

Z39.50 を用いた日本語書誌情報サーバの試作

安齋 宏幸 山本 毅雄 石塚 英弘

anzai@ulis.ac.jp, yamamoto@ulis.ac.jp, ishizuka@ulis.ac.jp

図書館情報大学

〒305 つくば市春日 1-2

ANSIによる標準情報検索プロトコルであるZ39.50での検索機能を有する、JAPAN/MARCサーバ、及び、図書館目録サーバの試作を行った。従来のWorld Wide Webによる情報検索サービスでは、検索結果集合の蓄積やそれらの中でのブール演算、ネットワーク上の複数のデータベースに対する同時検索などが行えなかった。Z39.50に基づいたシステム開発を行うことによって、これらの欠点を克服できる。試作の過程と成果について報告し、明確に単語を切り出し難い日本語の性質からくる、Z39.50の問題点についても指摘する。

AN EXPERIMENTAL IMPLEMENTATION OF Z39.50 FOR JAPANESE BIBLIOGRAPHIC DATABASES

Hiroyuki Anzai, Takeo Yamamoto and Hidehiro Ishizuka

anzai@ulis.ac.jp, yamamoto@ulis.ac.jp, ishizuka@ulis.ac.jp

University of Library and Information Science

1-2 Kasuga, Tsukuba-city, Ibaraki, 305 Japan

An experimental implementation of Z39.50, an ANSI standard for information retrieval, for two Japanese bibliographic databases, JAPAN/MARC and ULIS-OPAC, is described. The standard offers "stateful" searches which, unlike conventional WWW-based searches, allow the server to keep multiple result sets throughout a retrieval session and enables operations among the sets. The standard also allows for a client application to maintain sessions concurrently with multiple servers. Potential problems with using Z39.50 for Japanese language data are also discussed.

1 はじめに

著者らは標準情報検索プロトコルである Z39.50 を用いて情報提供サーバを試作した。Z39.50 は広く開かれた情報サーバを作成することを目標に、情報検索のための検索質問や検索結果、課金や認証について定めた情報検索のための ANSI 標準である [1]。

WWW(World Wide Web)などに代表される新しい情報システムの枠組みが、広く情報の提供と収集の手段として用いられている。著者らは先に、日本語が明確に分ち書きされた単語を持たない言語であることに注目して、日本語の個人における単語概念の揺れに対応した書誌情報提供システムを、複数ハッシュふるい分けによる単語切りわけアルゴリズムを用いて試作した [2]。

この際のデータには国立国会図書館の許可を得て JAPAN/MARC を採用し、システムの作成には AD-ABAS と NCSA httpd を用いた。この研究により、複数ハッシュふるい分けが単語の区切りを明確に定めるにくい日本語を対象にした情報の提供に優れていること、手探り検索とハイパー目録(書誌レコードをノード、共通の著者や主題をリンクとする)の実現が WWW で可能なことを示した。システムの全体構成を図 1 に示す。

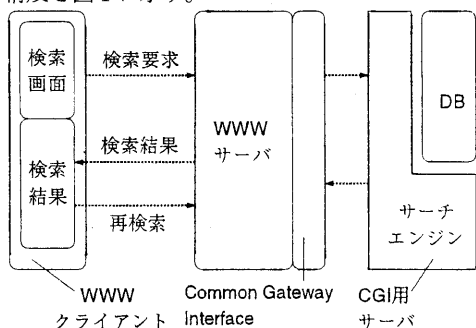


図 1 WWW を用いた情報検索の概念図

1.1 現行の WWW による情報検索システムの問題点

情報提供に WWW を用いるシステムでは、検索結果集合の蓄積とそれらとのブール演算ができない。またユーザは、関心を持つ情報を扱っている複数の情報サーバを順番にまわらなければならない。

検索結果集合が蓄積できない問題は、WWW を情報提供に用いる限り本質的解決をはかることができない。これは、WWW が HTTP(Hyper Text Transfer

Protocol) を利用した情報システムであることに起因している。HTTP は HTML(Hyper-Text Markup Language) の文書のリクエストを出してそれを受け取るという、コネクションレスのクライアント-サーバの関係で成り立っており、この過程では検索結果集合を作成するためのコネクションを維持した処理を行うことはできないからである [3][4]。

また、ユーザが複数の情報サーバを一度に検索することができない問題は、つぎの二点に起因している。

第一に、情報サーバが共通する検索質問構造を持っていない点である。それぞれの情報サーバは独自の検索質問フォームで動作している。複数の情報サーバに一度に検索をかけるためには、共通の機能について、対象とするサーバそれぞれに対応する検索式に変換して検索をかけなければならない。

第二に、WWW が検索結果を構造化されたデータとして返すための約束を用意していない点である。レコードは HTML 文書としてクライアントに返されるので、複数のサーバから受け取ったレコードの重複を削除したり、ソートするなどといった処理を行うことは難しい。さらにこの制約は、システムの障害や検索式のエラーといった情報サーバからのメッセージについても同様に課せられる [4]。

このように、現行の WWW を用いた情報検索はユーザの視点からみると多くの問題を抱えている。これらの問題は、情報検索のための標準プロトコルである Z39.50 を用いることで解決できる。

2 Z39.50 とは

Z39.50 は、クライアント-サーバ環境¹で、複数の情報サーバにたいして検索を行うことを支援するための ANSI(American National Standards Institute) 標準である "Information Retrieval Service Definition and Protocol Specification" の識別番号である [5]。これまでに、化学構造や一般科学情報、ビジネス情報やポストスクリプト、SGML、フルテキスト、イメージなどへの応用研究が進められている [5][6]。

Z39.50 は現在のところ国際標準ではない。ANSI の認可のもと NISO(National Information Standards Organization) が情報検索や流通の分野での標準化の一部として Z39.50 を整備し、1988 年以来米国議会図書館の The Network Development and

¹Z39.50 の仕様ではクライアントを origin、サーバを target、検索セッションを association と呼ぶ。本稿ではより一般的な語であるクライアント、サーバ、セッションという言葉を用いる。

MARC Standards Office が管理している。Z39.50-1992(Version.2) は国際的な情報検索プロトコル (SR プロトコル) である ISO 10162/10163—the Search and Retrieval(SR) Service Definition and Protocol Definition に対し上位互換性を有している [7] が、これは近く国際標準として採用される予定である [12]。Z39.50 クライアント-サーバの概念を図 2 に示す。

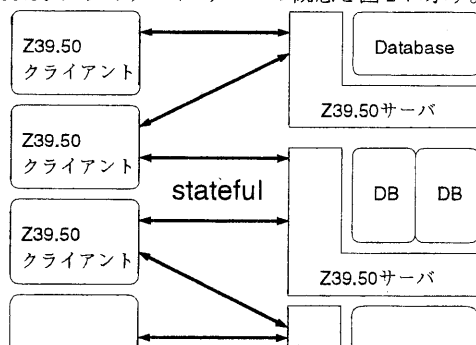


図 2 Z39.50 を用いたクライアント-サーバ概念図

Z39.50 は、遠隔地の情報資源に対する、検索のためのプロトコル (約束手順) を提供する。プロトコルはクライアントとサーバ間の相互作用についてのみを対象にしており、サーバにおける実際の検索処理などについては扱わない。しかし、Z39.50 に基づいたクライアントとサーバを設計することで、ユーザは遠隔地の一つまたは複数の分散したデータベースを、同じ手順で一度に検索できる。検索結果はクライアントとサーバの間で了承された構造データとして返される。

このように Z39.50 をシステムに実装することで、コネクションを意識した検索処理の実現、クライアントソフトウェアを特定しないサーバ構築、複数の情報サーバへの同時検索が可能となる。最新版は Z39.50-1995(Version.3) であり、本報告では Z39.50-1995 に基づいて試作サーバの開発を行った。以下に Z39.50 の経緯や機能のうち、本報告に関連する事柄について紹介する。

2.1 ASN.1 と BER

Z39.50 プロトコルはデータ型 (構造) 記述言語である Abstract Syntax Notation (ASN.1、抽象構文記法 1) を用いて記述され、Basic Encoding Rules (BER) に従ってコード化される。異なったアーキテクチャを持つコンピュータ間で構造をもつデータの交換を行うために OSI (Open Systems Interconnection) の

枠組みのなかで開発された規約で、複数のマシン間でのデータのやりとりを保証している。

Z39.50-1995 (Version.3) は主に TCP/IP プロトコルを用いて実装されているが、ASN.1 と BER の目的と機能は維持されている [8]。

2.2 stateless と stateful

WWW を支える HTTP が stateless、すなわちクライアントに対応するセッションの確保が行われないのに対し、Z39.50 は stateful (session-oriented) である。すなわち Z39.50 ではサーバが、クライアントとの累積する検索結果の積み上げと、セッション中の一貫したユーザの識別を行う [3]。

stateful な検索処理の代表的な長所には検索結果集合の蓄積とそれらの演算処理の実現、ならびにクライアント-サーバ間での検索機能の折衝とその維持がある。

2.3 Z39.50-1995 (Version.3) の機能

最新版である Z39.50 v.3 には、書誌情報検索という枠組みを越えて、イメージデータなどの binary を扱うための拡張機能をはじめ、認証、ソート機能、インデックスのスキニング等、豊富な機能が定められている。

これらのうち、検索時の文字セットの設定や機能の折衝を行う Initialize 機能、検索を実行する Search 機能、検索結果レコードのリトリブ (返戻) 要求を満たす Present 機能の 3 つをもつサーバを基本サーバと呼ぶ [3]。本報告ではこの基本サーバを作成した。

2.4 サーバへの検索質問の構造

Z39.50 の検索セッションで用いることのできる検索質問の構造として、Type-0, Type-1, Type-101, Type-102, Type-2 などが用意されている。しかし、現在まで主に用いられている検索質問の構造は Type-1 である。本試作でも、Type-1 Query での検索処理についてのみを実装した。

Type-1 Query は逆ポーランド記法で記述された、ツリー構造の検索質問である。leaf ノードはオペランドを、non-leaf ノードは複合オペランド (オペランドとオペランド、そしてそれらを結ぶ演算子) を表す。オペランドはデータベースのレコード集合を表しており、次の種類などが用意されている。

- AttributeList + term

検索語と、その検索語の (アトリビュート) に

よって検索された結果集合。

- ResultsetId

検索過程で Id を付与された検索結果集合。

用いるアトリビュートがどの集合であるかは、アトリビュートセットで表される。書誌検索のために作成された Bib-1 アトリビュートセットが、おもに用いられている。これらの枠組みに基づく定められた記号の組合せで、利用するインデックスの種類やトランケーションの形態を指定することによって、検索要求における検索語の属性を表現できる。

2.5 サーバから返されるレコードの構造

サーバからクライアントにレコードを返す際の標準文法が定められているのも、Z39.50 の大きな特徴である。

一番簡単な場合、検索結果の集合は Simple Unstructured Text Record Syntax(SUTRS)、つまり構造のない ASC II のテキストによってクライアントに返される。それ以外の代表的な返戻時のレコード構造には、米国議会図書館の機械可読形式書誌の構造である USMARC や、サーバ情報の提供に用いられる Explain などが用意されている。

3 試作システムの構成

本報告に際して、ふたつのサーバを試作した。

1 年分の JAPAN/MARC データベースのローマ字の情報について、Z39.50 の検索要求に応じて提供するサーバを構築した。この JAPAN/MARC システムは、先の報告の cgi(common gateway interface) 用サーバに Z39.50 インタフェースをフロントエンドとして開発し、実現した。

また、図書館情報大学附属図書館の目録データ (ULIS-OPAC) のサブセットを日本語データとして提供する Z39.50 基本サーバを構築した。大型汎用計算機 HITAC M660 で提供されていた ULIS-OPAC データベースからレコードをコピーし、UNIX WS 上で必要なデータの変換を行った。

これらのサーバやインタフェースは、C 言語で開発した。試作と動作環境には UNIX WS を用いた。

3.1 サーバ構成

JAPAN/MARC, ULIS-OPAC の双方ともに DBMS には ADABAS を使い、Z39.50 を実装するためのツールとして INDEX DATA 社の YAZ(Yet Another Z39.50 toolkit)[9] を採用した。

3.1.1 JAPAN/MARC サーバ

JAPAN/MARC サーバの構築は、Z39.50 の検索要求を変換して cgi のためのサーバにリクエストを行う、データベースインタフェースの開発で実現した。この方法は、すでに稼働している WWW 情報サーバを Z39.50 で検索可能にすることで、どのような効果をもつか確かめる有効な方法である。

cgi を用いた検索サーバは、受け取ったフォームからの検索要求を文字列として受け取り、それを処理して結果を HTML として返している。

Z39.50 データベースインタフェースは、受け取った Z39.50 検索要求を cgi 用のスクリプトに変換して検索を行う。またサーバから返された HTML 形式レコードを、Z39.50 に定められた形式に変換し、返戻する。

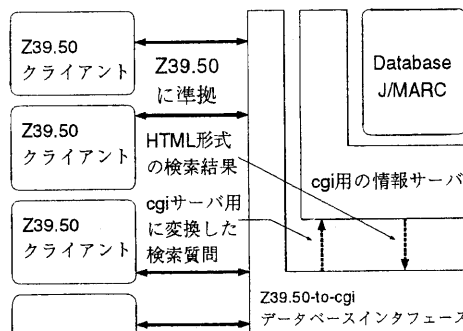


図3 Z39.50 DI による JAPAN/MARC システム

図3にクライアントを含む JAPAN/MARC システムの全体構成を示し、システムの動作を簡条書で説明する。なお、データベースインタフェースを DI と省略して記述することがある。処理は Search, Present に関連する項目について任意回数行われる。

JAPAN/MARC システムの動作

1. Z39.50 クライアントがセッションを OPAC サーバに要求; *Initialize Request*
2. プロトコルエンジン (YAZ) が要求を受信 (以降、プロトコルエンジンについての言及は省略する)
3. DI がセッションを開始
4. クライアントが検索を要求; *Search Request*
5. DI が検索要求を受信し、cgi 形式の質問に変換 (応えられない質問構造には Z39.50 で定められたエラーメッセージをクライアントに送る)
6. cgi 形式の質問を JAPAN/MARC サーバに送る
7. DI が検索結果件数などをクライアントに送信 (

この際には検索結果レコードも HTML 形式で JAPAN/MARC サーバから返されるので、これを一時的に DI で保存しておく)

8. クライアントは検索結果レコードの返戻を要求 ; *Present Request*
 9. DI が検索結果を HTML から SUTRS 構造に変換して送信
 10. トランスポート層の切断でDIのセッション終了
- データベースインタフェースはつぎの処理を行う。
- Z39.50 サーバとのセッション維持
 - Z39.50 検索要求を cgi サーバ形式へ、cgi 検索でのエラーを Z39.50 形式へ変換して転送
 - 検索結果 (HTML) を SUTRS に変換、返戻

3.1.2 ULIS-OPAC サーバ

試作したサーバはフロントエンドとして YAZ が Z39.50 リクエストを受信し、バックエンド DBMS である ADABAS をコール、検索結果を Z39.50 の SUTRS(Simple Unstructured Text Records) 形式に変換して返す処理を行う。

プロトコルエンジンは PDU(Protocol Data Unit) の転送、BER エンコーディングとデコーディング、接続状態の維持と監視などを行う。それぞれのリクエストをあらわす PDU は、コントロールモジュール内の対応するサーバロジックに渡される。

サーバロジックは受け取った Z39.50 リクエストを処理する、C でコーディングされた手続きに相当する。サーバロジックは Initialize, Search, Present に対応して ADABAS に対して必要な照会を行い、リクエストに対応した Z39.50 仕様のデータ構造を YAZ プロトコルエンジンに渡す。

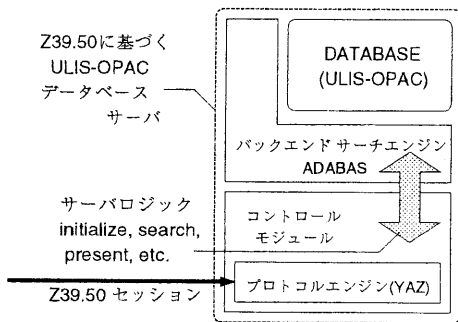


図4 Z39.50を用いたULIS-OPACサーバ

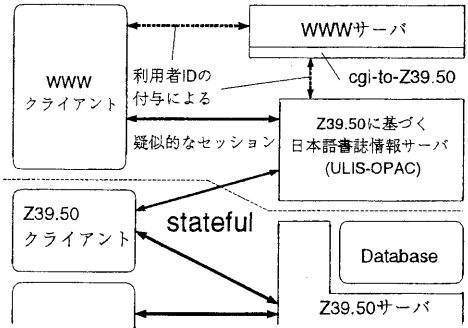


図5 ULIS-OPACシステム構成

図4に試作したULIS-OPACサーバの構成を、図5にクライアントを含むシステム全体構成を示し、システムの動作について箇条書で説明する。JAPAN/MARCシステムと同じく処理は *Search*, *Present* に関連する項目について任意回数行われる。

ULIS-OPACシステムの動作

1. Z39.50クライアントが検索セッションをOPACサーバに要求 ; *Initialize Request*
2. Z39.50プロトコルエンジン(YAZ)が要求を受信
3. コントロールモジュールがADABASデータベースにアクセス (以降2と3の記述省略)
4. データベースが利用可能なら、要求の受諾を送信
5. クライアントが検索を要求 ; *Search Request*
6. サーバが検索成否と件数を送信
7. クライアントは検索結果レコードの返戻を要求 ; *Present Request*
8. サーバはレコードをSUTRS構造で返す
9. トランスポート層の切断でセッション終了 (*Terminate* 機能を用いることもある)

3.2 クライアント構成

JAPAN/MARCの検索

試作したサーバのうち一方のJAPAN/MARCサーバでは、データベース中のローマ字に変換できる書誌情報についてをローマ字で提供している。この限りでは、日本語の文字コードの問題は生じない。ワシントン大学で開発されたZ39.50 GUIクライアントであるWillow[10]を用いた検索事例を紹介する。

ULIS-OPACの検索

もう一方のULIS-OPACサーバでは、日本語(和書誌)の書誌情報を提供している。日本語の文字コードを正しく伝えることのできるZ39.50クライアントソフトウェアについて、著者らは存在を確認していない。このため、当面サーバを評価する目的で

Z39.50 ULIS-OPAC サーバに対し cgi を通して疑似的に stateful な検索を行うことのできる検索ページを、WWW のフォーム機能を用いて作成した。

上の目的のため、cgi-to-Z39.50 インタフェースを開発した。この方法では、それぞれが別個である Z39.50 セッションを維持するために、cgi で呼び出されるコマンドはおのこの識別用 ID を引数にもつ。

クライアントとなる cgi の検索コマンドから Initialize 要求を受けた cgi-to-Z39.50 インタフェースは、自分の子プロセスを作成する。作成された子プロセスは親プロセスに代わって、Z39.50 サーバに Initialize 要求を行う。Initialize に成功すると、子プロセスはセッションの成功を親プロセスに伝える。親プロセスはセッションを維持する子供プロセスを識別するためのユニークな ID を作成し、Initialize 成功の連絡と ID を埋め込んだ検索用フォームを cgi の検索結果 HTML として WWW クライアントに返す。

以後の検索はすべて ID を通した WWW クライアントと Z39.50 サーバとの、疑似的な stateful セッションとして行われる。WWW クライアントからの検索終了の連絡が、一定時間の検索要求途絶で、子プロセスは Z39.50 セッションを切断し終了する。

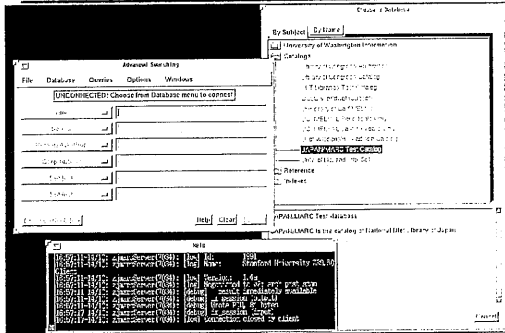
4 システムの稼働例

試作した JAPAN/MARC サーバ、ULIS-OPAC サーバのそれぞれに対する検索例を、クライアントからみた実行画面をもとに紹介する。

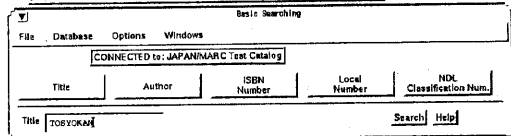
4.1 JAPAN/MARC システム

Z39.50 クライアントから、JAPAN/MARC test database を選択する¹(Initialize)。

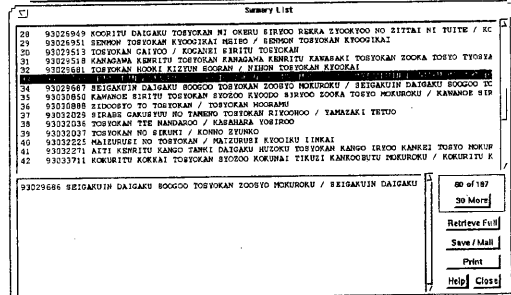
実行画面 1 接続するデータベースを選択する



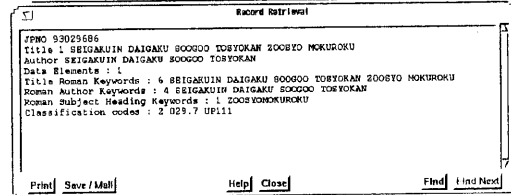
実行画面 2 アクセスポイントと検索語の入力



実行画面 3 検索結果の一覧



実行画面 4 検索結果レコードの詳細表示



検索のためのアクセスポイントを選択し、検索語を入力する²(Search)と、検索結果が一覧表示される³。必要なレコードは詳細表示を要求できる⁴(Present)。この例は tosyokan という単語を書名に含む書誌の検索を示す。

なお、Z39.50 サーバのアドレスや利用できる検索項目、対応するアトリビュートなどの情報はあらかじめ Z39.50 クライアントに登録されている。

4.2 ULIS-OPAC システム

cgi-to-Z39.50 インタフェースを用いた、WWW からの Z39.50 ULIS-OPAC サーバへの検索例を、実行画面をもとに紹介する。

実行画面 1 Z39.50 ULIS-OPAC サーバに接続
ULIS-OPAC DATABASE

CGI-TO-Z39.50 PROTOTYPE SEARCH FORM

安藤宏幸の修士論文に作る研究のために作成されました。
(本システムは竹塚図書館のサービスではありません。)
データの著作権は図書館情報科学大学附属図書館にあります。

検索を行うためには、まず Z39.50 サーバとの Initialize (セッション開始) が必要です。セッションを維持するのに必要なのは、以後あなたに提供される HTML の検索フォームのなかに埋め込まれます。

Request ボタンを押してください。



mpn@ulib.ac.jp

利用者はインタフェースを意識せず、情報サーバのあるページを開いて検索開始ボタンを押すことで、セッションを始めることができる¹(Initialize)。

実行画面 2 検索フォーム

検索条件を入力して、SEARCHボタンを押してください。

SEARCH



- 検索対象
- 検索語

実行画面 3 検索結果の表示

検索に成功しました。
該当レコードは10件 集合番号は1です。



実行画面 4 蓄積した検索集合同の演算

検索条件を入力して、SEARCHボタンを押してください。

SEARCH

集合番号: 1 | 演算子: OR | 集合番号: 1

検索対象項目をボタンで選び、検索語を入力して検索要求をZ39.50サーバに送信する²⁻³。また、セッションで蓄積した検索結果集合同で論理集合をとることも可能である⁴(Search)。

実行画面 5 結果レコードの返戻要求

表示するレコードを入力して、PRESENTボタンを押してください。

PRESENT

- レコード範囲の指定

実行画面 6 書誌レコードの簡略表示

No. 4 / BN0316472K: コンピュータ入門 日本電気情報処理教育部編
日本能率協会
No. 5 / BN045699K: コンピュータ重要語彙引(集新聞・雑誌からの情報検索 情報科学研究所編 情報科学研究所)
No. 6 / BN0012220: コンピュータ重要語彙引(集新聞(5版)・雑誌(60誌)からの情報検索1970年版 情報科学研究所編 情報科学研究所)
No. 7 / BN0012230: コンピュータ重要語彙引(集新聞(5版)・雑誌(60誌)からの情報検索1971年版 情報科学研究所編 情報科学研究所)



実行画面 7 書誌レコードの詳細表示

Record: 1
書誌レコードID: BN0012250
レコード作成日付: 19001027
刊年: 1974
本原語言語コード: ja
書名: コンピュータ重要語彙引(集新聞・雑誌からの情報検索)1974
年刊
著者名: 情報科学研究所編
出版事項1: 東京 情報科学研究所 1974
枚数: 40p
大きさ: 24cm
著者名標目: 情報科学研究所/分類番号: 007.09/件名: ドキュメンテーション/電子計算機/
所蔵情報所蔵レコード: 請求記号: 007.032-661974
物理型/ID: 790001520 利洋コード: 1 記号場所: 1 閲覧閲覧



検索結果集合同レコードの返戻を要求する⁵。簡略表示⁶と詳細表示⁷が、利用者に提供される(Present)。

実行画面 8 検索終了ボタン

検索を終了する場合は、TERMINATEボタンを押してください。

TERMINATE

実行画面 9 セッションの終了

セッションをクローズしました。

大事な注意

検索結果のページなどをurl(ブックマークなど)で保存することは出来ません。必ず出力するか、ファイルとして保存してください。



検索が終了したら、セッションを終了する⁸⁻⁹。クライアントからの検索要求が一定時間ない場合は、インタフェースがセッションを切断する。

5 システムの評価

5.1 JAPAN/MARC サーバ

本試作サーバでは、データベースインタフェースがZ39.50セッションを維持することで、既存のWWWの情報サーバによるZ39.50検索処理を実現した。

これによって、従来は特定のWWWのページからしか検索できなかったJAPAN/MARCサーバが、ユーザが望む任意のZ39.50クライアントで検索できるようになった。これまでのWWW情報システムの場合、ユーザは好みのクライアントこそ使えるが、情報サーバの検索フォームにあわせて、それぞれ異なったレイアウトに対する習熟と操作が必要だった。Z39.50データベースインタフェースの実装によって、この欠点を克服している。

しかしながら、このアプローチには2つの大きな問題点がある。ももとのcgi用サーバがstatefulに対応できない構造のため、検索集合の蓄積や集合同の論理演算などを実装することができない点と、検索結果をHTMLからZ39.50のレコードテキストに変換する際に、テキスト文字列に依存した複雑な処理が必要となる点である。

5.2 ULIS-OPAC サーバ

Z39.50に基づいてULIS-OPACサーバを作成したことで、過去の検索履歴から検索集合を取り出し、論理演算を行うことのできるstatefulなシステムを構築することができた。

我々の試作サーバでは、cgiで実現されている検索の各々について、結果集合が検索履歴として作業の終了まで保存される。ユーザはこれら検索結果を組み合わせてAND, OR, NOTなどの論理演算を行い、検索結果を絞り込むことができる。これまでのcgiを用いた検索システムでは、その都度つぎの検索条件のコマンド引数をフォームに与えることで、再検

索を促すことが限界であった。本システムでは、このような限界を克服している。

6 今後の課題

著者らは上のように日本語データベースに対して Z39.50 に基づく基本サーバを試作し、stateful な検索を実現した。しかしながら、将来 Z39.50 が ISO あるいは JIS に採り入れられる場合、検討すべき課題は多い。

6.1 英語以外の言語への対応

Z39.50 は米国においてデータベースサービス機関や図書館など、それぞれの専門家が要求や折衝を重ねてつくられたプロトコルであり、いまでも討議が続けられている。しかしアメリカなど英語圏の取り組みにおいても、ヨーロッパ言語に対する文字の排列や翻字の問題などが指摘されている [11]。語を空白で区切る習慣のない、日本語のように構造の大きく異なる言語では、問題はさらに大きくなる。

たとえば、検索語の構造を示す用意されたアトリビュート（属性）についても、それがワードなのかフレーズなのかといった、日本語では定義しにくい概念が用いられている。マルチバイトコードや漢字、フォントとその体系、外字処理などについても十分な検討が必要である。

6.2 レコード構造の標準化

書誌情報の提供の際に該当レコードを Z39.50 のクライアントに返戻する場合、米国では USMARC (米国における書誌情報の機械可読レコード標準) がレコード構造として定義されている。

一方、日本においてはそのような標準は存在しない。従って、今回のように便宜的に非構造テキストとしてレコードを返してやるか、亜流となるのを承知のうえで他のフォーマットに無理にデータ押し込むしか方法はない。これでは、検索結果の重複レコードの同定といった Z39.50 の大きなメリットを生かすことはできない。

検索結果をきちんとした構造化データとしてクライアントに渡すことが可能になり、複数の書誌情報サーバが日本で稼働するようになれば、分散したデータベースを一度に検索することを目指した Z39.50 の価値は飛躍的に高まる。

7 さいごに

新しい情報システムの枠組みに求められるのは、いかに効率的に情報を共有し利用するかという基本的な課題である。書架から始まって組織構造に至るまでの既存の枠組みがコンピュータとネットワークによって抜本的に変化し、組織や国家の力がこの「情報の共有」によって飛躍的に高められるだろうという認識が一般的となっている。情報の効率的共有のためには、標準化とオープン指向を避けて通ることはできない。このためのひとつの方向として、Z39.50 に基づいて日本語書誌情報を提供する情報サーバを試作した。

参考文献

- [1] ANSI/NISO Z39.50-1995. Information Retrieval (Z39.50): Application Service Definition and Protocol Specification.
<http://lcweb.loc.gov/z3950/agency/>
- [2] 安齋 宏幸, 山本 毅雄. WWW による JAPAN/MARC の提供実験. 情報処理学会情報学基礎研究会報告. Vol. 95, No. 106, p.9-16(1995.11)
- [3] John A. Kunze. Basic Z39.50 Server Concepts and Creation.
<http://lcweb.loc.gov/z3950/agency/articles.html>
- [4] S. Hammer. J. Favaro. Z39.50 and the World Wide Web.
<http://www.indexdata.dk/webz.html>
- [5] Z39.50 Server System: OCLC SiteSearch Software.
<http://www.oclc.org/>
- [6] 桂 啓壯. OPAC の変容: 欧米の動向を中心にして. 現代の図書館. Vol.33, No.4, p.264-273(1995)
- [7] Clifford A. Lynch. The Z39.50 Protocol in Plain English.
<http://ds.internic.net/z3950/pe-doc.txt>
- [8] Clifford A. Lynch. Request for Comments: 1729. Using the Z39.50 Information Retrieval Protocol in the Internet Environment.
<http://ds.internic.net/rfc/rfc1729.txt>
- [9] Index Data.
<http://www.indexdata.dk/>
- [10] Willow Information Center.
<http://www.cac.washington.edu/willow/home.html>
- [11] Makx Dekkers. Implementing Z39.50 in a multi-national and multi-lingual environment.
<http://lcweb.loc.gov/z3950/agency/articles.html>
- [12] 上田修一. Z39.50 の可能性と問題点. 1996. 私信 (1996年11月2日三田図書館・情報学会研究大会予稿集掲載予定).