メモリに、文字列の識別子と文字列オブジェクトの特徴を含む文字列オブジェクト情報と、文字列の各文字がどの文字列に含まれるかを示す文字情報と、を格納している。そして、単語区切り判定処理部では、元文書情報を参照しても隣接する二つの文字が同一の単語に含まれると断定できないときに、文字列オブジェクト情報及び文字情報を参照して、隣接する二つの文字が同じ文字列オブジェクトに含まれるかどうか確認し、当該確認結果に基づいて、文章中の隣接する二つの文字が同じ単語に含まれるかどうか判定する。
【００１４】
文字列オブジェクト情報は、さらに、それぞれの文字列の位置情報を含んでいる。そして、単語区切り判定処理部では、文字列オブジェクト情報及び文字情報を参照しも隣接する二つの文字が同一の単語に含まれると断定できないときに、それぞれの文字列の位置情報を参照して、文字列オブジェクトの間隔が空いているか近接しているか確認し、当該確認結果に基づいて、文章中の隣接する二つの文字が同じ単語に含まれるかどうか判定する。
【００１５】
さらなる本発明の特徴は、以下本発明を実施するための形態および添付図面によって明らかになるものである。
【発明の効果】
【００１６】
本発明によれば、定型でない言語（英語等）の文書からメタデータを抽出する際、単語ごとに空白文字で区切られた形で出力することができるようになる。
【図面の簡単な説明】
【００１７】
【図１】空白文字を無視した読み込み処理が適切である文書の例を示す図である。
【図２】英語の文書からメタデータを抽出する際、全ての単語がつながった形での出力となってしまう例を示す図である。
【図３】「単語の区切り」を調べるための直感的な方法を示す図である。
【図４】図３に示す直感的な方法では「単語の区切り」を正確に調べられない文書の例を示す図である。
【図５】本発明の実施形態による業務文書処理装置の概略構成を示す機能ブロック図である。
【図６】文書情報、文字情報および文字列オブジェクト情報のデータ構造例を示す図である。
【図７】線画情報、画像情報およびメタデータ情報のデータ構造例を示す図である。
【図８】業務文書処理装置において実行される処理の全体を説明するためのフローチャートである。
【図９】空白文字再挿入処理部において実行される詳細動作を説明するためのフローチャ

ートである。
【図１０】単語区切り判定処理部において実行される詳細動作を説明するためのフローチャートである。
【図１１】結果表示処理部で表示される確認画面例を示す図である。
【図１２】結果表示処理部で表示される確認画面例を示す図である。
【発明を実施するための形態】
【００１８】
以下、添付図面を参照して本発明の実施形態について説明する。ただし、本実施形態は本発明を実現するための一例に過ぎず、本発明の技術的範囲を限定するものではないことに注意すべきである。また、各図において共通の構成については同一の参照番号が付されている。なお、以下では、英語で作成された文書を例にして本発明の実施形態を説明するが、英語以外の、文書を構成する単語の区切りに空白文字が存在する言語（例えば、日本語、中国語、韓国語等以外の言語であって、フランス語、ドイツ語、イタリア語、ロシア語等の言語が含まれる）で作成された文書にも適用可能である。
【００１９】
＜序論＞
（１）抽出されたメタデータに空白文字を再挿入する際に、直感的には、下記の方法により元文書から「単語の区切り」を調べられるように思われる。
【００２０】
ｉ）元文書から「単語の区切り」を調べるための直感的な方法１
プレーンテキストやワード等の編集用アプリケーションで作成した元文書には空白文字があるため、空白文字の箇所をデータとして保持しておき、図３Ａに示すように、元文書で空白文字があった場所を「単語の区切り」とする。
【００２１】
ｉｉ）元文書から「単語の区切り」を調べるための直感的な方法２
オフィス文書作成ソフトウェアの保存形式、閲覧ソフトウェアの保存形式、印刷用データファイルなどでは、文字列オブジェクトの形で文書記載内容を保持している（文字列オブジェクトは、一つまたは複数の文字を含む）。そこで、元文書のデータ構造を参照し、図３Ｂに示すように、文字が保持されるオブジェクトが切り替わる時点を「単語の区切り」とする。
【００２２】
ｉｉｉ）元文書から「単語の区切り」を調べるための直感的な方法３
図３Ｃに示すように、文字の位置が離れている場所を「単語の区切り」とする。
【００２３】
ｉｖ）元文書から「単語の区切り」を調べるための直感的な方法４
図３Ｄ及びＥに示すように、単語辞書を用意して文字列とのマッチングを行い、辞書登録語の境界を「単語の区切り」とする。
【００２４】
しかし、実際には、上記いずれの方法によっても、単語の区切りを正しく調べることができない。
【００２５】
例えば、閲覧・保存の目的に特化したソフトウェアの保存形式、ＰＤＦの変換後のデータや、印刷用データファイルのようなファイルフォーマットでは、文書の外観のみ再現できれば十分である。このため、英単語の区切りを表現することは、文字の位置を離して描画するだけでも可能であり、必ずしも空白文字をデータとして保持する必要はない。図４Ａに示すように単語の区切りでも文書ファイル上は空白文字が保持されていないことがあるため、上記の方法１では正しく調べられない場合がある。
【００２６】
また、図４Ｂに示すようにオブジェクトが単語単位ではないことがあるため、上記の方法２では正しく調べられない場合がある。

【００２７】
さらに、図４Ｃに示すようにプロポーショナルフォントが利用されて文字が等間隔で並んでいなかったり、図４Ｄに示すようにカーニングが行われて文字間隔が変動したりすることがあるため、上記の方法３では正しく調べられない場合がある。
【００２８】
また、図４Ｅに示すように辞書に登録されていない単語があったり、図４Ｆ及びＧに示すように複数のマッチング可能性があったりすることがあるため、上記の方法４では正しく調べられない場合がある。
【００２９】
（２）そこで、発明者は、英語の表記ルール上、表１に示すような隣接する文字同士については、同じ単語に含まれるかどうか判断できることに着目した。この表記ルールに従って同じ単語に含まれるかどうか判断する処理を最初に実行することが重要である。そして、表記ルールに従って処理しても同じ単語に含まれるものか判断できない場合に、後述のように、元文書データに空白文字を挟んでいるか、同じ文字列オブジェクトに含まれるか、文字列オブジェクトの間隔が空いているかについて判断する。ここで、表記ルールとは、オックスフォードルールやシカゴマニュアル等、文章を作成する上で規範的なスタイルを規定したルールをいうものである。
【００３０】
【表１】

英語の表記ルールから同じ単語に含まれるかどうか判断できる例

項番	前の文字	後ろの文字	単語	例(該当箇所を下線で示す)
1	カンマ	英文字	違う	Japan, Co., Ltd.
2	英文字、数字、ピリオド	カンマ	同じ	Japan, Co., Ltd.
3	英文字	ピリオド	同じ	Japan, Co., Ltd.
4	コロン	英文字	違う	From: Toshiko Matsumoto
5	英文字、数字	コロン	同じ	From: Toshiko Matsumoto
6	セミコロン	英文字、数字	違う	…of RFID; makes shipping more accurate …
7	英文字、数字	セミコロン	同じ	…of RFID; makes shipping more accurate …
8	クエスチョンマーク	英文字、数字	違う	… How you received it? I have been…
9	英文字、数字	クエスチョンマーク	同じ	… How you received it? I have been…
10	コロン	スラッシュ	同じ	http://www.hitachisoft.jp
11	開き括弧	英文字、数字	同じ	… SaaS (Software as a Service) businesses …
12	閉じ括弧	英文字、数字	違う	… SaaS (Software as a Service) businesses …
13	英文字、数字、ピリオド	開き括弧	違う	… SaaS (Software as a Service) businesses …
14	英文字、数字	閉じ括弧	同じ	… SaaS (Software as a Service) businesses …
15	開きクォート	英文字、数字	同じ	… the “social innovation business” that fuses …
16	閉じクォート	英文字、数字	違う	… the “social innovation business” that fuses …
17	英文字、数字	開きクォート	違う	… the “social innovation business” that fuses …
18	英文字、数字	閉じクォート	同じ	… the “social innovation business” that fuses …

【００３１】
また、発明者は、サンプルファイルを調査した結果、英語の表記ルールからは同じ単語に含まれるかどうか判断できない場合には、間に空白文字を挟んでいるような文字同士については、異なる単語に含まれていると判断できることに着目した。
【００３２】

さらに、発明者は、サンプルファイルを調査した結果、英語の表記ルールからは同じ単語に含まれるかどうか判断できず、間に空白文字を挟んでいない場合には、同じ文字列オブジェクトに含まれる文字同士については、同じ単語に含まれていると判断できることに着目した。

【００３３】

また、発明者は、サンプルファイルを調査した結果、英語の表記ルールからは同じ単語に含まれるかどうか判断できず、間に空白文字を挟んでおらず、異なる文字列オブジェクトに含まれる場合には、文字列オブジェクトの間隔が空いているならば異なる単語に含まれており、近接しているならば同じ単語に含まれていると判断できることに着目した。

【００３４】

以上の着目点に従った処理を実行する、本発明の実施形態による業務文書処理装置について説明する。

【００３５】

＜装置構成＞

図５は、本実施形態による業務文書処理装置（文書処理装置）の概略的構成を示す機能ブロック図である。業務文書処理装置５０は、データを表示するための表示装置５００と、文書ＤＢ５０１と、表示されたデータに対してメニューを選択するなどの操作を行うためのキーボード５０２と、マウスなどのポインティングデバイス５０３と、必要な演算処理及び制御処理などを行う中央処理装置５０４と、中央処理装置５０４での処理に必要なプログラムを格納するプログラムメモリ５０５と、中央処理装置５０４での処理に必要なデータを格納するデータメモリ５０６と、文字列オブジェクトを処理した結果であるメタデータを格納するメタデータＤＢ５３０と、を有している。

【００３６】

中央処理装置５０４は、空白文字を無視してメタデータを文書から抽出する処理を行う空白文字無視メタデータ抽出処理部５０７と、空白文字が無視されて抽出されたメタデータに空白文字を再度挿入する処理を行う空白文字再挿入処理部５０８と、空白挿入処理されたメタデータを表示する結果表示処理部５０９と、を含んでいる。本実施形態では、各処理部はコンピュータ構成の少なくとも一部によって実現される。つまり、空白文字無視メタデータ抽出処理部５０７と、空白文字再挿入処理部５０８と、結果表示処理部５０９は、いずれもコンピュータ上で実行されるプログラムの機能の一部として実現される。従って、各処理部は各処理プログラムと読み替えることが可能である。これらのプログラムは、プログラムメモリ５０５に格納されている。なお、各処理部は、モジュール化することにより実現しても良い。

【００３７】

空白文字再挿入処理部５０８は、単語区切り判定処理部５１０を有している。また、単語区切り判定処理部５１０は、表１の表記ルールを参照しながら文字列に空白文字が存在するか確認する処理を実行する英語表記ルール確認処理部５１１と、文書情報５１５、文字情報５１６、及び文字列オブジェクト情報５１７を参照しながら空白文字列が存在するか確認する処理を実行する空白文字有無確認処理部５１２と、文字列オブジェクト情報５１７を参照してオブジェクトとして同一のものが含まれているか確認する処理を行う文字列オブジェクト同一性確認処理部５１３と、文字列オブジェクトの座標情報から２つの文字列オブジェクトの間隔（ピクセル数）を確認する処理を実行する文字列オブジェクト間隔確認処理部５１４と、を含んでいる。

【００３８】

データメモリ５０６は、文書情報５１５と、文字情報５１６と、文字列オブジェクト情報５１７と、線画情報５１８と、画像情報５１９と、メタデータ情報５２０と、を格納している。

【００３９】

＜データメモリ内の情報のデータ構造例＞

図６は、データメモリ５０６に含まれる文書情報５１５、文字情報５１６、及び文字列

オブジェクト情報５１７のデータ構造を示す図である。

【００４０】

文書情報５１５は、構成項目として、文書ＩＤ６００、文字データ６０１、文字列オブジェクトデータ６０２、線画データ６０３、及び画像データ６０４を含んでいる。ここで、文字データ６０１は、文書内に記載された文字の情報であり、文字情報構造体の配列の形で保持される。文字列オブジェクトデータ６０２は、文書内に記載された文字が含まれる文字列オブジェクトの情報であり、文字列オブジェクト情報構造体の配列の形で保持される。線画データ６０３は、ページ内に記載された線画（線分）の情報であり、線画情報構造体の配列の形で保持される。画像データ６０４は、ページ内に記載された画像の情報であり、画像情報構造体の配列の形で保持される。

【００４１】

文字情報５１６は、構成項目として、文字ＩＤ６０５、文字コード６０６、文字列オブジェクトＩＤ６０７、左下座標６０８、右上座標６０９、フォントサイズ６１０、フォント種類６１１を含んでいる。ここで、文字ＩＤ６０５は、各文字に一意に割り当てられたＩＤである。文字コード６０６は、その文字の内容を示す。文字列オブジェクトＩＤ６０７は、その文字が含まれる文字列オブジェクトのＩＤ６１２である。左下座標６０８は、文字の外接矩形の左下の頂点がページのどこに位置するかを示す座標である。右上座標６０９は、文字の外接矩形の右上の頂点がページのどこに位置するかを示す座標である。フォントサイズ６１０は、その文字の大きさである。フォント種類６１１はその文字のフォントの種類である。

【００４２】

文字列オブジェクト情報５１７は、構成項目として、文字列ＩＤ６１２、左下座標６１３、右上座標６１４を含んでいる。ここで、文字列ＩＤ６１２は、各文字列オブジェクトに一意に割り当てられたＩＤである。左下座標６１３は、文字列の外接矩形の左下の頂点がページのどこに位置するかを示す座標である。右上座標６１４は、文字の外接矩形の右上の頂点がページのどこに位置するかを示す座標である。

【００４３】

図７は、データメモリ５０６に含まれる線画情報１５８、画像情報５１９、及びメタデータ情報５２０のデータ構造を示す図である。

【００４４】

線画情報５１８は、例えば罫線等に関する情報であり、構成項目として、線画ＩＤ７００、始点座標７０１、終点座標７０２を含んでいる。ここで、線画ＩＤ７００は、各線画に一意に割り当てられたＩＤである。始点座標７０１は、直線の一方の端がページのどこに位置するかを示す座標である。終点座標７０２は、直線のもう一方の端がページのどこに位置するかを示す座標である。

【００４５】

画像情報５１９は、例えば印鑑の印面画像や挿絵等に関する情報であり、構成項目として、画像ＩＤ７０３、左下座標７０４、右上座標７０５、ピクセルデータ７０６を含んでいる。ここで、画像ＩＤ７０３は、各画像に一意に割り当てられたＩＤである。左下座標７０４は、画像の外接矩形の左下の頂点がページのどこに位置するかを示す座標である。右上座標７０５は、画像の外接矩形の右上の頂点がページのどこに位置するかを示す座標である。ピクセルデータ７０６は、画像のイメージを画像形式で保持する。

【００４６】

メタデータ情報５２０は、文書のメタデータを保持するためのデータ構造であり、構成項目として、メタデータＩＤ７０７、メタデータ種別７０８、文字データ７０９、空白文字再挿入フラグ７１０を含んでいる。ここで、メタデータＩＤ７０７は、各メタデータに一意に割り当てられたＩＤである。メタデータ種類７０８は、どの種類のメタデータであるかを保持する。文字データ７０９は、そのメタデータに含まれる文字それぞれについての文字ＩＤ６０５を配列の形で保持する。空白文字再挿入フラグ７１０は、再挿入処理の結果に対応して、文字データ７０９の間に空白文字が再挿入されるべきかどうかを配列の

形で保持する。
【００４７】
＜メタデータ抽出処理の概要＞
続いて、以上のように構成された本実施形態の業務文書処理装置において行われる処理について説明する。図８は、業務文書処理装置において行われるメタデータ抽出・登録処理の概要を説明するためのフローチャートである。
【００４８】
図８において、まず、ＯＣＲ等を用いて、文書情報の読み込み処理が行われる（ステップ８００）。読み込んだ結果は文書情報５１５に保持される。この段階ではまだメタデータは抽出されておらず、従ってメタデータ情報５２０は１要素も存在しない。
【００４９】
次に、空白文字無視メタデータ抽出処理部５０７は、空白文字を無視したメタデータ抽出を行う（ステップ８０１）。ここでの処理については、空白文字無視メタデータ抽出処理部５０７を用いて行われるものであり、非特許文献１、非特許文献２、特許文献１、特許文献２、特許文献３等に記載されている既存技術で行うことができる。よって、詳しい説明は割愛する。当該処理の結果は、メタデータ情報５２０に格納される。この時点では空白文字の再挿入処理は行われていないため、空白文字再挿入処理部５０８は、空白文字再挿入フラグ７１０（図７参照）の全ての配列要素についてｆａｌｓｅで初期化する。
【００５０】
続いて、空白文字再挿入処理部５０８は、空白文字をメタデータへ再挿入する処理を行う（ステップ８０２）。この処理の詳細については、図９を用いて詳細に説明する。
【００５１】
その後、結果表示部５０９は、処理結果を表示装置５００に表示する（ステップ８０３）。ここで表示される画面例については、図１１及び１２を用いて詳細に説明する。
【００５２】
そして、最後に、メタデータ登録処理部（図示せず）が、利用者の指示に応答して、空白文字が再挿入されたメタデータを処理結果としてメタデータＤＢ５３０に登録する。
【００５３】
＜空白文字の再挿入処理の詳細＞
図９は、図８の空白文字をメタデータへ再挿入する処理（ステップ８０２）の詳細について説明するためのフローチャートである。
【００５４】
まず、空白文字再挿入処理部５０８は、メタデータのインデックスm_idxを１で初期化し（ステップ９００）、メタデータ情報の数がm_idx以上であるか調べる（ステップ９０１）。m_idx未満である場合は処理を終了させる。一方、m_idx以上である場合は、空白文字再挿入処理部５０８は、文字のインデックスc_idxを１で初期化し（ステップ９０２）、m_idx番目のメタデータ情報は文字データ７０９としてc_idx+1以上の文字を持つか調べる（ステップ９０３）。c_idx未満である場合は、空白文字再挿入処理部５０８は、そのメタデータに含まれる全ての隣接する文字同士についての処理を終えているため、m_idxを１だけインクリメントして（ステップ９０４）、ステップ９０１に処理を戻す。
【００５５】
ステップ９０３でc_idx+1以上である場合は、空白文字再挿入処理部５０８は、c_idx番目とc_idx+1番目の文字は同じ単語に含まれるかどうか調べる（ステップ９０５）。この処理は、単語区切り判定処理部５１０で行われるものであり、図１０において詳細に説明する。
【００５６】
ステップ９０５の結果、違う単語であると判定された場合には、空白文字再挿入処理部５０８は、メタデータ情報の空白文字再挿入フラグ７１０のc_idx番目の要素にtrueを設定する（ステップ９０６）。その後、空白文字再挿入処理部５０８は、c_idxを１だけインクリメントして（ステップ９０７）、処理をステップ９０３に戻す。

【００５７】
＜同一単語に含まれるか否かについての判定処理＞
図１０は、図９の二つの文字が同じ単語に含まれるかどうか判定する処理（ステップ９０５）の詳細について説明するためのフローチャートである。
【００５８】
まず、空白文字再挿入処理部５０８は、英語表記ルール確認処理部５１１を用いて、英語表記ルールから二つの文字が同一単語に含まれるか判断を行う（ステップ１０００）。より具体的には、英語表記ルール確認処理部５１１が、二つの文字の関係として表１の中に該当する項目（ルール）があればそれに従って判断し、該当する項目がなければ断定できないとする。
【００５９】
断定できないと判断された場合（ステップ１００１）、空白文字再挿入処理部５０８は、空白文字有無確認処理部５１２を用いて、空白の有無からの判断を行う（ステップ１００２）。具体的には、空白文字有無確認処理部５１２が、文書情報に含まれる文字データ６０１それぞれについて、文字コード６０６から空白文字かどうかを調べ、左下座標６０８と右上座標６０９から二つの文字の間に挟まれているかどうかを調べる。空白文字で間に挟まれているものが見付かったら、二つの文字は異なる文字列に含まれると判断し、そのような文字がなければ断定できないとする。
【００６０】
断定できないと判断された場合（ステップ１００３）、空白文字再挿入処理部５０８は、文字列オブジェクト同一性確認処理部５１３を用いて、文字列オブジェクトの同一性からの判断を行う（ステップ１００４）。具体的には、文字列オブジェクト同一性確認処理部５１３が、二つの文字の文字列オブジェクトＩＤ６０７が同一であるかどうかを調べる。同一であれば二つの文字は同じ文字列に含まれると判断し、異なれば断定できないとする。
【００６１】
断定できないと判断された場合（ステップ１００５）、空白文字再挿入処理部５０８は、文字列オブジェクト間隔確認処理部５１４を用いて、文字列オブジェクトの間隔からの判断を行う（ステップ１００６）。具体的には、文字列オブジェクト間隔確認処理部５１４が、二つの文字の文字列オブジェクトＩＤ６０７と同じ値の文字列ＩＤ６１２を持つ文字列オブジェクト情報を探し、それらの左下座標６１３と右上座標６１４から間隔を調べる。例えば、間隔が所定値以上空いていれば異なる文字列、近接していれば（所定値未満であれば）同じ文字列と判断するようにすれば良い。
【００６２】
以上のように、表記ルールに従った処理を最初に実行し、それでも判断できない場合に、空白文字有無確認処理、文字列オブジェクト同一性確認処理、文字列オブジェクト間隔確認処理を順番に行うようにする。表記ルールによる判断を最初に実行するのは、様々な文書から英単語の区切りを正確に判断することができるからである。例えば、図４Ｂに示したような文書の記載内容の場合、最初に「隣接する文字同士が同じ文字列オブジェクトに含まれるかどうか」（文字列オブジェクト同一性確認処理を用いて）判断してしまうと、単語「Characters」の最後の「s」と単語「are」の最初の「a」は同じ文字列オブジェクトに含まれることから、同じ単語に含まれると誤判断してしまう。また、先に「間に空白文字を挟むかどうか」（空白文字有無確認処理を用いて）判断を行い、間に空白文字を挟まない場合についてのみ「隣接する文字同士が同じ文字列オブジェクトに含まれるかどうか」（文字列オブジェクト同一性確認処理を用いて）判断を行うという順番にすることで、このような誤判断を防いで正確に判断することができる。
【００６３】
＜結果表示画面例＞
図１１及び１２は、図８の結果の表示処理（ステップ８０３）において結果が表示される画面例を示す図である。

【００６４】
図１１では、メタデータ情報５２０に保持している内容について並べて表示が行われる（１１００）。ここでは、空白文字再挿入フラグ７１０の結果に基づき、文字データ７０９の内容を近接させたり空白文字を挟んだりして、単語ごとに空白文字で区切られた形で並べて表示する。このうち、選択したメタデータ（１１０１）について、単語区切りについての詳細な情報の表示を要求するユーザ操作（ボタン１１０２押下）されると、図１２に示す画面が表示される。
【００６５】
図１２では、メタデータ文字列が表示されると共に、着目している隣接する文字同士について下線による強調表示が行われる（１２００）。また、着目している隣接する文字同士のみを取り出して表示される（１２０１）。この文字同士について、図１０のフローチャートの処理で判断した結果が示される（１２０２）。図１０の処理では判断に成功したらその後の処理は行われないため、行われなかった部分についてはその旨（図中では、「－－－」）表示される（１２０３）。また、図１０の処理での最終的な判断結果が示される（１２０４）。着目している隣接する文字の変更を要求するユーザ操作を受付け（１２０５）、表示１２００～１２０４が更新される。なお、閉じるボタン１２０６が押下されると、図１２の画面は図１１の画面に切り替わる。
【００６６】
＜変形例＞
なお、本明細書では、文書のメタデータを単語ごとに空白文字で区切られた形で出力するための場合について述べた。英語の文書から全文データを単語ごとに空白で区切られた形で出力するための処理についても同様である。
【００６７】
また、文字情報については、図６で挙げた他にも、文字の外周の色（ＲＧＢ成分）、文字の塗りつぶしの色（ＲＧＢ成分）、斜体であるかどうか、太字であるかどうか、文字の背景色（ＲＧＢ成分）など様々な書式指定情報を持つことが考えられる。この場合も、非特許文献１、非特許文献２、特許文献１、特許文献２、特許文献３に記載されている既存技術で、空白文字を無視したメタデータ抽出処理を行うことができるので、本明細書で述べた場合と同様に取扱えば良い。
【００６８】
また、線画について本明細書では線分の場合について述べたが、矩形・多角形・ベジエ曲線・円弧などやその組み合わせが文書に含まれていることが考えられる。さらに、線の色、太さ、パターン（実線や点線など）、塗りつぶしの色など様々な書式指定情報を持つことが考えられる。この場合も、非特許文献１、非特許文献２、特許文献１、特許文献２、特許文献３に記載されている既存技術で、空白文字を無視したメタデータ抽出処理を行うことができるので、本明細書で述べた場合と同様に取扱えば良い。
【００６９】
また、本明細書では図１２で下線表示による強調表示を行う例について述べたが、強調表示の形態はこれに限らない。太字、文字色による強調表示なども可能である。
【００７０】
＜まとめ＞
本発明の実施形態では、英語の表記ルールを用いて、英語文書中の隣接する二つの文字が同じ単語に含まれるかどうか判定することを特徴とする。ここで、表記ルールとは、オックスフォードルールやシカゴマニュアル等、文章を作成する上で規範的なスタイルを規定したルールをいうものである。このようにすることにより、英語特有の表記方法に則った空白の有無を判断することができるようになる。なお、実施形態では、英語を例にして説明しているが、表記方法が特殊で、単語と単語の間に空白文字が存在する言語であればどのような言語にも本発明は適用することが可能である。
【００７１】
そして、表記ルールに従って判断しても空白の有無について断定できない場合に、二つ

の文字の間に空白文字を挟むかどうかという元文書の情報（読み込む文書データに含まれる空白文字についての情報）に基づいて、英語文書中の隣接する二つの文字が同じ単語に含まれるかどうか判定する。また、元文書も情報に基づいて判断しても空白の有無について断定的な判断ができない場合に、同じ文字列オブジェクトに含まれるかどうかという文字列オブジェクトの情報に基づいて、英語文書中の隣接する二つの文字が同じ単語に含まれるかどうか判定する。さらに、文字列オブジェクトの情報に基づいて判断しても空白の有無について断定的な判断が出来ない場合に、文字列オブジェクトの間隔が空いているか近接しているかの情報（各文字情報が有する座標情報から文字間の距離がどの位離れているかの情報）に基づいて、英語文書中の隣接する二つの文字が同じ単語に含まれるかどうか判定する。このように、表記ルールを用いても空白の有無について断定できない場合に初めて、他の方法によって空白文字の有無について判断することにより、また、このような順番で空白の判断することにより、より正確に空白の有無を判断することが可能となる。つまり、上述したように、最初に「隣接する文字同士が同じ文字列オブジェクトに含まれるかどうか」を用いて判断してしまうと、例えば、単語「Characters」の最後の「s」と単語「are」の最初の「a」は同じ文字列オブジェクトに含まれることから、同じ単語に含まれると誤判断してしまう。よって、先に「間に空白文字を挟むかどうか」を用いて判断を行い、間に空白文字を挟まない場合についてのみ「隣接する文字同士が同じ文字列オブジェクトに含まれるかどうか」を用いて判断を行うという順番にすることで、このような誤判断を防いで正確に判断することができるようになる。

【００７２】

なお、本発明は、実施形態の機能を実現するソフトウェアのプログラムコードによっても実現できる。この場合、プログラムコードを記録した記憶媒体をシステム或は装置に提供し、そのシステム或は装置のコンピュータ（又はＣＰＵやＭＰＵ）が記憶媒体に格納されたプログラムコードを読み出す。この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコード自体、及びそれを記憶した記憶媒体は本発明を構成することになる。このようなプログラムコードを供給するための記憶媒体としては、例えば、フレキシブルディスク、ＣＤ－ＲＯＭ、ＤＶＤ－ＲＯＭ、ハードディスク、光ディスク、光磁気ディスク、ＣＤ－Ｒ、磁気テープ、不揮発性のメモリカード、ＲＯＭなどが用いられる。

【００７３】

また、プログラムコードの指示に基づき、コンピュータ上で稼動しているＯＳ（オペレーティングシステム）などが実際の処理の一部又は全部を行い、その処理によって前述した実施の形態の機能が実現されるようにしてもよい。さらに、記憶媒体から読み出されたプログラムコードが、コンピュータ上のメモリに書きこまれた後、そのプログラムコードの指示に基づき、コンピュータのＣＰＵなどが実際の処理の一部又は全部を行い、その処理によって前述した実施の形態の機能が実現されるようにしてもよい。

【００７４】

また、実施の形態の機能を実現するソフトウェアのプログラムコードを、ネットワークを介して配信することにより、それをシステム又は装置のハードディスクやメモリ等の記憶手段又はＣＤ－ＲＷ、ＣＤ－Ｒ等の記憶媒体に格納し、使用時にそのシステム又は装置のコンピュータ（又はＣＰＵやＭＰＵ）が当該記憶手段や当該記憶媒体に格納されたプログラムコードを読み出して実行するようにしても良い。

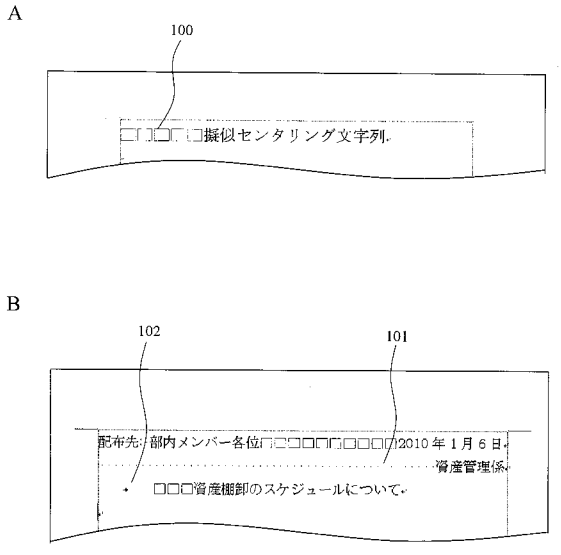
【符号の説明】

【００７５】

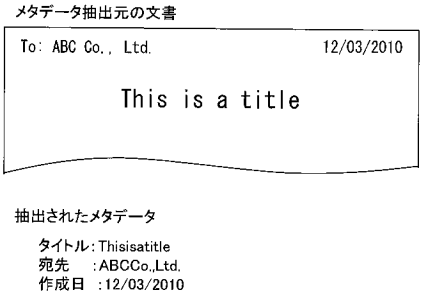
５００・・・表示装置
５０１・・・文書ＤＢ
５０２・・・キーボード
５０３・・・ポインティングデバイス
５０４・・・中央処理装置
５０５・・・プログラムメモリ

５０６・・・データメモリ
５３０・・・メタデータＤＢ

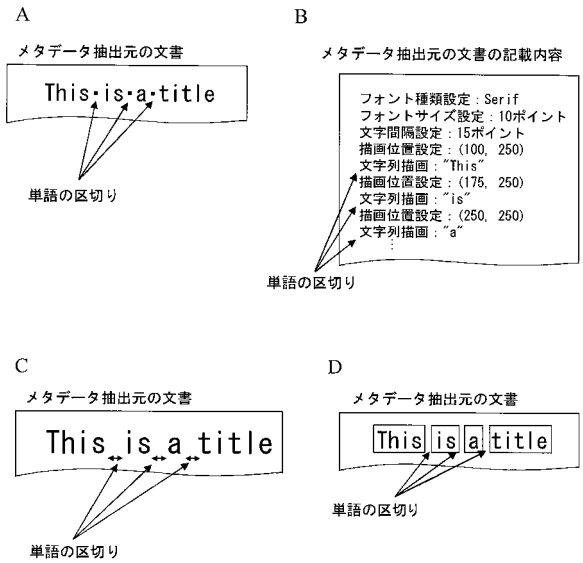
【図１】

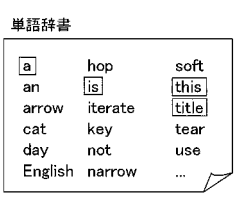


【図２】

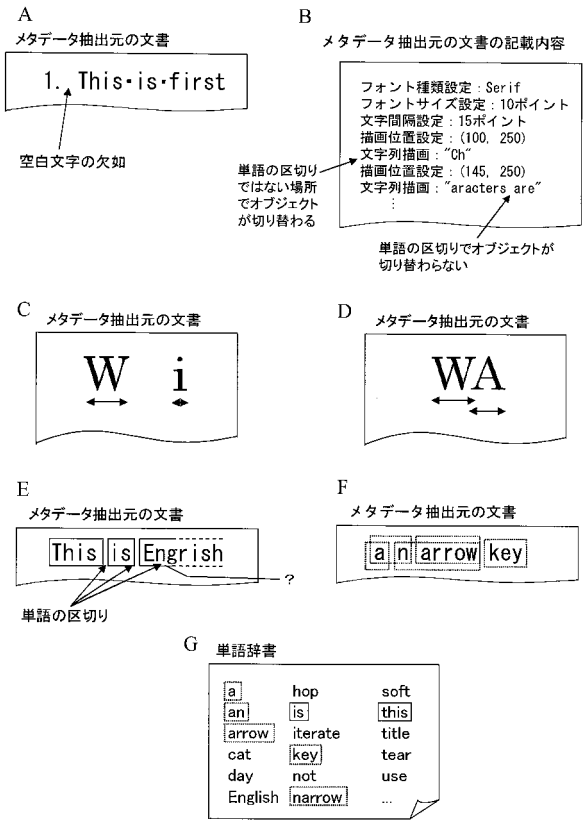


【図３】

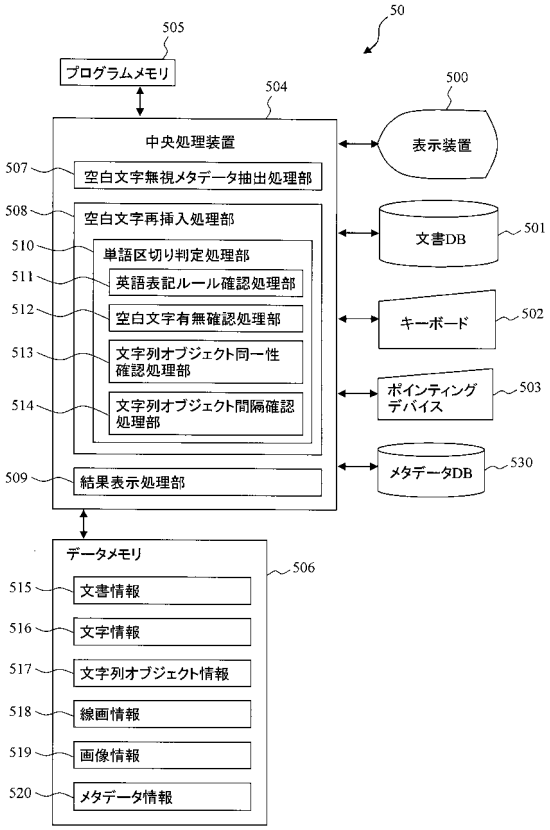




【図４】



【図５】



【図６】

文書情報

	メンバ名	値
600	文書ID	doc_0001
601	文字データ	[文字情報]
602	文字列オブジェクトデータ	[文字列オブジェクト情報]
603	線画データ	[線画情報]
604	画像データ	[画像情報]

文字情報

	メンバ名	値
605	文字ID	char_0001
606	文字コード	“a”
607	文字列オブジェクトID	str_01
608	左下座標	(500, 330)
609	右上座標	(510, 345)
610	フォントサイズ	10.5
611	フォント種類	Serif

文字列オブジェクト情報

	メンバ名	値
612	文字列ID	str_0001
613	左下座標	(500, 330)
614	右上座標	(580, 345)

【図７】

線画情報

	メンバ名	値
700	線画ID	line_0001
701	始端座標	(100, 250)
702	終端座標	(130, 250)

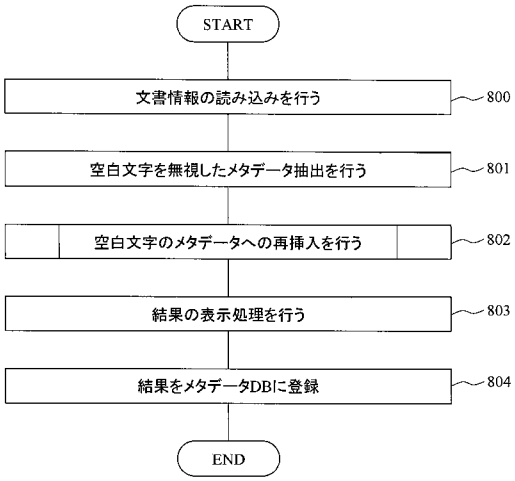
画像情報

	メンバ名	値
703	画像ID	img_0001
704	左下座標	(400, 200)
705	右上座標	(500, 550)
706	ピクセルデータ	画像イメージ

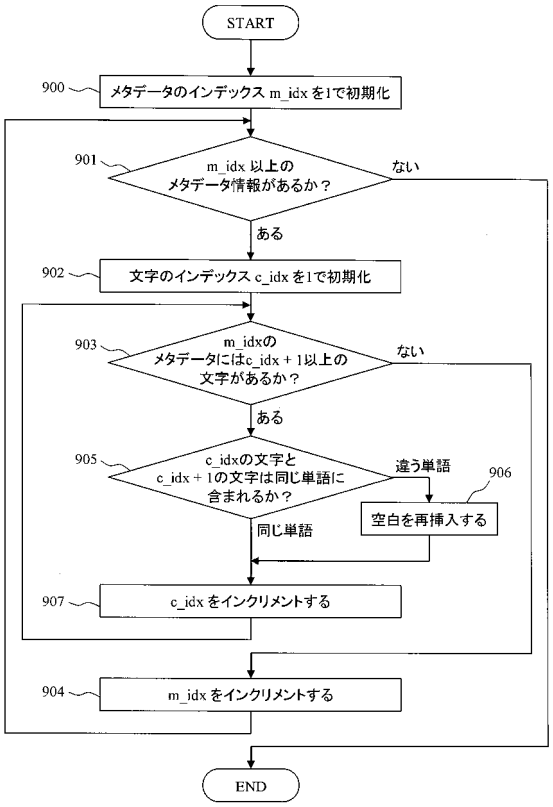
メタデータ情報

	メンバ名	値
707	メタデータID	meta_0001
708	メタデータ種別	"タイトル"
709	文字データ	[char_001, char_003, ...]
710	空白文字再挿入フラグ	[false, false, true, false, ...]

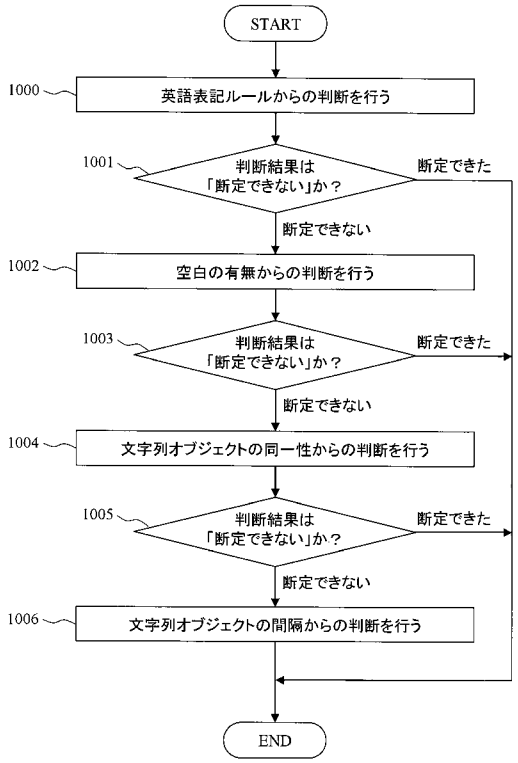
【図８】



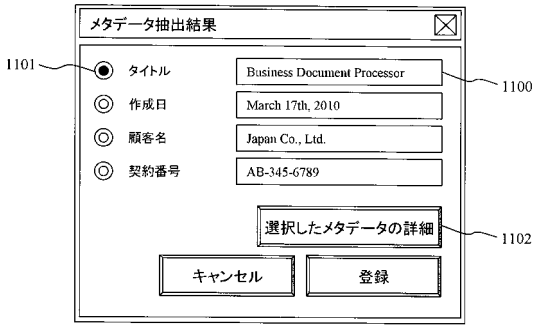
【図９】



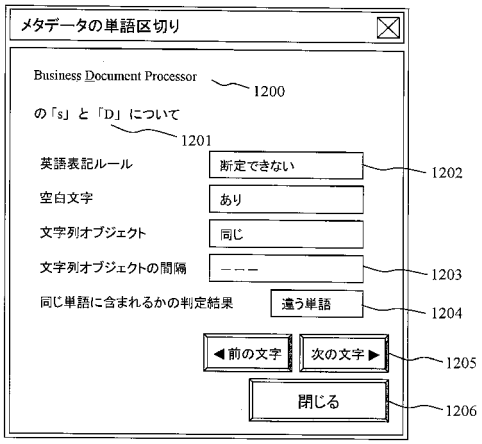
【図１０】



【図１１】



【図１２】



フロントページの続き

(56)参考文献 特開平０３－２０９５６４（ＪＰ，Ａ）
特開平０９－２３７３２０（ＪＰ，Ａ）
特開平０６－３４８９１１（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１７／２０－２８
Ｇ０６Ｆ　１７／３０
Ｇ０６Ｋ　９／００