オープンソースによる簡易WebOPACの開発

—全文検索ソフトNamazuの応用—

中尾泰士¹, 村上幸二²

著者抄録: 近年, 大学図書館をはじめ一般の図書館においては, インターネット上で検索できるOPAC (WebOPAC) を導入し, 情報サービスの拡充を図る機関が急増している。しかし多くの場合, それは図書館システムベンダーに依存しており, その導入やシステムの変更にあたっては多大なコストが必要となる。一方, オープンソースソフトウェアを利用したWebアーキテクチャを応用すれば, ほとんど経費をかけず, 独自にカスタマイズが可能なWebOPACの構築が可能である。われわれは本稿において, フリーウェアである全文検索エンジン “Namazu” を利用した簡易WebOPACの開発と, 同システムに付加したWebcat即時参照機能について紹介する。

キーワード: オープンソース, フリーウェア, Linux, WebOPAC, 全文検索エンジン, Namazu, NACSIS Webcat

An attempt to develop a simplified version of WebOPAC system by using open source software

—An application of the full-text search engine of “Namazu”—

NAKAO Yasushi¹, MURAKAMI Koji²

Author Abstract: In recent years, many libraries have set up WebOPAC systems to provide users with information of their databases. Such information systems supplied by vendors, however, tend to be quite expensive. Furthermore, it is difficult for the librarians to customize such a system for its own needs. On the other hand, we can construct a WebOPAC system by using open source software, although it may be limited to some functions. Such a hand-made system is easy to customize and can be maintained without much cost. This paper describes an attempt to develop a simplified version of WebOPAC, which is linked to the NACSIS Webcat to get further information on books and other material.

Key words: open source, freeware, Linux, WebOPAC, full-text search engine, Namazu, NACSIS Webcat

(情報管理 44 (6): 412-420)

1. はじめに

近年の図書館においては, インターネット上で検索できるOPAC (以下, WebOPAC) を導入し, 情

報サービスの拡充を図る機関が急増している¹⁾。また, 国立情報学研究所では, インターネット上で検索可能な総合目録データベースNACSIS Webcatを提供している²⁾。

¹奈良産業大学 経営学部

(〒636-8503 奈良県生駒郡三郷町立野北3-12-1) Tel. 0745(73)7800 E-mail: nakao@nara-su.ac.jp

¹Nara Sangyo University Faculty of Business Administration

(12-1, Tatsuno-kita 3-chome, Sango-cho, Ikoma-gun, Nara, 636-8503)

²奈良産業大学 図書館

(〒636-8503 奈良県生駒郡三郷町立野北3-12-1) Tel. 0745(73)7800 E-mail: murakami@nara-su.ac.jp

²Nara Sangyo University Library (12-1, Tatsuno-kita 3-chome, Sango-cho, Ikoma-gun, Nara, 636-8503)

これらのシステムは、それぞれの機関まで出向かなくてもインターネットを通じて資料の検索ができ、また、Webブラウザを利用したGUI（グラフィカル・ユーザ・インタフェース）によるユーザ・フレンドリーな操作性を実現しているのが特徴である。従来、図書館内設置の目録検索システム（以下、館内OPAC）は、図書館システムの専用端末として機能していた。そして、その検索インタフェースは図書館システムのメーカーによって様々な形が存在し、初心者が利用するには負担が大きかった。このため今日では、館内OPACにおいてもWebブラウザをインタフェースとしたOPACを導入する図書館が次第に増えてきている。

情報技術の進展が著しい今日、日々刻々と進化を続ける多様な技術を柔軟に取り入れ、情報を提供できるシステムを構築することは、現代の図書館にとっても必要なことである。図書館システムベンダーにおいても、情報技術の進歩に伴い、新しい機能を付加したシステムの開発が進められており、WebOPACの提供も最近の潮流となっている³⁾。

しかしながら、図書館システムベンダーに依存したWebOPACの場合、その導入にあたっては多大な費用を要するだけでなく、大規模なシステムで強固に構築される場合が多いため、システムオーナーである図書館側が新しい技術を実験的にシステムに取り込んだり、利用者の要望に応じてシステムに改良を加えたりすることは容易ではない。また、通常、異なるシステム間のデータベースには互換性がないことが多い。そのため、ベンダーの移行は、データベースの移植という多くの費用と労力を要する作業を伴い、結果として、最初に導入したシステムベンダーの制約を受けるということにもなりかねない。そのような観点に立てば、データベースの汎用性を確保しているシステムかどうかは、システム導入の際により重要視されてしかるべきであろう。

一方、提供するサービスを限定しさえすれば、システムベンダーによる館内OPACとWebによる情報提供システムが必ずしも連動している必要はない。例えば、蔵書データベースの静的な検索サービス、すなわち、ある時点でその図書館が所蔵している資料にはどのようなものがあるかを利用者に知らせるサービスであれば、館内OPACが提供するような資料の貸出・予約状況などの動的な情報までリンクさせる必要はなくなる。

筆者らは以上のような観点から、システムベン

ダーに依存しない簡易WebOPACの開発を試みた。システムの構築にあたって念頭においたのは、

- (1) 予算が制約されている、比較的小規模の図書館でも運用できるようにすること
 - (2) 各図書館が独自にカスタマイズ可能なシステムにすること
 - (3) データベースの汎用性を確保すること
- の3点である。次に本システムの構成について述べる。

2. システム構成

本システムは、Webサーバ（厳密に言えばHTTPサーバ）、検索ソフトウェア、蔵書データ、および関連するユーティリティから構成される。

2.1. Webサーバ

予算に制約のある小規模図書館においては、高額なサーバを購入し、そのメンテナンスを日常的に行うことは決して容易ではない。しかし幸いにも、最近では比較的廉価なパソコンであっても、サーバとして十分な能力を発揮することができるようになってきている。ただし、ここでは1度に多数のアクセスが予想されるような状況は想定していない。そこでOSは“Linux”、Web（HTTP）サーバ・プログラムは“Apache”を採用した。LinuxはL. Torvalds氏が作成したUNIXクローンなフリーのOSであり、膨大なフリーソフト資産を擁するUNIX文化をほぼそのまま利用することができる。また、Apache⁴⁾も同じくフリーソフトウェアでありながら、多くの商用サーバでも利用されている定評のあるものである。

本システムに使用したハードウェアは、富士通製ノートパソコンFMV-BIBLO 5233 NU/Y（CPU：MMX Pentium 233 MHz、メモリ：32 MB、ハードディスク容量：3.2 GB）で、これにVine Linux 2.1⁵⁾を導入した。Apacheのバージョンは1.3.14である。

2.2. 検索ソフトウェア

蔵書検索システムにおいて中心となるのは、言うまでもなく検索ソフトウェア部分である。ここでは、

- (1) フリーウェアである
 - (2) 一般によく利用されていて、情報の入手が容易である
 - (3) カスタマイズが可能なオープンソースである
- などの理由から、全文検索エンジン“Namazu”⁶⁾を

採用した。

Namazuは1997年に高林哲氏によって開発され、現在では“Namazu Project”によって開発が続けられている検索エンジンで、数多くのWebサイトで採用されている。検索の速さなど、その性能については定評があり、もともとCGIとして設計されているので、Webサービスとの親和性も高い。

ところで通常、蔵書データベースはリレーショナル・データベースで構築されることが多いが、実は図書館の書誌データにリレーショナル型は合わないことも指摘されている⁷⁾。また、リレーショナル・データベースの構築には、データベース処理の知識が不可欠であり、小規模図書館が独自で行うには困難が伴うものであった。しかし、全文検索システムを採用すれば、高度なデータベースの知識をそれほど必要とせずにサービスを行うことができる。このことはNamazuを採用することの大きな利点と言えるだろう。

Namazuは、インターネット上のNamazuホームページから入手できるし、また最近のLinuxのパッケージには付属している場合が多い。Namazuを利用してCGIによる検索システムを構築する方法については、いくつかの文献が存在する^{8,9)}ので、詳しくはこれらを参考にとよいだろう。なお、我々が使用したNamazuのバージョンは2.0.5である。

2.3. 蔵書データの抽出とフィルタリング処理

蔵書データは、図書館システムのデータベースから抽出したものを利用した。これは図書館の目録業務において、国立情報学研究所のNACSIS-CATに自動登録をするためのアップロード・ファイルとなるものである。具体的には、ISBN、フルタイトル、出版者、著者名、登録番号、請求記号が入力されている。図1にこの蔵書ファイルの構成を示す。

抽出データは各項目がTabによって区切られ、1行に一物理単位の書誌データが記述されている。このようにデータベースを1度、取り扱いやすいテキス

トデータにすると、様々な加工が容易である。また、図書館のローカル・データベースとWebで提供するデータベースを完全に分離することで、セキュリティ面でも都合がよい。万が一、Webで提供するデータが破壊されても、元のデータから何度でも再構築が可能になるからである。

ところで、図書館システムから抽出したデータには、通常インターネット上の文書データとしては不適切な文字が存在する場合がある。半角カナの問題もその1つである。半角カナの全角化を行うにはいくつかの方法があるが、本システムでは“kcc” (Kanji Code Converter) を使用した。“kcc”はインターネットからダウンロードして、Linux上にインストールすることができる (使用したのはkcc-2.3-0v12.i386.rpm)。そのほか、変換が必要な文字については、フィルタ・プログラムをPerlによって自作して対応した。

2.4. 蔵書レコードの生成

図書館システムから抽出した蔵書データは、複数の書誌がまとまった1つの (あるいは複数の) テキストファイルである。このファイルから、一物理単位の書誌ごとに“登録番号.html”のファイル名を持つHTMLファイルを作成する。そのためのプログラムはPerlなどを使えば簡単に作ることができる。

書誌ごとにHTMLファイルを作成するのは、Namazuの検索仕様に合わせるための処置であるが、データの追加・削除が、ファイルの追加・削除に対応しているので、高度なデータベース処理の知識を必要としないという利点も生じる。さらに、ファイルの集合体としてデータベースが構築されているので、データベース全体が、致命的なダメージを受ける可能性が低いことも利点としてあげられるだろう。また、画面の表示において、1書誌に対応するHTMLファイルを自由にデザインすることができる。本システムにおける書誌詳細画面は、シンプルに、書名・著者名・出版者・ISBNの書誌

ISBNKEY/S	FTITLEKEY/S	PUBLKEY/S	AUTHKEY/V	RGTN	CLN
4826801392	現代戦略論	勁草書房	道下 徳成/[ほか]著	0150015	391.3 MIC
0275968594	Silicone spills	Praeger	Mary White Stewart	0150067	495.46 STE

図1 蔵書ファイルの構成

情報と、ローカル情報としての請求記号、そして第4章で詳述するWebcat参照ボタンを配置することにした。

なお、Namazuは指定されたキーワードを中に含む文書ファイルを検索するという仕様であるから、すべてのHTMLファイルにある「詳細表示」や「タイトル」などの文字はそのまま検索インデックスとして登録されてしまう。もしこれらの文字が検索キーワードとして指定された場合、すべてのHTMLファイルが該当してしまう（実際には、Namazuの仕様で、ある一定数以上のヒット件数があった場合には、検索結果エラーを返すことになっている）。そこで本システムでは、これらを検索キーとして登録されない形に工夫している（図2）。

2.5. 検索インデックス作成

Namazuでは、あらかじめ検索対象のデータベース（実は文書ファイル集合体）に対して、キーワードごとのインデックスを作成しておき、検索はそのインデックスを元に行うという形をとる。検索のレスポンスが速いのはそのためである。

検索の精度は、このインデックスに大きく依存するため、インデックス作成時に使用される「かな漢字変換辞書」の性能が問題になる。本システムでは、Namazuに附属する辞書を使用しているが、辞書の独自拡張も可能である。インターネット上には、特定の分野に特化した辞書の拡張を提供するホームページがいくつか存在するので、それらを利用する方法もある。特定の分野の専門書を中心に所

```
<html>
<head>
<META NAME="ROBOTS" CONTENT="NOINDEX,NOFOLLOW">
<title>報酬管理の日本的展開: 宮坂 純一／著: 晃洋書房 </title>
</head>
<body>
<h2>詳<!--細-->表<!--示--></h2>
<FORM action=http://webcat.nacsis.ac.jp/cgi-bin/krkproc method=post>
<INPUT name=db type=hidden value=all>
<table>
<!--title-->
<tr><td>タ<!--イ-->ト<!--ル--></td><td>:</td><td>報酬管理の日本的展開</td>
<INPUT name=title type=hidden value=報酬管理の日本的展開></tr>
<!--author-->
<tr><td>著<!--者-->名</td><td>:</td><td>宮坂 純一／著</td>
<INPUT name=auth type=hidden></tr>
<!--publisher-->
<tr><td>出<!--版-->者</td><td>:</td><td>晃洋書房</td>
<INPUT name=pub type=hidden value=晃洋書房></tr>
<!--year-->
<INPUT name=year type=hidden>
<!--isbn-->
<tr><td>I<!--S-->B<!--N--></td><td>:</td><td>4771004560</td>
<INPUT name=isbn type=hidden value=4771004560></tr>
<!--free word-->
<INPUT name=kennei type=hidden>
</table>
<hr>
N<!--A-->C<!--S-->I<!--S--> W<!--e-->b<!--C-->A<!--T-->
書<!--誌-->情<!--報-->へ --&gt;<INPUT type=submit value=参照>
</FORM>
<hr>
請<!--求-->記<!--号-->: 336.45 | MIY
</body>
</html>
```

図2 HTMLソースファイルの例

蔵している図書館のような場合は、その分野の専門用語を認識できる辞書を用いることによって、検索精度の向上が期待できるだろう。

2.6. 検索画面

検索画面および検索結果の詳細表示画面は、通常のWebOPACで表示されるデザインに近く、かつシンプルさを重視した。大多数のWebOPACに見られる検索項目別の入力フィールドは、多すぎることも指摘されており¹⁰⁾、最近では入力フィールドが1つのタイプと併用するシステムも増えてきている。そこで本システムでは、Namazuの機能も考慮して、入力フィールドは1つとし、入力された検索要求に対して、書名、著者名、出版者、ISBNのどの項目に対しても検索をかけるようにした（実際には、Namazuはフィールドごとの検索機能を備えている。しかし、蔵書検索はもともとNamazuが想定していない使用法であり、フィールド検索を可能にするには、より細かなカスタマイズが必要になる）（図3）。

以上のようなシステム構成によって、テキストファイルの抽出元である図書館ローカル・システムに入力されている全蔵書分（15万件強）を本システ

ムに載せた。データ量が多いほど、Namazu検索用のインデックス作成に時間を要することになるが、1度作成すれば、後は追加された蔵書に対する追加作業のみでインデックスの更新が可能である。

3. 検索機能の検証

次に、本システムの検索能力を検証するために、館内OPACと検索の比較実験を試みた。ただし、両システムの検索環境を等しくするために、館内OPAC側の「件名」や「内容注記」などの検索項目は除外した状態で比較した。また、両システムとも、検索入力フィールドには記号などを用いた検索式を展開せず、単に語句を入力した形で検索パフォーマンスを比較した。

まず、書名タイトル中のキーワードを想定した検索実験を行った。一例をあげると表1ようになる。館内OPACでは拾えない書誌も、本システムではヒットしている。また、うろ覚えの書名をもとに検索することを想定し、書名の一部を用いた検索を行って見たところ、やはり館内OPACで拾えない書誌を検索することができた（表2）。

図3 検索画面

表1 タイトル・ワードを想定して、“環境会計”で検索

館内 OPAC	本システム
環境会計入門 環境会計と環境報告書作成の実務 環境会計と情報開示 環境会計の現状と課題 環境会計の構築 環境会計 (以下、省略)	(←左記のもの以外に) 日本の会計と会計環境 農業における環境会計の理論と実践 よくわかる環境会計 入門・環境会計 ドイツ環境会計論 (以下、省略)
25 件	30 件

表2 書名の一部を想定して、“歴史的展開”で検索

館内 OPAC	本システム
	韓国工業化の歴史的展開 ドイツ中世都市の歴史的展開 イギリス企業経営の歴史的展開 社会権の歴史的展開 中国経済の歴史的展開 (以下、省略)
0 件	18 件

表3 出版者を想定して、“文芸社”で検索

館内 OPAC	本システム
イスタンブールの東では 文芸社 旧制官立高等教育機関の沿革粗描 文芸社 こうすれば個店は生き残れる 文芸社 職場における「空回り」の構図 文芸社 二十一世紀の新経済学 文芸社 凡人の哲学 文芸社	(←左記のもの以外に) 名僧の素顔 近代文芸社 水への旅の道しるべ 近代文芸社 メアリー・バートン 近代文芸社 証拠の問題 現代文芸社 (以下、省略)
6 件	46 件

さらに、フルタイトル、著者名、ISBNのそれぞれを想定した検索実験を行ったが、館内OPACとわれわれが開発したシステムとの間で違いは見られなかった。これらの検索項目に関するかぎり、本システムは館内OPACシステムをかなりよく再現できるといえるだろう。

しかし、出版者を想定した検索の場合、本システムは多くの検索ノイズを出す結果となった。たとえば、「文芸社」という検索要求に対し、本システムは「文芸社」だけでなく、「近代文芸社」のように、語句中に「文芸社」を含むようなデータも抽出して

しまう(表3)。これは、本システムが全文検索によって構築されていることを考えれば致し方ないところであろう。

以上の結果をまとめたものが表4である。

両検索システムにおけるこのような違いは、それぞれの検索システムの成り立ちに大きく依存している。本学図書館の館内OPAC検索システムは、書誌1件ごとにインデックスとなるキーワードを(自動生成プログラムによって)作成している。一般的にキーワード検索システムによるインデックスは、利用者の検索要求をある程度予測した上で調整さ

れる。一方、全文検索システムを利用した本システムでは、HTMLファイル群中出现する語句すべてを検索対象のインデックスとしている。したがって、どうしても検索ノイズが発生しやすい。しかし、調整されたインデックスを持つシステムが常に要求された条件を満たす結果となるとは限らない。例えば、実験にみられるように、書名の一部による検

索要求に対しては、調整されたインデックスが無力な場合もある。そのような検索要求に対応するためには、全文検索システムが力を発揮するだろう¹¹⁾。

4. NACSIS Webcat参照機能

例えば、NCR1987R（日本目録規則1987年版改訂

表4 検索項目による検索結果の違い

想定事項	入力語句	館内 OPAC	本システム
タイトル・ワード	環境会計	25 件	30 件
書名の一部	歴史的展開	0 件	18 件
フルタイトル	現代の金融政策	2 件	2 件
著者名	村上春樹	11 件	11 件
ISBN	4900900486	2 件	2 件
出版者	文芸社	6 件	46 件

図4 検索結果画面とWebcat詳細表示画面

版)やAACR2R(英米目録規則第2版改訂版)では、目録における記述の水準にいくつかのパターンを設定している¹²⁾。一方、本システムで表示されるデータは、図4に示した項目で構成されるものであり、書誌の記述の精度は高くはない。これは、図書館データベースからテキストデータとして抽出できる情報量に依存しているものである。

そこで本システムでは、検索の結果として表示されるHTMLレコードからWebcatの書誌にリンクできる設計にし、書誌データの補完をWebcatに求めるようにした。

具体的には、それぞれの書誌に対するHTMLファイル内に、あらかじめWebcat向けの検索要求フォームを埋め込んでおく。そして、詳細表示画面のWebcat参照ボタンをクリックすると、インターネットを通じてWebcatに検索要求を行い、該当する書誌データを回答してもらう仕組みである。この検索要求フォームは、図書館ローカル・システムから抽出したテキストデータを1書誌ごとにHTMLファイルに分割する際に使用したPerlプログラムによって自動的に埋め込まれる。

これらのことをまとめると、図5のようになる。

5. おわりに

以上、Namazuを利用したWebcat連動型簡易WebOPACの開発の概要を紹介した。本研究によって、フリーウエアのみを使用したシステムにおいても、相当の性能が期待でき、十分に実用に耐えうるものが作成できることが実証された。しかし、いくつかの問題点もある。

まず、全文検索という性格上、第3章でも述べたように、検索ノイズが発生しやすい。逆に、検索もれが起こる可能性は低く、絞り込み検索などによって目的の検索結果に到達することは可能である。しかしユーザにとって、余分な手間をかける欠点があることは否めない。

また現時点において、本システムでは件名、内容注記などを入力していないため、例えば、論文集などにおける個々の著者名を対象とした検索などが不可能となっている。しかし、これは図書館データベースに入力されているすべての書誌項目をテキストデータとして抽出できれば、解決できることである。

以上のような問題があるにしても、多大な費用を

図5 システム概念図

要さずに、WebOPACの機能を持つシステムを構築できることを示した本研究の意義は大きいと思われる。フリーウェアで構築したシステムは、原理的には誰にでも再現可能なシステムであり、必要とあれば、本稿で示したように独自のカスタマイズも可能である。また、導入にほとんど資金を必要としないことから、システムの廃棄や他のシステムへの移行が比較的スムーズに行える。逆に言えば、ベンダーが提供する高額なシステムを導入する際には、そのシステムでなければ提供できないサービスは

何であるか、それが本当に必要かということなど、先々を十分に見越した検討が必要になるといえるだろう。

本稿は、奈良産業大学図書館において簡易版WebOPACを構築する個人的な試みの成果をまとめたものである。なお、Webcat参照機能については事前に国立情報学研究所の了承を得た。最後に、Namazuをはじめとする有用なソフトを開発し、それをオープンソースとして公開している世界中のプロジェクトグループに敬意を表したい。

参考文献

- 1) 林 賢紀. 日本国内図書館OPACリスト. (<http://ss.cc.affrc.go.jp/ric/opac/opacolist.html>), (参照 2001-07-09).
- 2) 国立情報学研究所. NACSIS Webcat. (<http://webcat.nacsis.ac.jp/>), (参照 2001-07-09).
- 3) 上田修一. メーカー別, システム別の大学図書館OPAC (v.2). (<http://www.slis.keio.ac.jp/~ueda/ir/opacsoft2.html>), (参照 2001-07-09).
- 4) The Apache Software Foundation. (<http://www.apache.org/>), (accessed 2001-07-09).
- 5) Vine Linux Home Page. (<http://www.vinelinux.org/>), (accessed 2001-07-09).
- 6) 全文検索システムNamazu. (<http://www.namazu.org/>), (参照 2001-07-09).
- 7) 坂本徹朗. 図書館とコンピュータ. 第3版. 東京, 日本図書館協会, 1996, 318p.
- 8) 馬場肇. 日本語全文検索システムの構築と活用. 東京, ソフトバンク, 1998, 258p.
- 9) 西村めぐみ. オープンソースソフトウェアによる全文検索・データベースWebの作り方. 東京, ソシム, 2000, 395p.
- 10) 上田修一. 大学図書館のホームページとOPACを採点する. 情報の科学と技術, Vol.49, No.12, 1999, p.618-625.
- 11) 山本毅雄ほか. 全文検索: 技術と応用. 東京, 丸善, 1998, 124p.
- 12) 北 克一, 村上泰子. OPACにおける目録構造. 図書館界. Vol.51, No.2, 1999, p.100-108.