鹿児島純心女子短期大学研究紀要　第32号，153－162　2002

全文検索を前提とした場合の電子化保存形式に関する比較検討

吉　田　尚　史

A Comparison on Digital File Formats for Preparing of Full-Text Searching

Naofumi Yoshida

電子化された情報を公開されたネットワーク上に提供することの重要性は，必要な情報がどこに存在するのかを物理的な制約なしに知ることができることにある。公開される文書の内容に対して書誌情報のような一部分ではなく，全文に対する情報検索が必要であり，それによって得られた必要な文書をネットワーク上で閲覧できることが必要である。

これらのことから，電子化した学内出版物を蓄積し，それらの内容全文に対しての検索機能が必要がある。本稿では検索エンジンとして namazuに代表されるインデックス検索型のシステムやデータベースシステムを利用した場合のユーザビリティの高さ，管理運用の容易さといった観点に立ち，文書の公開を行う場合の情報の電子化保存形式について考察を行う。

Key words：［文書の電子化］［全文検索］［保存形式］［pdf］

(Received November 5, 2001)

1．はじめに

前稿『学内出版物の電子化保存形式に関する一考察』において，学内出版物を電子化し保存する際の保存形式について代表的なものについて作業手順の点から比較した。本稿では，前稿での前提を継承し，次のような観点に立つ。すなわち，学内出版物を電子化し公開することの意義とその際に前提となるファイルの閲覧に対する制限，および作業段階の簡略化である。これらの点に立脚した上で，本稿では電子化されたファイルを全文検索するということを前提とし，その観点から再び電子化保存形式について考えてみたい。

2．問題の所在と前提

学内出版物の電子化保存形式を考えることは，いわゆる電子図書館，すなわち提供する情報とその情報の検索・閲覧手段の全てを電子化しネットワーク化された図書館を考える場合に重要な要素となるものである。

各館のOPACシステムの普及が利用者を空間的・物理的な制約から開放し，瞬時に各館の書

＊　鹿児島純心女子短期大学生活学科生活学専攻生活ビジネスコース　（〒890-8525 鹿児島市唐湊４丁目22番１号）

誌情報へのアクセスを可能とした。さらにこのシステムがWebサービス上で提供されることにより，専用の文字情報によってのみ提供されるサービスの制約を取り払うこととなった。

しかしながら現在の電子図書館論が構想しているような，いながらにして所蔵された情報そのものを得るような段階へはいまだ到達していない。これは所蔵された情報を無制限に公開する権利を付与されないためであるが，さらにそれらを電子化し利用に供する膨大な作業量とそれによって発生するコストの問題でもある。

すでに多くの大学図書館などで提供されている電子ジャーナルのサービスでは，印刷物として流通する情報の電子化，およびその検索と提示のためのシステムの構築，電子化された情報を利用することによって発生する著作権使用料の管理までを一括のシステムに含めて利用し対価を支払うものである。そのため導入する図書館としてはその利用料という形での経済的なコストのみを負担し，それ以外の運営上の人的労力などの面におけるコストを高くすることなくサービスを提供することができる。これらのサービスは一部のジャーナルなどの雑誌類を対象とするものが多いが，Internet上で運営されるいわゆるオンライン大学の図書館のように多くのハードカバーの書物まで電子化し提供される場合もある。また利用者を特定の者，大学図書館であればその大学に所属するものなどに限定しすることで無制限な著作権利用とならないようにされている。すなわち物理的な書籍を購入し利用者に提供する代わりに電子化された情報を提供するサービスを購入するものである。

このようなサービスで提供される電子化された情報は,物理的に市場で流通する書籍であり,学会や大学の研究紀要など一般には流通しない印刷物がそれらの商用サービスによって提供されることはない。本稿で考察の対象としている大学紀要など学内出版物は通常であれば，他の図書館への寄贈などの形をとり，物理的な移動を行うが，書庫のスペースの問題などにより所蔵数が減りつつあるのが現状である。したがってこれらを電子化し公開することは，利用者が空間的なあるいは物理的な制約なくこれらの情報を利用することを可能にするものである。さらに各館における公開がなされれば，元の情報が空間的に，あるいは地理的に物理的にどこに所蔵されているものか，ということを気にすることもなく利用者はその希少な情報を手にすることが可能となる。

以上のような意味で学内出版物を電子化し何らかの形で公開し提供することは必要かつ重要な意味を持つものであるが，電子化された情報が元の情報から抽出された部分であること，すなわち文書のタイトル，著者，あるいは数語のキーワードなどいわゆる書誌情報であることは，現実世界における図書館において図書目録のみで図書を検索することと同様のことをネットワーク上に展開したにすぎない。その書誌情報を元にして実際の所蔵資料を手に取りその中に求めている情報が含まれているかを確かめるという作業は，それがネットワーク上で行われたにせよ，検索されたものを印刷されたものであれ電子化されたファイルであれ同様である。

電子化された情報を公開されたネットワーク上に提供することの重要性は，必要な情報がどこに存在するか物理的な制約なしに，あるいは時間と空間の制約なしに知ることができることにこそある。コンピュータのような情報機器とネットワークによって構成されるシステムにあっては，既存の図書館で行われている利用形態・サービスの提供形態をただ情報機器を利用したものに置きかえるのではなく,IT/ICTによってのみ提供できるサービスの提供をそこに含

みこむことこそが電子化された図書館を構想することに大きな意味を持つものである。したがって学内出版などの電子化と公開を大学図書館の電子化における独自の作業であると考えれば，そこで公開される文書の内容に対する書誌情報のような一部分の情報提供ではない全文に対する情報検索が必要であるし，それによって得られた必要な文書へのネットワーク上での閲覧が可能であることが必要である。

これらのことから，電子化した学内出版物を蓄積し，それらの内容に対して何らかの形で検索を行うための機能を持たせる必要がある。またこれらのシステムは一般ユーザが利用することを前提とし，情報を公開する側・利用する側両方におけるデータの更新，ソフトウェアの設定，運用の容易さなどが求められるものである。本稿ではユーザビリティ[1]の高さ，管理運用の容易さといった観点に立った場合の，全文検索を前提とした学内出版物の公開を行う場合の情報の電子化保存形式について比較し考察を行う。

3．全文検索を提供する際の構成形式

全文検索を行うための構成として，文書の内容に対してインデックスを作成し高速な検索を行う namazu システム[2]と，PostgreSQL [3]などのデータベースシステムを検索エンジンに利用した場合について考察するが，それらのバックエンドに対して利用者が実際に検索を行うためのフロントエンドとしては，専用のアプリケーションからではなくサーバー側のWebサービス上で稼動するCGI（Common Gateway Interface）などを経由した問い合わせを前提とする。これは次のような理由からである。

第 1 にユーザビリティの問題である。namazu はもともとコマンドラインで用いるコマンドとして開発されており，すなわちキャラクターベースのソフトウェアである。またデータベースシステムにはそれぞれフロントエンドとして動作するツールが存在する。例えばPostgreSQLにおける psql コマンドや Tkライブラリを前提とした PgAccess などである。前者はnamazuと同様にキャラクターベースのソフトウェアである。これらのものはネットワーク上であればターミナルソフトウェアを利用することで，それらのソフトウェアがインストーされているコンピュータにtelnetで接続し，そのコマンドラインで起動，操作するように作られているものである。さらに namazu であればその引数として検索語句と検索に利用するインデックスファイルへのパス＝そのファイルがディスク上で保存されている場所を，psql では起動の際に接続するデータベース名，もしそれが設定されていればデータベースに接続するためのユーザー名を必要とする。起動後データベースに対して検索を行う際には一般的にSQL（Structured Query Language）と呼ばれる問い合わせ言語を使用するが，これを利用者自ら入力することになる。したがって提供される情報をこれらのソフトウェアあるいはコマンドを用いて検索を行うためには，キャラクターベースでのサーバーへのログイン方法，シェル環境でのコマンドの起動方法，さらにデータベースシステムがバックエンドの検索エンジンとして用いられている場合には，フロントエンドコマンドの操作方法と，SQLによる問い合わせの方法，SQLの文法を理解している必要があり，また出力結果もターミナルソフト上に表示されるため，その結果を用い，必要なファイルを自分が利用しているコンピュータに転送し，そこで表示するため

の手順を理解している必要がある。

またnamazuにおけるTkNamazuやsearch-sのようなGUIを前提としたソフトウェアを用いる場合であってもインデックスがディスク上にどこに存在するのか，そのパスを指定するためにはどのように表記すればよいのかの知識を必要とする。PostgreSQLにおけるGUIフロントエンドである，PgAccessを利用する場合でもSQLの知識が必要であるし，いずれにせよ利用者は通常のコンピュータを利用する以外のリテラシーを要求される。

第2にソフトウェアおよび公開するためのデータの管理・運用の問題である。上述のソフトウェアを利用するためにはキャラクタベースのものの場合，サーバーにログインする必要があるが，そのためには利用者ごとに当該サーバーに利用資格としてのアカウントを用意する，すなわちユーザー名とパスワードを登録する必要がある。これは不特定多数の利用者を前提とする場合には不適切である。公開に際して同一のユーザー名，パスワードをすべての利用者に使用させることは可能であるが，セキュリティ上不適切である。またログインシェルとして検索を実行するようなシェルスクリプトやプログラムを指定しその他の操作を行わせないような設定も可能であるが，これもセキュリティ的な問題と，そのようなスクリプトあるいはプログラムを準備するための提供者側の負担を増加させるものである。一方GUIベースのものであってもサーバー側のアカウントが必要であるのは変わりなく，またnamazuは基本的にそれが動作しているマシン上のインデックスしか検索できないため，TkNamazuやsearch-sをネットワーク上で一般公開した情報を検索するためのフロントエンドとしては利用できない。

第3にセキュリティ上の問題である。上述の2つの理由においてもセキュリティに関する問題があったが，さらに検索エンジンとして利用するバックエンドプロセスが公開するポートがセキュリティー上の欠陥をもたらす可能性が大きい。一般にバックエンドのプロセスと協働した処理が行われる場合，バックエンドのプロセスは特定の番号をふられたポートを用意し，そこに接続してきた他のプログラムとデータの授受を行う。このプロセスどうしのデータ授受の仕組みをインターネットソケットと呼ぶ。一方他のプログラムが同一のコンピュータ上にある場合にUNIX系のオペレーティングシステムでは，2つのプロセス間を結ぶ場合にUNIXドメインソケットと呼ばれる仕組みを用いる。後者を用いる場合にはCUIフロントエンドを利用する場合で述べたように，当該コンピュータのアカウント情報に基づいたユーザー認証が必要となるため相応のセキュリティが得られるが，前者を用いた接続の場合，予期せぬセキュリティ上の陥穽のために，正当な権限，正当な利用者以外の利用を許してしまう，いわゆるセキュリティーホールの危険性を高めることになる。したがって必要最低限のポートのみを開放し，不用不急のポートは閉鎖することがセキュリティを確保する上で常道であるが，バックエンドの検索エンジンプロセスに対して検索を命令するためにこのポートをインターネット上に開放することは危険性をはらむものである。

以上3点から，サーバーローカルな環境でCUI/GUIのフロントエンド（namazuにおけるnamazuコマンド，TkNamazu，searhch-s），あるいはUNIXドメインソケットでローカルな環境で検索を行う，または INETソケットを利用してバックエンドに対して検索を行うフロントエンド（PostgreSQLにおけるpsqlコマンドやPgAccess）を用い，インターネットに公開された環境で，不特定多数の利用者に対して情報を提供することは不可能ではないがセキュリティ

的にも運用の負担の面からも困難である。

一方CGIなどのサーバーサイドスクリプトによる検索では，利用者はHTMLブラウザを通して，スクリプトに対して検索語や検索対象の指示を行い，実際の検索を実行するのはサーバー上のスクリプトである。検索結果はまたHTTPサーバーを通じて利用者のブラウザへと送信される。利用者は直接サーバーにログインする必要がなくそのため利用者の管理の必要がない。また利用者は検索を実行するスクリプトあるいはプログラムに対して検索語や検索の振る舞いを決定するオプションを設定することが可能であるが，ログインした後に検索用のフロントエンドを利用することなどに比べれば利用者の自由度が低い分セキュリティ的には問題が少ない。

また利用者のユーザビリティの点からも，検索のフロントエンドにHTMLブラウザを用いることで，特定のソフトウェアや新たなインターフェースに対する習熟，SQLコマンドに関する知識など常用するソフトウェア以外の部分に対する学習が少なくてすむ。管理・運営についても，実際に検索を行うユーザーインターフェースを準備するという意味でサーバー側だけの設定によってこれを行うことができるというメリットがある。

4．検索エンジン形式の比較

CGIなどのサーバーサイドスクリプトによる検索を前提とした場合，利用者による操作と，サービスを提供するサーバー側のデータの流れは一般的に次の図1のようになる。namazuのような検索専用システムを使用する場合と，PostgreSQLのような汎用のデータベースシステムを使用する場合，公開するデータファイルの処理に相違点がある。

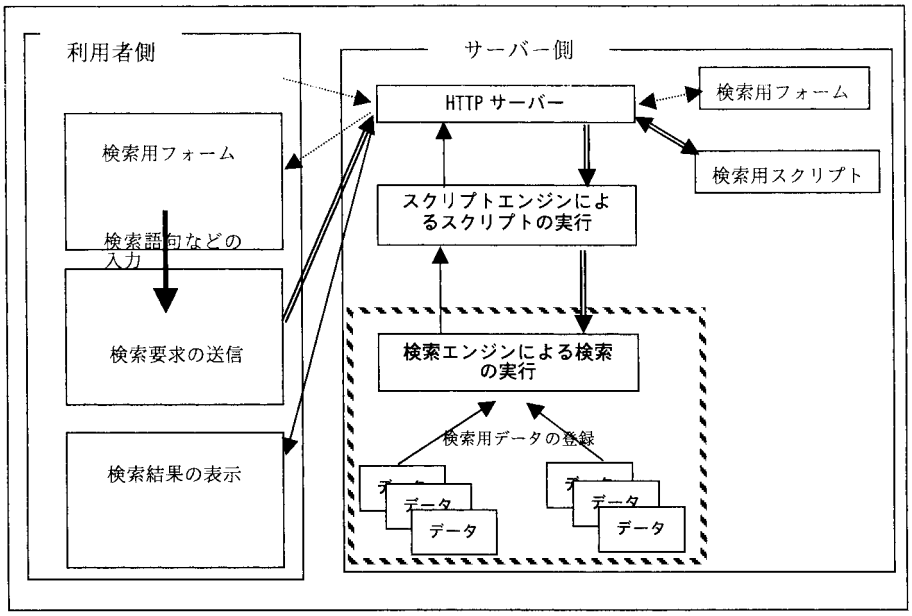

図1　全文検索処理の要求と結果の流れ

(1) 検索エンジンにnamazuを用いる場合

namazuシステムは大量の文書を高速に検索することを目的に開発されたソフトウェアであり，あらかじめインデックスを作成しておきWebブラウザ経由の検索要求に対して インデックスを検索し結果を返すものである。Webサービスを利用した全文検索を利用しているサイトでは非常に多くの利用されている[4]。

namazuを検索エンジンとして用いる場合には，namazuが検索を行うインデックスファイルを作成する必要があるがこれには mknmz コマンドを利用する。mknmzコマンドは基本的にテキストファイルのデータを対象にインデックスを作成するように作られており，通常のテキストファイルの他，HTML，HDML，Mail/Newsなどを対象にすることができる。また外部コマンドを利用することで，Adobe Acrobat の文書形式であるPDF（Adobe Portable Document Format）やMicrosoft Word の保存形式であるDOC形式の他，Microsoft Excelや，Microsoft Powerpointなど一般によく利用される電子化文書で，通常のテキスト形式ではない保存形式のファイルからテキストを抜き出しインデックスを作成することが可能である[5]。著作権的な観点から公開する文書情報の閲覧のみを可能としコピーや印刷に制限を設けたい場合には，PDF形式やWord文書の形式で保存する必要があるが，公開すべき情報が電子化され上述したような種類のデータファイルの形で保存されていれば，mknmzコマンドを実行するだけでインデックスファイルが作成され，検索を行うことが可能である。図 2 はnamazuを利用した場合の図 1 の点線で囲んだ部分の処理の流れの詳細である。

サービスを提供する側の作業は，HTMLを用いて検索フォームを作成することと，実際に検索を行うためのスクリプトやプログラムを作成する必要がある。このスクリプトやプログラムは，検索フォームに入力された検索語句および検索のオプションなどの情報を用いて実際に検索エンジンに対して問い合わせを行い，その検索結果を利用者に対して表示し，検索条件に適合したファイルを利用者に対して提示するための結果画面の作成を行うものである。

検索フォームの作成や検索結果の表示を行うためのスクリプトを作成するという作業手順も，namazuのパッケージに含まれるnamazu-cgiを利用することで作業量を最小限に押さえることが可能である。namazu-cgiはHTTPサーバーから呼び出されるCGIスクリプトであり，HTMLで記述されたWebページからnamazuの検索をコントロールすることができ，また検索結果をリンクとして表示することができるため，このリンクをたどることで利用者は目的の情報をおなじブラウザの画面の中で得ることができる。すなわち利用者側は新たなソフトウェアのインストールや設定，新たな操作に対する習熟なしに目的の情報を含む文章を自分が操作する画面に呼び出すことができる。

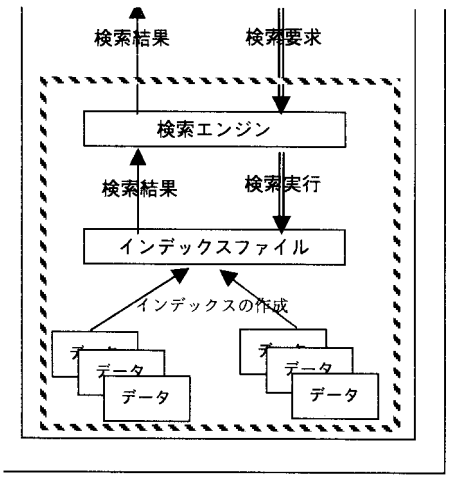

図 2　namazuを利用した際の検索の流れ

⑵ PostgreSQLなどデータベースシステムを用いる場合

データベースシステムを検索エンジンとして用いる場合には，まず公開する電子化された文書をデータベースへ登録することが必要となる。図３はデータベースを利用した場合の図１の点線で囲んだ部分の処理の流れの詳細である。

一般にデータベースが扱うことが可能なデータ形式は単純に言えばプレーンテキストか数値データであるが，これ以外のいわゆるバイナリ－データを登録することが可能なデータ型が用意されている場合がある。PostgreSQLの場合，BLOB（Binary Learge Object）型を用いることで非数値，非文字型のデータを扱うことができる。したがって公開するデータがテキスト形式以外の保存形式をとっていてもデータベースのデータとして扱うことが可能であるが，このようなバイナリー形式のデータの場合，データベースシステム自体がこれを検索の対象とすることができない。そのためこれらの形式で保存されたファイルの場合は，これを一度テキスト化し検索の対象として扱えるようにする必要がある。

上述のように公開するデータに何らかの制限を設ける場合，プレーンテキストで公開することは考えにくいため，公開用のデータとデータベースに登録するためのデータの２種類のものを用意することになる。公開する元となる文書がテキスト形式で作成されており，これを元にして公開用のデータを編集・作成する場合，元々のテキストを利用することで作業の手順を簡略化することができるが，原稿がWord形式などの場合にはテキスト形式への変換の作業を行わなくては成らない。これはExcelやPowerPointなどバイナリー形式でデータが作成される場合に共通である。

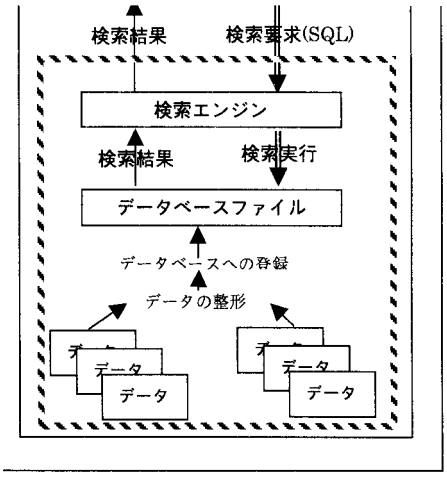

図３　データベースを利用した際の検索の流れ

検索サービスを提供する側の作業としてはデータの登録以前に，データを登録するためのデータベーステーブルを設計する必要がある。このテーブルには表示・公開用のデータおよび検索対象とするためにテキストに変換した全文データの登録を行い，全文データを登録するフィールドに対してデータベース内にインデックスを作成するなどの手順が必要である。またこの他に検索フォームの作成，検索の実行・表示用データへのリンクを含む検索結果のHTML化を行うスクリプトあるいはプログラムの作成が必要であるが，その中でデータベースシステムへの問い合わせを発行するSQL文の作成がさらに必要である。

５．ファイルの保存形式による作業手順の比較

公開するデータの元となる原稿の形式に対して全文検索機能をともなって公開するまでの作業の手順を概念化したものが次の図である。上段の「作業段階」の欄が公開までの一般的な作

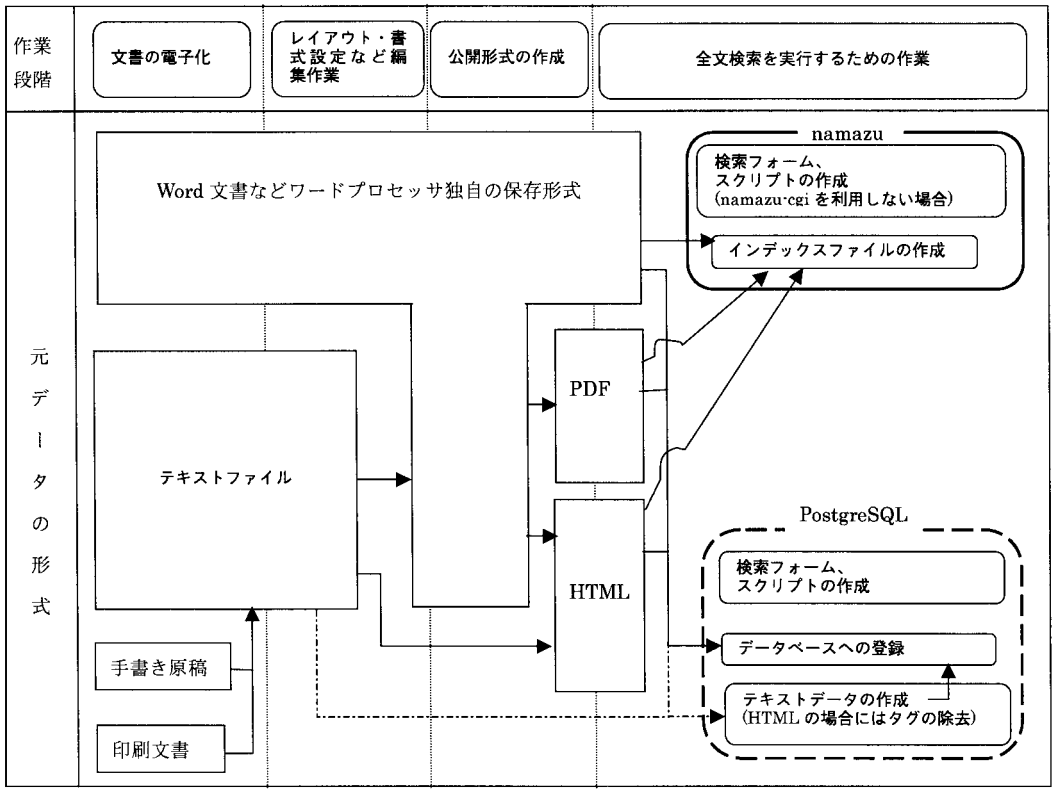

図 4　原稿のデータ形式と作業段階

業の過程を，下段の「元データの形式」の欄が，それぞれのデータ形式を処理し，検索可能な形式で公開されるまでの実際の過程である。

ここでは元データの保存形式を namazu で検索可能な形式で保存されているワードプロセッサなどの文書形式，HTMLを含むテキストファイル，手書き原稿，印刷文書の四種類についてそれぞれの処理の過程を段階的に追っている。文書の電子化では手書き原稿や印刷された文書を元にしてコンピュータを用いて処理可能な文字データを入力する作業を経てテキストファイルへ，またレイアウトや書式設定などの編集作業では文字のサイズや書体，1ページの行数・文字数などの情報を付加していく作業を想定し，公開形式の作成では，全文検索を可能にしながら利用者に対して文書の全文を公開するために通常良く用いられる 3 種類の文書形式を想定している。

この図では元のデータから最終的な文書の公開までの間のそれぞれの段階を結ぶ矢印が少ないほど作業の段階が少ないことを意味する。ただし作業の段階の少なさと，作業に要する時間と労力の少なさは必ずしも等価ではない。例えばテキストファイルからHTMLへの変換の際にすべてのHTMLタグを手作業で付加するとすれば膨大な作業量である。

従がってこの図からは元のデータがWord形式である場合ならWord形式で公開し，検索エンジンはnamazuを，検索を実行するためのスクリプトは namazu-cgiを利用する場合が最も作業

の段階が少ない。次に段階が少ないのはテキストファイルからHTMLへの変換，およびWord形式などからPDFやHTMLに変換する場合である。いづれも検索エンジンとしてnamazu，フロントエンドとしてnamazu-cgiを用いた場合である。手書き原稿，印刷文書を元データとした場合にはそれらを電子化された情報に変換する作業が発生するために作業段階が増える。

検索エンジンとしてPostgreSQLなどデータベースシステムを用いた場合には，全文検索を行うためにプレーンテキストの情報が必要となるため，公開形式からテキストデータを作成するか，あるいはテキストファイルが元のデータ形式の場合には変換前のデータを利用しこれを公開形式とともにデータベースへ登録する必要がある。そのため作業の段階は増加する。またnamazu における namazu-cgi のように，Webサービスを経由したフロントエンドが用意されているわけではないので，検索フォームや検索実行・結果の表示などを行うためのスクリプトの作成が必ず必要となる。

6．まとめ

公開された情報を全文検索し目的の文書を参照するためのフロントエンドとしてWebサービスを用いた場合の，提供者側の作業段階と公開する文書の電子化保存形式についていくつかの比較を行った。

前項で述べたように作業段階数のみに着目した場合，namazu-cgi をバックエンドに利用した場合が最も容易に全文検索機能を提供可能である。提供する情報の入稿時のデータ形式がインデックス作成コマンドである mknmz に対応する形式で，かつ利用者に対して閲覧や印刷，コピーなどの制限を考慮した上でそのデータ形式での公開が可能であれば，インデックスの作成以外の作業が不要となる。

一方，PostgreSQLなどのデータベースシステムを利用した場合であるが，データベーステーブルの設計の必要，公開形式とそのプレーンテキスト形式の二重登録の手間，検索フォーム・検索スクリプトの作成などの多くの作業段階と労力が必要となり，本稿で取り上げたような形式を公開形式とする場合にはメリットが見出し難い。

公開する情報の保存形式については，利用者側が目的の文書を閲覧する際にそれがどのようなコンピュータ環境であれ閲覧可能である文書の公開形式としてはHTML形式かPDF形式が適している。HTML形式の方が利用者の環境を選ばない。どのようなプラットホームであれWebサービスを利用したHTML文書を閲覧可能な環境であれば利用者は必要な情報を閲覧することができる。しかし文字のサイズやレイアウトまで細かく再現する用途には適さないし，上述のような閲覧時の制限を行うのは不可能である。

PDF形式の文書ではHTMLほどではないが多様なプラットホームにおいて閲覧環境が用意されている。またレイアウトや書式情報をどのようなプラットホームでも同じように閲覧可能に設計された文書形式であり，閲覧している文書を印刷したり，内容をコピーすることを制限することが可能である。またインターネット上での検索の実行と公開されたファイルへの参照であることを考えると保存された文書の容量が小さいほど公開形式として適していると考えられるが，この点でも書式情報をもった他の文書形式に比べれば最も小容量のファイルを作成で

きる(6)。

Word形式の文書を公開形式に用いる場合は，最も作業段階が少ないというメリットがあるが，HTML形式やPDF形式に比して対応するプラットフォームが少ないため，利用者の環境によっては提供されている元々のレイアウトや書式が表示不可能であるだけでなく，文書の閲覧そのものが不可能である場合も考えられる。すべての利用者にあまねくサービスを提供することが前提ならばまったくのプレーンテキストかHTML文書での情報提供が望ましいが，それでも何らかの条件，例えば印刷されたものと電子化されたものとの書式の一致の必要性など，があるのであればより対応するプラットフォームが多いPDF形式が適していると考えられる。

註

(1) 以下文中でユーザビリティという語を，「ハードウエアやソフトウエアの使いやすさ」という意味で用い，利用者側から判断する作業効率の良さや，満足度などの高さを表す。

(2) http://www.namazu.org/

(3) http://www.postgresql.org/

(4) 『日本語全文検索エンジンソフトウェアのリスト』(http://www.kusastro.kyoto-u.ac.jp/%7Ebaba/wais/other-system.html#prologue) ではnamazuを検索エンジンとして使用例として259のサイトがあげられている。

(5) テキスト変換用の外部コマンドとして wvWare (http://www.wvware.com) を利用した場合，Word形式の文書をさまざまなファイル形式に変更可能である。mknmz はwvWareがインストールされていればWord形式の文書をテキスト化し検索対象とすることができる。またExcel文書やPowerPointの文書にに対するxlHtmlも同様である。PDF形式の場合には Xpdfに含まれるpdf2text コマンドを用いることでテキスト化を行っている。

(6) A4白紙の文書1ページのWord文書は約19KB， PDF形式の場合は4KB。この文書（A4，11ページ，約10500文字）の場合，Word文書形式で約69KB，PDF形式で約42KB（PDF Writerを用いてデフォルトの設定でPDF化）