(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4990431号
(P4990431)
(45)発行日　平成24年8月1日(2012.8.1)　(24)登録日　平成24年5月11日(2012.5.11)

(51)Int.Cl.　F I
G06F 17/30 (2006.01)　G06F 17/30 419A

請求項の数 6　(全 17 頁)

(21)出願番号	特願2000-519371 (P2000-519371)
(86)(22)出願日	平成10年10月21日 (1998.10.21)
(65)公表番号	特表2001-522097 (P2001-522097A)
(43)公表日	平成13年11月13日 (2001.11.13)
(86)国際出願番号	PCT/US1998/022331
(87)国際公開番号	WO1999/023581
(87)国際公開日	平成11年5月14日 (1999.5.14)
審査請求日	平成17年10月18日 (2005.10.18)
審判番号	不服2010-9702 (P2010-9702/J1)
審判請求日	平成22年5月7日 (2010.5.7)
(31)優先権主張番号	08/963,372
(32)優先日	平成9年11月3日 (1997.11.3)
(33)優先権主張国	米国(US)

(73)特許権者　501438485
ヤフー！　インコーポレイテッド
アメリカ合衆国　カリフォルニア州　９４０８９　サニーヴェイル　ファースト　アヴェニュー　７０１
(74)代理人　100059959
弁理士　中村　稔
(74)代理人　100067013
弁理士　大塚　文昭
(74)代理人　100082005
弁理士　熊倉　禎男
(74)代理人　100084009
弁理士　小川　信夫
(74)代理人　100086771
弁理士　西島　孝喜

最終頁に続く

(54)【発明の名称】階層状に複式化されたドキュメントからの情報検索

(57)【特許請求の範囲】

【請求項１】

検索要素を含みさらにドキュメント階層に組織されている、ドキュメント保管庫に記憶されたドキュメントの検索をコンピュータにより実行する方法であって、

前記コンピュータが、

一組の検索要素の記述を含む検索表現を検索エンジンに提供するステップと；

直接的に合致するドキュメント及び間接的に合致するドキュメントを検索するステップであって、直接的に合致するドキュメントとは前記検索表現に合致するドキュメントであり、また間接的に合致するドキュメントとは、前記階層において、前記間接的に合致するドキュメントをその先祖のドキュメントと組み合わせた時に、前記検索表現との合致が生じるドキュメントであり、

複数のドキュメントをリンクする階層内のパスとして表現されている複数のドキュメントから、前記検索表現と合致する少なくともひとつのドキュメントを含んだリストを生成するステップと；及び

検索結果として前記リストを出力するステップと

を実行することを特徴とする方法。

【請求項２】

前記検索要素が語(words)およびドキュメントであり、そして少なくともいくつかのテキストを含む、請求項１に記載した方法。

【請求項３】

前記コンピュータは、前記リストを生成するステップでの合致のために、前記検索表現の構成要素を要素索引で検索することを含む、請求項１に記載した方法。
【請求項４】
前記コンピュータは、階層トリーにおける各ドキュメントに、その階層トリーのどの分岐内でもドキュメントナンバーが連続するように、ドキュメントナンバーを指定することを含む、請求項１に記載した方法。
【請求項５】
前記検索表現は演算子およびオペランド（演算対象）を含む式であり、該オペランドは検索要素又はワイルドカードを含み、該演算子はＡＮＤ、ＯＲ、ＡＤＤ、又はＭＩＮＵＳを含む、請求項１に記載した方法。
【請求項６】
ドキュメントが明確に示された題目を表わしているストリングであり、前記階層は題目の階層である、請求項１に記載した方法。
【発明の詳細な説明】
【０００１】
この特許文書の説明部分は版権保護を受ける資料を含んでいる。版権所有者は、特許商標局の特許ファイルまたは記録となっている正確な形式での特許書類あるいは特許説明内容のいかなる人による複写再生に異議はとなえないが、その他の点では版権権利の全ては何でも留保している。
【０００２】
（技術分野）
本発明は電子的なドキュメントの格納と管理の分野に関する。より詳細には、本発明のひとつの実施態様は、複式ドキュメントを記憶しさらに記憶された複式ドキュメントを検索するシステムを提供する。
（背景技術）
情報は最近不足した有用品からあり余る有用品への変化を受けている。不足した有用品の時は、努力は有用品を集めることに集中されるが、しかるにあり余る有用品の時は、努力はそれをもっと価値あるものとするために有用品を選別することに集中されている。この現象の主な例は“インターネット”として知られるネットワークの全世界相互ネットワークへの成長からもたらされる情報の爆発的急増である。インターネットに接続されたネットワークおよびコンピュータは送信元ノードから送信先ノードへ確実にデータパケットを通すためにＴＣＰ／ＩＰ（Transport Control Protocol／Internet Protocol）を使用してデータを通過させる。高位レベルのいろいろなプロトコルがデジタルデータの対象を運ぶためにＴＣＰ／ＩＰの上位で使用され、それぞれのプロトコルは対象の性質に依存している。例えば、ｅ－メールはシンプルメールトランスポートプロトコル（ＳＭＴＰ）およびポストオフィスプロトコル３（ＰＯＰ３）を使用して運ばれ、一方ファイルはファイルトランスファープロトコル（ＦＴＰ）を使用して運ばれる。ハイパーテキストドキュメントおよびその関連する結果物はハイパーテキストトランスポートプロトコル（ＨＴＴＰ）を使用して運ばれる。
【０００３】
多くのハイパーテキストドキュメントが他のハイパーテキストドキュメントとリンクされるときには、集合として“ｗｅｂ”構造の形をとっており、これはＨＴＴＰを使って運ばれ得るハイパードキュメントの集合に対して“ワールドワイドウェブ”（しばしば“ＷＷＷ”または“Ｗｅｂ”と短縮される）の名前となっている。勿論、ハイパーリンクはドキュメントにおいてＨＴＴＰを使用して運ばれることを必要としない。実際には、いかなる対象も、ＨＴＴＰの要求に合致する限りにおいては、ＨＴＴＰを使用して運ばれ得る。
【０００４】
ＨＴＴＰの代表的な使用では、ブラウザがウェブサーバーにユニフォームリソースロケーター（ＵＲＬ）を送り、さらにウェブサーバーはブラウザが表示するためのハイパーテキストマークアップランゲージ（ＨＴＭＬ）ドキュメントを返送する。ブラウザはＨＴＴＰ

クライアントの１例であり、さらにそれが返送されたハイパーテキストドキュメントを表示し、そして返送されたドキュメントの中で参照される他のハイパーテキストドキュメントを選びかつ表示する機会を使用者に与えるが故に、そのように名付けられている。ウェブサーバーは、ＨＴＴＰクライアントから要求されたハイパーテキストドキュメントを返送するインターネットノードである。

【０００５】

ウェブサーバーの中には、静的なドキュメントを送達することに加えて、動的なドキュメントを返送できるものがある。静的なドキュメントは、ドキュメントに対する要求が行われる前からウェブサーバー上に存在し、さらにその要求に対しウェブサーバーは単に要求にもとづく静的ドキュメントを送出する。静的ページＵＲＬは代表的には“ホスト・サブドメイン・ドメイン・ＴＬＤ／パス／ファイル”または同様のものの形である。静的ページＵＲＬは、“ホスト・サブドメイン・ドメイン・ＴＬＤ”のドメイン名を持つマシン上のパス“／パス／”上で見出される“ファイル”と名付けられたドキュメントを参照する。実際のドメイン“ｗｗｗ．ｙａｈｏｏ．ｃｏｍ”は、“．ｃｏｍ”トップレベルドメイン（ＴＬＤ）におけるドメイン“ｙａｈｏｏ”で“ｗｗｗ”として示されるマシン（あるいは複数マシン）を参照する。これと反対に、動的なドキュメントは、ウェブサーバーが動的なドキュメントに対する要求として同一であると確認する特定されたＵＲＬを受取ると、ウェブサーバーによって作り出されるドキュメントである。

【０００６】

多くのウェブサーバーが、１人の人間あるいは法人によって管理されたリンクされたハイパーテキストドキュメントの集合を提供する“ウェブサイト”を動かしている。ウェブサイトは１人の人間あるいは法人によって管理されているから、ハイパーテキストドキュメントは、これに関連してしばしば“ウェブページ”と呼ばれ、共通する外見と固有の事項をもっている。特に品物やサービスを売る商業上の関係者によって築かれたウェブサイトの場合には、ウェブサイトを形作るハイパーリンクされたドキュメントは、関係者によって管理されていないページへは殆んどリンクしないかあるにしてもごくわずかであろう。語句の“ウェブサイト”および“ウェブページ”はしばしば同じ意味に使われているが、ここでは“ウェブページ”はウェブサイトの部分を作る単一のハイパーテキストドキュメントをさし示し、そして“ウェブサイト”は特定のトピックについてのサイトを共同して提供する単一の法人またはグループの法人によって管理され（つまり変えられ得る）ているひとつあるいはそれ以上のウェブページをさし示している。

【０００７】

数千万のインターネットのユーザーが彼等のウェブサーバーを通して役立たせている多くのサイトならびにページのすべてにとって、特定のトピックについての特定のページを見つけたりあるいは情報を見つけるところを決定することはしばしば困難となっている。何が使えるかと言う“正式”なリストするものが全く無い、なぜなら誰でも彼等のウェブサーバー上になんでも並べることが出来、そしてそれを正式な機関に報告する必要がなく、さらにウェブが非常に早く変化するからである。正式な“内容表”の欠如のためにウェブに索引をつけるいくつかのアプローチが提案されている。

【０００８】

ひとつのアプローチは、いたるところで見出される全てのウェブドキュメントに索引をつけることである。このアプローチは、たまにしか議論されないトピックまたは共通性のない姓あるいは名に関係するドキュメントを見つけるのには有益であるが、それはしばしば“当たり（hit）”の過度な数をひき起こしている。他のアプローチはウェブドキュメントを要約し、部類分けし、かつ要約を部類によって探し得るようにしている。

【０００９】

どちらの場合においても、典型的な検索エンジンが各々の予想されるドキュメントの検索項目を探し検索規準に合うドキュメントのリストを返送する。残念なことに、ドキュメントの相互関係から得られるべき情報は喪失される。上記のことから、ドキュメント間の相互関係を考慮に入れた改良型の検索システムが必要とされることが判る。

【００１０】
（発明の開示）
リンク全域で探すことによって、ドキュメント間の相互関係を考慮に入れた改良された検索システムが本発明の効力によって提供されている。本発明のひとつの実施態様において、ドキュメントはキーワードとトピック検索に対して使われる階層的なドキュメント保管庫に関係している。検索照会が階層に加えられ、階層は検索照会項目に直接的に合致するか、または照会項のすべてあるいは部分に合致する親ドキュメントから階層上では子ドキュメントであることによって、間接的に検索照会項目に合致するドキュメントを返送する。好ましい実施態様では、返送されるドキュメントは照会項の少なくともひとつの下位項目に直接に合致する。
【００１１】
本発明のひとつの利点はそれが階層的なデータの効率のよい格納庫を提供し、検索が階層でのデータ要素間の相互関係を考慮して遂行され得ることである。
本発明の本質と利点のさらなる理解が、詳細内容と添付図面の残りの部分を参照することによってここで実現されるであろう。
【００１２】
（発明を実施するための最良の形態）
本発明は、多くのドキュメントを含むドキュメント保管庫において、選ばれるドキュメントを探すための装置に関係する。本発明のひとつの実施態様による検索エンジンは、検索表現を受け、その検索表現にもとづき、ドキュメント保管庫の中で合致するドキュメントを検索し、そして合致するドキュメントあるいは合致するドキュメントの各々に関するリストのいづれかを返送する。検索表現が２つまたはそれ以上の下位項目による複雑な論理機能である場合は、検索エンジンは、いくつかの下位項目にただ間接的に合致するドキュメントを返送するであろう。例えば、検索表現は３つの下位項目の“ＡＮＤ”であるかも知れない。３つのすべての下位項目を含むドキュメントをただ単に返すかわりに、検索エンジンは、残りの下位項目がドキュメントから根元のノードにいたる階層上のパスに添うドキュメントのどこかで見つかったならば、下位項目のひとつまたは２つのみをもつドキュメントをも返送するであろう。いくつかの変形実施態様では、下位項目のすべてに対して間接的にのみ合致するドキュメントが返送されるが、しかし好ましい実施態様においては、返送されるドキュメントは少なくともひとつの下位項目に直接的に合致していなければならない。
【００１３】
本発明がある特定なタイプのドキュメントに関連してここで述べられるが、ここで述べられる本発明及び実施態様は、多くの他のドキュメントでも使用できると言うことが理解されるべきである。
この中の主な例で述べられるドキュメントは、検索データベースにおけるレコードである。検索データベースは部類およびサイトを照会する階層的な構造として組織されている。構造は自動的に作り出されるであろうが、Ｙａｈｏｏ！検索データベースとして知られる実施態様では、部類およびサイト照会は経験やサイトの意見者からの提案を使って編集スタッフにより階層の適当な位置に置かれている。
【００１４】
部類およびサイト照会は構造のノードとして集合的に照会される。いくつかの部類ノードは親ノードであり、そこではそれらは、親ノードによって表わされた部類のもっと詳細な下位部類ノードを表わしている他の部類ノード（子ノード）をさし示している。サイトノードは部類ノードからの子ノードである（とは言え、ある特定のサイトは、複数の部類にリストされ、いくつかのサブトリーにおける子ノードであるかも知れない）。
【００１５】
ここで、あるノードが親、子、他のノードの先祖または子孫として述べられるであろう。あるノードＮに関して、親ノードは階層においてノードＮより１レベル上位のノードであり、Ｎの子ノードは階層においてノードＮより１レベル下位のノードであり、Ｎの先祖ノ

ードはノードＮより上位のあらゆるレベルにあるノードであり、さらにＮの子孫ノードはノードＮより下位のあらゆるレベルにあるノードである。一般的には、階層は先祖ノードは持たないで子孫ノードとしてすべての他のノードをもつ根元ノードを持っている。

【００１６】

ここで述べられている実施態様では、部類ノードは子ノードとして部類ノード、サイトノードまたはその両方をもつことが出来るが、サイトノードは子ノードをもつことは出来ない。部類ノードの中には子ノードをもたないものがあるかも知れないが、そのような空の部類は好ましくは削除されるか隠される。また、必ずしもすべての部類ノードが子ノードをもつことを要求されている訳ではないが、しかし空の部類は不必要な混乱をさけるためにむしろ除かれるか隠される。

【００１７】

第１図は、その様な検索データベースがそこで照会されるクライアントサーバシステム１０の１例を示している。システム１０はインターネット１６を経由して検索サーバ１４と接続されたＨＴＴＰクライアント１２を含んで示されている。検索サーバ１４はドキュメント保管庫２０およびワード索引２２とつながれてそして検索要求３０に対し検索結果３２により応答する。

【００１８】

この特定の例において、ＨＴＴＰクライアント１２はブラウザであるが、検索用バックエンドプロセッサのような他のＨＴＴＰクライアントも、ブラウザの代わりに使うことが出来よう。また、システム１０は、インターネット１６をＨＴＴＰクライアント１２と検索サーバ１４との間の別の通信チャンネルで置換えることによっても構築され得ることが理解されるべきである。さらに検索サーバ１４はＨＴＴＰサーバであるが、異なったプロトコルがＨＴＴＰクライアント１２またはそれに代わるものによって理解される限りは、全く異なったプロトコルを使って要求を処理出来ることが理解されるべきである。簡潔のために、ひとつのＨＴＴＰクライアント、ひとつの要求およびひとつの応答のみが示されているが、しかし実際には、多くのクライアントがそれぞれがひとつまたはそれ以上の検索要求をもって検索サーバ１４に実体として同時にアクセスするであろうことは理解されるべきである。実際に、もし充分な理由があれば、検索サーバ１４のタスクは複数のマシンにばらまかれるかも知れない。もしタスクが複数のマシンにばらまかれれば、より好ましい配置は単一の論理マシンとしてクライアントに提供される複数マシンを持ち、クライアントのアクセスを簡単化することである。

【００１９】

動作においては、ブラウザにおけるユーザ、または他のＨＴＴＰクライアントは検索表現を含む要求３０を検索サーバ１４に送り、そこで検索サーバ１４は検索表現を解剖しさらに、検索表現が有効な形式であれば、検索表現に合致するドキュメントをドキュメント保管庫２０において見つけるために検索表現を使用する。検索サーバ１４は合致するドキュメントのリストかまたはドキュメントそのものかのいづれかにより応答する。ワード索引２２はドキュメント保管庫２０においてドキュメント検索をスピードアップするために使用される。

【００２０】

第２図はドキュメント保管庫内のドキュメントがいかにして論理的に配列されているかを示している。この例において、ドキュメントは通常は関心のＷＷＷサイトに配置されている検索データベースの要素である。各々のドキュメントはトピックの部類かあるいはサイトを表わしかつ各ドキュメントは他のレコードと親または子の関係にある階層構造においてレコード３８として示されている。各レコード３８はドキュメント番号４０および内容４６とともに示されている。部類であるドキュメントの場合には、内容４６は部類や隠されたキーワード、同意語、記述などのような他のテキスト（示されていない）のタイトルであり、一方サイトへ照会するドキュメントの内容はタイトル、ＵＲＬ、記述、隠されたキーワード、同意語等を含んでいる。勿論、これらの要素のいくつかは固有かあるいは希望によって空白であり得る。上で説明されたように、Ｙａｈｏｏ！の検索データベースで

は、ドキュメントは編集スタッフにより階層構造に位置付けされる。代表的な手順では、サイトの促進者が編集スタッフにサイトのタイトル、サイトのＵＲＬ、階層での申込まれた位置、記述などのようなサイト情報を提起するであろう。編集スタッフはその提起を評価し、もしより適当な位置が存在すれば示された位置を変え、必要とされるようにリンクし、さらにいくつかの場合には、隠されたキーワード、同意語および／またはドキュメントの重要度重み値を加える。

【００２１】

レコード間のリンクが、より一般的なトピック（親ノード）とより詳細なトピックまたはサイト照会（子ノード）とをつないでいる各々のリンクでもって、第２図に示されている。例えば、ドキュメント＃５は、“碁”に関係しているＷＷＷサイトに対するサイト照会であり、“碁”は盤のゲームでありそれ故に“盤のゲーム”トピックのサブトピックであり、これは“ゲーム”トピックのサブトピックであり、“ゲーム”トピックは“娯楽”トピックのサブトピックである、等々。

【００２２】

ドキュメント＃５および＃６のような、いくつかのサイト照会は葉の部類から離れたノードであり（つまり、子の部類ノードを持たないノード）、ドキュメント＃２１のような他のドキュメントは葉のない部類から離れたノードである。集合として、リンクは下記に説明されるように、どのサブトリーにあるドキュメントもサブトリーのトップにあるドキュメントのドキュメントナンバーに続いて連続的に番号付けされるように番号付けされるトリーおよびサブトリーを定義する。

【００２３】

さて第３図を参照して、第２図で示された情報とは異なる様相が表わされている。第３図は情報が、第２図におけるトリー構造のノードに対応するデータテーブルにおけるレコード３８でもってそれが記憶されている様に表わしている。第３図のデータ構造は多くの可能なデータ配列のうちのひとつを表わしていることは理解されるべきである。少しのレコード３８のみが示されているが、実際には多くの数百万のレコードが存在するであろう。

【００２４】

レコード３８に対して示されるフィールドはドキュメント番号４０、サブトリーにおける最終ノードを示すサブトリーポインタ４２（記憶され、要求されるように急いで作り出されるかまたはメモリ配列から得られる）、親ノードを示す親ポインタ４４、レコードによって表わされているドキュメントのテキスト（ここではタイトル４６および記述４７として示されている）、ドキュメントに関連するひとつまたはそれ以上のキーワードの付加的なセット４８、およびレコードが部類またはサイトのいづれに対するものかを示すブーリアン表示５０である。第２図の概観にあるように、いくつかのノードはＷＷＷサイトを指しそして他のノードはサイトの要素がひとつまたはそれ以上の部類の要素に関連する階層的なトピックの部類構造における部類を表わしている。この例はトピックならびにサイト照会のトリー構造であるが、ここで述べられるシステムはもっと複雑なドキュメントを探し出せることがこの記述から明らかである。

【００２５】

部類ノードに対して、レコード３８はタイトル４６、部類の記述４７、および時には隠されたキーワード４８のセットを含んでいる。サイトノードに対しては、レコードはタイトル、サイトの記述（たぶん空白）、および照会されるサイト／ページを示すＵＲＬポインティングを含んでいる。どちらも、ドキュメントナンバー４０、サブトリーポインタ４２および親ポインタ４４がレコード間のつながりを記述している。例えば、ドキュメント＃２はそのサブトリーポインタとして“８”をもち、３（ドキュメントナンバープラス１）から８（サブトリーポインタ値）の番号を付けられたすべてのドキュメントがドキュメント２の下位のサブトリーに存在することを示しており、さらに親ポインタとしての“１”はドキュメント１がドキュメント２の親ドキュメントであることを示している。ドキュメント＃２に対するレコードの他のフィールドは、その内容が“ゲーム”であり、リストされるキーワードをもたずさらにそれは部類（サイト照会に相対するように）であることを

示している。サブトリーにおける最後のドキュメントナンバーを使用してサブトリー全体を詳述することはドキュメントナンバー順位のこの指定のために可能である。

【００２６】

ドキュメント保管庫２０はドキュメントナンバーによって要求されたドキュメントを返すための必要な処理をする論理を含み、そしてドキュメント保管庫２０あるいは検索エンジン３６のどちらかが照会項目に合致するフィールド値の例示に対してレコードを探すための処理する論理を含んでいる。

【００２７】

さて第３図に示されるワード索引２２を参照して、ワード索引２２の内容の小さなサブセットがそこで示されている。ワード索引２２は複数のレコードとして組織され、ドキュメント保管庫２０のドキュメントにおいて生じるワードあたりひとつのレコードであり、ワードによってアルファベット順にソートされている。ワード索引２２における各レコード５２が、ワードを確認するタグとともに示され、ドキュメントナンバーのリストによって伴われている。これらのドキュメントナンバーは、そのワードを含むドキュメントのリストを表わしている。ワードが、語句の“ the ”のように有用性が限られた検索項目である程に普通である場合は、そのレコードはワードを含むすべてのドキュメントではなくて、そのワードが無視されると言う表示のみを記している。

【００２８】

好ましい実施態様では、検索エンジン３６は検索スピードを改善するために、ドキュメントプロフィールアレイ４９を使用している。代表的にはアレイ４９は速いアクセスのためのメモリに記憶されている。アレイ４９はドキュメントあたりひとつのレコードをもち、そして各レコードは、ドキュメントナンバー、ドキュメント保管庫ポインタ、時間スタンプ、子レコードの範囲および重要度重み値のためのフィールドを含んでいる。ドキュメントナンバーはドキュメント保管庫２０のドキュメントナンバーに対応し、さらにドキュメント保管庫ポインタはドキュメント保管庫２０のドキュメントの物理的なディスク位置に対応しており、だからアレイ４９はドキュメント保管庫２０それ自体へのアクセスを必要としないドキュメントで、いくつかの動作を遂行するように使われる。時間スタンプはドキュメントが最後に変更された時間を識別する。子の範囲フィールドは、レコードが階層においてそのドキュメントよりも下にあることを示しており、その結果、検索エンジン３６はドキュメント保管庫を余計に参照しないで合致リストを迅速に作ることができる。

【００２９】

重要度重み値はある特定の部類またはサイトが他の部類およびサイトといかに有益におよび／または適切に関係があるかを示すために、自動的に、または編集スタッフによってセットされる値である。レコードの重要度重み値は外部の催物あるいはサイトの重要性にもとづいて調整されるであろう。例えば、現在新聞記事になっている特定のグループに関係するサイトは、より高い重みを与えられるであろうし、あるいは編集スタッフがそのサイトが人気があるかまたはよく企画されていると決定すれば、サイトはより高い重みを与えられるであろう。第３図に示されるアレイ４９における部類レコードは重み付けされた部類をもっていないが、重み付けされた部類は有用であるかも知れない。例えば、ボートシーズンの間は、ドキュメント＃９（部類“ ボートのり ”）に対する重みは増加するであろう。レコードの重み付けは、表示ドキュメントがその重みの値によって順に表示されるから、複数のドキュメントが検索の結果として表示されるときに、役割を演じることになる。勿論、以下に述べられるように、他の重み付けの要因が重要度の重み付けを押え、あるいはオーバーオールでの重み付けをつくるために組合わされるであろう。

【００３０】

ドキュメント保管庫２０およびワード索引２２の使用が例を参照してここで述べられる。この例においては、ユーザーはドキュメントを探しており照会ストリング“ 碁のゲーム ”をもつ検索要求を与える。検索エンジン３６はワード索引２２にある各々の語句を調べる。非常にありふれているから、“ the ”および“ of ”は検索エンジン３６によって無視されるかあるいは、上述されたように、ワード索引２２がこれらの語句を無視する指示を返

すかのいづれかである。検索エンジン３６はそれから“game”および“go”に対してドキュメントリストを読み、各々の語句に対する合致リストを作り出し、さらに第６図との関係で以下に述べられるように、合致リストに“ＡＮＤ”の運算記号を適用する。

【００３１】

合致リストは、直接的あるいは間接的のいづれかでリストの合致タグを含んでいる全部のドキュメントのリストである。合致タグは照会項目が何であるかに依存して、語であるかまたは他の検索項目または検索要素である。第２図はなぜ間接的な合致が重要なのかを図示している。この記述を通して使用される例は、黒と白のマーカーを使用するよく知られた盤ゲームである“碁”のゲームに関係のある部類とサイトを探すことである。ゲームの名前はたまたま英語におけるありふれた語句と同じであるから、“ｇｏ”を探すことはあまりにも多くの無関係な合致をひき起こすであろう。しかしながら、見られるように、同じドキュメントの中に“ｇｏ”および“ｇａｍｅ”を探すことは合致しない結果を引き起こすであろう。そのために、各ドキュメントは、あたかもそれがすべての先祖ドキュメントのすべての検索要素（この場合には検索要素は語である）を含んでいるかのように検索される必要がある。勿論、すべての先祖ドキュメントは、そのサブトリーの中の各々の子孫ドキュメントの中へ挿入され得るが、しかし大きなトリーでは、このアプローチは無駄が多く実用的でない。

【００３２】

第２－３図を再び参照して、合致リストの各項目は直接的な合致の場合には、ひとつのドキュメントを参照し、または間接的な合致の場合にはドキュメントの範囲を参照する。第４図は合致リストのいくつかの例を示している。第１の、第４図（ａ）は第２－３図に示されているこのドキュメントに対応する合致リスト６０である。合致リスト６０は３つの項目、または合致レコードを含んでいる。第１はドキュメント＃３が合致タグに合致することを示す直接的な合致レコードであり、さらに第２は＃４から＃８に至るドキュメントが間接的に合致タグと合致することを示している間接的な合致レコードである。

【００３３】

この例では、合致タグは、テキストであるから、“合致”はドキュメントが合致タグをドキュメントの内容においてストリングまたはサブストリングとして含むときに生じる。ある場合には、認識されているサブストリングの形式のみが“右手のワイルドカード”サブストリングであり，これは“語”の形式である。第２－３図から判るように、ドキュメント＃３は、合致リスト６０の合致タグ“盤”にまさに直接的に合致している。ドキュメント＃４から＃８は直接には“盤”の語を含まないが、その語を含むドキュメント／ノードからの子ドキュメント／ノードである。それらはその語を含む親からの子であり、さらに子はその語を含んでいないので、それらはそれ故に間接的な合致である。

【００３４】

合致リスト６０は第３の合致レコード“ヌル（null）”をもっており、これは単に合致リストの終りを示している。リストの終りにヌル項目を使うことは、良く知られたコンピュータでの技術であり、また多くの他のリスト操作技術がここで述べられた特定のものに代って使用され得る。

【００３５】

合致リストの直接的な合致レコードは、ワード索引２２から生じる。間接的な合致レコードは、ドキュメント保管庫２０にあるドキュメントレコード、または各々の直接的な合致に対してアレイ４９にあるドキュメントサマリーレコードを調べることにより得られる。もし直接的な合致のドキュメントレコードが、ドキュメントがサブトリーをもつことを示せば、間接的な合致レコードがサブトリーのドキュメントの範囲に対して作り出される。サブトリーのドキュメントもまた直接的合致である場合には、それは間接的な合致の範囲から除外される（これは範囲が２つの間接的な合致レコードに分割されることに帰着するかも知れない）。各々の直接的合致は合致リストに加えられるから、合致リストは合致リスト上に既に存在する間接的な合致（範囲）が、直接的な合致と重複するかどうかを決めるためにチェックされる。これは合致項目が先祖のドキュメントと子孫のドキュメントの

両方に現れるところで生じる。先祖のドキュメントは子孫ドキュメントよりも下位のドキュメントナンバーをもっているから、先祖のドキュメントは最初に処理される。直接的な合致レコードは先祖のドキュメントに対して作り出され、それから間接的な合致レコードが直接的な合致ドキュメントより下位の子孫ドキュメントのグループに対して作り出される。子孫の直接的な合致ドキュメントが処理される時には、それもまた直接的合致レコードに記名されるであろうし、さらにそれ故間接的な合致の範囲に含められるべきでない。各々の間接的な合致をただひとつのドキュメント範囲に関連づけるために、間接的な合致レコードは２つの間接的な合致レコード、ひとつは子孫の直接合致ドキュメント側のものと、それを除外したものに分かれている。勿論、もし子孫ドキュメントが間接の範囲のひとつの境界に存在するならば、ただひとつの新しい間接的な合致レコードが作り出されるであろう。このひとつの新しい間接的合致レコードは単に境界にあるひとつのドキュメントナンバーによって減らされる間接の範囲であるであろう。

【００３６】

これは第４図（ｂ）－（ｃ）に図示されている。第４図（ｂ）に示される合致レコード６２は、ドキュメント１２に対する直接的な合致レコードと、ドキュメント＃１３から＃１７までに対する間接的な合致レコードとを持っている。もしドキュメント＃１５および＃１６が合致タグを直接に含むように変更されるならば、サブトリーは２つの間接的合致と、直接的合致の各側におけるひとつによって表わされるであろう。勿論、直接的合致のひとつの側にも合致するものが無ければ、ただひとつの間接的な合致レコードが必要とされるであろう。

【００３７】

ドキュメント範囲のひとつ“１７－１７”は、ただひとつのドキュメントを含んでいることは注意されるべきである。これは直接的な合致レコードと間接的な合致レコードとを見分けるためのものである。勿論、代り得る別の仕組みが使用され得る。例えば、単純な場合では各々の合致レコードは、フラグとドキュメントナンバーを含み、このフラグはドキュメントナンバーが直接的合致か間接的合致かのいづれに関係するかを示すものである。直接的合致に対しては、単独の番号が合致するドキュメントの番号となり、間接的合致に対しては、その数はその範囲の最後のドキュメントの数である。これは、範囲の最初のナンバーがすぐ前の直接的合致レコードのナンバーの続くナンバーである場合には、まぎらわしくない。これは、ドキュメントがこの特性をもつように順番づけされるので、ドキュメントナンバーが欠けていない場合であろう。

【００３８】

第３図を再び参照して、あるシステムでは、どのくらい度々ドキュメントが変更されるか、およびどのくらい度々検索項目が使われるかによって、直接的な合致および間接的な合致の両方が前もって計算され、そしてレコード５２に記憶されるであろう。そうでなければ、それらは必要とされるように急いで作り出される。もしそれがなされるならば、検索エンジン３６は、検索照会がドキュメントの中の語の位置またはワード索引２２から得られない他のフィールドの情報の検査を要求しなければ、ドキュメント保管庫２０にアクセスする必要はない。ワード索引２２を使用する利点は、与えられた検索要求を素速く探すために、合致タグがあると言うことである。

【００３９】

いちど合致リストが得られるか、または検索エンジン３６によって作られると、それは出力リスト３１を返送する。出力リスト３１は記入されたドキュメント自体か、またはドキュメントナンバーであり得る。もし出力リスト３１がドキュメント自体であり、さらに適切にフォーマットされていれば、出力リスト３１はブラウザ１２へ送られる検索結果となるであろう（第１図参照）。

【００４０】

第５図は、照会ストリング“碁のゲーム”から生じる検索結果の表示５３の例を示している。表示５３上で、合致する部類ドキュメント５４が上部に示され、そして合致するサイトのドキュメント５８と分けられ、部類トリーへ通じるそれらのパス５６とともに示され

ている。第５図は部類構造を通した実際の検索、およびＹａｈｏｏ！，Ｉｎｃ．のサイト帳表、当アプリケーションの割当者を表している。明確のために、第５図に示されるすべての合致が必ずしも他の図で表されている訳ではないし、また見つかった実際の１７７サイトの合致がすべて示されている訳でもない。

【００４１】

表示５３のいくつかの楽に使える特徴は、注目されるべきである。各々の“ヒット”または合致（５４、５８）は、合致までのパスを明示する部類の一連のタイトルとともに示されている。この例は、第５図に示される合致する部類のドキュメント５４によって示されている。読み易さをさらに改善するために、合致するドキュメントからの子ノードである合致するドキュメントは、示されていない。もしあれば、合致する部類ドキュメント５４より下位のすべてのレコードが示されていたであろう。

【００４２】

さて、直接的および間接的合致レコードをもつ合致リストおよびその成生が記述されたから、合致リストの“ＡＮＤ”、“ＯＲ”、“加算（ＡＤＤ）”、“減算（ＳＵＢＴＲＡＣＴ）”と言うような演算の適用が次に述べられる。これらの演算は、検索エンジンが各々の検索の下位項目に対して合致リストが有効である複数の検索下位項目を含む複雑な検索表現に対する合致リストを作り出す必要がある場合では、有益である。上述の例を続けて、検索エンジンは、検索表現“碁”および“ゲーム”に対する合致リスト（またはドキュメントリスト）に到達するために、ＡＮＤ演算を使用して検索項目の“碁（ｇｏ）”および“ゲーム（ｇａｍｅ）”に対する合致リストを結びつけるであろう。

【００４３】

第６図は新しい合致リストを作り出すために、２つまたはそれ以上の合致リストを“ＡＮＤ”するプロセスのフローチャートである。明らかなように、結果としての合致リストは、それから検索結果を作り出すために使われ得るか、あるいは合致リストでのその後の論理演算のための入力として使用される。もしその後の論理演算がなされないのならば、出力は単にドキュメントのリストとなる。第６図において、ステップはＳ１、Ｓ２等々にラベルづけられており、一般にステップの実行順序を表わしている。この記述を読んで明らかなように、ステップの他の配置が、実質的に同じ結果を得るために、実質上同じ機能を遂行するかも知れない。

【００４４】

２つまたはそれ以上の合致リストを“ＡＮＤ”する要求は、検索ストリングが“表現ＡＡＮＤ 表現Ｂ　ＡＮＤ・・・・”形式の表現を含む場合に起こる。ひとつの合致リストは、その下位項目を含むドキュメントを示している“表現Ａ”に対して得られ、他は“表現Ｂ”に対して、等々である。結果としての合致リストは、“ＡＮＤ”された表現のすべてを含むすべてのドキュメントのリストであり、さらに少なくともひとつの検索下位項目を直接に含んでいる。これらの要求の他の変化は、この記述を読んだ後で明らかになるこのプロセスの変更によって扱い得ることは明らかである。

【００４５】

幅広い項目において、第６図に示されるプロセスは、どのドキュメントがすべての合致リストにおいて見い出され、さらに少なくともひとつの直接的な合致レコードにおいて見い出されるかを見つけるために、多数の合致リストを走査することに対する効率的なプロセスである。これを行うために、プロセスは最初にひとつの合致リストにおける直接的な合致レコードの位置を決め、そしてドキュメントがそれらのリスト上で見い出されるかどうかを決めるために、すべての他の合致リストをチェックすることを含んでいる。ひとつの合致リストがドキュメントを持たないことが判ると、ドキュメントカーソルが合致リストの次のドキュメントに対して増加される。

【００４６】

再び第６図を参照して、プロセス変数がステップＳ１で初期化される。ループカウンタ、ＬＯＯＰ＿ＣＮＴ、はゼロに初期化される。ループカウンタの使用は下記で説明される。加えて、合致リストにおけるドキュメントをさし示すドキュメントカーソル（Ｄ＿ＣＵＲ

）が、１と同じにセットされ、見い出された合致の数を数える採集カウンタ（ＣＯＬＬ＿ＣＮＴ）はゼロにセットされ、さらにひとつの合致リストを指すリストポインタ（Ｌ＿ＰＴＲ）はひとつの合致リストを指すようにセットされる。Ｌ＿ＰＴＲは、しかし必要ないことだが、“ＡＮＤ”されている最初に記名された下位項目に対する合致リストがさし示されるかも知れない。Ｌ＿ＰＴＲによってさし示される合致リストは、ここで“現在の合致リスト”あるいは“現在リスト”として参照される。

【００４７】

ステップＳ２で、現在の合致レコードが現在リストから得られる。これはここで“現在の合致レコード”として参照される。現在の合致レコードはＤ＿ＣＵＲより大きいか、あるいは最も低い同じドキュメントナンバーをもっている現在リストにおける合致レコードである。もし現在の合致リストが空であれば、プロセスは単純に終了する。なぜならドキュメントが見い出されないからである。ステップＳ２へ通じる最初の時は、Ｄ＿ＣＵＲは１であろう、だから現在の合致レコードは現在の合致リストにおける最初のレコードであろう。

【００４８】

ステップＳ３において、現在の合致レコードが、それが直接的な合致であるかまたはグループ合致であるかどうかを決定するためにチェックされる。第４図の記述から明らかであるように、合致レコードが直接的合致であるかあるいはそうでないかを決めることは簡単な問題である。なぜなら間接、すなわちグループ合致は、ひとつまたはそれ以上のドキュメントナンバーの範囲として表わされるからである。もし現在の合致レコードが直接的な合致であれば、それは現在の合致レコードによって表わされたドキュメントに対し少なくともひとつの直接的な合致があると言う要求を満足する。最初の直接的な合致が見い出される前は、ＣＯＬＬ＿ＣＮＴはゼロであり、プロセスは合致リストからドキュメントをまだ“集め”始めていないことを示している。もしＣＯＬＬ＿ＣＮＴがステップＳ４でゼロであり、さらに直接的な合致レコードが見つかったならば、ＣＯＬＬ＿ＣＮＴは直接的な合致が見つかったことを示すために１にセットされる（Ｓ５）。

【００４９】

ひとたび最初の直接的な合致が見つかると、ＬＯＯＰ＿ＣＮＴはステップＳ６でゼロにリセットされ（ＬＯＯＰ＿ＣＮＴはある状況下で違ったふうに起るであろう無限ループを避けるために使用されており、その使用は下に説明される）、Ｌ＿ＰＴＲが回わされて次の合致リストを指す（Ｓ７）。それに続いて、プロセスは現在の合致リストである次の合致リストをもってステップＳ２へループバックする。ステップＳ２では、現在の合致レコードが新しい現在の合致リストで見出される。上述されたように、見出されたレコードはＤ＿ＣＵＲに同じかまたはそれより大きい一番低いドキュメントナンバーをもつものであり、そしてもし何も見つからなければ、プロセスは終了する。

【００５０】

もしＣＯＬＬ＿ＣＮＴがＳ４でゼロでなく、ドキュメント収集の状態を示しておれば、Ｄ＿ＣＵＲが現在の合致レコードのドキュメントナンバーに対してチェックされる（Ｓ８）。もしそれらが同じなら、それは前の合致リストから集められているドキュメントが、現在の合致リストの今のリストに対するものと同じであることを示している。もしそれがその場合であれば、ＣＯＬＬ＿ＣＮＴは増やされ（Ｓ９）、別の合致が見つかっていることを示す。もしすべてのＮ合致リストより少ないリストが処理されたのであれば、ＣＯＬＬ＿ＣＮＴはＮより少ないであろうし、だから合致リストが現在の合致リストとなるステップＳ６／Ｓ７でプロセスは続く。これはＣＯＬＬ＿ＣＮＴがＮに等しくなるまで続くであろう。

【００５１】

ＣＯＬＬ＿ＣＮＴがＮに達する時は、Ｄ＿ＣＵＲに同じドキュメントナンバーが合致リストのＮすべてで見つかったことを意味し、それ故にこれは出力する合致リストに存在するべきドキュメントナンバーである。この結果、現在の合致レコードは出力され（Ｓ１０）、そしてプロセスはステップＳ６／Ｓ７へ続く（プロセスは現在の合致リストを変えない

でステップＳ２へループバックすることによりさらに続くであろう）。ステップＳ１０では、ＣＯＬＬ＿ＣＮＴはドキュメントナンバー探しの次のサイクルのためにゼロにリセットされる。

【００５２】

もし、ステップＳ８で、現在の合致レコードのドキュメントナンバーがＤ＿ＣＵＲと同じでなければ、それは現在の合致リストがＤ＿ＣＵＲと同じドキュメントナンバーをもつ合致レコードを持っていなかったこと、そしてより大きなドキュメントナンバーが選ばれたことによる。その場合には、現在のドキュメントがなお集められつつあるが、それは新しいより大きなドキュメントナンバーである。Ｄ＿ＣＵＲは、その新しいドキュメントナンバーにセットされる（Ｓ１１）。どれだけの合致リストがこの新しいドキュメントナンバーを持っているかについてのこん跡を残すために、ＣＯＬＬ＿ＣＮＴは１にリセットされ（Ｓ５）、そしてプロセスは上述のように続いていく。

【００５３】

もし、ステップＳ３で、現在の合致レコードが直接的な合致の代りにグループ合致であれば、レコードの処理はプロセスの状態、即ち、ドキュメントが“採集”されているかどうかに依存する。これはＣＯＬＬ＿ＣＮＴをチェックすることにより決められる（Ｓ１２）。もしもＣＯＬＬ＿ＣＮＴがゼロでなければドキュメントは採集されつつあり、この場合には現在の合致レコードがＤ＿ＣＵＲと比較される（Ｓ１３）。もしもＤ＿ＣＵＲが現在の合致レコードの範囲内であれば（このステップをとるグループレコードであるに違いない）、その時はＣＯＬＬ＿ＣＮＴは増加され（Ｓ９）、そして次のリストが上述のようにチェックされる。

【００５４】

もし、ステップＳ１２で、ＣＯＬＬ＿ＣＮＴがゼロであれば、プロセスはステップＳ１４へ続く。また、もしステップＳ１３で、Ｄ＿ＣＵＲが合致レコードの範囲内でなければ、プロセスはＣＯＬＬ＿ＣＮＴをゼロにセットした後でステップＳ１４へ続き、ドキュメントが採集されていないことを示す。ステップＳ１４で、ＬＯＯＰ＿ＣＮＴは増加されそしてＮと比較される（Ｓ１５）。もしもＬＯＯＰ＿ＣＮＴがＮと同じでなければ、プロセスは次のリストとともにステップＳ７へ続く。もしもＬＯＯＰ＿ＣＮＴがＮに等しければ、Ｎリストのすべてが調べられさらに合致リストが各々で見い出されたが、合致のどれもが直接的な合致でないことを示している、その他の点ではＣＯＬＬ＿ＣＮＴはゼロでないであろう。

【００５５】

もしＬＯＯＰ＿ＣＮＴがＮに等しければ、Ｄ＿ＣＵＲに入っているグループ（間接）合致レコードが、合致リストの各々で見つかったことおよびそれ故直接的な合致はＤ＿ＣＵＲに対しては存在しないことを意味している。Ｄ＿ＣＵＲに入っている各々のグループは、最も低い最後のドキュメントナンバーをもつグループを見つけるために調べられる。

他に、検索エンジンは各々の合致リストが調べられるように、最も低い最後のドキュメントナンバーのこん跡をちょうどもち続けるかも知れない。Ｄ＿ＣＵＲは最も低い最後のドキュメントナンバーより１大きくセットされ（Ｓ１６）、そしてドキュメントに対する検索がステップＳ６に続き、そこではＬＯＯＰ＿ＣＮＴがゼロにセットされる。ステップＳ６は直接的な合致が見つかり、出力レコードが出力されるかまたはＬＯＯＰ＿ＣＮＴがＮに到達し、さらにＤ＿ＣＵＲが現在のグループの最後に到達することによって無限ループが避けられるときに、ＬＯＯＰ＿ＣＮＴをリセットするために置かれている。好ましい実施態様においては、少なくともひとつの直接的な合致が要求される。しかしながら、直接的な合致が要求されない実施態様においては、プロセスはＬＯＯＰ＿ＣＮＴがＮに達する時に合致レコードを出力するであろう。

【００５６】

ステップＳ２を通過する時、現在の合致リストの終りに達している時には、このプロセスが終るのに続いて、ＡＮＤ演算の要求を満たすドキュメントのための合致レコードが、ステップＳ１０を通過する際に出力されるであろう。他に、もしそれ以上の論理演算がなさ

れないのであれば、出力は合致するドキュメントのドキュメントナンバーのまさに帳表だけとなるであろう。

【００５７】

さて第７図を参照して、下位項目Ａ、Ｂ、・・・等に対する合致リストから“Ａ　ＯＲ　Ｂ　ＯＲ・・・”形式の検索表現に合致するドキュメントの出力リストを作り出すためのプロセスが、ここで示されている。幅広い項目において、このプロセスは、検索表現をその下位項目に解剖すること、さらに各下位項目に対して合致リストを鑑定すること、それから出力リスト上の各ドキュメントが少なくともひとつの下位項目を含む出力リストに合致リストを結合することを含んでいる。

【００５８】

好ましい実施態様において、出力リストの各ドキュメントは、すくなくともひとつの直接的な合致をもつという追加の要求が課せられ、そこで追加の要求に合うドキュメントは必然的にＯＲ表現と直接に合致するから、間接的な合致は存在しないであろう。好ましい実施態様において、出力リストは各々が関連ある合致カウントをもつ直接的な合致のリストである。合致カウントは、いくつのＯＲ下位項目が直接的にまたは間接的に合致するかを示しており、それ故そのドキュメントの相対的な関連の表示である。

【００５９】

第７図のフローチャートにおいて、プロセスのステップはＳ３０、Ｓ３１等のラベルを付けられており、示されている場所ではなくて数字順に実行される。プロセスは、ステップ３０で始まり、そこでは下位項目が検索表現から抽出されそして合致リストカウンター、Ｎ、が下位項目の数と同じにセットされる。ステップＳ３１では、ひとつの合致リストが下位項目に対して作り出されるか、またはリストがもし先在するリストであれば持ってこられる。

【００６０】

ステップＳ３２で、ひとつのカーソルが各々の合致リストに対して初期設定され、カーソルは関連するリストの最初のドキュメントをさし示す。ステップＳ３３では、各リストからの最初のドキュメントがＮの一部の推積（ｈｅａｐ）に加えられる。

次に、推積の内容は、ドキュメントナンバーによって整理される。好ましい実施態様では、推積がある限られたドキュメントナンバーに対する直接的な合致と、さらにその同じドキュメントナンバーで始まる範囲をもつ間接的な合致を含む場合には、直接的合致は間接的な合致の前に整置される。もし推積が同じ最初のドキュメントナンバーをもつひとつ以上の間接的な合致をもつならば、彼等はその終りのドキュメントナンバーによってソートされる。

【００６１】

いちど推積がソートされると、一番上の推積項目は推積から移される（Ｓ３５）。もし一番上の推積項目が、直接的な合致項目でありその直接的合致項目のドキュメントナンバーが今まで出力リストになければ、それは出力リストに加えられ（Ｓ３６）、さらに今移された項目の合致リストから他の項目を推積に加えることによって、プロセスは続く（Ｓ３７）。もし合致リストが、それ以上の項目を持たなければ、新しい項目は推積に加えられない。ついには、推積は空になるだろう。もしステップＳ３８で、推積が空となればプロセスは完了し、そして終結する。そうでなければ、プロセスはステップＳ３４へループバックし、そこでは推積が再び整理される。

【００６２】

もし移された項目が、出力リストに既にあるドキュメントのドキュメントナンバーをもつ直接的な合致のレコードであれば、そのドキュメントナンバーに対する合致のカウントが増やされ（Ｓ３９）、そしてプロセスは上述されたようにステップ３４へ続く。もし移された項目が間接的な合致のレコードであれば、それは出力リストに置かれずに、間接的な合致のレコードのドキュメント範囲内にある各ドキュメントに対して合致のカウントが増やされ（Ｓ３９）、そしてプロセスはステップＳ３４へ続く。間接的な合致は出力リストに加えられない、なぜなら少なくともひとつの直接的な合致をもつという要求に合うその

間接的な合致に対するドキュメント範囲にあるどんなドキュメントも、すでに出力リスト上にあるからである。これは、合致レコードが推積上に移されるとき順番に合致リストから取られ、そして推積上の項目もまた順番に持ち去られ、さらに直接的合致は同じナンバーからスタートする間接的合致の前にとられる理由によるものである。

【００６３】

推積が空になると、出力リストは、ＯＲ規準に合致するすべてのドキュメントを含むであろう。すべての出力リストの入口は直接的合致であり、そして関連する合致のカウントをもつであろう。もし各々の合致は少なくともひとつの下位項目を直接に含むという要求が課せられないならば、出力リストはそれ以上の処理に適した合致リストの形式となるであろう。合致のカウントは、単独あるいは重要度の重み付けと結びつけて、関連によってドキュメントを整理するために使用され得る。

【００６４】

さて“加算（ＡＤＤ）”および“減算（ＳＵＢＴＲＡＣＴ）”演算へ転じて、これらはもっと簡単である。“加算”演算に対しては、リストに加えられるドキュメントナンバーが単に挿入されるのみである。勿論、直接的な合致のドキュメントナンバーが入っている間接的な、グループ合致を含むリストに直接的な合致が加えられれば、グループ合致レコードは上述のように分割する。“減算”演算に対しては合致レコードが単に合致リストから削除される。もしドキュメントがグループ合致レコードの範囲内にあるところでドキュメントナンバーが減じられるならば、グループ合致レコードは上述のように分割する。

【００６５】

上の記述は説明に役立てるものであり限定するものではない。発明の多くの変形例は、この開示の吟味によって技術者には明らかとなるであろう。例えば、ドキュメントの階層構造は、上述の階層による検索構造の代りにインターネットのドキュメントの入り組んだ仕組みであるかも知れない。発明の範囲は、それ故上記の記述に関して決められるのではなくて、同等のすべての範囲をもつ添えられた請求項に関して決められるべきである。

【図面の簡単な説明】

【図１】　本発明のひとつの実施態様による検索エンジンをもつクライアントサーバシステムの概略図である。

【図２】　ドキュメント保管庫の各部に相応するドキュメントのトリー図である。

【図３】　図１に示されるクライアントサーバシステムの要素のより詳細に示された図であり、ドキュメント保管庫、ワード索引ならびに検索エンジンのさらなる詳細を示している。

【図４（ａ）】　図３に示される検索エンジンにより使用される合致リストの例を示す図である。

【図４（ｂ）】　図３に示される検索エンジンにより使用される合致リストの例を示す図である。

【図４（ｃ）】　図３に示される検索エンジンにより使用される合致リストの例を示す図である。

【図５】　本発明のひとつの実施態様による検索結果のブラウザ表示によるスクリーン画面を示す図である。

【図６】　検索エンジンで遂行されるＡＮＤ演算のフローチャートを示す図である。

【図７】　検索エンジンで遂行されるＯＲ演算のフローチャートを示す図である。

【図１】

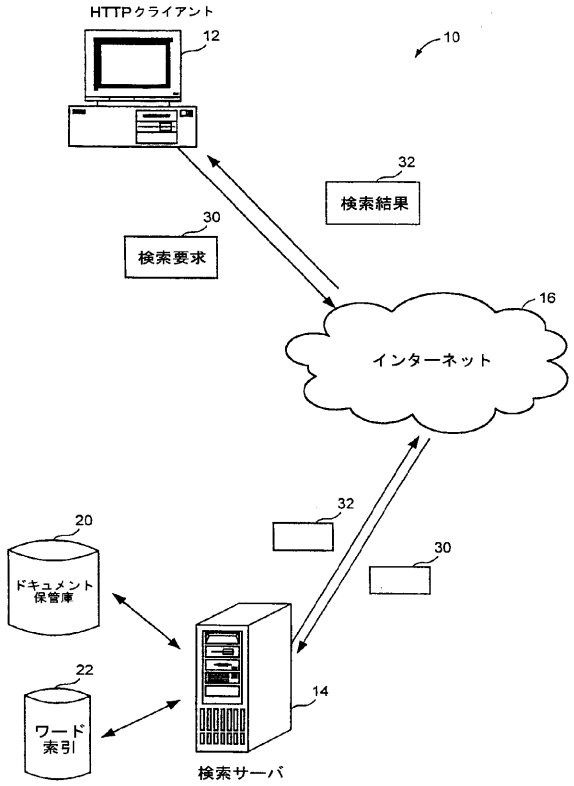


FIG. 1

【図２】

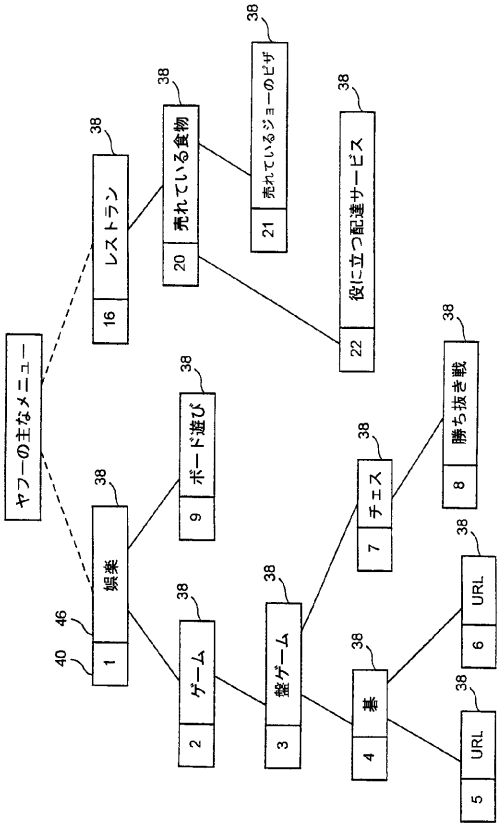


FIG. 2

【図３】

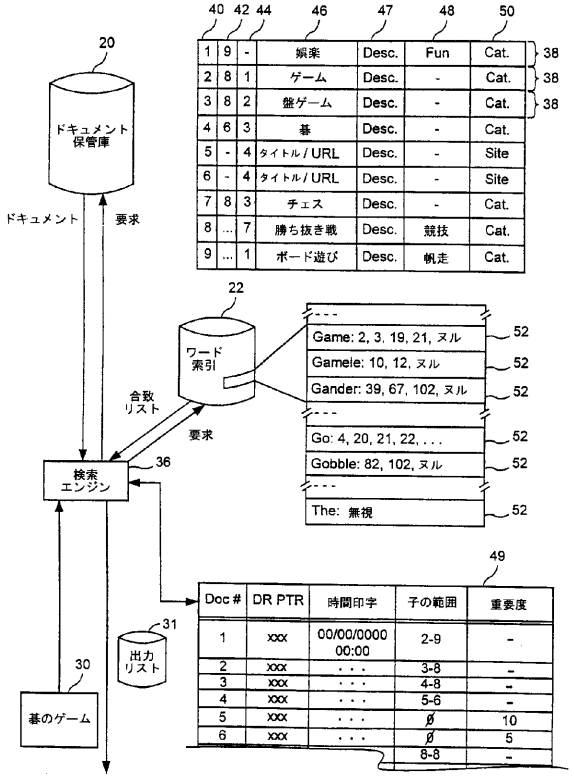


FIG. 3

【図４】

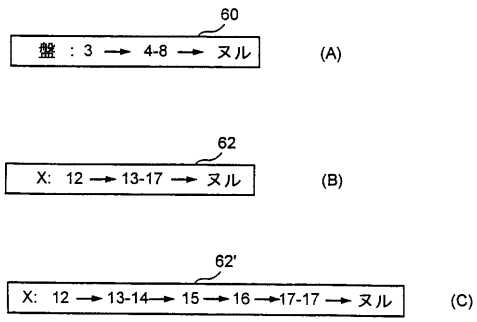


FIG. 4

【図５】

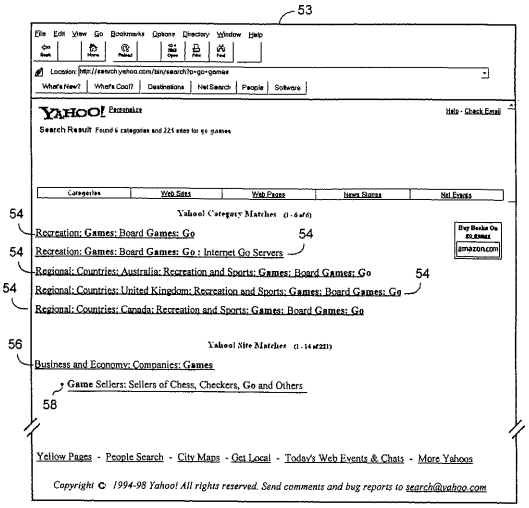


FIG. 5

【図６】

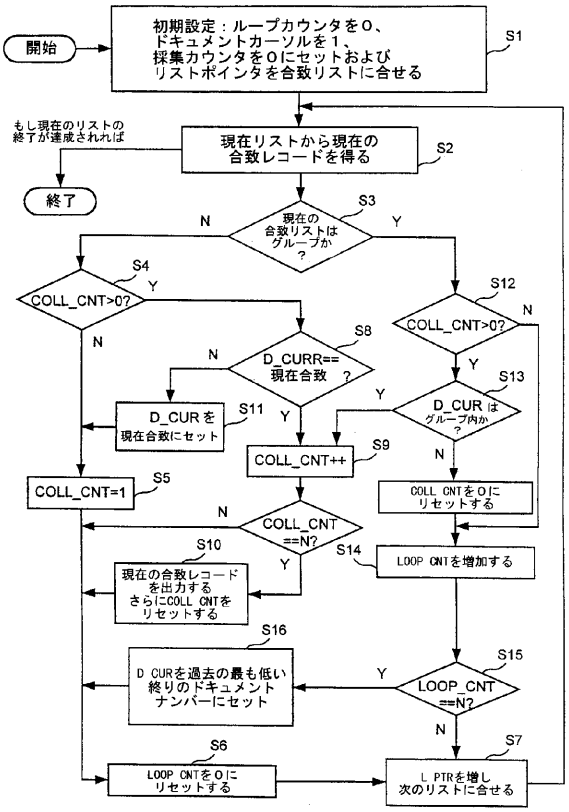


FIG. 6

【図７】

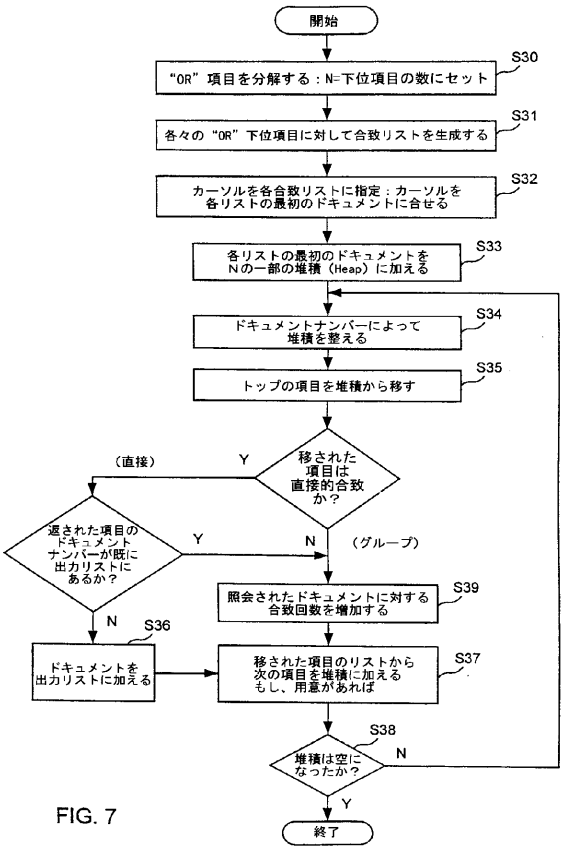


FIG. 7

フロントページの続き

(74)代理人 100084663
弁理士 箱田 篤
(74)復代理人 100122563
弁理士 越柴 絵里
(72)発明者 ウー イーオン
アメリカ合衆国 カリフォルニア州 ９４５５５ フリーモント ガードウォール コモン ３４３０５

合議体
審判長 小曳 満昭
審判官 加内 慎也
審判官 岩崎 伸二

(56)参考文献 特開平８－３３９３８０（ＪＰ，Ａ）
特開平６－２４３０１８（ＪＰ，Ａ）
特開平９－１２８３８０（ＪＰ，Ａ）
鈴木 尚志，Ｙａｈｏｏ！ ＪＡＰＡＮ アメリカの超人気サーチエンジンの日本語版，ＩＮＴＥＲＮＥＴ Ｓｕｒｆｅｒ，日本，エーアイ出版株式会社，１９９６年8月1日，第1巻 第3号，ｐ．４１
寺田 慶治，ゼッタイ見つかる！秘伝インターネット情報検索術，ＰＣｆａｎ，日本，株式会社毎日コミュニケーションズ，１９９７年 8月１５日，第4巻 第１５号，ｐｐ．６８－７７

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F17/30