

データベース構築促進及び技術開発に関する報告書

インハウスデータベース用CGI作成の 調査研究

平成8年3月

財団法人 データベース振興センター

委託先 日本電子計算株式会社

KEIRIN

00

この事業は、競輪の補助金を受けて実施したものです。

序

データベースは、わが国の情報化の進展上、重要な役割を果たすものと期待されている。今後、データベースの普及により、わが国において健全な高度情報化社会の形成が期待される。さらに海外に対して提供可能なデータベースの整備は、国際的な情報化への貢献および自由な情報流通の確保の観点からも必要である。しかしながら、現在わが国で流通しているデータベースの中でわが国独自のものは1/3にすぎないのが現状であり、わが国データベースサービスひいてはバランスある情報産業の健全な発展を図るためには、わが国独自のデータベースの構築およびデータベース関連技術の研究開発を強力に促進し、データベースの拡充を図る必要がある。

このような要請に応えるため、(財)データベース振興センターでは日本自転車振興会から機械工業振興資金の交付を受けて、データベースの構築および技術開発について民間企業、団体等に対して委託事業を実施している。委託事業の内容は、社会的、経済的、国際的に重要で、また地域および産業の発展の促進に寄与すると考えられているデータベースの構築とデータベース作成の効率化、流通の促進、利用の円滑化・容易化などに関係したソフトウェア技術・ハードウェア技術である。

本事業の推進に当って、当財団に学識経験者の方々に構成されるデータベース構築・技術開発促進委員会(委員長 前山梨学院大学教授 蓼沼良一氏)を設置している。

この「インハウスデータベース用CGI作成の調査研究」は平成7年度のデータベースの構築促進および技術開発促進事業として、当財団が日本電子計算株式会社に対して委託実施した課題の一つである。この成果が、データベースに興味をお持ちの方々や諸分野の皆様方のお役に立てば幸いである。

なお、平成7年度データベースの構築促進および技術開発促進事業で実施した課題は次表のとおりである。

平成8年3月

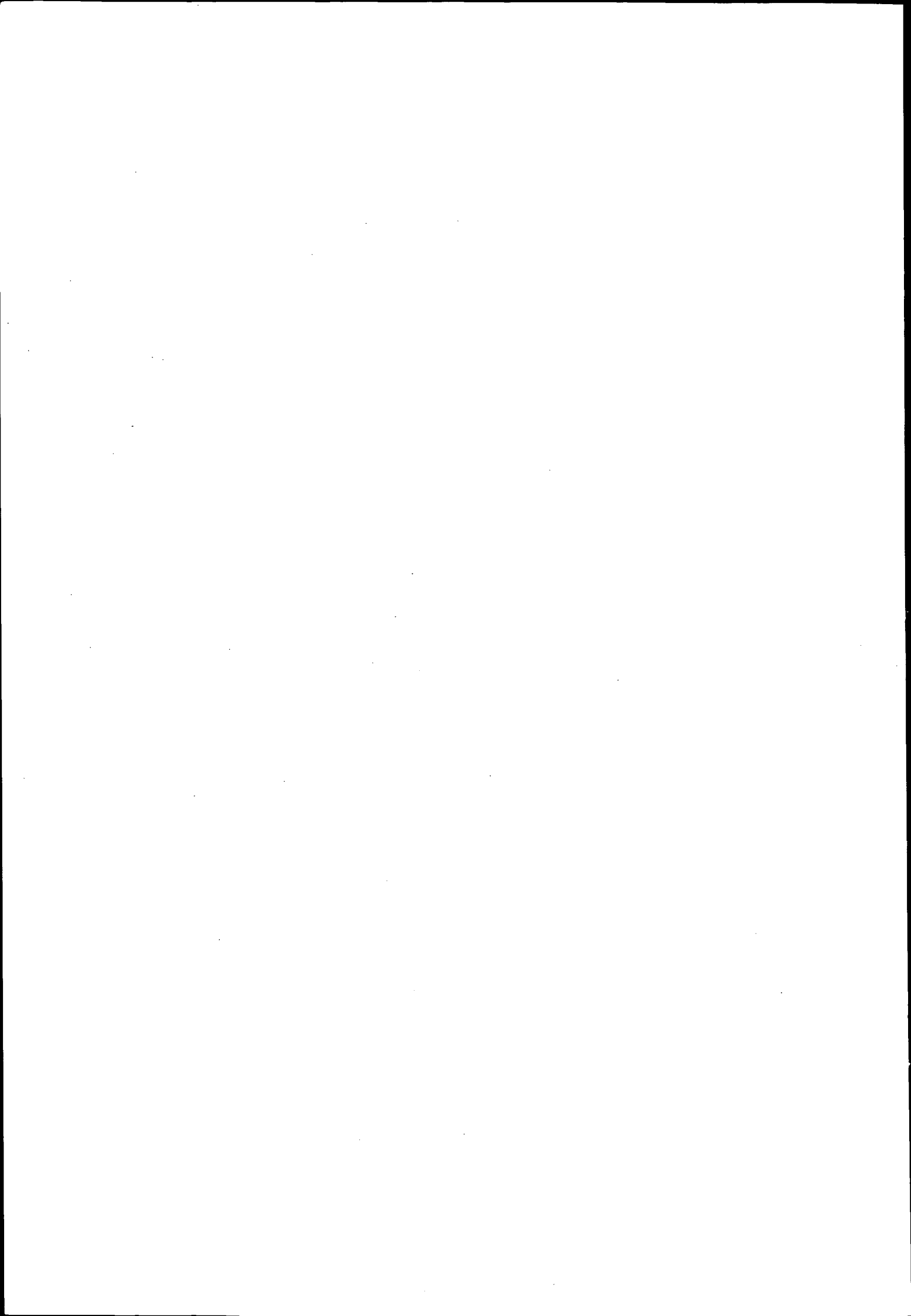
財団法人 データベース振興センター

平成7年度 データベース構築・技術開発促進委託課題一覧

分野	課題名	委託先
社会	1 法的データベースにおける多分野データベースの統合体的管理とオフ・オンラインの融合化に関する調査、研究 2 新聞記事分類キーワードの標準モデル構築と自動付与に関する調査研究	(株)日本法律情報センター (株)エレクトロニック・ライブラリー
中小企業振興 地域活性化	3 パソコンを用いた地図データベースの基礎構築 4 景観シミュレーション用樹木のデータベース構築 5 包装機械データベースの構築 6 新規事業創出支援のためのデータベース構築に向けた基礎調査 7 高効率化先進材料ファクトデータベースのパッケージ化 8 阪神・淡路大震災の情報デジタル・アーカイブ	(有)朝日データサービス (株)ストゥディオサカイ (社)日本包装機械工業会 (株)日本インテリジェントトラスト (財)次世代金属・複合材料研究開発協会 神戸マルチメディア・インターネット協議会
技術	9 M o s a i c の利用によるマルチメディアデータベース検索システムの構築 10 大規模データベースにおける構造化情報抽出方式の調査 11 モバイルデータベースシステムに関する調査研究 12 情報収集ロボットによるInternetでのWWW所在検索データベースの構築 13 インハウスデータベース用C G I 作成の調査研究	日本電子開発(株) (株)日本総合研究所 (株)イフ・アドバタイジング 日外アソシエーツ(株) 日本電子計算(株)

目 次

1. 調査研究の目的	1
2. 実験システムの概要	2
2. 1 設計方針	2
2. 2 ハードウェア・システムの構成	6
2. 3 ソフトウェア・システムの構成	8
3. ワークステーション用C G I（日本語対応）の詳細	10
3. 1 データ整備	10
3. 2 検索システム	13
3. 2. 1 検索ロジック	13
3. 2. 2 プログラムの表記と説明	15
3. 2. 3 画面の構成	28
4. パーソナル・コンピュータ用C G I（英語対応）の詳細	34
4. 1 データ整備	34
4. 2 検索システム	38
4. 2. 1 検索ロジック	38
4. 2. 2 プログラムの表記と説明	38
4. 2. 3 画面の構成	51
4. 2. 4 W e bサーバの構築	58
5. 成果と課題	59
5. 1 実験システムの評価	59
5. 2 成果	60
5. 3 課題	61



1. 調査研究の目的

インターネットの普及にともないWWWサーバ（Webサーバ）による情報提供は急速に進んでおり、企業のみならず行政機関を含む公的機関も、従来の紙による情報提供からインターネットでの情報提供を積極的に行っている。

ネットワークを使用することで、多様で最新の情報を迅速にかつ低コストで多数のユーザに対してサービスが可能なことから、新しい情報提供手段の一つとしてあげられ、さらに静止画像、動画や音声等のネットワークでは高コストでしかも困難であったマルチメディア対応も可能となっている。

またこの機能を利用して、最近ではイントラネットと呼ばれる社内LANでのWebサーバを利用した組織内向け情報サービスが盛んとなり、一層の情報提供機能の拡充に拍車をかけている。

これらのWebサーバを利用したサービスでは、提供されるデータはHTMLエディタ等により容易に作成でき、また階層的な構造を持つファイルのリンクによる他の情報のアクセスやURLによる他の情報源への到達が可能となり、さらに、これらの検索・情報を表示するソフトウェアであるブラウザは、操作が簡単で標準化がなされていることから、初心者でも容易に利用することができる。

しかしながら、Webサーバでの情報提供が進むにつれ、従来の表層的なHTML形式によるデータのみならず、大量のデータの提供や、これらのデータをユーザが指定したキーワードで検索可能とする機能が必要となってきた。また、組織内では共有するデータをデータベース化していることが多く、これらのデータを容易に、他の部門や外部へ提供する傾向が強くなってきた。

そこで、本調査研究では、既に構築済みのデータベースのデータを利用し、これらのデータファイルとWebサーバ間にCGI（Common Gateway Interface）を組み込み、容易にインハウスデータベースの内容をサーバで可能とする機能を調査研究し、その実現を図るとともに、この成果を直ぐに応用できることを実証するため、(財)データベース振興センターで作成している「データベース台帳総覧」等のデータを使用して、インハウスデータベースの利用普及を促進することとした。

2. 実験システムの概要

2. 1 設計方針

調査研究の実施にあたり、作成される実験システムの設計方針を以下のように定めた。

(1) Webサーバの構成

一般的な検索機能を有するWebサーバでは、下図のようにHTTPサーバ・ソフトウェアに付属して、CGI、データベース・ソフトウェアとCGIとの連携を図るゲートウェイ・ソフトウェアがあり、さらにそのゲートウェイ・ソフトウェアにデータベース・ソフトウェア（検索、表示、更新）が接続されている。

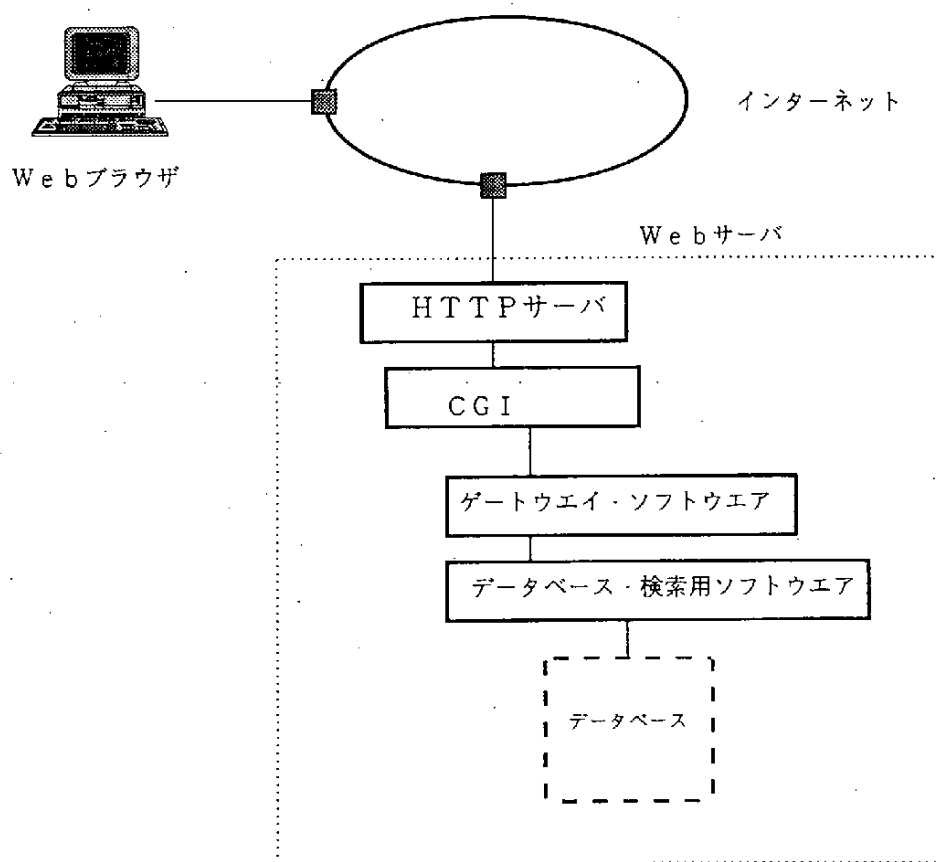


図1. 1 一般的なCGI機能の構成図

これらのゲートウェイ・ソフトウェアとデータベース・ソフトウェアは高価であるとともに、それぞれのゲートウェイやデータベース・ソフトウェアに依存したCGIを作成することが必要であることから、今回の調査研究では、これらのソフトウェアを使用せずに、CGIに検索・表示機能を持たせ、パソコンに構築されたデータベースに蓄積されたデータを一度ファイルに落とし、このファイルをCGIが検索等の処理をするようにした。

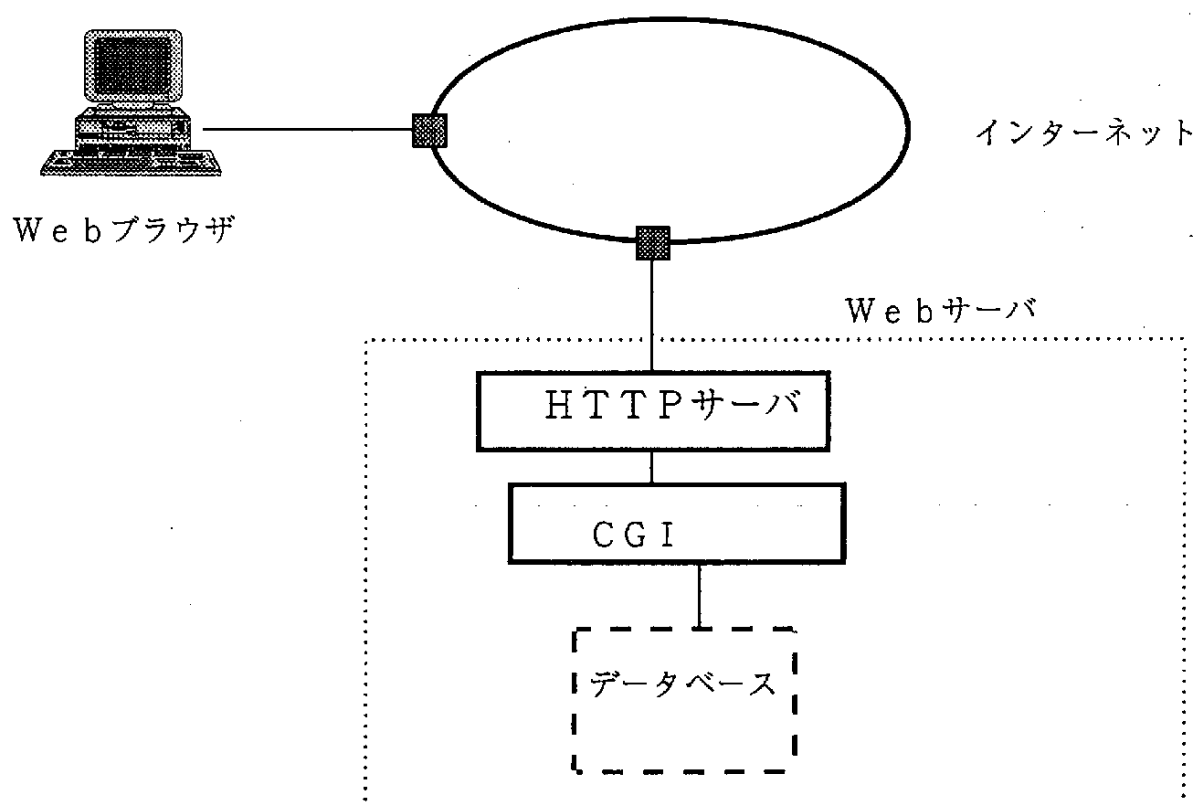


図 1. 2 実験用 CGI 機能の構成図

(2) Webサーバの構築

Webサーバは、ワークステーション用のものとパーソナル・コンピュータ（Macintosh）用のものとの二つを作成し、ワークステーション用では日本語（漢字データ）を対象にし、パーソナル・コンピュータ用では英語を対象データとした。

- ・ワークステーション用 `httpd` (CERN)
- ・パーソナル・コンピュータ用 `MacHTTP`

(3) 検索機能等の選定

①検索機能

- ・ 検索条件については、論理和、論理積及び否定の指定を可能とし、検索キーワードは最大5語の組み合わせを可能とした。
- ・ 検索キーワードの一致処理は、全文を対象に1文字ずつ比較するため、前方一致や後方一致の記述はしない。検索キーワードの語の中にはワイルド・カード等は使用しない。
- ・ アルファベットの大文字・小文字の処理については、全て小文字とする。

②論理式の表記方法

論理式の表記方法は様々なものがあるが、この実験では論理式の複雑な表現を避けるためと入力を単純化するため、以下の表記とした。

- ・論理積の表現 検索キーワードの前に「+」を付与する。
 (当該のレコードに必ず検索キーワードが存在すること)
 - ・論理否定 検索キーワードの前に「-」を付与する。
 (当該のレコードに必ず検索キーワードが存在すること)
 - ・論理和 とくに付与はしない。
 (当該のレコードに検索キーワードが存在すること)
 - ・その他 検索キーワードの間はスペースを1つ入れる。
-
- ・表記例
 - ・単純な検索 日本電子計算
 +日本電子計算 (結果は「日本電子計算」と同じである)
 -日本電子計算
 - ・複雑な検索

日本電子計算 医薬 医療
 (日本電子計算 or 医薬 or 医療)

+日本電子計算 医薬 医療
 (日本電子計算 and 医薬) or (日本電子計算 and 医療)

+日本電子計算 +医薬 医療
 (日本電子計算 and 医薬 and 医療)

+日本電子計算 +医薬 +医療
 (日本電子計算 and 医薬 and 医療)

-日本電子計算 +医薬 医療
 not日本電子計算 and (医薬 and 医療)

-日本電子計算 -医薬 医療
 (not日本電子計算 and not医薬) and 医療

③表示機能

- ・表示は、検索結果の概要を示す画面と、この概要の一部をクリックすることで、当該の詳細データが表示される画面の二つとする。
- ・HTML形式への変換は、それぞれの処理に従いそのつど変換することとし、検索対象ファイル等については事前のHTML化作業は行わない。

(4) ブラウザの選定

Mosaic,Netscape等の一般的なブラウザを対象とし、日本語の入力、出力が可能なバージョンのものを想定した。

(5) 対象とするデータの選定

組織内で利用されるデータベースは、顧客管理、経理処理等の数値データを主体にした事務処理で多く利用されているが、最近ではワープロの普及にともない文書情報(文字情報が主体)の電子化が進展し、これらを収集しデータベース化している。

そこで、本調査研究ではこれらの文書情報で構成されるデータベースをソースとするため、一般的な文書形式である「データベース台帳総覧」(以下台帳データという)及びこの台帳の中から日本製データベースを英訳した「Directory of Japanese Databases」(以下ディレクトリ・データという)を対象とした。

処理の手順としては、「データベース台帳総覧」等のデータを市販のデータベース・パッケージの「ファイル・メーカー・プロII」でデータベース化し、これを原データとした。

このデータベースから検索・表示に必要なデータ項目を抽出し、Web サーバ用の

データ作成のためにVisual Basic、Macperlで、以下の加工編集を行った。

・台帳データ（日本語）

①検索部分と表示部分の区分けとそのファイル作成

- ・ 検索対象及び表示対象になるデータ項目の選定

②表示ファイルの編集

- ・ データ項目の分割（データ項目の先頭に” # ”を付与）

③検索ファイルの編集

- ・ データ項目の分割（データ項目の先頭に” # ”を付与）
- ・ 英文字の統一（大文字をすべて小文字に統一）

・ディレクトリ・データ（英語）

- ・ データ項目の分割（データ項目の先頭に” # ”を付与）
- ・ 表示ファイルと検索ファイルは同一のものを使用する。
- ・ 大文字・小文字の統一は検索CGI内で行う。

2. 2 ハードウェア・システムの構成

以下のハードウェアをLAN上で使用した。

- | | |
|----------------|--|
| ・インハウス・データベース用 | Macintosh Quadra610 32MB
(漢字TALK 7.1) |
| ・データ加工用 | HP-600C
(Windows 3.1) |
| ・Webサーバ用 | |
| ・WS | SPARC station 5
・ CPU micro SPARC-II (85MHZ)
・ MAIN MEMORY 32MB
・ HARD DISK 2.1GB
(Solaris 2.4) |
| ・パソコン | Macintosh Quadra610 32MB
(漢字TALK 7.1) |
| ・ネットワーク | LAN Ethernet 10B
インターネット接続 64k |

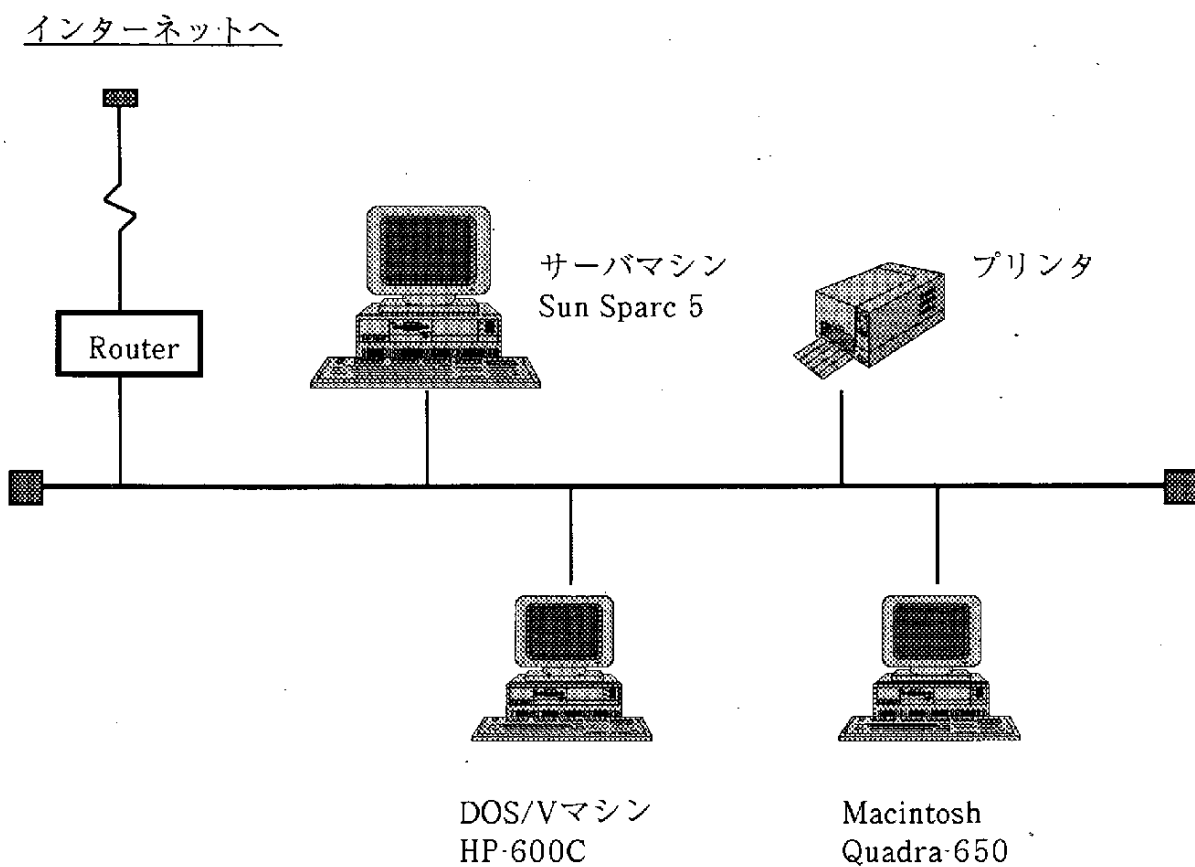


図 1 . 4 ハ ー ド ウ エ ア 構 成 図

2. 3 ソフトウェア・システムの構成

利用及び作成したソフトウェアは以下通りである。

- ・ インハウス・データベース (Macintosh)
 - File-Maker Pro-II
 - Mac-Ms-dos変換ソフトウェア
- ・ データ加工 (DOS/V)
 - Visual Basic (新規作成 作成プログラム 2 本)
- ・ データ加工 (Macintosh)
 - MacPerl (新規作成 作成プログラム 2 本)
- ・ W e b サーバ
 - ・ W S
 - H T T P server CERN httpd 3.0
 - CGI 用スクリプト言語 jperl version 4.019 + 1.3(SJIS)
 - perl CGI ライブラリ cgi-lib.pl 1.8
 - ・ データベース検索 (新規作成)
 - ・ データ表示 (新規作成)
- ・ パソコン
 - H T T P server MacHTTP 2.2
 - MacPerl 5.0.7
 - ・ データベース検索 (新規作成)
 - ・ データ表示 (新規作成)

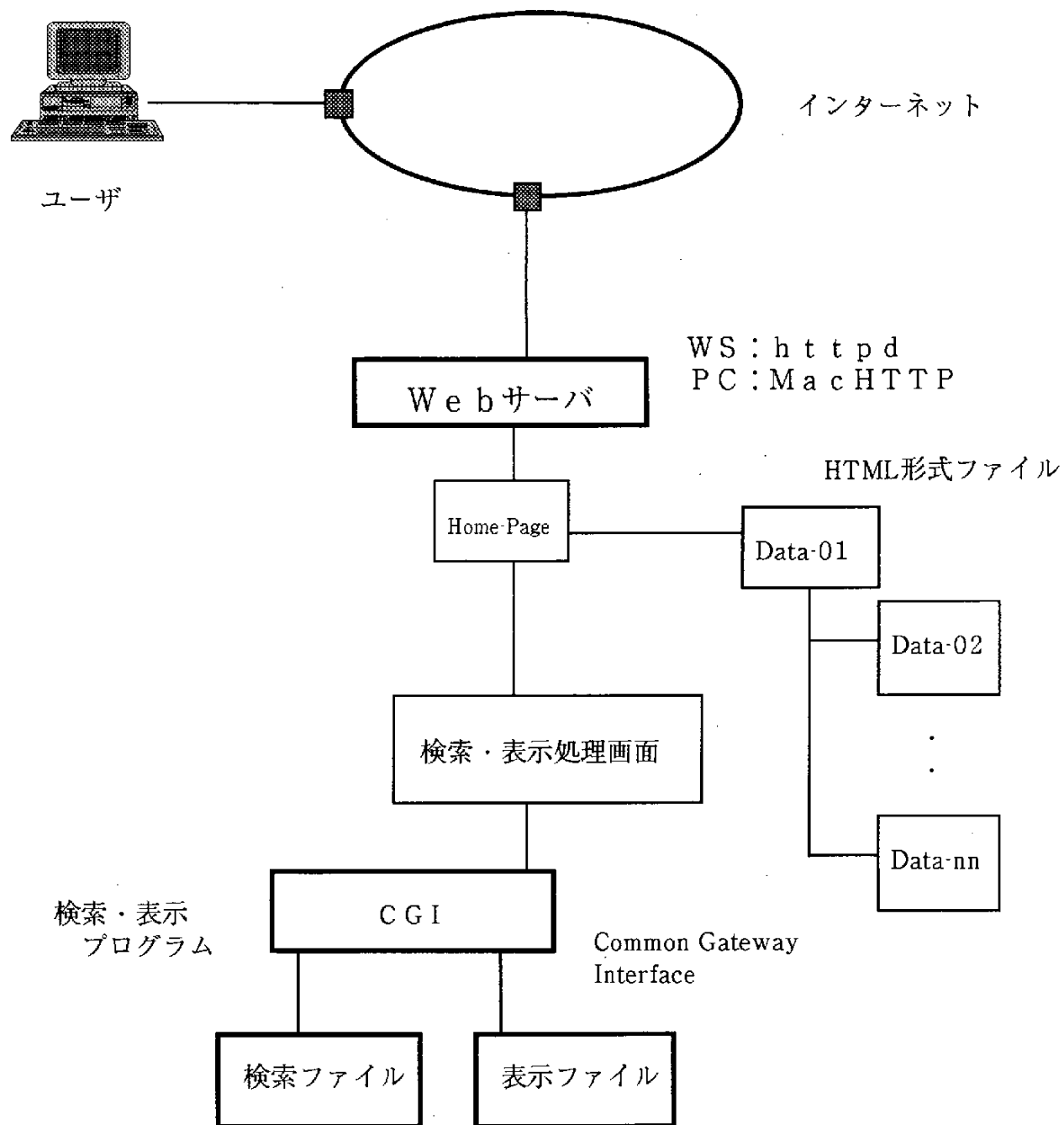


図 1. 3 実 験 シ ス テ ム 図

3. ワークステーション用C G I（日本語対応）の詳細

3. 1 データ整備

使用したデータ「データベース台帳総覧」は以下の構成となっている。

(1)原データの構成

インハウス・データベースから表示・検索に必要なデータ項目を以下のとおり抽出した。

- (01)レコード番号：
- (02)データベース名：
- (03)プロデューサ名：
- (04)国名：
- (05)内容：
- (06)タイプ：
- (07)言語：
- (08)サービスシステム名：
- (09)サービス企業名：
- (10)申告企業名：
- (11)住所：
- (12)担当部署：
- (13)キーワード群：

- ・(01)のレコード番号はファイル内ではユニークである。
- ・レコード番号を除いて、他のデータ項目は全て2バイト系で表記されている。
- ・データ項目は全てTabコードで区分されている。

(2)表示ファイルの処理と例

上記のデータ構成から必要なデータ項目を抽出し、データ項目の先頭に”#”を付与した。

#4001

#The Palm Beach Post

#The Palm Beach Post

#米国#「Palm Beach Post」は首都圏地域のスポーツ、ビジネス・産業を完全に収録する一般日刊紙です。本紙で扱われる主な分野は、航空・航空宇宙、コンピュータ・通信技術、農業および金融です。この地域に所在している大企業にはPratt & Whitney社およびAtlantic Sugar Millがあります。全コラムの全文、投書、論説（地域人の訃報を含む）、葬儀告知、地域・ニュースの通信社記事が本ファイルに収録されます。（PAPERS参照）

#1 一般。（新聞）

#報道

#文書（全文）。

#英語。

#DIALOG

#Knight-Ridder Information, Inc.

#ON.

#株式会社 紀伊國屋書店

#156 東京都世田谷区桜丘5-38-1

#電子情報部

#4109

#半導体（一般電子部品含む）データ・ベース

#電子デバイス情報サービス（株）

#日本

#半導体及び一般電子部品を含む電子デバイスに関わる情報をデータベース化して提供する。国内・海外の電子デバイスメーカより提供される、製品に関するデータシート・データブックの抄録・原文情報を網羅的に蓄積するほか、新製品ニュース、開発ニュース、保守・廃止品情報、イベント案内、メーカ・代理店ガイド等、広範な情報を蓄積し、提供する。

#2 自然科学・技術。（電気／電子／情報）

#電気・電子産業

#文書（全文）．文書（抄録）．図表．

#日本語．英語．

#ELISNET

#電子デバイス情報サービス（株）

#ON. OFF. (MT. CD. COPY.)

#電子デバイス情報サービス株式会社

#102 東京都千代田区平河町2-4-14 平河町KSビル

#企画部企画担当

(3)検索ファイルの処理と例

画面ファイルと同様の処理を行うとともに、英文字についてはすべてを小文字化した。

#4001

#the palm beach post

#the palm beach post

#米国

#「palm beach post」は首都圏地域のスポーツ、ビジネス・産業を完全に収録する一般日刊紙です。本紙で扱われる主な分野は、航空・航空宇宙、コンピュータ・通信技術、農業および金融です。この地域に所在している大企業にはpratt & whitney社およびatlantic sugar millがあります。全コラムの全文、投書、論説（地域人の訃報を含む）、葬儀告知、地域・ニュースの通信社記事が本ファイルに収録されます。（papers参照）

#1 一般。（新聞）

#報道

#文書（全文）.

#英語.

#dialog

#knight-riddler information, inc.

#on.

#株式会社 紀伊國屋書店

#4109

#半導体（一般電子部品含む）データ・ベース

#電子デバイス情報サービス（株）

#日本

#半導体及び一般電子部品を含む電子デバイスに関わる情報をデータベース化して提供する。国内・海外の電子デバイスメーカより提供される、製品に関するデータシート・データブックの抄録・原文情報を網羅的に蓄積するほか、新製品ニュース、開発ニュース、保守・廃止品情報、イベント案内、メーカ・代理店ガイド等、広範な情報を蓄積し、提供する。

#2 自然科学・技術。（電気／電子／情報）

#電気・電子産業

#文書（全文）. 文書（抄録）. 図表.

#日本語. 英語.

#elisnet

#電子デバイス情報サービス（株）

#on.off. (mt.cd.copy.)

#電子デバイス情報サービス株式会社

3. 2 検索システム

3. 2. 1 検索ロジック

- (1)本システムはインハウス向けという事もあり、検索対象となるデータは比較的小規模(2～3MB)なテキストのみのものを前提とする。全データを主記憶にロードする為、データ規模は主記憶容量に依存するが、読み込みのオーバーヘッドを踏まえると実用的には9～10MB程度が限界と考えられる。
- (2)実際の検索は主記憶にロードされた全件データ(検索ファイル)を対象に perl の関数 grep を用いてユーザによって指定された検索キーワードそれぞれについて行ない、得られた集合間の論理演算によって検索結果を導出する。
- (3)検索対象となるテキストデータベースは1件1レコードで構成され、項目は任意のセパレータ(“#”)により区切られているものとする。
- (4)検索の実質的な意味は1レコードを1つのオブジェクトとしてみなし、1つのキーワードに対応する(ヒットした)レコードオブジェクトからなる部分集合を生成することである。
- (5)最終的に得られた検索結果から、必要なデータ項目を抽出し、HTML形式に変換の後、クライアントに返す。詳細画面の検索に必要なレコード番号も出力対象とする。
- (6)詳細画面の生成は、クライアントのクリックにより与えられたレコード番号により、表示ファイルを検索し、必要な項目を抽出・編集したのち、クライアントに返す。

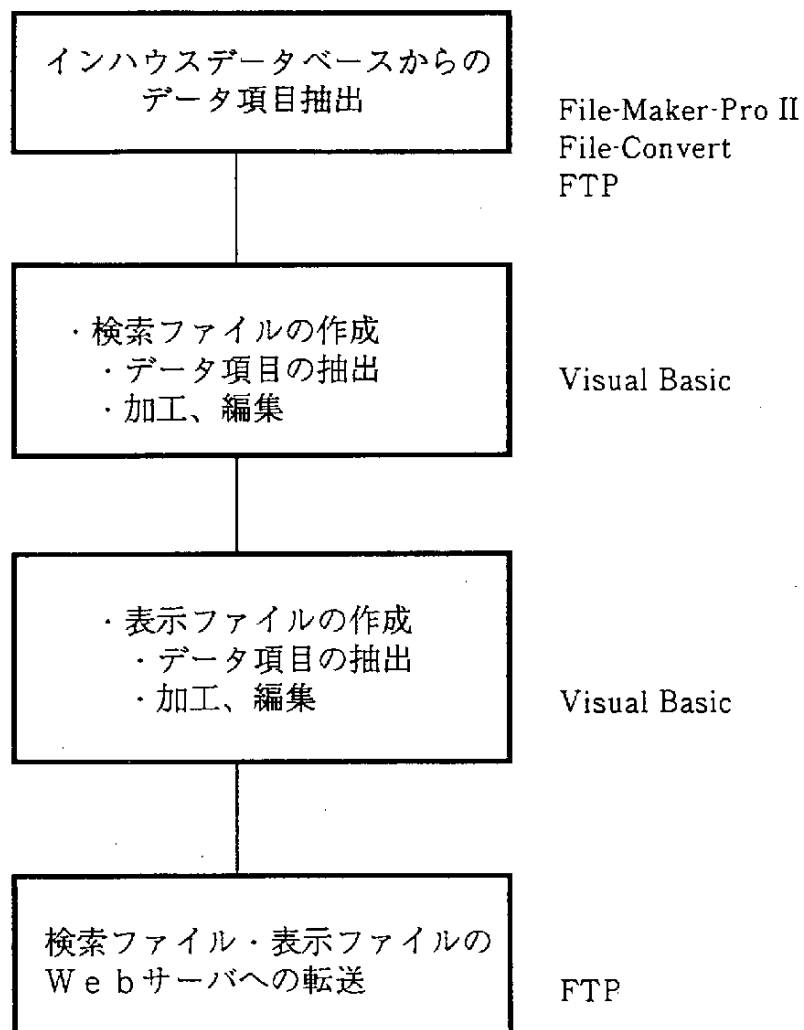


図 3 . 1 データ処理の流れ

3. 2. 2 プログラムの表記と説明

(1) 検索プログラム

```
#!/usr/local/bin/jperl
```

```
# CGI ライブラリの読み込み
```

```
require "cgi-lib.pl";
```

```
# CGI パラメータの読み込み
```

```
&ReadParse(*input);
```

```
# 環境設定
```

```
# リモートホスト
```

```
$RHOST = $ENV{'REMOTE_HOST'};
```

```
# ログファイルパス
```

```
$log = "/usr/tmp/dbgrep.log";
```

```
# データベースパス
```

```
$db = "/mnt/WWW/daityo/daityo";
```

```
# 検索キーワード
```

```
$tquery = $input{'query'};
```

```
# @tquery = @ARGV;
```

```
# 検索キーワードをSHIFT JIS に変換
```

```
open(QRY, "echo '$tquery' | /usr/local/bin/nkf -x") || die "open jconv $!\n";
```

```
read(QRY, $_, 4096);
```

```
$_ =~ s/ / /g;
```

```
close(QRY);
```

```
# 検索キーワードを配列に変換
```

```
@querylog = @query = split;
```

```
# データベースの読み込み
```

```
open(DB, "$db") || die;
```

```
while (<DB>) {
```

```
    push(@db_all, $_);
```

```
}
```

```
close(DB);
```

```

# 検索キーワードの分類
for ($i=0; $i<=$ # query; $i++) {
# 先頭に「+」がある場合。
if ( $query[$i] =~ ^+ / ) {
    substr($query[$i], 0, 2) = "";
# andq → AND 検索条件
    push(@andq, $query[$i]);
} elseif
# 先頭に「-」がある場合。
    ( $query[$i] =~ ^- / ) {
    substr($query[$i], 0, 2) = "";
# notq → NOT 検索条件
    push(@notq, $query[$i]);
# 普通のキーワード
} else {
# orq → OR 検索条件
    push(@orq, $query[$i]);
}
}

# AND 検索条件がある場合
if ( $ # andq >= 0 ) {
    for ($i=0; $i<=$ # andq; $i++) {
# 一時集合の生成
        @tmpresult = grep(/$andq[$i]/, @db_all);
        local(%mark);
        if ( $i == 0 ) {
# 最初は一時集合が AND 条件の結果集合
            @andresult = @tmpresult;
        } else {
# 一時集合との AND 演算
            grep($mark[$_]++, @tmpresult);
            @andresult = grep(@mark[$_], @andresult);
        }
    }
}

# NOT 検索条件がある場合

```



```

if ( $# notq >= 0 ) {
  for ($i=0; $i<=$# notq; $i++) {
    # 一時集合の生成
    @tmpresult = grep(!/$notq[$i]/, @db_all);
    local(%mark);
    if ($i == 0) {
      # 最初は一時集合が NOT 条件の結果集合
      @notresult = @tmpresult;
    } else {
      # 一時集合との AND 演算
      grep($mark[$_]++, @tmpresult);
      @notresult = grep(@mark[$_], @notresult);
    }
  }
}

# AND/NOT 条件中間結果の生成
# AND 検索条件と NOT 検索条件が共にあった場合
if ( $# andq >= 0 && $# notq >= 0 ) {
  local(%mark);
  # AND 検索条件の結果集合と NOT 検索条件の結果集合との AND 演算
  grep($mark[$_]++, @andresult);
  @anresult = grep(@mark[$_], @notresult);
}

# AND 検索条件だけがあった場合
if ( $# andq >= 0 && $# notq < 0 ) {
  @anresult = @andresult;
}

# NOT 検索条件だけがあった場合
if ( $# andq < 0 && $# notq >= 0 ) {
  @anresult = @notresult;
}

# OR 検索条件があった場合
if ( $# orq >= 0 ) {
  # AND 検索条件か NOT 検索条件が指定されていた場合
  if ( $# andq >= 0 || $# notq >= 0 ) {

```

```

for ( $i=0; $i<=$ # orq; $i++ ) {
#       一時集合の生成
    @tmpresult = grep(/$orq[$i]/, @db_all);
    local(%mark);
    grep($mark$ _]++, @tmpresult);
#       AND/OR 条件中間結果との AND 演算
    @orresult = grep(@mark$ _], @anresult);
    if ( $i == 0 ) {
#       最初はそのまま最終結果へ代入
        @theresult = @orresult;
    } else {
#       最終結果との OR 演算
        @theresult = &merge(@orresult);
    }
} else {
#   AND 検索条件か NOT 検索条件が指定されていなかった場合(単純 OR 検索)
    for ( $i=0; $i<=$ # orq; $i++ ) {
#       一時集合の生成
        @tmpresult = grep(/$orq[$i]/, @db_all);
        if ( $ # theresult < 0 ) {
#       最終結果が空ならばそのまま最終結果へ代入
            @theresult = @tmpresult;
        } else {
#       最終結果との OR 演算
            @theresult = &merge(@tmpresult);
        }
    }
} else {
#   AND/OR 検索条件しか指定されていなかった場合
    if ( $ # anresult >= 0 ) {
        @theresult = @anresult;
    }
}

# ヘッダの出力
&print_header;
# 検索条件式の出力

```

```

&print_query;
if ( $# theresult < 0 ) {
#   該当なしの場合
&print_noresult;
} else {
#   検索結果の出力
&print_data;
}
#   トレイラの出力
&print_trailer;
#   ログの出力
&writelog;

#   デバッグ用
sub printno {
local(@result) = @_;
local($i, $no, @body) ;
for ($i=0; $i<=$# result; $i++) {
($no, @body) = split(' # ', $result[$i]);
printf "%2d:%04d¥n", $i,$no;
}
}

#   OR 演算
sub merge {
local(@tmpresult) = @_;
local(%mark1, %mark2, @result1, @result2, @result3);
grep($mark1{$_}++, @tmpresult);
@result1 = grep(@mark1{$_}, @theresult);
@result2 = grep(!@mark1{$_}, @theresult);
grep($mark2{$_}++, @theresult);
@result3 = grep(!@mark2{$_}, @tmpresult);
@theresult = "";
@theresult = @result1;
push(@theresult, @result2);
push(@theresult, @result3);
@theresult;
}

```

検索条件式の出力

```
sub print_query {
    local($i, $baseq);

    for ($i=0; $i<=$# andq; $i++) {
        if ( $i != 0 ) {
            $baseq .= " AND " ;
        }
        $baseq .= $andq[$i] ;
    }
    for ($i=0; $i<=$# notq; $i++) {
        if ( length($baseq) > 0 ) {
            $baseq .= " AND " ;
        }
        $baseq .= "NOT $notq[$i]" ;
    }
    if ( $# orq >= 0 ) {
        for ( $i=0; $i<=$# orq; $i++ ) {
            if ( length($lastq) > 0 ) {
                $lastq .= " OR " ;
            }
            if ( length($baseq) > 0 ) {
                $lastq .= "( $baseq ";
                $lastq .= " AND $orq[$i] ) ";
            } else {
                $lastq .= $orq[$i] ;
            }
        }
    } else {
        $lastq = $baseq ;
    }
    print "<hr>";
    print "検索式： $lastq¥n" ;
    print "<hr>";
    $all = $# db_all + 1 ;
    $hit = $# theresult + 1 ;
    print "該当 $all 件中 $hit 件。¥n";
    print "<hr>";
}
```

ログの出力

```
sub writelog {  
    open(LOG, ">>$log") || die "open log $log $!¥n";  
    ($sec,$min,$hour,$mday,$mon,$year,$yday,$isdst) = localtime(time);  
    $mon++;  
    printf LOG "%s/%s/%s %s:%s ", $year, $mon, $mday, $hour, $min ;  
    print LOG " $RHOST <$ # theresult> ";  
    foreach $i (@querylog) { print LOG "$i"; }  
    print LOG "¥n";  
    close(LOG);  
}
```

ヘッダの出力

```
sub print_header {  
    print &PrintHeader;  
    print "<HTML>¥n";  
    print "<HEAD><TITLE>DATABASE LIST</TITLE></HEAD>¥n";  
    print "<BODY>¥n";  
}
```

トレイラの出力

```
sub print_trailer {  
    print "</BODY>¥n";  
    print "</HTML>¥n";  
}
```

```
sub print_noresult {  
    print "該当データはありません。¥n";  
}
```

検索結果の出力

```
sub print_data {  
    # 最終結果について  
    for ($i=0; $i<=$ # theresult; $i++) {  
        ($dbno, $dbname, $producer, $country, $character, $field, $keyword, $type, $lang,  
        $sysname, $company, $online, $state, $address, $dept) = split(' # ', $theresult[$i]);  
        # 詳細表示CGIへのリンク・パラメータとしてDB番号を渡す  
        print "<A HREF=¥"/cgi-bin/dbdata?$dbno¥"><br>¥n";  
    }  
}
```

```
print "$dbno<br>¥n";  
print "$dbname<br>¥n";  
print "$producer<br>¥n";  
print "$country</A><P>¥n";
```

```
}
```

```
|
```

```
*****
```

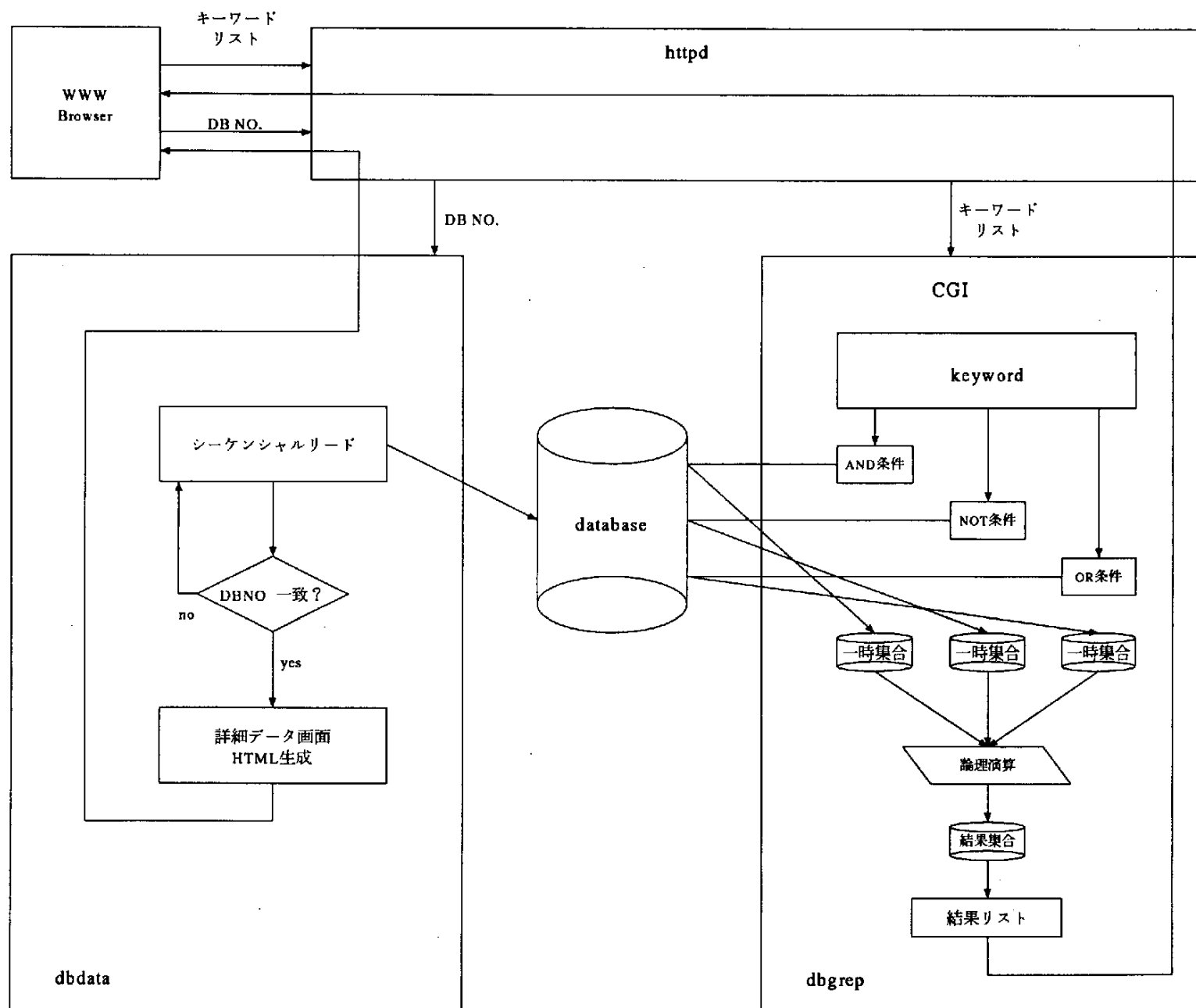


図3.2 検索及び表示処理フロー

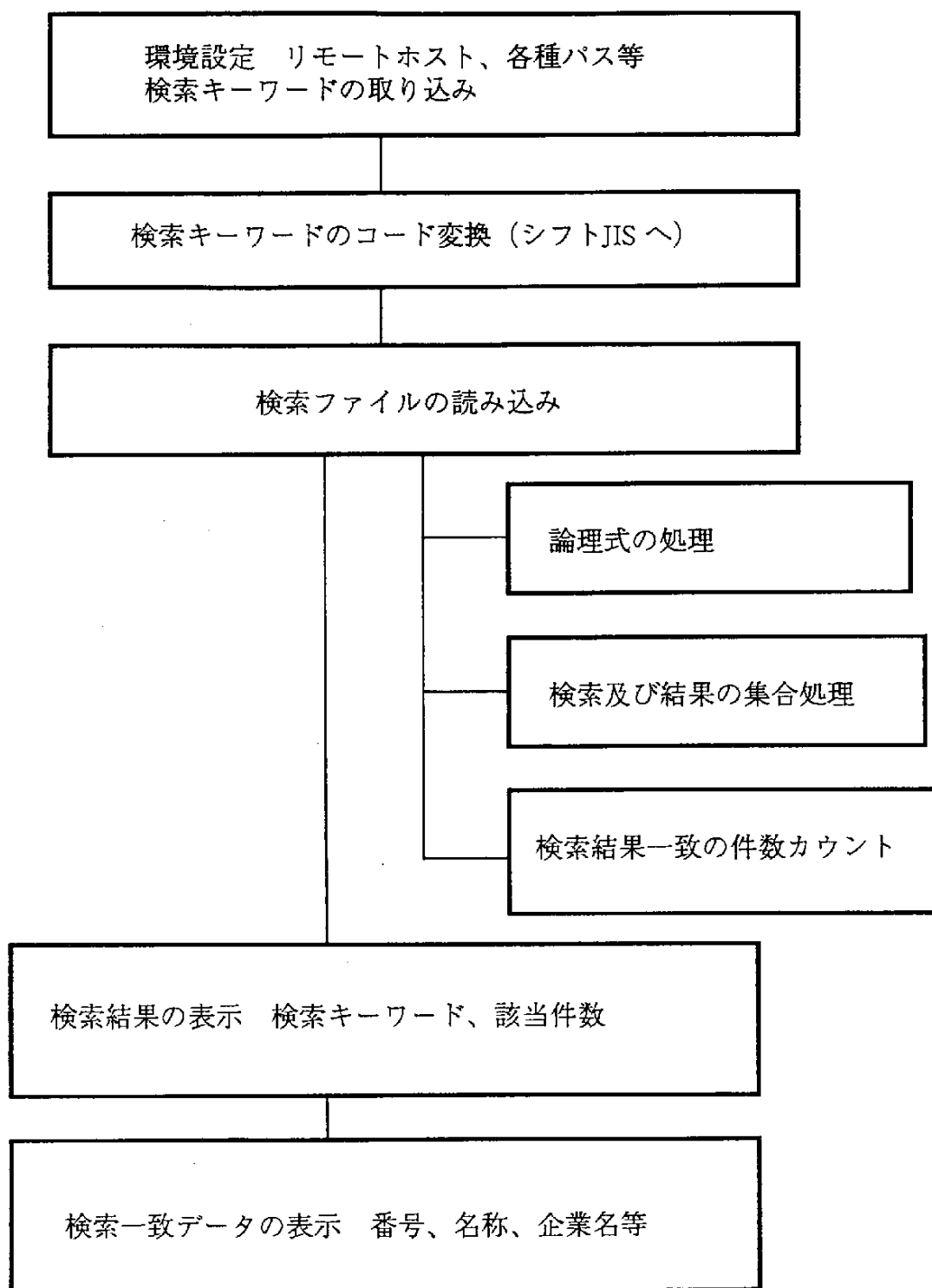


図 3. 3 検 索 処 理 の 概 要

(2) 表示プログラム

```
#!/usr/local/bin/perl
```

```
# データベースパス
```

```
$db = "/mnt/WWW/daityo/daityo";
```

```
# パラメータで指定されたDB番号
```

```
$argno = shift(@ARGV);
```

```
# HTML ヘッダの出力
```

```
print "Content-type: text/html¥n";
```

```
print "¥n";
```

```
print "<HEAD><TITLE>DATABASE DETAIL</TITLE></HEAD>¥n";
```

```
print "<BODY>¥n";
```

```
# データベースのオープン
```

```
open(DB, "$db") || die ;
```

```
while (<DB>){
```

```
# 一行ずつ読んで
```

```
( $dbno, $dbname, $producer, $country, $character, $field, $keyword, $type, $lang,  
$sysname, $company, $online, $state, $address, $dept) = split(' # ', $_);
```

```
# DB番号が一致したら終わり
```

```
last if ( $dbno == $argno );
```

```
}
```

```
close(DB);
```

```
# データの出力
```

```
print "<B>データベース名:</B> $dbname¥n";
```

```
print "<P>¥n";
```

```
print "<B>内容:</B> ¥n";
```

```
print "<P>¥n";
```

```
print "$character¥n";
```

```
print "<P>¥n";
```

```
print "<B>プロデューサ名:</B> $producer      <B>(国名):</B> $country¥n";
```

```
print "<P>¥n";
```

```
print "<B>分野:</B> $field¥n";
```

```
print "<P>¥n";
```

```
print "<B>タイプ:</B> $type¥n";
```

```
print "<P>¥n";
print "<B>言語：</B> $lang¥n";
print "<P>¥n";
print "<B>サービスシステム名：</B> $sysname¥n";
print "<P>¥n";
print "<B>サービス企業名：</B> $company¥n";
print "<P>¥n";
print "<B>問い合わせ先：</B> ¥n";
print "<P>¥n";
print "<BLOCKQUOTE>";
print "<B>住所：</B> $address¥n";
print "<P>¥n";
print "</BLOCKQUOTE>¥n";
print "<B>キーワード：</B> $keyword¥n";
print "<P>¥n";
print "</BODY>"
```

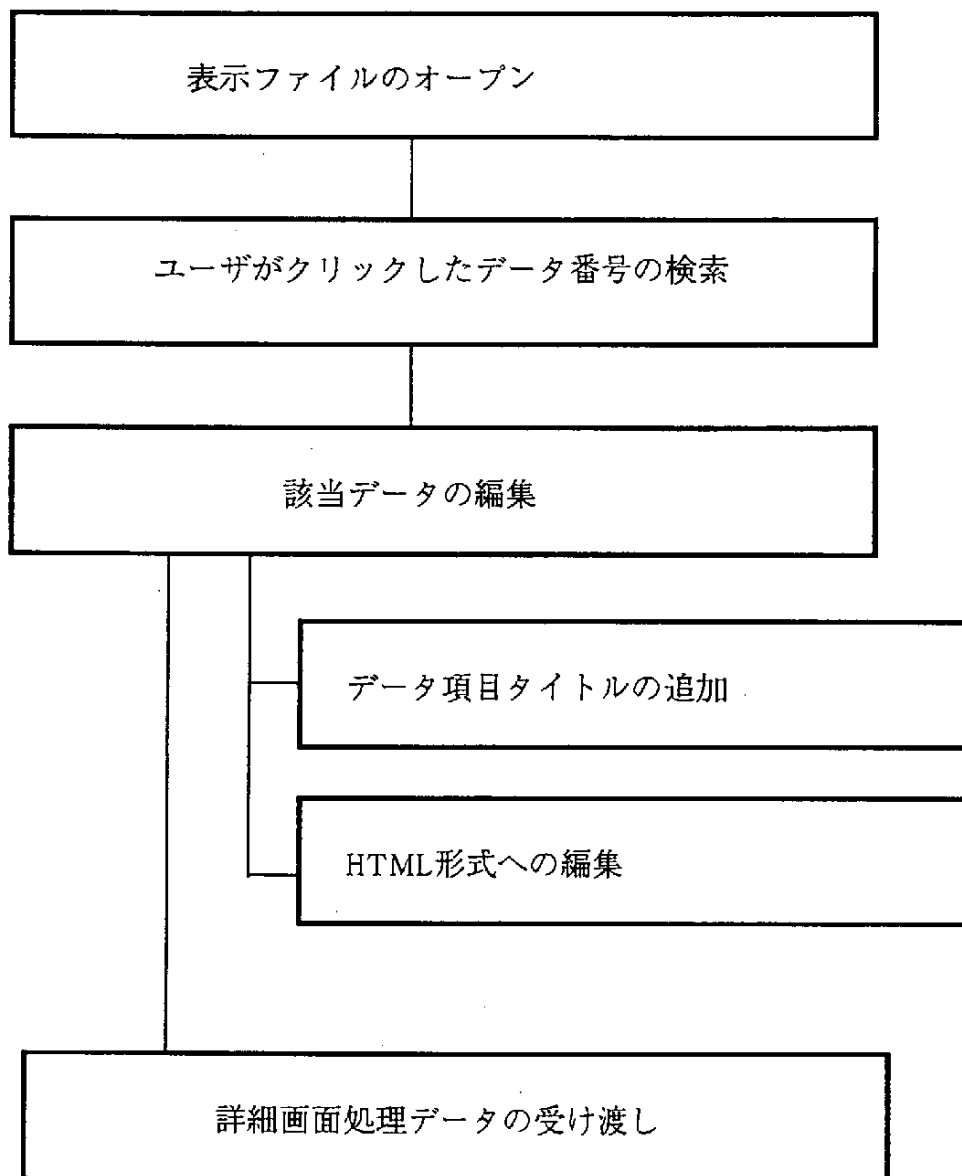


図 3. 4 検索結果表示処理の概要

3. 2. 3 画面の構成

以下に検索キーワード入力画面、検索結果画面、詳細画面を示す。

(1) 検索キーワード入力画面

The screenshot shows the NCSA Mosaic Document View (L10N+) window. The title bar reads "NCSA Mosaic: Document View (L10N+)". The menu bar includes "File", "Options", "Navigate", "Annotate", "News", and "Help". The address bar shows "Title: DATABASE SEARCH" and "URL: http://www.sci.jip.co.jp/sci/soc/dbgrep.html". The main content area displays the heading "データベース総覧検索" (Database Overview Search). Below this, there is a prompt "●検索語を入力して下さい。" (Please enter search terms). A text input field contains "医薬 + 承認". Below the input field are buttons for "検索" (Search) and "再入力" (Re-input). A list of links is provided: "データベース総覧について" (About Database Overview), "検索式について" (About Search Form), and "データ項目について" (About Data Items). At the bottom of the main area is the email address "www-admin@sci.jip.co.jp". The status bar at the very bottom contains buttons for "Back", "Forward", "Home", "Reload", "Open...", "Save As...", "Clone", "New Window", and "Close Window".

NCSA Mosaic: Document View (L10N+)

File Options Navigate Annotate News Help

Title: DATABASE SEARCH

URL: http://www.sci.jip.co.jp/sci/soc/dbgrep.html

データベース総覧検索

●検索語を入力して下さい。

医薬 + 承認

検索 再入力

- データベース総覧について
- 検索式について
- データ項目について

www-admin@sci.jip.co.jp

Back Forward Home Reload Open... Save As... Clone New Window Close Window

図 3. 5

```
<HEAD>
<TITLE>DATABASE SEARCH</TITLE>
</HEAD>□ <BODY>
<CENTER><H1>データベース総覧検索</H1></CENTER><P>
<HR>
<FORM METHOD="POST" ACTION="/cgi-bin/dbgrep">
●検索語を入力して下さい。<P>
<INPUT TYPE="text" SIZE=60 MAXSIZE=60 NAME="query"><P>
<INPUT TYPE="submit" VALUE="検索">
  <INPUT TYPE="reset" VALUE="再入力"></FORM>
<P>
<HR>
<UL>
<LI> <A HREF="daityou.html">データベース総覧について</A>
  <LI> <A HREF="kensaku.html">検索式について</A>
  <LI> <A HREF="item.html">データ項目について</A>
</UL>
<HR>
<ADDRESS>www-admin@jip.co.jp</ADDRESS></BODY>
```

(2) 検索結果画面

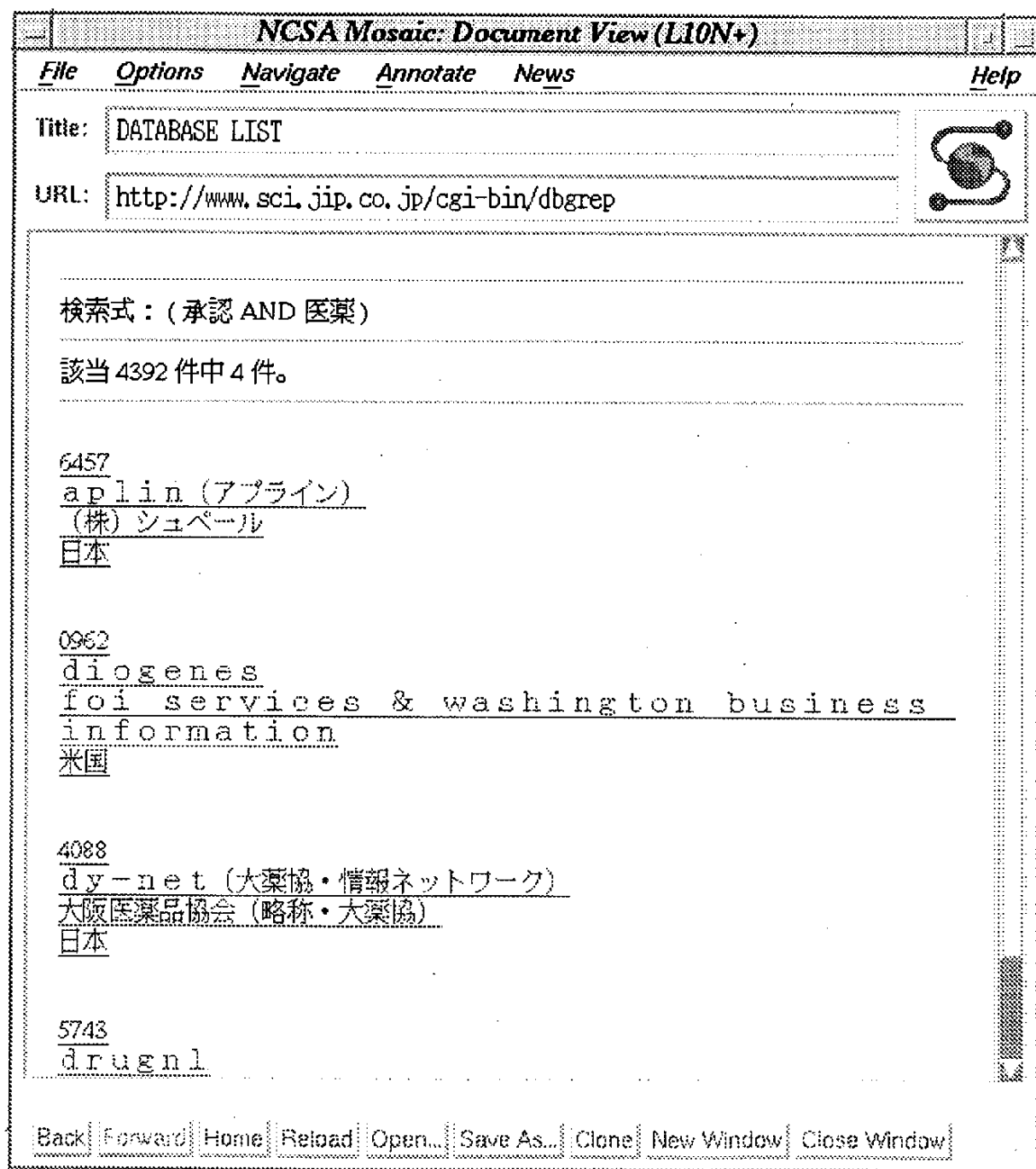


図 3. 6

<HTML>

<HEAD><TITLE>DATABASE LIST</TITLE></HEAD>

<BODY>

<hr>検索式：(承認 AND 医薬)

<hr>該当 4392 件中 4 件。

<hr>

6457

a p l i n (アプライン)

(株) シュペール

日本<P>

0962

d i o g e n e s

f o i s e r v i c e s & w a s h i n g t o n b u s i n e s s i n f o
r m a t i o n

米国<P>

4088

d y - n e t (大薬協・情報ネットワーク)

大阪医薬品協会 (略称・大薬協)

日本<P>

5743

d r u g n l

i m s w o r l d p u b l i c a t i o n s l t d .

英国<P>

</BODY>

</HTML>

(3) 詳細画面

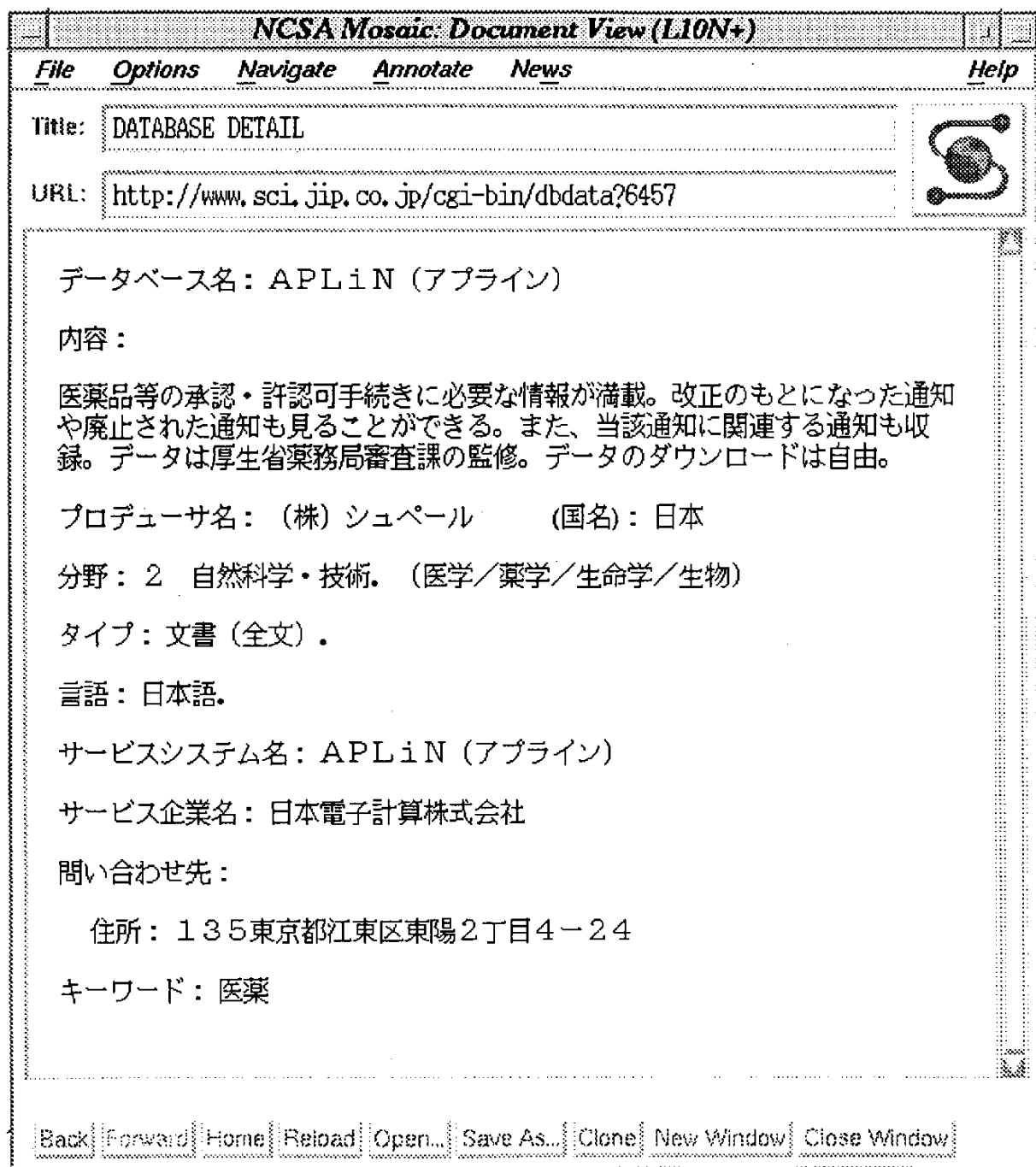


図 3. 7

<HEAD>

<TITLE>DATABASE DETAIL</TITLE>

</HEAD>

<BODY>

データベース名: A P L i N (アプライン)

<P>

内容:

<P>

医薬品等の承認・許認可手続きに必要な情報が満載。改正のもとになった通知や廃止された通知も見ることができる。また、当該通知に関連する通知も収録。データは厚生省薬務局審査課の監修。データのダウンロードは自由。

<P>

プロデューサ名: (株) シュペール (国名): 日本

<P>

分野: 2 自然科学・技術. (医学／薬学／生命学／生物)

<P>

タイプ: 文書 (全文) .

<P>

言語: 日本語.

<P>

サービスシステム名: A P L i N (アプライン)

<P>

サービス企業名: 日本電子計算株式会社

<P>

問い合わせ先:

<P>

<BLOCKQUOTE>住所: 1 3 5 東京都江東区東陽 2 丁目 4 - 2 4

<P>

</BLOCKQUOTE>

キーワード: 医薬

<P>

</BODY>

4. パーソナル・コンピュータ用C G I（英語対応）の詳細

個人の情報管理の道具としてデータベース作成及び利用が普及し、今では様々な汎用パッケージを使用した個人用のデータベースが作られている。さらに社内L A Nを整備している事務所等では、これらのデータを流通させ情報の共有化が進んできた。

また、パソコンの性能が向上し、またWindows版やMacintosh版のHTTPサーバ用のフリーソフトやシェアウェア・ソフトウェアも出回り、情報の幅広い提供が可能となってきた。

米国等ではすでにパソコンによるC G Iの利用例があり、また中には市販のデータベース・ソフトウェア（例えば、File-Maker）を対象にしたゲートウェイ・ソフトウェアを含むC G Iが公表されている。

そこで、本調査研究の一部として、Macintoshを利用してW e bサーバで利用可能なC G Iの作成を行った。Macintoshを選定したのはWindows版のperlが見あたらないためと、海外の例ではMacintosh版のC G I作成の例が多く見られたことによるものである。

・ハードウェア構成

Macintosh Quadra 610 32MB（L A N接続）

・ソフトウェア

基本ソフトウェア

漢字Talk 7.1

MacTCP（TCP Adminを含む）

MacPerl 5.0.7 C G I作成用 言語

MacHTTP 2.2 W e bサーバ

File-Maker Pro II（日本語版）データベース・ソフトウェア

・設計条件

・基本的な仕様は前述したW Sで作成したものと同様とする。

・MacPerlは日本語処理をサポートしていないため英語のデータを処理する。

（W S用perlでは日本仕様のjperlがあるが、Macintosh用は現在のところ見あたらない）

4. 1 データ整備

今回使用した「Directory of Japanese Databases」のデータは既にFile-Maker Pro IIによりデータベース化されているため、必要なデータ項目を抽出し、そのファイルをMacPerlで作成したプログラムで以下の処理を行った。

(1)原データの構成

- ①データベース名
- ②キーワード（分野別）
- ③データベースの内容説明
- ④記述言語
- ⑤プロデューサ名
- ⑥ディストリビュータ名

(2)ファイルの処理と例

・ファイルの処理内容

- ・一連データ番号の追加付与
- ・タブ記号から項目区分記号である‘#’への変換
- ・検索時の問題となるアルファベットの大文字・小文字の統一化は、ここでは行わず、検索プログラムの中で、パターン・マッチング時に小文字に統一した。

<原データの例>

データベース名：APLiN

キーワード：medicine

内容説明：This database contains abundant information necessary for obtaining approvals or permissions of medical products. Notices that have led to revisions and notices no longer effective are also available. Cross references are made among notices. The data is reviewed by the Examination Division, Pharmaceutical and Supply Bureau of the Ministry of Health and Welfare. Data can be downloaded freely.

言語：Japanese

プロデューサ名：SUPER,Co.,Ltd.

ディストリビュータ名：Japan Information Processing Service Co.,Ltd. (JIP)

データベース名：ARIS

キーワード：pharmacy, bioscience, clinical medicine

内容説明：Excerpts of Japanese literatures concerning side effects in easy-to-understand format. Also included are examples of symptoms of side effects reported in major foreign magazines translated into Japanese in same format.

Range of literatures is as wide as to cover poisoning, interaction, and side effects of healthy foods. Technical terms of side effects and names of medical products are coded for easy access.

言語 : Japanese

プロデューサ名 : Odajima Inc.

ディストリビュータ名 : Japan Information Processing Service Co.,Ltd. (JIP)
¥Odajima Inc.

<加工・編集結果 C G I の検索／表示用ファイルの例>

0001 # APLiN # medicine # This database contains abundant information necessary for obtaining approvals or permissions of medical products. Notices that have led to revisions and notices no longer effective are also available. Cross references are made among notices. The data is reviewed by the Examination Division, Pharmaceutical and Supply Bureau of the Ministry of Health and Welfare. Data can be downloaded freely. # Japanese # SUPER, Co., Ltd. # Japan Information Processing Service Co., Ltd. (JIP)

0002 # ARIS # pharmacy, bioscience, clinical medicine # Excerpts of Japanese literatures concerning side effects in easy-to-understand format. Also included are examples of symptoms of side effects reported in major foreign magazines translated into Japanese in same format. Range of literatures is as wide as to cover poisoning, interaction, and side effects of healthy foods. Technical terms of side effects and names of medical products are coded for easy access. # Japanese # Odajima Inc. # Japan Information Processing Service Co., Ltd. (JIP) ¥Odajima Inc.

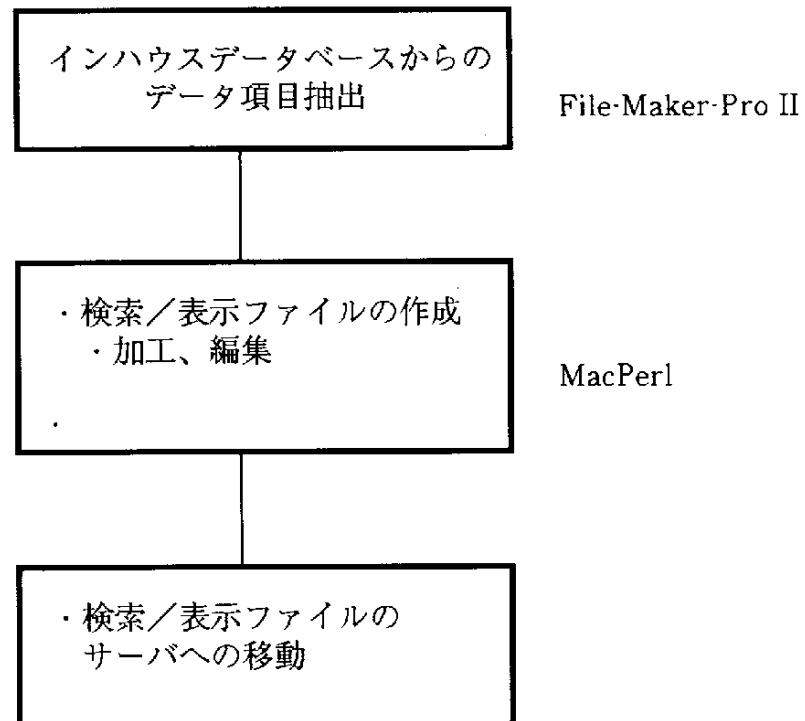


図 4 . 1 デ ー タ 処 理 の 流 れ

4. 2 検索システム

4. 2. 1 検索ロジック

- (1)原則的にはWS版と同様としたが、以下の変更を行った。
- (2)WS版では検索対象データを全てメモリに蓄積し、その後検索を行っているが、パソコンではメモリが少ないことから、レコードを1件ずつ読みながら検索を行う方式に変更した。
- (3)検索キーワードや検索レコードの文字処理やパターンマッチング処理で、日本語用PerlとMacPerlでは記述が異なるところを変更した。

4. 2. 2 プログラムの表記と説明

(1) 検索プログラム

```
# /perl
# database data retrieval with Macintosh
# input-file
# データベース・パスの設定
$db = "machd:jip-add:en-all.txt";
# ログ・ファイルの設定
$log = "machdd:log.txt";
# 検索キーワードの取り込み
if ($len = $ENV{"CONTENT_LENGTH"}) {
    sysread(STDIN, $in, $len);
}
# 不要ヘッダの削除
$_ = substr($in,6);
# スペース、プラス記号の変換
$_ =~ s/%20/ /g;
$_ =~ s/%2B/+/g;

# input query
# 検索キーワードの配列化
@querylog = @query = split(/ /);
# キーワードの加工編集
```

```

# query-set
&query_set;

        # 入力件数の初期化
$totalin=0;

# main-rtn
        # ファイルの読み込み
open(DB, "$db") || die ;
while ($s_ = <DB>){
    @db_all = $s_;
    $totalin++;
        # 検索ルーチン
    &rep_rtn;
}
close(DB);
#
# result print - header # 検索結果の出力
    &print_header;
# query-print
    &print_query;
if ( $# total < 0 ) {
# result - nothing
    &print_noresult;
} else {
# result
    &print_data;
}
# trailer                #
    &print_trailer;

        # 主処理終了 HTTPサーバへ
# =====
# debug
sub printno{
    local(@result) = @_ ;
    local($i, $no, @body) ;
    for ($i=0; $i<=$# result; $i++){
        ($no, @body) = split('#', $result[$i]);
        printf "%2d:%04d¥n", $i,$no;
    }
}

```

```

}
}
# =====
# retrieval # 検索処理
sub rep_rtn{
#
@theresult="";

# AND
if ( $# andq >= 0 ) {
  for ($i=0; $i<=$# andq; $i++) {
    #
    @tmpresult = grep(/$andq[$i]/i, @db_all);
    local(%mark);
    if ( $i == 0 ) {
      #
      @andresult = @tmpresult;
    } else {
      #
      grep($mark{$_}++, @tmpresult);
      @andresult = grep($mark{$_}, @andresult);
    }
  }
}

# NOT 検索条件がある場合
if ( $# notq >= 0 ) {
  for ($i=0; $i<=$# notq; $i++) {
    # 一時集合の生成
    @tmpresult = grep(!/$notq[$i]/i, @db_all);
    local(%mark);
    if ($i == 0) {
      # 最初は一時集合が NOT 条件の結果集合
      @notresult = @tmpresult;
    } else {
      # 一時集合との AND 演算
      grep($mark{$_}++, @tmpresult);
      @notresult = grep($mark{$_}, @notresult);
    }
  }
}

```



```
}  
}
```

```
# AND/NOT 条件中間結果の生成
```

```
# AND 検索条件と NOT 検索条件が共にあった場合
```

```
if ( $# andq >= 0 && $# notq >= 0 ) {
```

```
    local(%mark);
```

```
# AND 検索条件の結果集合と NOT 検索条件の結果集合との AND 演算
```

```
grep($mark{$_}++, @andresult);
```

```
@anresult = grep($mark{$_}, @notresult);
```

```
}
```

```
# AND 検索条件だけが合った場合
```

```
if ( $# andq >= 0 && $# notq < 0 ) {
```

```
@anresult = @andresult;
```

```
}
```

```
# NOT 検索条件だけが合った場合
```

```
if ( $# andq < 0 && $# notq >= 0 ) {
```

```
@anresult = @notresult;
```

```
}
```

```
# OR 検索条件があった場合
```

```
if ( $# orq >= 0 ) {
```

```
# AND 検索条件か NOT 検索条件が指定されていた場合
```

```
if ( $# andq >= 0 || $# notq >= 0 ) {
```

```
    for ( $i=0; $i<=$# orq; $i++ ) {
```

```
#        一時集合の生成
```

```
@tmpresult = grep(/$orq[$i]/i, @db_all);
```

```
local(%mark);
```

```
grep($mark{$_}++, @tmpresult);
```

```
#        AND/OR 条件中間結果との AND 演算
```

```
@orresult = grep($mark{$_}, @anresult);
```

```
if ( $i == 0 ) {
```

```
#        最初はそのまま最終結果へ代入
```

```
@theresult = @orresult;
```

```
} else {
```

```
#        最終結果との OR 演算
```

```
@theresult = &merge(@orresult);
```

```

    }
}
} else {
#   AND 検索条件か NOT 検索条件が指定されていなかった場合(単純 OR 検索)
  for ( $i=0; $i<=$#orq; $i++ ) {
#       一時集合の生成
    @tmpresult = grep(/$orq[$i]/i, @db_all);
    if ( $#theresult < 0 ) {
#       最終結果が空ならばそのまま最終結果へ代入
      @theresult = @tmpresult;
    } else {
#       最終結果との OR 演算
      @theresult = &merge(@tmpresult);
    }
  }
}
} else {
#   AND/OR 検索条件しか指定されていなかった場合
  if ( $#anresult >= 0 ) {
    @theresult = @anresult;
  }
}

# 検索結果の集積
$th = $#theresult;
if ( substr($theresult[$th],0,2) || "" ) {
  push (@total,$theresult[$th]);
}

} # end of rep_rtn
# =====
# OR 演算
sub merge {
  local(@tmpresult) = @_;
  local(%mark1, %mark2, @result1, @result2, @result3);
  grep($mark1[$_]++, @tmpresult);
  @result1 = grep($mark1[$_], @theresult);
  @result2 = grep(!$mark1[$_], @theresult);
  grep($mark2[$_]++, @theresult);
  @result3 = grep(!$mark2[$_], @tmpresult);
  @theresult = "";
}

```

```

@theresult = @result1;
push(@theresult, @result2);
push(@theresult, @result3);
@theresult;
}
# =====
# 検索条件式の出力
sub print_query {
    local($i, $baseq);

    for ($i=0; $i<=$# andq; $i++) {
        if ( $i != 0 ) {
            $baseq .= " AND ";
        }
        $baseq .= $andq[$i];
    }
    for ($i=0; $i<=$# notq; $i++) {
        if ( length($baseq) > 0 ) {
            $baseq .= " AND ";
        }
        $baseq .= "NOT $notq[$i]";
    }
    if ( $# orq >= 0 ) {
        for ( $i=0; $i<=$# orq; $i++ ) {
            if ( length($lastq) > 0 ) {
                $lastq .= " OR ";
            }
            if ( length($baseq) > 0 ) {
                $lastq .= "( $baseq ";
                $lastq .= " AND $orq[$i] ) ";
            } else {
                $lastq .= $orq[$i];
            }
        }
    } else {
        $lastq = $baseq;
    }
    print "<hr>";
    print "$lastq¥n";
}

```

```

print "<hr>";
$all = $totalin ;
$hit = $#total + 1 ;
print " Total record $all ... Hit record $hit ¥n";
print "<hr>";
}
# =====
# ログの出力 (debug)
sub writelog {
    open(LOG, ">>$log") || die "open log $log $!¥n";
    # ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) =
        localtime(time);
    $mon++;
    # printf LOG "%s/%s/%s %s:%s ", $year, $mon, $mday, $hour, $min ;
    print LOG " $RHOST <$#theresult> ";
    foreach $i (@querylog) { print LOG "$i"; }
    print LOG "¥n";
    close(LOG);
}
# =====
# header # 結果出力 ヘッダ
sub print_header {
    # print &PrintHeader;
    print "<HTML>¥n";
    print "<HEAD><TITLE>Japanese Databasee List</TITLE></HEAD>¥n";
    print "<BODY>¥n";
}
# =====
# トレイラの出力
sub print_trailer {
    print "</BODY>¥n";
    print "</HTML>¥n";
}
# =====
# 該当なし
sub print_noresult {
    print "No Data";
}
# =====

```

```

# 該当データの出力
sub print_data {
#
for ($i=0; $i<=$ # total; $i++) {
($ddmy, $dbn, $dbnam, $scattg, $scnstt, $slang, $prod, $disx, $ddn) =
split(/ # /,$total[$i]);

#                                     # 詳細画面出力用 C G I の指定
print "<A HREF=¥"/libwww.cgi/en-put.cgi?$dbn¥"><br>¥n";
# D B 番号
print $dbn;
print "¥n";
# D B 名称
print $dbnam;
print "¥n";
# プロデューサ名
print $prod;
print "¥n";
# ディストリビュータ名
print $disx;
print "¥n";
}
}
# ===== #
# query-set
sub query_set {
# AND
for ($i=0; $i<=$ # query; $i++) {
#
if ( $query[$i] =~ /\{+\/ ) {
substr($query[$i], 0, 1) = "";
# andq -> AND
push(@andq, $query[$i]);
} elsif
# NOT
( $query[$i] =~ /\^+\/ ) {
substr($query[$i], 0, 1) = "";
# notq -> NOT
push(@notq, $query[$i]);
}
}
}

```

```
# OR
} else {
#      orq -> OR
  push(@orq, $query[$i]);
}
}
}
# =====end of prog=====
```

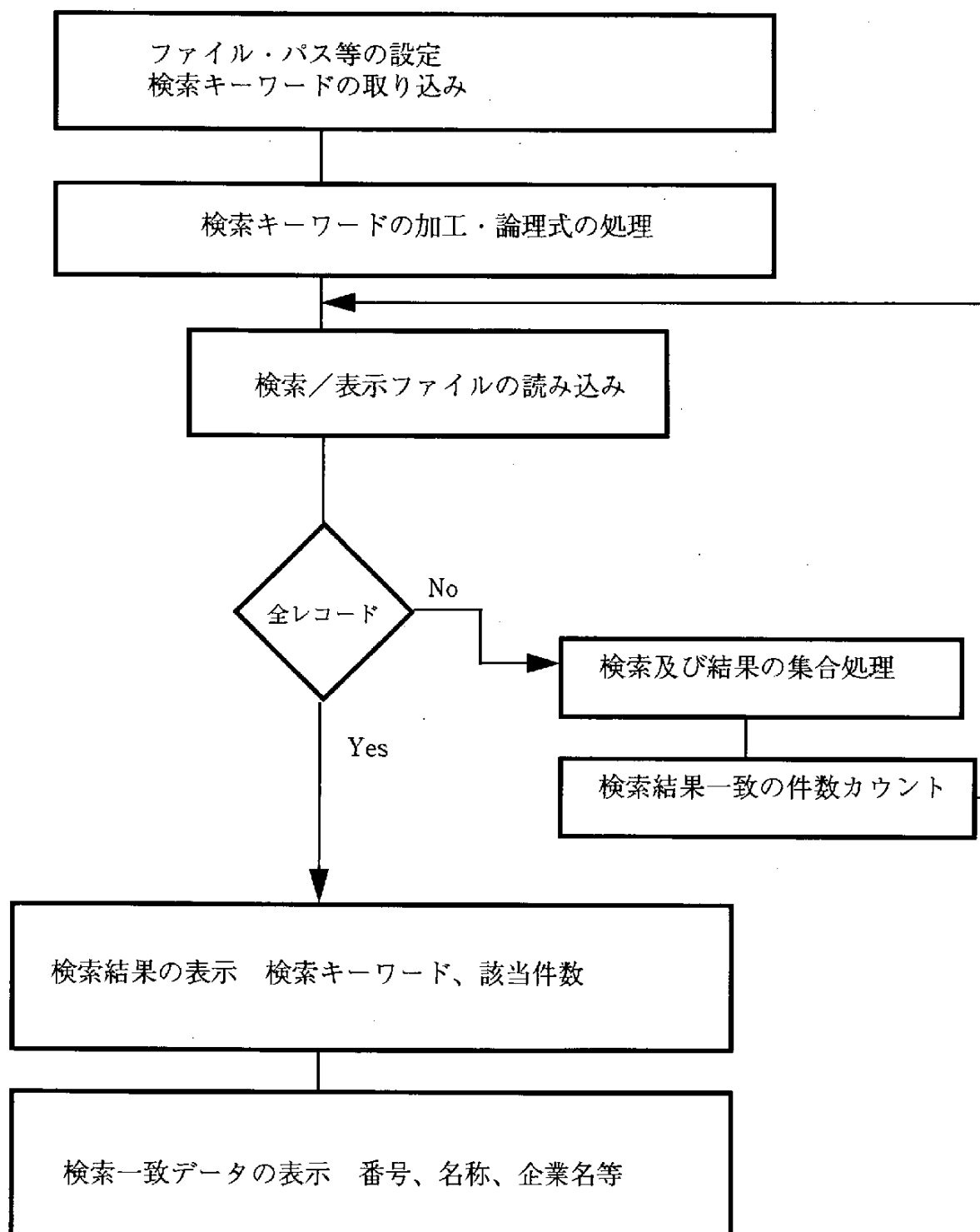


図 4. 2 検 索 処 理 の 概 要

(2) 表示プログラム

```
# /perl
# 詳細画面出力用 C G I
# input file
# データベース・パスの設定
Sdb = "machd:jip-add:en-all.txt";
# parameter
# D B 番号の取り込み
Sargnox = $ENV["QUERY_STRING"];
Sargno = substr($Sargnox, 0, 4);

# HTML header
print "Content-type: text/html¥n";
print "¥n";
print "<HEAD><TITLE>Ja[anese DAatabase Detail</TITLE></HEAD>¥n";
print "<BODY>¥n";

# ファイルの読み込み
open(DB, "$Sdb") || die ;
while (<DB>){
# item-set
($dummy, $Sdbno, $Sdbname, $subject, $sconts, $Slang, $Sprod, $Sdist) =
    split(/ # /, $_);
# D B 番号の比較
last if ( $Sdbno == $Sargno );
}
close(DB);

# 詳細データの編集・出力
print "<B>DBno.:</B> $Sdbno ¥n";
print "<P>¥n";

print "<B>Title:</B> $Sdbname ¥n";
print "<P>¥n";
```



```
print "<B>Content:</B> ¥n";
print "<P>¥n";

print "$conts¥n";
print "<P>¥n";

print "<B>Language:</B> $lang";
print "<P>¥n";

print "<B>Subject:</B> $subject";
print "<P>¥n";

print "<B>Producer:</B> $prod¥n";
print "<P>¥n";
# print "<BLOCKQUOTE>";
print "<B>Distributor:</B> $dist¥n";
print "<P>¥n";
# print "</BLOCKQUOTE>¥n";
print "</BODY>"
# end-of-prog
```

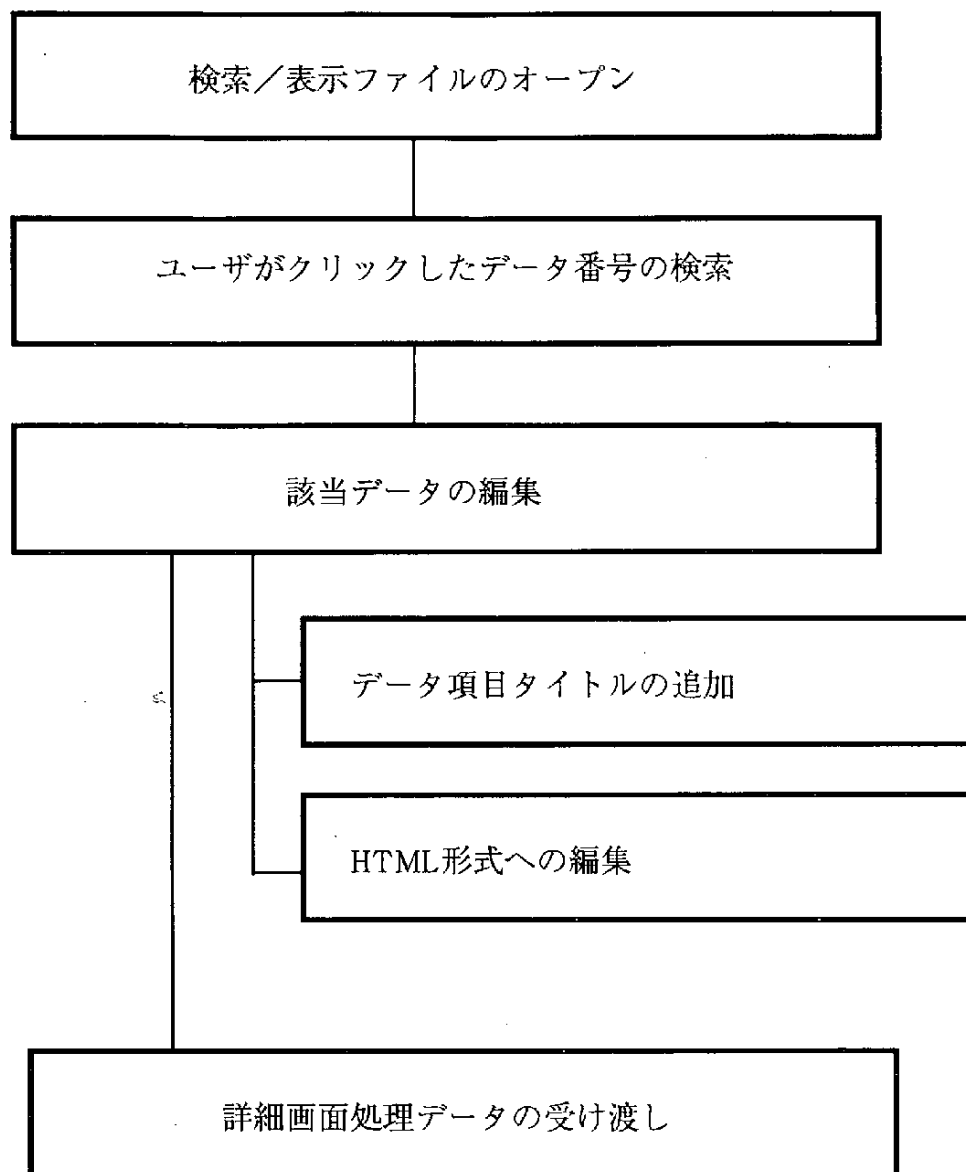


図 4 . 3 検索結果表示処理の概要

4. 2. 3 画面の構成

以下に検索キーワード入力画面、検索結果画面、詳細画面を示す。

(1) 検索キーワード入力画面

The screenshot shows a web browser window with the title "DATABASE SEARCH". The address bar displays "URL: http://mac1.jip.co.jp/". The main content area has the heading "Japanese Databases Retrieval". Below the heading is a text input field labeled "Enter keywords)". To the left of the input field are two buttons: "Go" and "Clear". Below the input field is a list of links:

- Directory of Japanese Databases
- How to use this engine
- Example of data

At the bottom left of the page, the email address "www-admin@sci.jip.co.jp" is displayed. The browser's status bar at the bottom shows navigation icons.

図 4 . 4

```

<HEAD>
<TITLE>DATABASE SEARCH</TITLE>
</HEAD> <BODY>
<CENTER><H1>Japanese Databases Retrieval</H1></CENTER><P>
<HR>
<FORM METHOD="POST" ACTION="/libwww.cgi/en-srh.cgi">
                                (C G I のパス指定)
Enter keywords<P>
<INPUT TYPE="text" SIZE=60 MAXSIZE=60 NAME="query">    <P>
                                (クエリ・エリア等の定義)
<INPUT TYPE="submit" VALUE="go">
<INPUT TYPE="reset" VALUE="re-enter"></FORM>
<P>
<HR>
<UL>
  <LI> <A HREF="daityou.html">Directory of Japanese Databases</A>
  <LI> <A HREF="kensaku.html">How to use this engine</A>
  <LI> <A HREF="item.html">Example of data</A>
</UL>
<HR>
<ADDRESS>www-admin@jip.co.jp</ADDRESS></BODY>

```

(2) 検索結果画面

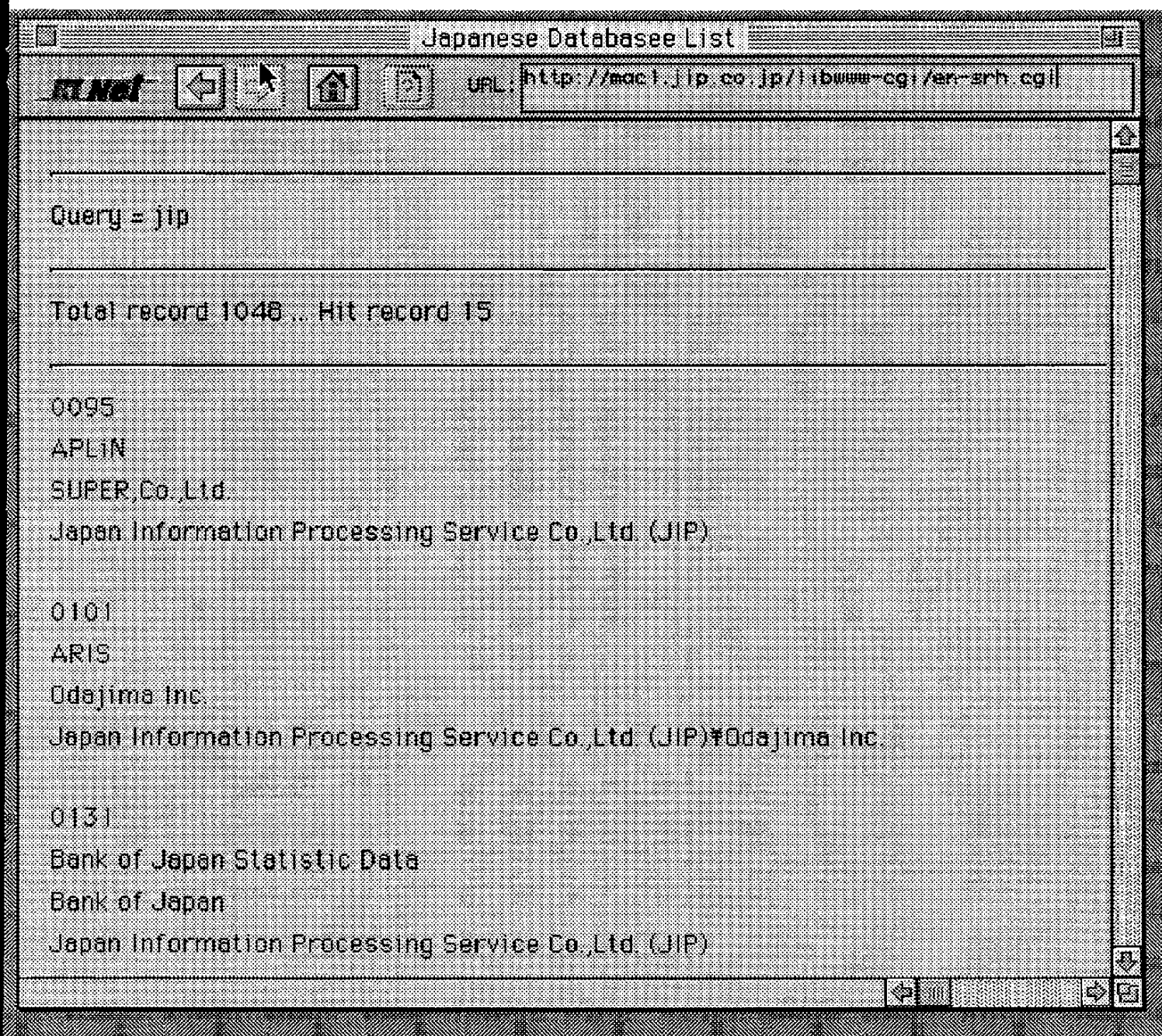


図 4. 5

<HEAD>
<TITLE>Japanese Databasee List
</TITLE>
</HEAD>
<BODY>
<HR> Query = jip
<HR> Total record 1048 ... Hit record 15
<HR>

 0095

 APLiN

SUPER,Co.,Ltd.

 Japan Information Processing Service Co.,Ltd. (JIP)

<P>

 0101

 ARIS

 Odajima Inc.

 Japan Information Processing Service Co.,Ltd.
(JIP)¥Odajima Inc.

<P>

 0131

 Bank of Japan Statistic Data

 Bank of Japan

 Japan Information
Processing Service Co.,Ltd. (JIP)

<P>

 0146

 Bond Information

 Japan Information Processing Service Co.,Ltd.

(JIP)

 Japan Information Processing Service Co.,Ltd. (JIP)

<P>

・ (途 中 省 略)

 0920

 Securities-Information

 Japan Information Processing Service

Co.,Ltd. (JIP); The Stock News

 Japan Information Processing Service

Co.,Ltd. (JIP)

<P>

</BODY>

(3) 詳細画面

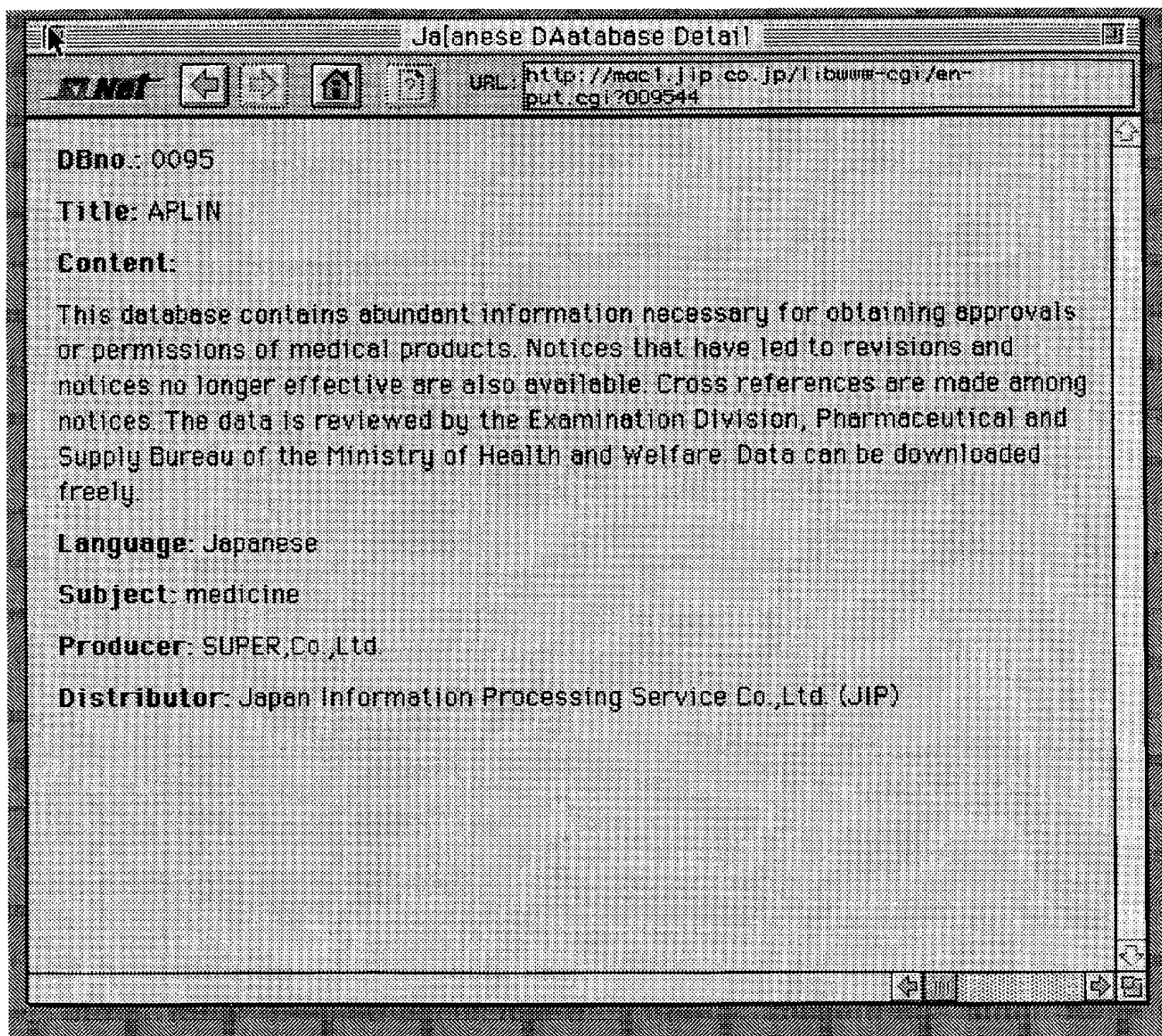


図 4 . 6

<HEAD>

<TITLE>Japanese DAatabase Detail

</TITLE>

</HEAD>

<BODY>

DBno.:

 0095

<P>

Title:

 APLiN

<P>

Content:

<P>

This database contains abundant information necessary for obtaining approvals or permissions of medical products. Notices that have led to revisions and notices no longer effective are also available. Cross references are made among notices. The data is reviewed by the Examination Division, Pharmaceutical and Supply Bureau of the Ministry of Health and Welfare. Data can be downloaded freely.

<P>

Language:

 Japanese

<P>

Subject:

 medicine

<P>

Producer:

 SUPER,Co.,Ltd.

<P>

Distributor:

 Japan Information Processing Service Co.,Ltd. (JIP)

<P>

</BODY>

4. 2. 4 Webサーバの構築

(1)MacHTTPの準備 (バージョン MacHTTP WWW server v2.0.3)

- ①市販のインターネットに関連する雑誌のCD-ROM、またはインターネットからソースを入手する。
- ②添付されているマニュアルに従ってインストールを行う。
- ③CGI作動に必要なファイルを決められたシステムフォルダの機能拡張へコピーする。

(2)MacPerlの準備 (バージョン Perl 5.0.7r1m)

- ①インターネットからソースを入手する。
- ②添付されているマニュアルに従ってインストールを行う。

(3)MacHTTP CGI scriptの入手

- ①市販のインターネットに関連する雑誌のCD-ROM、またはインターネットからソースを入手する。
- ②ソースをMacPerl Extensions フォルダへコピーする。

(4)CGI作成準備

- ①MacHTTPフォルダ内にCGI用のフォルダを作成する。
ここでは libwww.cgi とした。
- ②Macperl内にCGI作成用のフォルダを作成する。
ここでは cgi-source とした。

(5)CGI作成と格納

- ①MacperlでCGI用プログラムを作成する。
- ②完成したプログラムは所定の属性を付け、MacHTTP の libwww.cgiへ格納する。

(6)実行

- ①処理に従ったホームページを作成する。
- ①MacHTTPをたちあげる。
- ②ブラウザで確認する。

5. 成果と課題

5. 1 実験システムの評価

(1)ワークステーション用CGI

①環境

・ハードウェア	Sun SPARC Statiton 5
	CPU micro SPARC-II (M/M 32MB)
・データ量	検索ファイル 1, 984 KB 4, 576件
	平均レコード長 430バイト
	表示ファイル 2, 371 KB 4, 576件
	平均レコード長 518バイト

②評価

- ・今回の実験ではデータ量が少ないが、検索及び表示処理の応答速度は平均2～3秒であった。
- ・データ量を5倍にしても、それほどの応答速度の低下はみられなかった。
- ・全体的には、データベースからのデータの加工・編集プログラムやCGI等の使用するソフトウェアが整備されていれば、パソコン・ユーザでも簡単にWebサーバで自分のデータが提供できると考えられる。

(2)パーソナル・コンピュータ用CGI

①環境

・ハードウェア	Macintosh Quadra 650 (M/M 32MB)
・メモリ使用量	システム (漢字Talk 7.1) 3,470 KB
	MacHTTP 1,150 KB
	MacPerl 2,048 KB
	検索用CGI 128 KB
	表示用CGI 128 KB
・データ量	検索／表示ファイル 523 KB 1, 048件
	平均レコード長 500バイト

②評価

- ・ 1048件のデータでは、検索処理は13秒程度で応答があり、また表示については、最大7秒程度で詳細データが表示された。
- ・ データ量を5倍（5240件）にして実験したところ、検索には40秒から45秒程度かかり、詳細データの表示にも最大30秒程度かかった。
- ・ パソコンを使用したことで、CPUやディスクの速度がWSに比して遅いため、データ量がある程度の大きさ3000件（1.5MB）位が適当な大きさであると思われる。
- ・ データベースからのデータ項目抽出から、Webサーバへのデータ・ファイルへの複写までの処理をルーチン化することにより、容易に情報が提供できる。
- ・ WSに比べて、処理が簡単なことやコンピュータのある程度の知識があれば、このパソコンCGIは十分に活用できると思われる。

5. 2 成果

- (1)小規模なデータベースから、簡単にWebサーバ用のデータを作成することが可能となった。
- (2)処理ルーチンやデータの表現方法を明確にしたことで、個人や組織内でのインターネットやイントラネットでのデータベース情報提供仕組みや理解が容易となった。
- (3)現在の表層的な情報提供から、深みのある大量のデータを提供できることで、情報の共有化が容易となった。
- (4)WSの低廉化やPCでのUNIX利用が可能になっていることから、今後さらにWebサーバとCGIの組み合わせによる情報提供が容易になるだけでなく、提供する情報量の拡大や個人レベルの情報を容易に発信することが可能となり、埋蔵していた新しい情報資源が活用できる。

5. 3 課題

(1) ユーザ・インターフェイスの向上

今回の実験では、原データの内容は変更せずに直接その内容を検索対象としたが、内容の表現や表記はデータ提供者に依存していることから、記述されている内容は正規化されておらず、ユーザの検索結果への満足度は低くなりがちである。

そのため、このシステムの追加機能として、データベースに蓄積されているキーワードの表示や、同義語さらにはシソーラス等の追加によりユーザ・インターフェイスを拡充する必要がある。

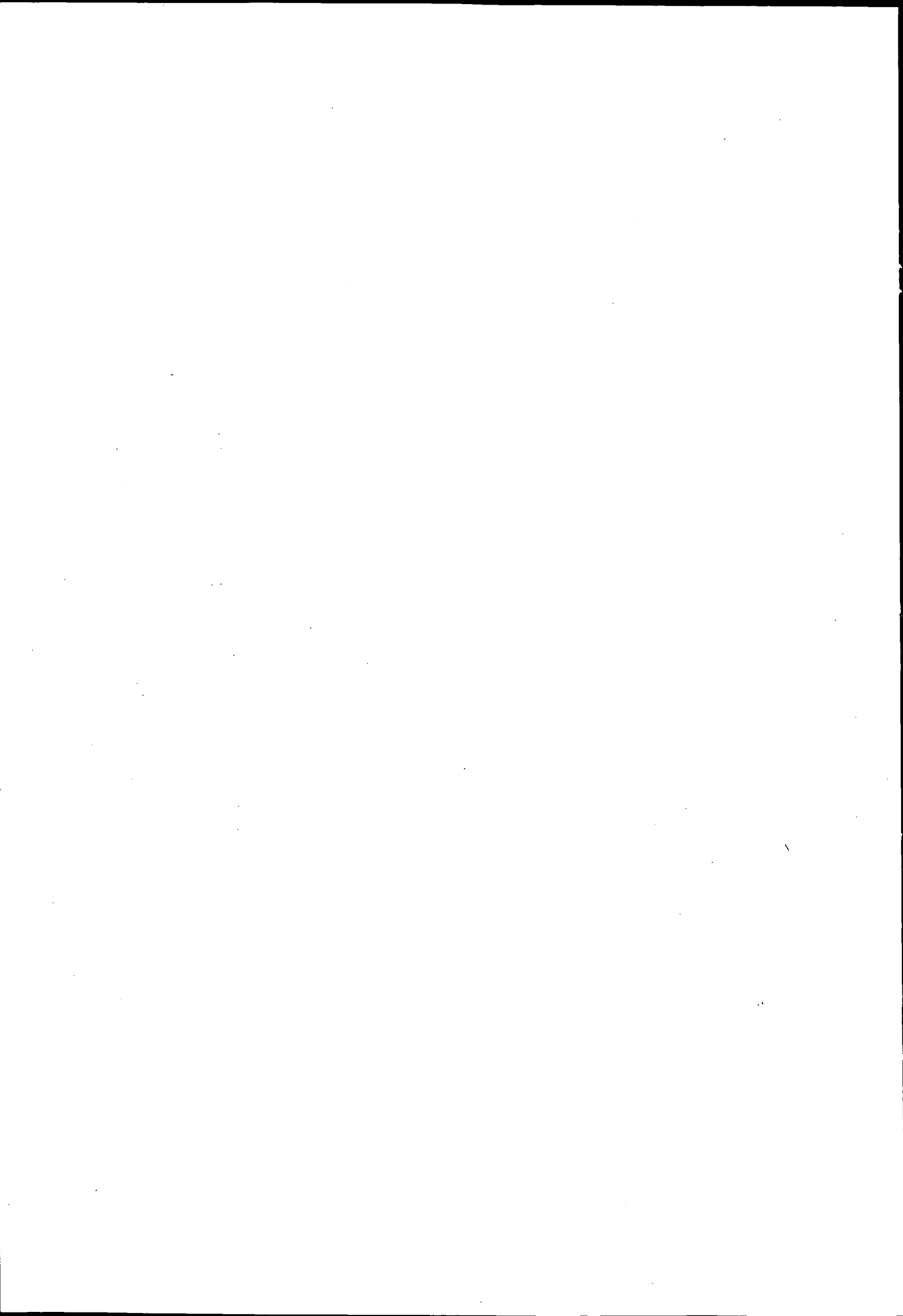
しかしながら、これらの機能を充実するには、標準的なシソーラスの作成、日本語解析のための環境やソフトウェアの進展が必要となる。とくに、英語処理の環境においては、上記の機能はすでに整備されていることから早急な解決が望まれる。

(2) プラットフォームの整備

今後は組織内LANの整備とともに、組織や個人での小規模のデータベースの構築作成は一層進展することが考えられるが、これらに蓄積されたデータは将来組織内のみならず、広く外部へ提供されると予想され、今回の実施した調査研究の成果はこれらに十分に対応できると思われる。

しかしながら、コンピュータに習熟していない人々がこの成果を容易に利用するには、もっと簡単にしかも経費がかからない方策を検討する必要がある。このためには、オペレーティングシステムやプログラミング言語をよりわかりやすいものにすることや、現在市販されているソフトウェア・パッケージ等がさらに簡単に利用できる環境の整備が望まれる。

また、パソコン用perlの日本語処理を可能にすることや、Windows版での利用ができること、さらには、普及しているデータベース・パッケージや表計算ソフトウェアを直接CGIで操作できることも、Webサーバを通しての情報流通に大きな効果があると思われる。



—— 禁 無 断 転 載 ——

平成 8 年 3 月発行

発 行 財団法人 データベース振興センター
東京都港区浜松町二丁目 4 番 1 号
世界貿易センタービル 7 階
TEL 03-3459-8581

委託先 日本電子計算株式会社
〒135 東京都江東区東陽 2-4-24
TEL 03-5690-3203

印刷所 日邦印刷株式会社
〒104 東京都中央区入船 3-9-5
TEL 03-3553-4039

