Z39.50 プロトコルを用いた検索クライアントの開発

94130 江草 由佳

1998 年 1 月

目次

表一覧

図一覧

1　はじめに

今日使われている大多数の商用オンラインデータベース検索サービスでは、大型計算機(mainframe) に検索システムを搭載し、端末の制御を大型機から行なっている [1]。

これらのオンラインデータベースを検索するには、システム毎のコマンドやデータベース構造を覚えなければならないということが大きな問題としてあげられる [2]。しかし、1 つのデータベース機関が多数のデータベースの提供する DIALOG のようなサービスが登場し、多様なシステム、多様な検索方法という問題は解決され、大きな問題とはなっていなかった。

現在、 WWW の普及により多くの OPAC が WWW 上で提供されるようになってきた。WWW 上で提供されている OPAC は、従来の OPAC よりユーザインタフェースに優れ、使いやすい [3]。 yahoo や infseek に代表されるサーチエンジンなど WWW 上で提供されている情報検索システムも現れた。 WWW は HTTP という検索を目的としていないプロトコルを使っているために、大型計算機時代からの履歴を利用した検索を行なえない。このように、ネットワーク上に OPAC や WWW 上のサーチエンジンなど様々な情報検索システムが存在し、アクセスできるようになった。しかし、これらの検索システムに対して検索を行なうためには、検索システム独自の検索クライアントからアクセスしなければならず、検索システムごとに違ったインタフェースやシステムに習熟しなければ検索できない。ユーザーが検索システム固有の検索クライアントからしかアクセスできない理由は大きく 2 つあげられる。

- 検索システムごとに検索質問の構造が違う。
それぞれの検索システムは独自の検索式の構造を持っており、 1 つの検索クライアントからアクセスするには、検索システムごとの検索式の構造に変換しなければならない。

- 検索結果の構造が検索システムごとに違う。
検索システムごとに違う検索結果を利用するには、検索システムごとの検索結果の構造を解析しなければならない。

これに対し、 Z39.50 は検索クライアント検索サーバ間のやり取りを標準化しているので、 Z39.50 を用いた検索クライアントを使うことにより、ユーザは固有のインタフェースやシステムの違いを意識せずに Z39.50 を用いた検索サーバにアクセスし、検索できる。また、 Z39.50 はセッションを維持したままの検索を約束しているので、履歴を利用した検索もできる。 Z39.50 を用いた検索システムは、アメリカ議会図書館の Z39.50 管理機構に登録されているだけでも 100 を越えており、増加の一途をたどっている。

そこで、このような背景をふまえて、本研究ではクライアントサーバ間のやりとりを標準化し、一元的にアクセスできる「情報検索プロトコル Z39.50」を用いた検索クライアントの開発を行なう。図 1は Z39.50 検索システムのイメージを表している。左にクライアント、右にサーバがあり、その間のやりとりを Z39.50 で行なう。クライアントは、特定のサーバだけでなく、 Z39.50 を用いたサーバであればどのサーバでもアクセスでき、

また、サーバも特定のクライアントからだけではなく、 Z39.50 を用いたクライアントであればどのクライアントからでもアクセスできる。

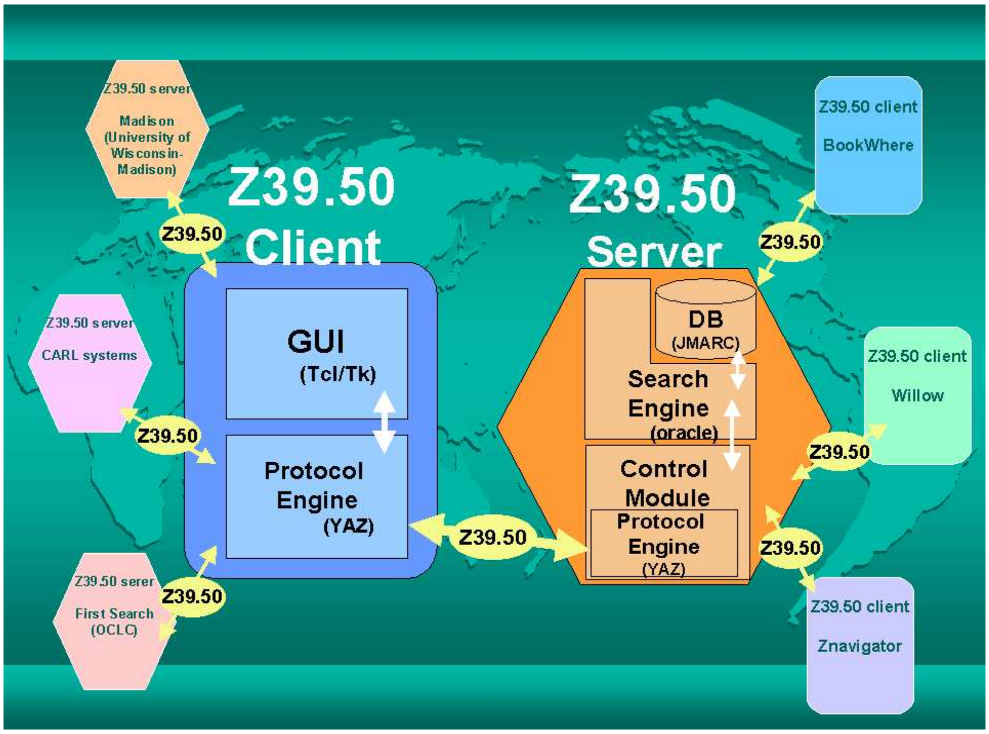

図 1: Z39.50 の全体像

2 Z39.50

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標準検索プロトコルである[4]。Z39.50は、検索質問、検索結果、課金、認証などクライアント、サーバ間のやりとりのみを標準化しており、クライアント、サーバはそれぞれ自由に構築できる(図2)。

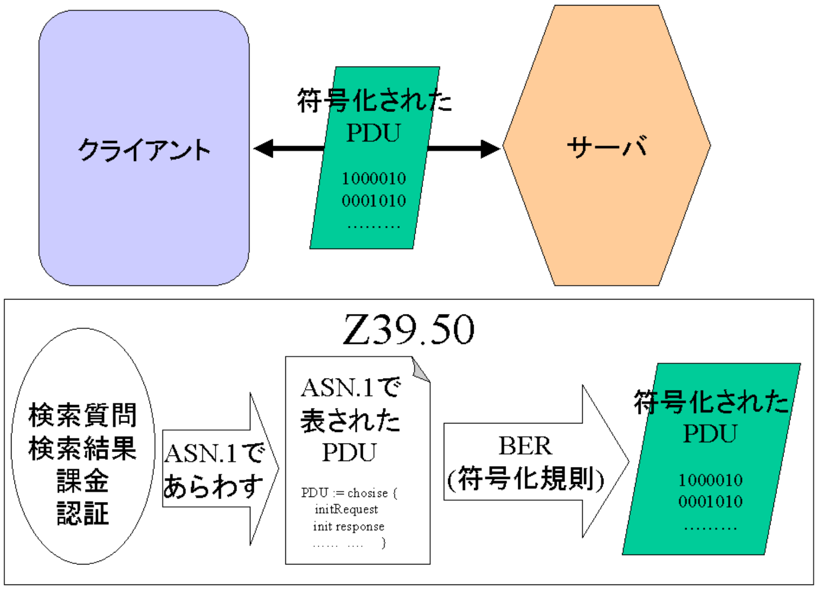

図 2: Z39.50

Z39.50はクライアントサーバ間でやり取りする値、つまり、接続の時に必要な情報であるIPアドレス、検索の時に必要な検索式の構造、検索結果を返す時に必要な検索結果の構造、こういったものすべてをPDU(Protocol Data Unit: プロトコルデータ要素)として、機種やアプリケーションに依存せずにデータが記述できるASN.1を用いて記述し、BERで符合化して転送する。

2.1 ASN.1

Z39.50では、機種に依存しない通信を可能とするために、ASN.1を使用している。ASN.1(Abstract Syntax Notation.1 ：抽象化構文記法１)とは「PDUのデータ構造およびそのPDUを転送する場合のオクテット列の形式を規定する言語」、「計算機に独立なデータ表現手法」[5]としてのISO標準である。以下はZ39.50のPDUの最上位部分をASN.1で表している。

```
PDU ::= CHOICE{
 InitRequest                      [20] IMPLICIT InitializeRequest,
```

```
InitResponse                    [21] IMPLICIT InitializeResponse,
searchRequest                   [22] IMPLICIT SearchRequest,
searchResponse                  [23] IMPLICIT SearchResponse,
presentRequest                  [24] IMPLICIT PresentRequest,
presentResponse                 [25] IMPLICIT PresentResponse,
deleteResultSetRequest          [26] IMPLICIT DeleteResultSetRequest,
deleteResultSetResponse         [27] IMPLICIT DeleteResultSetResponse,
accessControlRequest            [28] IMPLICIT AccessControlRequest,
accessControlResponse           [29] IMPLICIT AccessControlResponse,
resourceControlRequest          [30] IMPLICIT ResourceControlRequest,
resourceControlResponse         [31] IMPLICIT ResourceControlResponse,
triggerResourceControlRequest   [32] IMPLICIT TriggerResourceControlRequest,
resourceReportRequest           [33] IMPLICIT ResourceReportRequest,
resourceReportResponse          [34] IMPLICIT ResourceReportResponse,
scanRequest                     [35] IMPLICIT ScanRequest,
scanResponse                    [36] IMPLICIT ScanResponse,
                             -- [37] through [42] reserved
sortRequest                     [43] IMPLICIT SortRequest,
sortResponse                    [44] IMPLICIT SortResponse,
segmentRequest                  [45] IMPLICIT Segment,
extendedServicesRequest         [46] IMPLICIT ExtendedServicesRequest,
extendedServicesResponse        [47] IMPLICIT ExtendedServicesResponse,
close                           [48] IMPLICIT Close}
```

2.1.1 BER

ASN.1 はデータ型を表現する言語なので、 ASN.1 で記述された PDU を、ネットワークで転送するためには、バイナリーデータに変換する必要がある。 BER は (Basic Encoding Rules：基本符合化規則) ASN.1 で規定された構造のデータをバイナリデータに変換する符合化規則である。

2.2 request と response

Z39.50 では、クライアントからの要求を request、サーバのクライアントの対する結果を response といい、クライアントから request が出されると、サーバからは 必ず response を返す。(図 3)

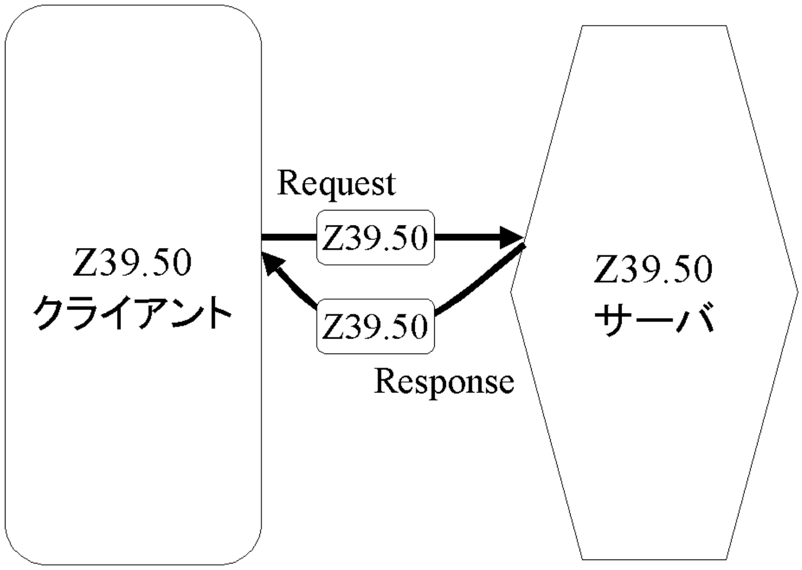

図 3: クライアントからの要求とサーバからの結果

クライアントから出される要求である request とサーバから出される response は、Initialize, Search, Present, DeleteResultSet, AccessControl, ResourceControl, ResourceReport, Scan, Sort, Segment, ExtendedServices, Close の１２あり、最も基本的なものは Initialize , Search , Present の３つである。

2.3 Initialize 機能 (接続機能)

クライアントは接続するための情報である接続要求 (Initialize request) を出し、サーバはクライアントからの接続要求を受けるか拒否するかの情報、接続結果 (Initialize response) を返す。 Initialize request には、プロトコルバージョン、サポートする機能、クライアントの情報などがある (表 1)。 Initialize response にも同様に、プロトコルバージョン、サポートする機能、サーバの情報などがある (表 2)。

表 1: Initialize request

必須のもの	Version(プロトコルのバージョン) Options(サポートする機能の宣言) Preferred-message-size(メッセージサイズ) Exceptional-record-size(例外メッセージサイズ)
省略可能なものとして	Id/authentication(ID 認証) Implementation-id(実装者の ID) Implementation-name(実装者の名前) Implementation-version(バージョン) User-informatation-field(利用者情報のフィールド) Other-information(その他情報) Reference-id(参照 ID)

表 2: Initialize response

必須のもの	Version(プロトコルのバージョン) Options(サポートする機能の宣言) Preferred-message-size(メッセージサイズ) Exceptional-record-size(例外メッセージサイズ) Result(接続開始の許諾もしくは拒否)
省略可能なものとして	Implementation-id(実装者の ID) Implementation-name(実装者の名前) Implementation-version(バージョン) User-information-field(利用者情報のフィールド) Other-information(その他情報) Reference-id(参照 ID)

2.4 Search 機能 (検索機能)

クライアントが検索要求には、検索質問、検索質問の種類、データベース名など (表 3) の検索要求 (Search Request) を出すと、サーバは検索結果 (Search Response) を返す。検索結果には、ヒットした件数などが含まれる (表 4)。

表 3: Search request

必須のもの	Query-type(検索質問の種類) Query(検索質問) Database-name(検索対象とするデータベースの名称) Result-set-name(検索結果集合につける名前) Replace-indicator(同じ名前の検索結果を置き換えるか否か) Small-set-upper-bound(Small 集合の最大レコード数) Large-set-lower-bound(Large 集合の最小レコード数) Medium-set-present-number(Medium 集合の present レコード数)
省略可能なものとして	Small-set-element-set-names(Small 結果集合を SearchResponse APDU に含める場合のエレメントセット) Medium-set-element-set-names(Medium 結果集合を SearchResponse APDU に含める場合のエレメントセット) Prefers-record-syntax(present されるレコードに適用するレコードシンタックス) Additional-search-information(付加検索情報) Other-information(その他情報) Reference-id(参照 ID)

2.4.1 検索式の構造

検索式の種類には、 type-0,type-1,type-2,type-100,type-101 が用意されており、一般的には type-1query が使われる。 type-1 は逆ポーランド記法 (RPN Query) で記述されたツリー構造の検索質問である。以下は検索式の構造を BNF で表している [6]。

```
RPN-Query ::= Argument
| Argument + Argument + Operator
         Argument    ::= Operand | RPN-Query
         operand     ::= AttributeList + Term
| ResultSetId | Restriction
         Restriction ::= ResultSetId + AttributeList
         operator    ::= AND | OR | AND-NOT | Prox

The notation above is used as follows:
```

```
::=       means "is defined as"
       |         means  "or"
       +         means  "followed by", and + has precedence over |
                         (i.e., + is evaluated before |).
```

検索語が書名であるとか、前方一致で、などを表すアクセスポイントは、 Attribute List(図4) と呼ばれる整数値の組合せの集合と、その整数値の組合せがどういう意味かを定義している Attribute Set ID で表される。例えば (1-1, 2-3) という Attribute List があったとすると、 Attrubute Set ID が書誌情報用の bib-1 ならば (Use - Personal Name, Relation - equal) というアクセスポイントを表すし、 ext-1 ならば (Use - UserId, Permission - ModifyPermissions) というアクセスポイントを表す。

- Attribute
  検索語の属性を表す。 Attribute List と、 Attribute Set Id から構成される。
- Attribute Set
  Attribute がどの属性に属しているのかを表す。 (書誌、全文など)
- Attribute set ID
  Attribute Set に Z39.50 が一意につけた ID で、書誌情報用に作られた bib-1 が通常使われる。
- Attribute Type
  整数値で表される。 Attribute set ID bib1 では Attribute Type は 表 5、表 6のように整数値が割り当てられている。
- Attribute Value
  整数値で表される。 Attribute set ID bib1 では Attribute Value は 表 5、表 6のように整数値が割り当てられている。

表 4: Search response

必須のもの	Result-count(ヒットした件数) Number-of-records-returned(返戻された結果の数) Next-result-set-position(次に返戻し始める場所) Search-status(検索状態)
省略可能なものとして	Response-Records(返戻されるレコード) Result-set-status(リザルトセットの状態) Present-status(返戻状態) Additional-search-information(付加検索情報) Other-information(その他情報) Reference-id(参照 ID)

- Attribute Element
  Attribute Type、 Attribute Value の組合せで表される。
  例えば、 Attribute set ID bib1 では以下のようになる。
  - 1-1 Use Personal name(個人名)
  - 1-4 Use Title(タイトル)
  - 4-1 Structure word(ワード)
  - 4-2 Structure prase(フレーズ)
  - 5-1 Truncation right Truncation (左方向のトランケーション)
- Attribute List
  Attribute Element の集合で表される。
- Result Set ID
  検索結果集合の ID

例をあげると、
「書名に Library という語 がつく本を探したい」という場合、
(1-4 , 4-2)"Library" ,bib1 となる。
1-4 は書名を、 4-2 は 検索語が語であることを表している。
1-4,4-2 は Attribute List、
1-4 や 4-2 は Attribute Element、 1-4 の 1 や 4-2 は の 4 は、 Attribute Type
1-4 の 4 や 4-2 は の 2 は、 Attribute Element を表している。
bib1 は Attribute set ID を表している。

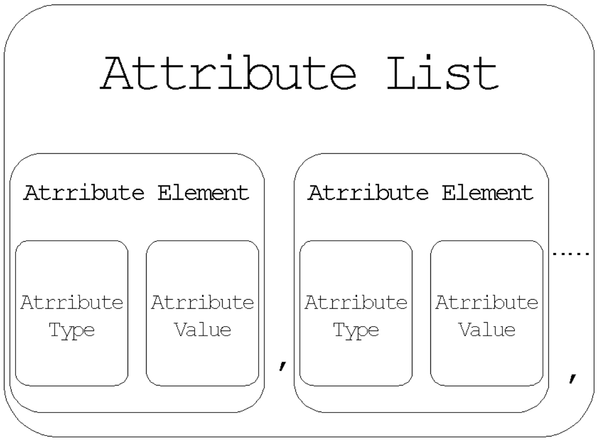

図 4: Attribute List

表 5: bib1 2-1

Attribute Type		Attribute Value	
use	1	Title	4
		ISBN	7
		ISSN	8
		Date	30
		Author	1003
		Any	1016
		Publisher	1018
Relation	2	less than	1
		less than or equal	2
		equal	3
		greater or equal	4
		greater than	5
		not equal	6
		phonetic	100
		stem	101
		relevance	102
		AlwaysMatches	103
Position	3	first in field	1
		first in subfield	2
		any position in field	3
Structure	4	phrase	1
		word	2
		key	3
		year	4
		date (normalized)	5
		word list	6
		date (un-normalized)	100
		name (normalized)	101
		name (un-normalized)	102
		structure	103
		urx	104
		free-form-text	105
		document-text	106
		local number	107
		string	108
		numeric string	109

表 6: bib1 2-2

Attribute Type		Attribute Value	
Truncation	5	right Truncation	1
		left truncation	2
		left and right	3
		do not truncate	100
		process # in search term	101
		regExpr-1	102
		regExpr-2	103
Completeness	6	incomplete subfield	1
		complete subfield	2
		complete field	3

2.5 Present 機能 (返戻機能)

検索した結果の書誌レコードを得る機能であり、 search and retrieve の retrieve にあたる。クライアントは検索した結果の書誌レコードを要求する返戻要求 (Present request) (表 7) を出す。サーバは要求されたレコード返戻結果 (Present response)(表 8) を返す。一般的には、検索するとヒットした件数だけではなく、書誌事項も同時に表示するが、Z39.50 では通常、検索クライアントが検索要求を出すと、検索サーバはヒットした件数などを返し、書誌レコードは返さない。検索クライアントが検索結果の書誌レコードを得るには、検索要求とはまた別に返戻要求を検索サーバに出さなければならない。検索結果の形式にはプレインテキストである SUTRS や、従来から使われている USMARC 形式がある。エレメントセットネームには簡略形式を表す B や、詳細形式を表す F があり、Z39.50 検索クライアント・検索ーサーバは最低でも、この 2 つのエレメントセットネームをサポートしなければならない。

表 7: Present request

必須のもの	Number-of-records-requested(present されるレコード数) Result-set-start-position(present される最初のレコード番号) Result-set-id(結果集合 ID)
省略可能なものとして	Additional-ranges(追加して present されるレコード番号) Element-set-names(エレメントセットネーム) Preferred-record-syntax(レコードシンタックス) Comp-spec(レコードコンポジション) Max-segment-count(最大セグメント) Max-segment-size(最大セグメントサイズ) Max-record-size(最大レコード長) Other-information(その他情報) Reference-id(参照 ID)

表 8: Present response

必須のもの	Number-of-records-returned(返戻したレコードの数) Next-result-set-position(次に返戻するレコード番号) Present-status(返戻の状態)
省略可能なものとして	Response-Records(返戻されるレコード) Other-information(その他情報) Reference-id(参照 ID)

3　Z39.50 クライアント

3.1　システムの構成

本システムは大きくプロトコルエンジンと GUI 部分の 2 つから構成される (図 5)。

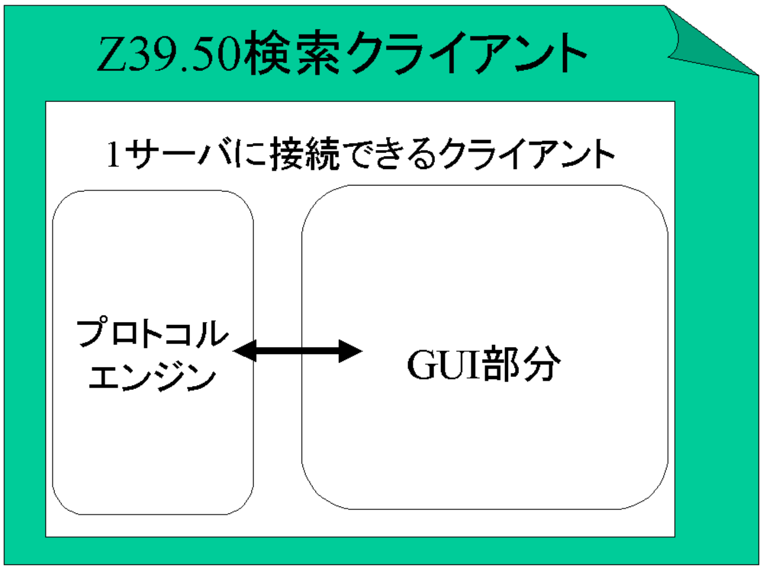

図 5: システムの構成図

プロトコルエンジンとはネットワークに近い部分で検索質問を Z39.50 に基づいた形式に変換したり、検索結果を Z39.50 に基づいた形式を解析し、本システムで加工しやすい形にする部分である。プロトコルエンジンには Index Data 社が開発した YAZ という Z39.50 に基づいた検索システムを作るためのツールキット [7] を利用して開発した。 YAZ は C 言語で書かれた Z39.50 を用いた検索システム (検索サーバ、検索クライアント) を作るためのプログラム群である。簡単なデモ用のプログラムも用意してあり、そのデモプログラムを改変するという形でプロトコルエンジンを開発した。実際に改変した部分は GUI がユーザーから受けとった接続要求、検索要求、返戻要求を受けとる部分と、サーバから受けとった接続結果、検索結果、返戻結果を GUI に返す部分である。

GUI を設計する方針は、次にする動作を色を変えて強調するなど「誰にでもマニュアルなしで使えるようにすること」とした。 GUI は Tcl/Tk[8] を使用して開発した。

3.2 システムの開発的な構成

本プログラムは以下のファイルから構成される (図 6)。

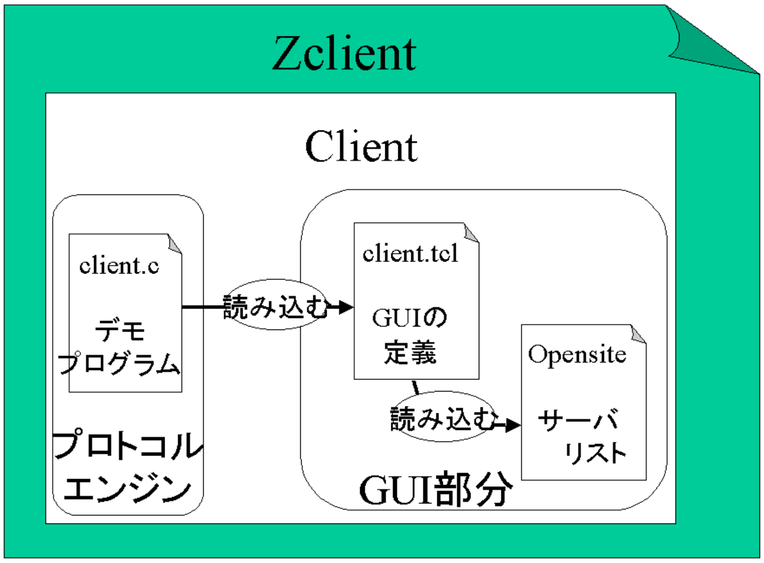

図 6: ソースファイルの関係

- client.c
  C 言語で書かれた YAZ のデモプログラムに、 client.tcl とつなげる部分を改変したファイル。
- client.tcl
  GUI 部分を Tcl/Tk 言語を使って定義したファイル
- opensite
  client.tcl に読み込まれる、接続できるサーバのリスト
- zclient
  デモプログラムでは 1 回の起動で 1 つの検索サーバにしかアクセスできなので、システムを終了させることなく、いろんなサーバに次々とアクセスできるようするためのシェルスクリプト
- client
  実行ファイル (client.c,client.tcl をコンパイルしたもの)

3.3 YAZ のデモプログラムの機能

用意してあるデモプログラム client は、一通りの機能をサポートしており、コマンドラインでの Z39.50 クライアントであるこのプログラムを動かすとすべての動作はコマンドを打つことによってできる。 client と入力するとこのデモクライアントは立ち上がり、 Z > というプロンプトを表示する。

```
d130@voyager(2)% client
Z>
```

次に、 open というコマンドを使って検索サーバに接続要求をだす。

- open tcp か osi: ホスト: ポート番号
  プロトコルは tcp か osi どちらかを選択する。一般的には tcp である。
  ホストは 検索サーバの IP アドレスを指定する。
  ポート番号には検索サーバが指定するポート番号を指定する。 Z39.50 では 210 番が推奨されている。

- 例： OCLC の FirstSearch(プロトコル tcp、 IP アドレス tikal.dev.oclc.org、ポート番号 210) に接続
  Z> open tcp:tikal.dev.oclc.org:210

以下は、ベル研究所の検索サーバに接続しているところである。接続が成功すると検索サーバの ID などが表示されプロンプトが返ってくる。

```
Z> open tcp:z3950.bell-labs.com:210
Connecting...Ok.
Sent initrequest.
Connection accepted by target.
ID      : AT&T-Test
Name    : Library Network Z39.50 Server
Version: 0.1
Z>
```

検索は find コマンドを用いて行なう。アクセスポイントを指定した検索や、履歴を利用した検索ができる。

- find コマンドは 検索式をポーランド記法で記述する。
  (RPN Query は逆ポーランド記法だが、デモプログラムの検索式の入力方式がポーランド記法)

- Attribute Element は “@attr Type=Value” と表す (例: 書名 @attr 1=4)。

- 検索結果集合番号 (Result Set ID) は “@set 検索結果集合番号” と表す。
  (例: 検索結果集合番号 2 @set 2)

- アンドやオアはそれぞれ @and, @or と表す
  (例:@and “library” “information”)。
- 複数にまたがる語は ”(ダブルコーテーション) でくくる。
  (例:“library and information science”)
- 例: World Wide Web が書名にあるものと検索結果集合番号 10 とのアンドがとりたい
  Z> @and @attr 1=4 “World Wide Web” @set 10

```
Z> find @attr 1=4 library
Sent searchRequest.
Search was a success.
Number of hits: 1570, setno 1
records returned: 0
Z> find @and @set 1 information
Sent searchRequest.
Search was a success.
Number of hits: 241, setno 2
records returned: 0
Z>
```

返戻 (検索結果のとりよせ) は show コマンドを使う。

- show とだけ打つと 直前の検索結果の最初の 1 レコードが返ってくる。
- show 開始レコード番号 + レコードの数
- 例 : 45 件ヒットした最前の検索の 5 番目のレコードから 10 個のレコード (5 番目 ~ 14 番目のレコード) を返戻
  Z> show 5+10

```
Z> show
Sent presentRequest (11+1).
Received presentResponse.
Records: 1
[books]Record type: USmarc
001   116200
035    $a  116200E
092    $a  025.3 $b  S49s
100 10 $a  Settel, B., ed.
245 10 $a  Subject description of books: a manual of procedures for
```

```
            augmenting subject descriptions in library catalogs.
260 0  $a  Sch. of Inform. Studies, Syracuse U. $c  1977.
300    $a  1v. (var. pagings)
440  0 $a  Syracuse University. School of Information Studies.
           Research Studies ; $v  no. 3.
690    $a  Library catalogs
690    $a  Subject headings
690    $a  Subject catalogs
Z>
```

応用的な機能である scan 機能は scan を使う。 scan 機能はインデックスワードなどのスキャンニング (ブラウジング) 機能である。

```
Z> scan library
SCAN: 20 entries, position=5
  libraries' (1)
  libraries--a (2)
  libraries/date (1)
  libraries/networking/data (1)
* library (2424)
  library administration (54)
  library administration division (1)
  library administrators (1)
  library and information research group  great britain (1)
  library and information technology association  us (3)
  library applications (23)
  library architecture (6)
  library association (4)
  library associations (1)
  library binding institute (1)
  library branch (1)
  library buildings (6)
  library catalogs (18)
  library catalogs and readers (1)
  library cooperation (7)
Z>
```

3.4 ウインドウ構成

3.4.1 接続ウインドウ

接続ウインドウを用いて検索サーバと接続する手続きを行なう (図 7)。

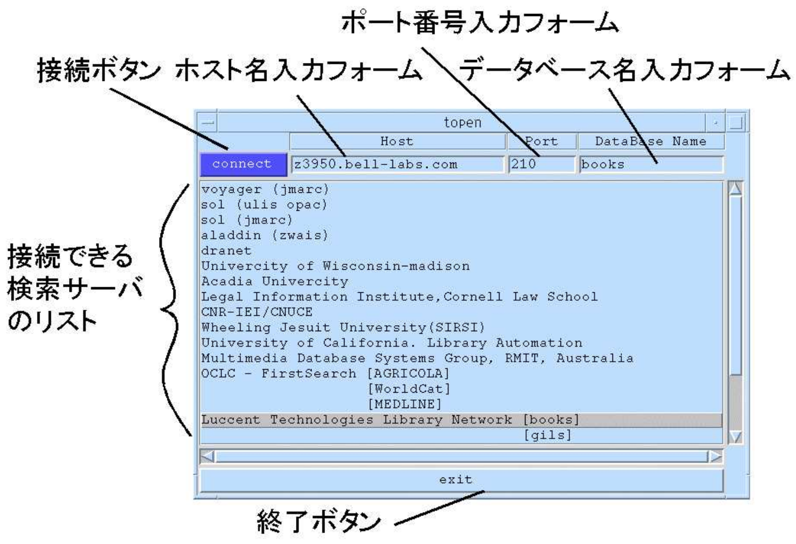

図 7: 接続ウインドウ

接続ウインドウは以下から構成される。

- 検索サーバ情報 (ホスト名、ポート番号、データベース名) の入力フォーム
  キーボードから検索サーバ情報を入力する。
- 接続できるサーバのリスト
  接続できるリストから一つ選択してをクリックすると、検索サーバ情報が入力フォームに入力される。
- 接続ボタン
  接続ボタンをクリックすると検索サーバに接続される。
- 終了ボタン
  終了ボタンをクリックすると本システムが終了する。

3.4.2　検索ウインドウ

検索ウインドウを用いて検索及び返戻を行なう (図 8)。

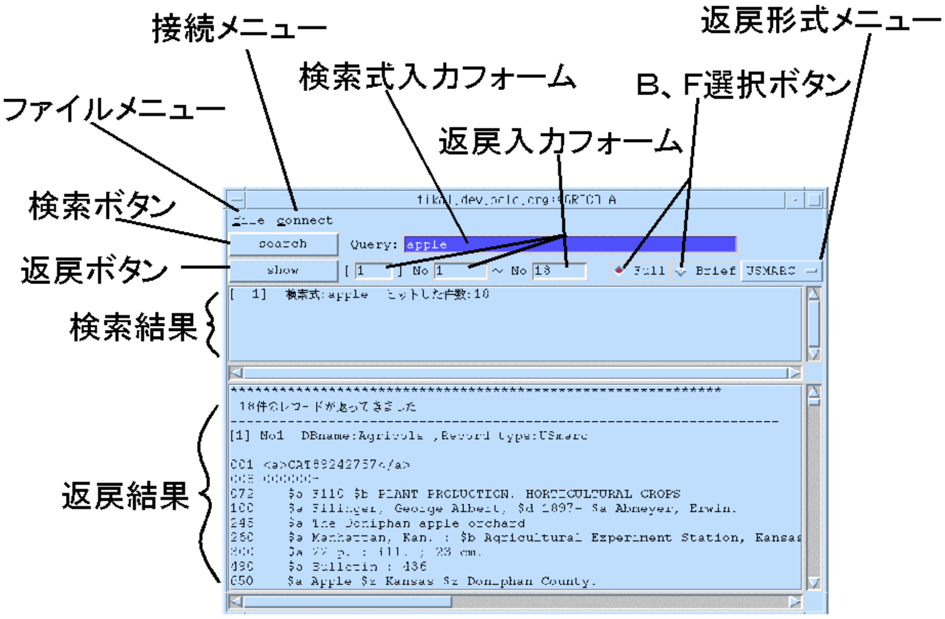

図 8: 検索ウインドウ

検索ウインドウは以下のように構成される。

- ファイルメニュー
  ファイルメニューをプルダウンし quit を選択すると本システムを終了する。

- 接続メニュー
  接続メニューをプルダウンし disconnect を選択すると現在接続している検索サーバと接続を切り、接続ウインドウが起動される。

- 検索ボタン
  検索ボタンをクリックすると検索を開始する。

- 検索式入力フォーム
  検索式入力フォームには、検索式をキーボードから入力できる。検索式は、 YAZ のデモプログラムと同じである。 Ctrl を押しながら\ を押すことで、日本語も入力できる。

- 返戻ボタン
  返戻ボタンをクリックすると返戻を開始する。返戻入力フォームに入力しなければ、最前の検索結果のすべてのレコードを返戻する。

- 返戻入力フォーム
  返戻入力フォームには返戻したいセットナンバー、レコードナンバーなどをキーボードから入力できる。

- B、F 選択ボタン
  B、F 選択ボタンは、返戻結果の簡略 (B)・詳細 (F) を選択できる。B を選択すると F が選択できなくなり、同様に F を選択すると B が選択できなくなっている。
  BF 選択ボタンのデフォルト値は接続ウインドウで、検索サーバを選択した時にあらかじめ設定されている。

- 返戻形式メニュー
  返戻形式メニューは、プルダウンして返戻形式を選択できる。返戻形式メニューには、USMARC,SUTRS などがある。
  返戻形式メニューのデフォルト値は接続ウインドウで、検索サーバを選択した時にあらかじめ設定されている。

- 検索結果表示覧
  検索結果表示覧には、検索式やヒットした件数が検索するごとに表示される。一番左の [ ] は、セットナンバーを表している。

- 返戻結果表示覧
  返戻結果表示覧には、検索サーバから返ってきたレコード数と返戻結果が表示される

3.5 システムの実行例

1. 本システムを立ち上げると、検索サーバに接続するための接続ウインドウ (図 9) が立ち上がる。

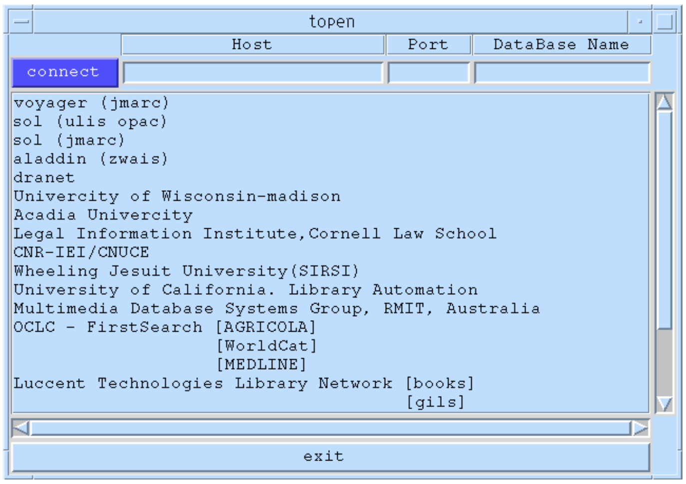

図 9: 接続ウインドウ：初期状態

2. OCLC FirstSearch をクリックすることで検索サーバの情報を入力 (図 10)connect ボタンをクリックして接続を開始する。

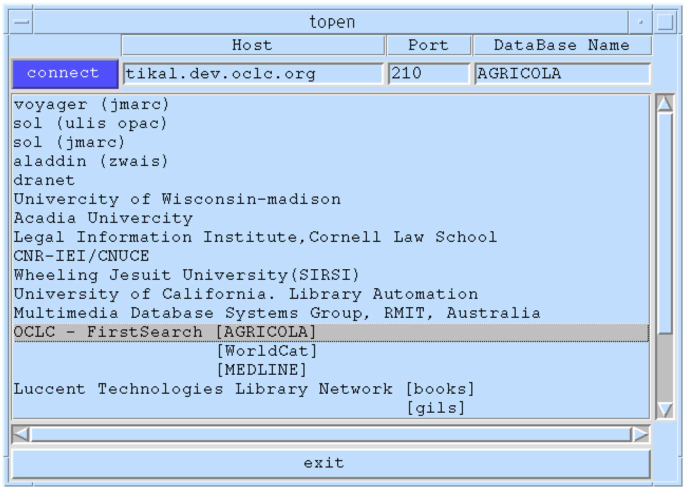

図 10: 接続ウインドウ：検索サーバ、 OCLC FirstSearch を入力

3. 接続が成功するとつぎに、タイトルバーに接続したホスト名である tikal.dev.oclc.org とデータベース名 AGTICOLA を表示した OCLC First Search を検索するための検索ウインドウが立ち上がる。検索ウインドウが立ち上がると検索式の入力フォームが青くなっており、つぎに入力する動作であることを強調する。検索式の入力フォームはクリックしなくてもすぐに入力できる状態にある。(図 11)。

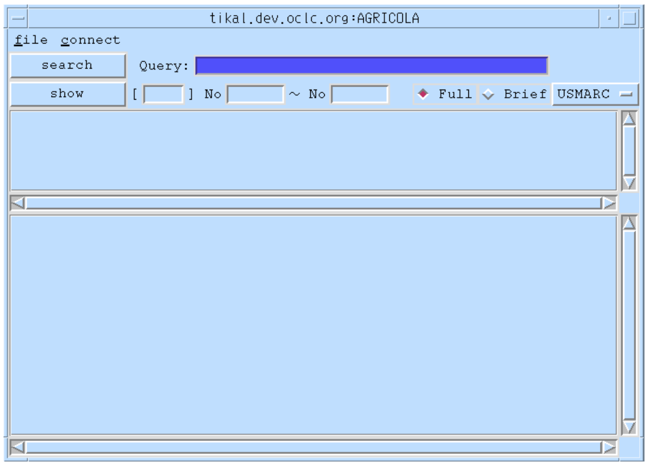

図 11: 検索ウインドウ： OCLC FirstSearch 初期状態

4. 検索式の入力フォームに検索式を入力し search ボタンをクリックすると検索を開始する。検索されると 検索結果表示蘭に検索履歴番号 1 、検索式 apple 、ヒットした件数 18 が表示される (図 12)。検索式 apple は、 apple という語が書名や著者名などどこかに含むものを検索という意味である。

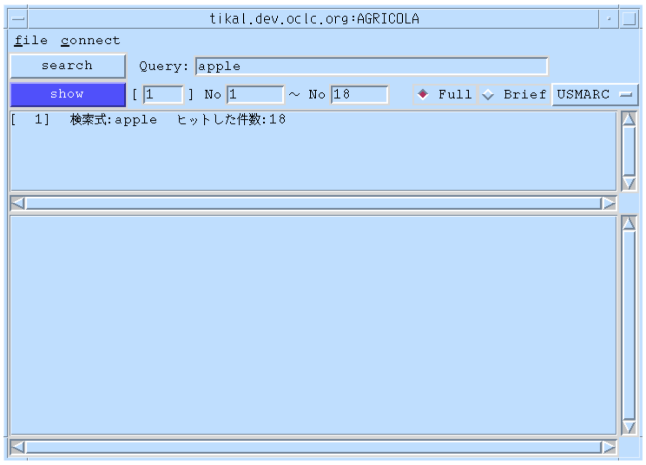

図 12: 検索ウインドウ：検索 1

5. show ボタンをクリックすると直前の検索結果のすべてを返戻する (図 13)。

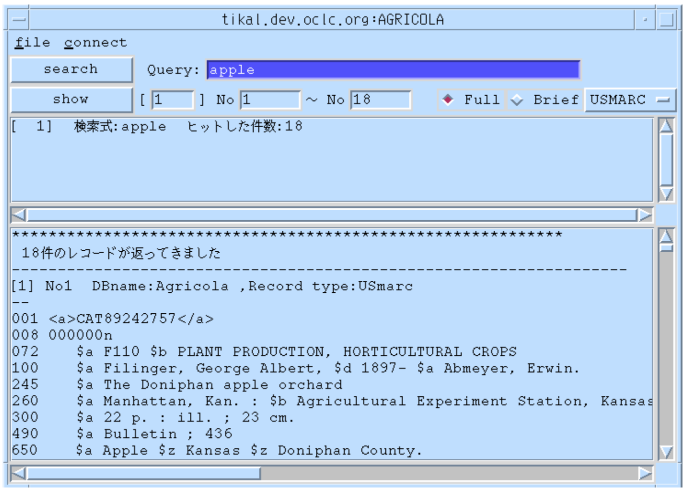

図 13: 検索ウインドウ：返戻 1

6. BF 選択ボタンの B をクリックして選択し、 show ボタンをクリックすると簡略形式が返戻される (図 14)。

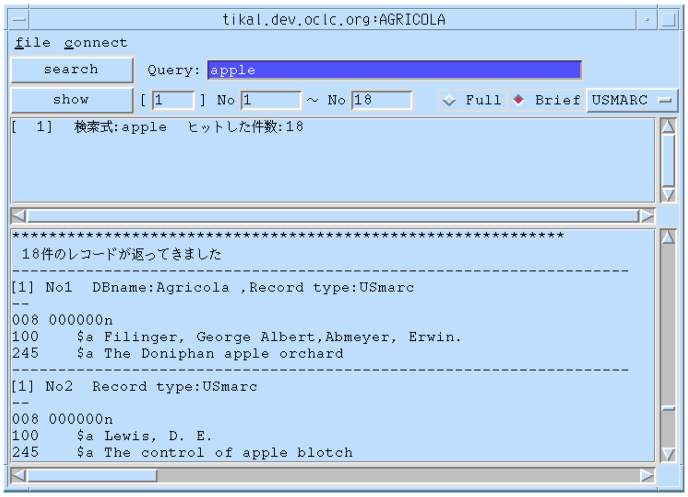

図 14: 検索ウインドウ：返戻 2 簡略表示

7. 返戻形式メニューをプルダウンしてSUTRSを選択し、 showボタンをクリックすると返戻形式SUTRSが返戻される(図15)。

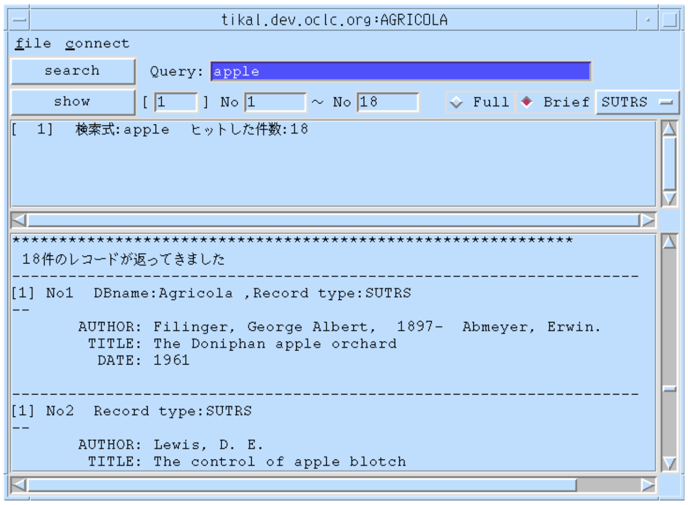

図 15: 検索ウインドウ：返戻 3 SUTRS

8. connect メニューをプルダウンして disconnect を選択すると、現在接続しているサーバ (OCLC FirstSearch) との接続を切り、他の検索サーバに接続するために、再び接続ウインドウが立ち上がる (図 9)。

9. 続けて、ベル研究所に接続を開始 (図 16)。

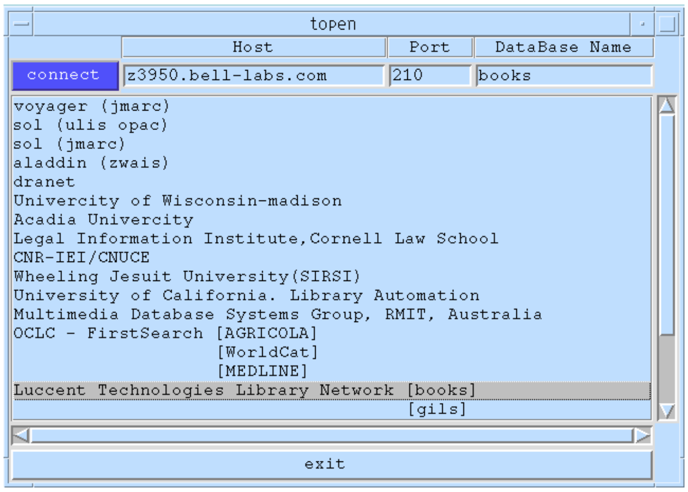

図 16: 接続ウインドウ：検索サーバ、ベル研究所を入力

10. 検索式 library で検索し、次に 検索式 internet で検索。この二つの検索結果集合のアンドをとる (図 17)。

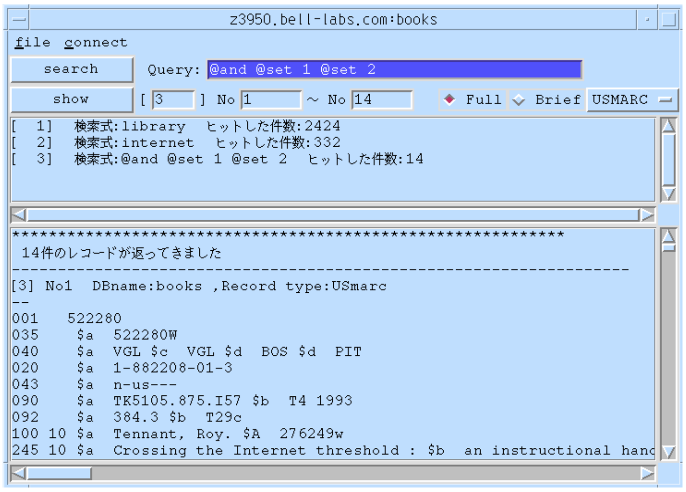

図 17: 検索ウインドウ：検索 履歴を利用した検索

11. 次に、真野が開発した JMARC の日本語検索サーバ [9] に接続する (図 18)。

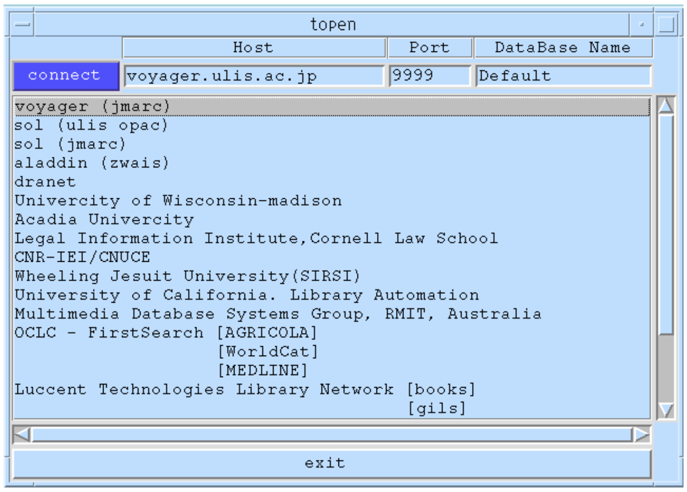

図 18: 接続ウインドウ：検索サーバ、 JMARC 検索サーバ

12. 検索式 図書館 で検索。 Ctrl+\ で日本語を入力開始し、変換は変換キーでする (図 19)(図 20)。

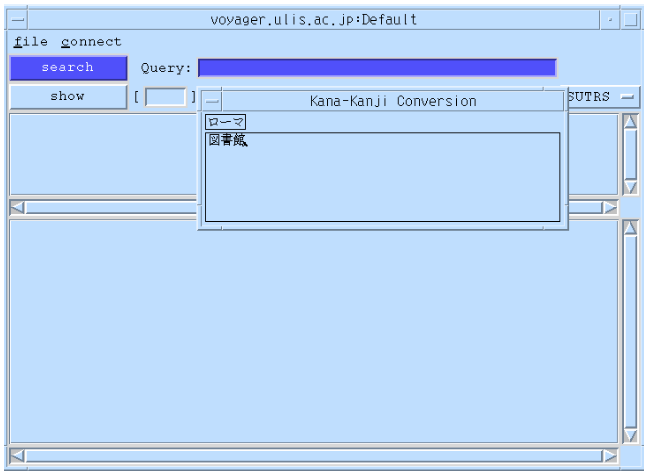

図 19: 接続ウインドウ：検索 日本語入力

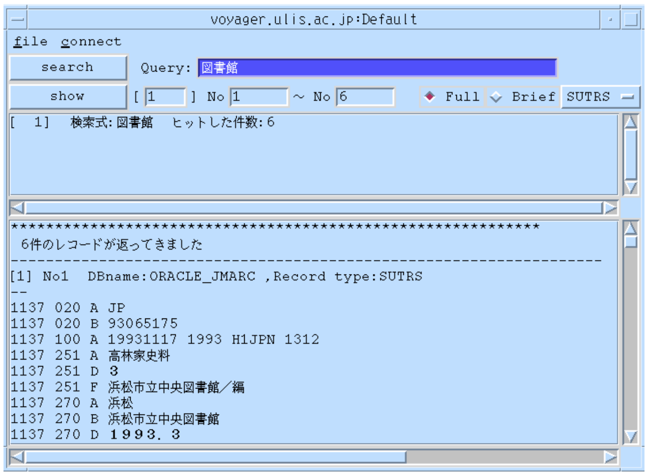

図 20: 接続ウインドウ：検索 JMARC 検索サーバ

13. file メニューをプルダウンして quit する。

4　考察

これまで本研究で開発した Z39.50 クライアントの構成について述べてきたが、つぎに Z39.50 クライアントの１つである willow と、 WWW 上のクライアントの１つであるアメリカ議会図書館の Z39.50 の gateway クライアントと本研究で開発したクライアントを比較検討する。本システムでは、誰にでもマニュアルなしですぐに使うことができることをインタフェースを設計する時の方針とした。そのため細かな機能は設定せず、最低限必要な機能だけを実装したり、次にする動作をわかりやすくするために、次にクリックするボタンや、次に入力する入力フォームの色を変えるなどした。誰にでもマニュアルなしで、使うことができるインタフェースがあるのは、アメリカ議会図書館の Z39.50 gateway である (図 21)(図 22) (図 23)。

WWW 上のシステムの場合、ユーザーは良く知った親しみやすいインタフェースなので、次に行なう動作を指定しやすい。また、エラーが出た時も、 WWW ブラウザに慣れたものにとってわかりやすい。また、アメリカ議会図書館の Z39.50 gateway は、簡単な機能しか持っていないので、動作が単純でわかりやすい。

それとは対象的に、 willow は大変多機能である (図 24)(図 25) (図 26)。

そのため、ボタンやメニューがたくさんあり、次に何をすれば良いのかが大変わかりいにくい。しかし、機能がたくさんあるので、設定を変えたり、検索結果を保存したりなど、様々に応用できる。アメリカ議会図書館の gateway や本システムは、機能が少なく、始めてのユーザには良いかも知れないがもっと複雑な検索をしたいユーザにとっては物足りないものになっている。

現在、 Z39.50 Client Survey [10] に登録されている Z39.50 クライアントは 13 ありるが日本語が使えるものは調査した限りでは存在しない。その他 WWW ブラウザをクライアントに見立てて、 WWW 上に Z39.50 とのゲイトウエイという形で作られているサーバもある。本システムを使うことにより、 Z39.50 を用いた検索サーバにアクセスすることが可能となったが、 Z39.50 は多言語化についての規定はされておらず、話し合われている段階である。 Z39.50 を用いた日本語化検索システムを開発する時に問題となるのは、文字コード、ワード、フレーズ、返戻のレコード形式である。文字コードは日本に特有なものだけでも何種類もある。よって検索要求を出す時、検索結果を受けとった時など、文字コードが異なった場合は文字化けを起こして読めない問題がある。日本語はスペースで区切られた言語ではないため、語に区切ることができない言語である。しかし、 Z39.50 はアメリカで規定されたという経緯があるため、検索属性を表す Attribute Set には、 word や phrase といった指定をするため、このようなスペースで語が区切られる言語特有のものとどのように、折り合いをつけていくか、という問題がある。 Z39.50 には、返戻レコードの形式として、 USMARC,UNIMARC などが定められているが、 JAPAN/MARC は定められていない。よって、現在は SUTRS（プレインテキスト）で送るしかない状態である。世界中の Z39.50 検索システムを利用するためには、この多言語問題の解決が待たれる。

5　終りに

以上述べてきたように本研究では、 Z39.50 プロトコルを用いた検索クライアントを開発し、世界各地の検索サーバや、日本語の Z39.50 検索サーバにアクセスできるようになった。

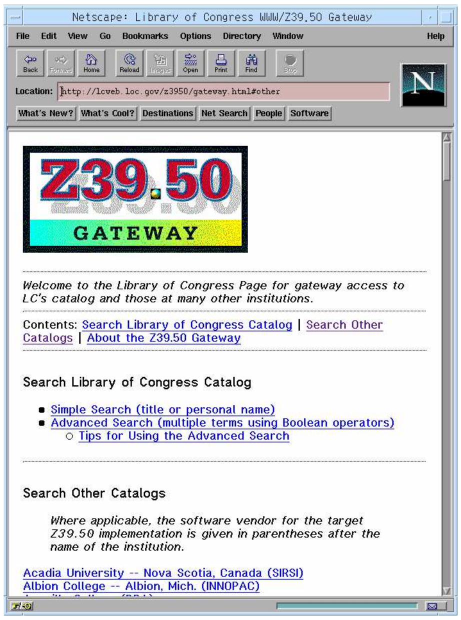

図 21: アメリカ議会図書館 Z39.50 getway：接続

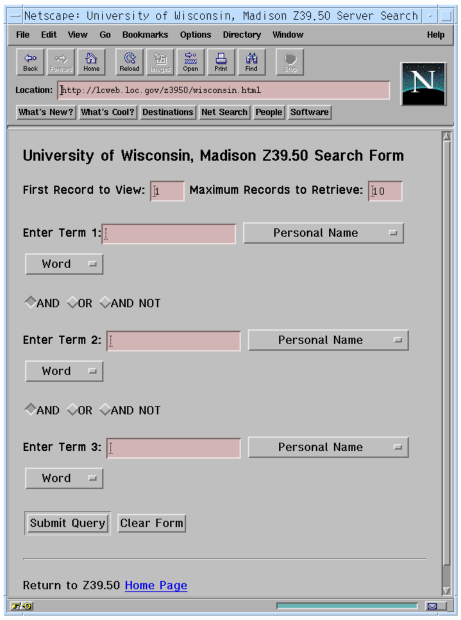

図 22: アメリカ議会図書館 Z39.50 getway：検索

図 23: アメリカ議会図書館 Z39.50 getway：返戻

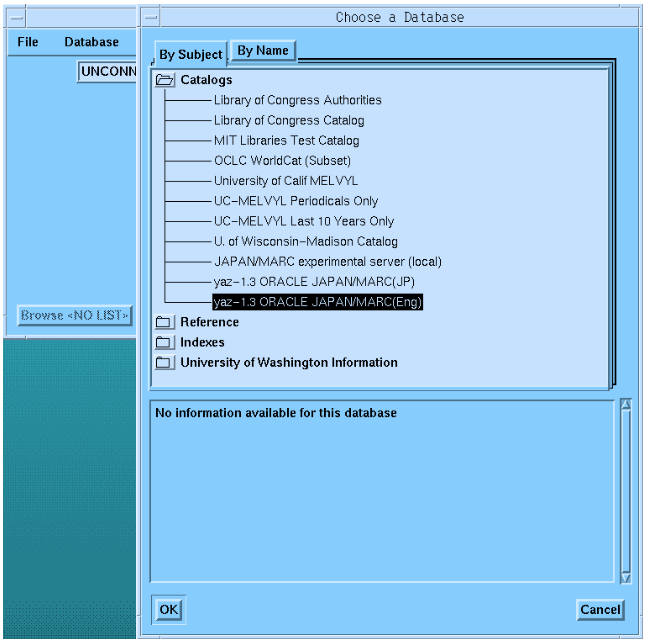

図 24: willow：接続

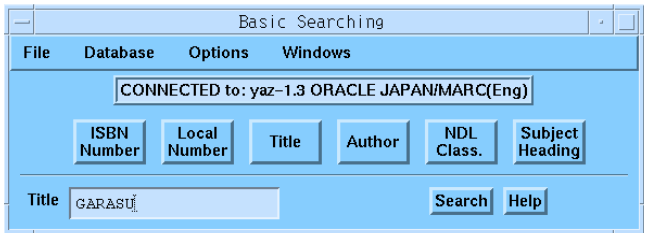

図 25: willow：検索

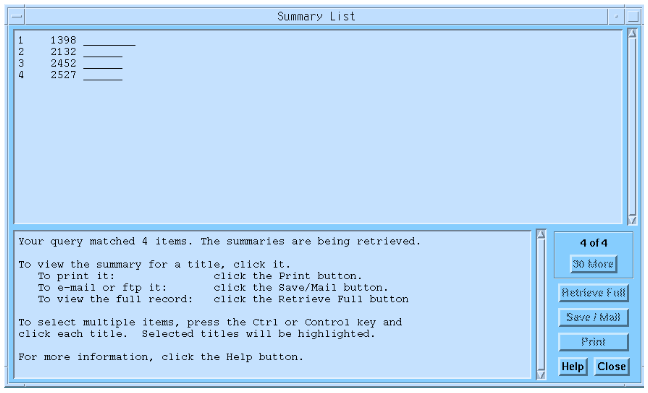

図 26: willow：返戻

謝辞

本研究全般にわたり、御指導、御教示頂きました 石塚 英弘先生、宇陀 則彦先生に感謝致します。また、石塚ゼミの皆さん、本研究を進めていくにあたりお世話になりましたすべての方々に感謝致します。

参考文献

[1] 上田　修一.　Z39.50 の可能性と問題点.　三田図書館・情報学会研究大会予稿集 (1996.11).

[2] 上田 修一. Z39.50 とは何か. 電子ライブラリー. Vol.5, No.3/4, p.62-65 (1997.5).

[3] 桂 啓壮 . 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 (1995.4).

[4] 安斎 宏幸. インターネット環境における日本語書誌情報　システムの構築. つくば, 図書館情報大学,1997. 修士論文

[5] プロトコルハンドブック編集委員会. 新プロトコルハンドブック. 東京. 株式会社アトソン.1994.241-251p.

[6] ANSI/NISO Z39.50-1995.Information Retrieval (Z39.50) : Application Service Definition and Protocol Specification
<URL:http://lcweb.loc.gov/z3950/agency/>

[7] Index Data.Index Data homepage.
<URL:http://www.indexdata.dk/>

[8] 久野 靖. 入門tcl/tk. 東京, 株式会社アスキー,1997,242p.

[9] 真野 泰久.Z39.50 プロトコルを用いた検索サーバの開発　. つくば, 図書館情報大学,1998. 学士論文

[10] Z39.50 Client Survey
<URL:http://www.dstc.edu.au/RDU/reports/zreviews/z3950-client-survey.html>