

## 独立したリンク情報と検索対象 Metadata を備える Hub-Metadata の提案

Warwick Framework に代わる Metadata アーキテクチャ

大矢一志

千葉大学自然科学研究科

ohya@cogsci.l.chiba-u.ac.jp

土屋 俊

千葉大学総合情報処理センター

tutiya@kenon.ipc.chiba-u.ac.jp

### 概要

本稿では、Metadata に ID の機能をもたせ、意味による検索を可能にするほか、さらにリンク情報を独立させることで、管理対象となる情報の特性に依存しない、共通した書式で Metadata を記述できる、Hub-Metadata を提案する。

## Hub-Metadata

Core-Metadata compounded with link information of a document

OHYA Kazushi

TUTIYA Syun

Chiba University

### Abstract

In this paper we suggest a new type of metadata description format called Hub-Metadata. The Hub-Metadata not only works like a ID, so we can use it as a keyword set for retrieval, but works as a hub-document since information on link in a content object is included in it, so we can manage all kinds of data types in a same manner on the Hub-Metadata.

## 目次

- 1 はじめに
- 2 Metadata
- 3 先行事例
- 4 Warwick Framework
- 5 Hub-Metadata
- 6 さいごに

## 1 はじめに

本稿では、Metadata に ID の機能をもたせ、意味による検索を可能にするほか、さらにリンク情報を独立させることで、管理対象となる情報の特性に依存しない、共通した書式で Metadata を記述できる、Hub-Metadata を提案する。

## 2 Metadata

### 2.1 Metadata に必要な特性

Metadata には、「電子化された情報のより広い範囲を十分に記述でき、かつ簡潔な表現であること<sup>1</sup>」が求められている。

その特性としては、以下のものが要求される。

**柔軟性** ネットワーク上のあらゆるデータタイプに対する記述力を持つこと。また、あらゆるデータタイプを相互に関連付けられること。

**継承性** 電子化されていない情報に対しても、カタログ情報を記述することが出来ること。

**拡張性** 将来に現れるであろう情報タイプや、新たな記述項目の発生にも対応できること。

**簡潔性** 人が確実に操作できるもので、作業の誤りを導かないのに十分な簡潔さを持っていること。

**連携能力** URN や URC と連携できること。

さらに、次の特性も必要であろう。

**唯一性** 地球規模で唯一の ID として働くもの。

**継続性** 将来に渡っても、ID 変更の必要がないこと。

以上の特性は、URN の要求事項<sup>2</sup>と似たものである。特に後の 2 項目は URN の第一の特徴である、いわば

<sup>1</sup>Metadata Workshop in Dublin

<sup>2</sup>Requirements for Uniform Resource Names, RFC1737

ID 性を示す条件であるが、Metadata においても同じように重要になる。なぜなら、Metadata はネットワーク上の検索対象として捉えるのであれば、ID として URN と同じ機能を要求することは正当だからである。

### 2.2 求められる機能

先述の Metadata に必要とされる特性は、以下にある機能によって備えることが可能になる。

#### Core-Metadata と Complete-Metadata

検索対象となる Core-Metadata の各項目は、全ての検索対象データにおいて共通であることが望ましく、従って Core-Metadata が頻繁に変更されることとは、極力避けなければならない。しかし、新しい情報形態に対応していくためには、Metadata の記述項目の修正は必須の事態である。この対応策として、Metadata を、検索対象となる Core-Metadata と、詳細な情報を記述する Complete-Metadata の 2 つの部分に分けることで、例えば、新技術の開発や新たなデータタイプの出現で、より詳細な Metadata の記述が必要になった場合には、Complete-Metadata の内容のみを変更していくという方法がある。これにより、柔軟性と拡張性が獲得される。

また、この Complete-Metadata の方は、独自の記述書式が認められる。これにより、現在ある様々なローカルの書式で記述されている膨大な資産を生かせることになる。また、変更が必要な場合でも、その内容に修正を加えるような書式の変更ではなく、DTD の変更で済むことが多い。これにより、継承性が獲得される。

#### Markup による記述

Core-Metadata も Complete-Metadata も、その内容は共に markup されている必要がある。markup の記述書式として、Complete-Metadata は、できれば SGML を利用するのが望ましい。なぜなら、DTD を利用することで異なる書式間での可換性が保証され、また、各項目への参照の際に、その要素が自由記述中のある部分というだけではなく、より効率的な参照が可能になるからである。これにより、拡張性が獲得される。

#### リンク情報の独立

Metadata が記述する対象は様々な媒体上に存在する。例えば、その情報は書き不可の属性を持つかもしれないし、また画像データのように、リンクのような付

加情報を直接データに書き込めないようなものもある。また、ネットワーク上では接続に制限があることもある。このような事態に対しては、情報本体からリンク情報を独立させることで、リンク情報とその制約情報を、リンクされる情報の属性に関わらず、共通した書式で記述することが可能になる。さらに、リンクという機能の生かすことで、例えば、情報内容の各項目と Metadata の項目との関連性や、情報内容同士の関連性、Metadata の項目間の関連をリンクで示すことが可能になる。これにより、拡張性と、継承性が獲得される。

このように、リンク情報が独立してあるものを、Hub-document と呼ぶ。Hub-document とは、既存の情報を、Hypertext に変えるための手段になる。

### URC の機能

URC の機能としては、URN から URL を返すことが提案されているが、まだ論議中のものであり、具体的な仕様は決められていない。しかし、URN、URL、URC の論議は、それぞれが名前、所在、属性によって、ある情報の存在を特定しようというものであり、これはネットワークで使われる ID の種類をより豊かにする試みである。従って、URC の論議は、単に URN から URL を引き当てるといった機能だけではなく、属性情報 (Characteristic) が ID としての機能をいかに持つかということになる。この属性情報とはすなはち Metadata のことであり、つまり、Core-Metadata が外部からの検索対象となる働きをもつことと同じである。これにより、連携能力と ID 性が獲得される。

## 3 先行事例

### 3.1 記述項目

#### 3.1.1 Full Description

従来のカタログ情報の記述力をそのままに、ネットワーク上の情報に対応した、カタログデータの交換形式には、MARC、TEI、EAD 等がある。

#### MARC

現在、図書館にある電子化されたカタログデータの殆どが何らかの MARC 形式で記録されている。この資産を継承していくことは重要である。

元々 MARC 自体は情報の交換形式全般を規定したものであるが、内容レベルでの交換形式として、SGML

による DTD が公開されている。ネットワーク上の情報には Field856 を用意することで対応している。

項目は多岐に渡りその記述能力は高いが、サブフィールドの記入項目が指定されているなど、記述方法が厳密に規定されている。従って、拡張が難しく、データ作成には専門家が必要になってくるという問題がある。

#### TEI header

TEI header とは、記述書式として SGML を用いて記述内容に関する提案を行った TEI プロジェクトによる、カタログデータに関する記述項目である。TEI は、文化資産の扱いを、学術的・体系的に検討したもので、その内容の充実と信頼性は他になく、Metadata の記述を提案する際には、基礎とすべき成果である。

難点をいえば、その目標とする要件は高度であり普遍的であるが、SGML の記述能力の限界が結果として記述法を難解なものにしてしまっている。

#### EAD

EAD とは、現在では LC を拠点に開発されている、Metadata の記述書式である。検索補助としてのカタログデータを、SGML ベースで記述しようとする試みで、MARC、TEI header の成果を反映したものである。現在は最終評価版が公開されている段階である。SGML の次期記述方式と見なされている HyTime で採用されたリンクに関する記述書式の基本部分を視野に入れた、柔軟な関係情報の記述を目指している。ネットワークでの利用法を考慮しながら、現在あるリソースで実現可能な記述書式を整備するという、意欲的かつ現実的な対応をしているプロジェクトである。

#### 3.1.2 Core Description

#### ISBN

新刊書籍に割り当てられる世界的な ID<sup>3</sup>である。販売形態を単位となるように作られたものなので、対象は限定されており、また Metadata としてみると、情報量は少なくかつ信頼性に欠けるものである。

#### Dublin Core

ネットワーク上で検索される際に、必要と思われる必要最小限の項目をまとめたものである。既存の Catalogue data の記述形式との共存可能な要素が利用されている。現時点での項目数は 15 項目である。Metadata を ID として利用する際、この 15 項目で唯

<sup>3</sup>ISO 2108:1971

一性が保たれるかの検証は必要であるけれども、実務者による現実的な対応を考慮した上で限定された検索項目の提案は重要であり、今後の共通理解を進める上の基礎となる提案である。

### 3.2 運用様式

PICT, PURL, GILS, そして NDLP での研究等多数あり、Metadata 研究は主にこの分野の関連領域として広く行われている。その中でも Dublin Core の運用様式として提案された Warwick Framework は、Metadata の書式から運用様式を検討するというアプローチを探っており注目される。

#### Warwick Framework

Warwick Framework とは、Dublin Core の提案に統いて翌年開かれた OCLC/NCSA 2nd Metadata Workshop の成果で、Metadata の運用に関するアーキテクチャとして提案されたものである。複数の Metadata 書式を container にまとめるというアイディアで、検索対象としての Dublin Core と Full Description としての MARC を共に備えることが出来るというものである。しかし、その記述書式等を厳密に規定したものではなく、またそれぞれの Metadata 間の関係をどのように実現するかなど、これから検討されるべき課題が多い。

実装例として、HTML2.0、MINE、SGML を利用したものが挙げられており、それぞれで開発目標の一つである簡易性をよく実現している。

今後の動向としては、簡易性を保ちながらどのように実装案がまとめられるのかが注目される。

本稿ではこの Warwick Framework の開発姿勢を尊重しながら、その拡張案を提案する。

## 4 Warwick Framework

Dublin Core をどのように利用していくのか、次の点を目標にして、その仕様が提案された。

1. ネットワーク対応であること
2. 様々な Metadata の記述書式を同時に扱えること
3. ユーザーやシステムに依存しないこと
4. Metadata と Data を、同じように扱えること
5. Metadata を共有できること

3番目のユーザに依存しないという姿勢が、Warwick Framework の論議を一貫して現実的対応に配慮させたものとしている。4番目の Metadata と Data は共に同じく扱われるべきであるという態度は、今までの Metadata の論議にはなかった目標であり、秀逸である。これは、単に Metadata も元の Data と同じように重要であるという話ではなく、ある情報について記述したもの、例えば作品の解説文なども、Metadata と位置づけられるということである。この考えに従えば、Metadata とは、関連する情報間のつながりを記録するためのものと捉えることができ、すなわち、あらゆる情報は Metadata を介して管理されるということになる。これは、情報管理システムを考える際の重要な視点となりうる。

#### 基本仕様

Metadata の集合体である container と、その構成要素である各 Metadata を収めた package からなる。各 package は何らかの URI を通じて情報 (a content object) と関連付けられる。

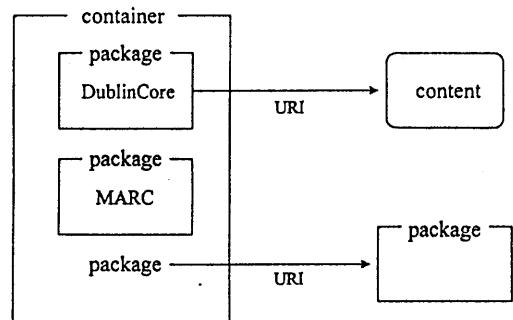


図 1:

Warwick Framework で残された課題は、以下のものが提示されている。

- Metadata 項目間の関連付け
- 各 Metadata の記述書式をどのようにするか (package level)
- パッケージ間の関連付けをどのように記述するか (container level)
- 情報本体 (content object) のやりとりの際に、container 全体を同時にやり取りすることによるトラフィックの増加

項目間の関連付けには2つの問題がある。ひとつは記述語彙間の調整で、いわゆるシソーラスの問題と関連するものである。もうひとつは、目標とする情報へのアクセス情報をどのように記述するかである。前者の問題は、Metadataの仕様の論議から外れる。後者の問題は、リンク使った書式を用意することで、対応可能であるが、Warwick Frameworkではこのリンクの部分を、簡単にURIの問題としてしまい、解決を避けている。これは、package levelでの記述書式を確定しないでいることに因る。

トラフィックの増加という問題は、content objectと同時に、Metadata全てが情報の受渡しの対象となるという前提から生じている。ここでは、package単位ではなく、container単位でのデータ管理が行われることが前提とされている。しかし、全てのMetadataが常に同時に管理される必要はない。例えば、Dublin Coreのような特定のMetadata以外は、要求が発生した時点で参照されれば、トラフィックの問題は深刻なものではなくなる。すなわち、package単位での運用をcontainerが管理できれば、この問題は解決する。

#### 評価

先述の残された課題の所で問題とした、記述書式を特定することへの懸念が、問題解決を妨げている。例えば、「containerとcontent objectsとの結び付きは、実装の仕方で異なる」という立場を探り、具体的には「URIに任せる」としているが、これは事実上の解決の棚上げであり、先に挙げた項目間の問題と同様である。様々な記述単位が互いに関連づけられることによって得られる柔軟性は、重要である。そして、項目間の関連付けを、より自由なものにするための手法として有用なのは、リンクによる管理である。リンクは、意味構造を示し得る機能を持つ、ネットワークの特性を生かす上で重要な手法である。そして、リンクに関する規定は、実装仕様の問題ではなく、記述書式のレベルで行われるものである。従って、Metadataの枠組を作ろうとするのであれば、リンクの記述書式を規定することは避けられない。ここに、Warwick Frameworkで採られている、記述書式を限定しないという方針の限界がある。リンクの記述の他、例えば、各項目の記述形式の独立性と可換性を保つことを今後の課題としているが、これはとともにSGMLのDTDで保証される等、記述書式を規定することによる問題解決は多い<sup>4</sup>。

<sup>4</sup>確かに、Warwick Frameworkで懸念される「記述書式を規定すれば、目標が達成できる」という命題は、システム依存を高める

## 5 Hub-Metadata

以上の論議を踏まえ、新たなMetadataの記述書式とその実装様式としてHub-Metadataをここに提案する。Hub-Metadataとは、Warwick Frameworkに、IDとしての役割と、リンク情報を担えるようにしたものである。Hub-Metadataは、urc、coreMetadata、linkPackから構成されている。urcにはHub-Metadataを起点として関連する情報単位の所在情報が記述される。coreMetadataには、Dublin Coreの各項目が記述され、内容による情報検索の際に第一に参照される項目になる。linkPackは、元情報に張られたリンクに関する情報がまとめられている。

#### urc

IDとしての機能を担う項目であり、urn、metadata、urlで構成されるリンク名である。すなわち、urn自体は構成要素を持つものではなく、ある情報のIDとなりうる、名前、特性、所在に関する情報の所在情報を、単に結びつけているリンクの名前になっている<sup>5</sup>。urnは下位要素として、localNameとurNameを持つ。localNameは情報制作者によって普段使用されているもので、システム依存や日常の変更が許されるものである。urNameは、今後決められるURN規則に従って名付けられたものである。metadataは、Complete-Metadataがある全ての所在を、リスト形式で記述する項目であり、Warwick Frameworkのpackageに当たるものである<sup>6</sup>。urlは、リスト形式で元情報の所在を示す項目である。Warwick Frameworkでは、

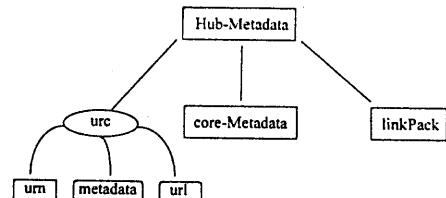


図 2:

だけでも他の目標が達成できるという選択であり、これを慎重に拒むという態度には共感を持つけれども、記述書式の規定はシステム依存にはならない。独立性は、記述書式を規定することによる特定性に余る程、十分に保たれている。

<sup>5</sup>従って、むしろURIと名付けた方が適切であるかもしれない。

<sup>6</sup>packageに含まれるMetadataと所在情報を一つの項目で扱えることから、より直截的なmetadataという名前を採ることにした。

元情報の所在は、各 Metadata の記述書式にあるそれぞれ項目毎に収録されることになるか、またはそれらを変数<sup>7</sup>として別に立てることになるが、それについては実装様式毎の形式に任されている。Hub-Metadata では、Metadata にある元情報の所在を示す全ての項目は、この url を指示参照することになる。

### coreMetadata

検索対象となる Metadata を記述する項目であり、ここでは Dublin Core の 15 項目が下位要素となる。Dublin Core の記述規則が固まっていないので、DTD は以下のようになる。

```
<!ELEMENT dublinCore - -
  (title | creator | subject | description
  | publisher | contributor | date | type
  | format | identifier | source | language
  | relation | coverage | rights)* >
<!ATTLIST dublinCore version CDATA *IMPLIED>
```

### linkPack

content object にあるリンクに関する情報の全てがまとめられる。その際、バーザは、object content の DTD からリンクタイプを解釈し、その情報を linkPack でのリンク形式へと変換する。以下にある DTD では、HyTime の構造形式を利用している。

### Hub-Metadata DTD

Hub-Metadata の DTD は以下のようになる。

```
<!-- Hub-Metadata.DTD except dublinCore element -->
<!-- and localDef entity -->
<!NOTATION SGML PUBLIC
"ISO 8879:1986//NOTATION Standard Generalized
 Markup Language//EN">
<!NOTATION HyTime PUBLIC
"+//ISO/IEC 10744:1992//NOTATION Hypermedia/
 Time-based Structuring Language//EN" >
<!ENTITY % a.global
  ' type CDATA *IMPLIED
  id ID *IMPLIED"uncontrolled"'>
<!ELEMENT
  hubMetadata - - urc, coreMetadata, linkPack>
<!ATTLIST
  hubMetadata
    id ID *FIXED hub>
<!ELEMENT urc - 0 EMPTY>
<!ATTLIST urc
  HyTime NAME ilink
  id ID *IMPLIED
  anchrole NAMES *FIXED (urn, metadata, url)
  linkends IDREFS *FIXE (lurn, lmetadata, lurl)>
<!ELEMENT urn - - localName, urName>
<!ATTLIST urn id ID *FIXED lurn>
<!ELEMENT localName - 0 (*PCDATA)>
<!ATTLIST localName % a.global;>
<!ELEMENT urName - 0 (*PCDATA)>
<!ATTLIST urName % a.global:>
```

<sup>7</sup>SGML では ENTITY 言語を行うことになる。

```
<!ELEMENT metadata - 0 nmlist*>
<!ATTLIST metadata
  HyTime NAME nameloc
  id ID *FIXED lmetadata>
<!ELEMENT url - 0 nmlist*>
<!ATTLIST url
  HyTime NAME nmlist
  id ID *FIXED lurL>
<!ELEMENT nmlist - - (*PCDATA)>
<!ATTLIST nmlist
  HyTime NAME nmlist
  type NAME *REQUIRED CDATA>
<!ELEMENT coreMetadata - - dublinCore>
<!ATTLIST coreMetadata %a.global;>
<!ELEMENT linkpack - - link*>
<!ATTLIST linkpack
  linkpack
  linkpack %a.global;>
<!ELEMENT link - 0 EMPTY>
<!ATTLIST link
  HyTime NAME ilink
  id ID *IMPLIED
  linkends IDREFS
  anchrole NAMES *FIXED (target1, target2)
  source NAMES *REQUIRED>
```

urc は HyTime の構造形式タイプ ilink になる。ilink は、従来のリンクタイプのように方向性や振る舞いを既定することなくリンク付けを行うので、urn、metadata、url 間での方向性に関する制約はない。metadata、url は複数の所在情報を要素内容として記述できる。またその所在のタイプはここでは規定していない。

### 実際例

上記の仕様によって、先に挙げた求められる機能と、Metadata の特性の殆どが実現される。一例として図 3 のような場合を想定した。

ある情報とその Metadata は、Hub-Metadata の linkPack 中で結びつけられる。例えば、doc1 にあるパラグラフで、〈p id=p1〉と markup されているところの Metadata は、Hub-Metadata にある id=l1 の要素により、その M 所在は Metadata.doc1 の〈meta-p id=m-p1〉と markup されている部分と関連付けられている。

同じように、ある情報が他の情報へ関連付ける際にも linkPack を通して行われ、例えば、doc2 中の要素〈p id=p2〉は、linkPack 中の要素〈link id=l3〉により、doc1 の要素〈fig id=f1〉と関連づけられている。

また、ある content object に直接埋め込まれている静止画像データのある領域と関連づけることも可能であり、その際も linkPack を通して行われる<sup>8</sup>。

content object 中のリンク情報は linkPack にまと

<sup>8</sup>例中にある ilink は multilink タイプになる。

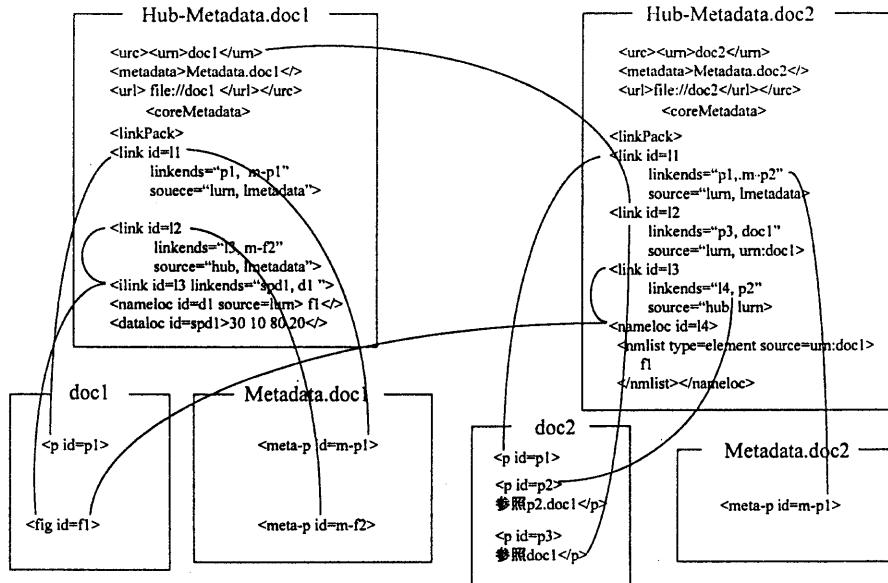


図 3:

められるが、元のリンク情報は消されるのか、また linkPack でのリンク法については、各アプリケーションの裁量で決められる。リンク部での情報の継続性は必要でなく、むしろ用意に変更されることが重要であり、従って、urc 部の継続性と巧く連関しながら記述されることが望ましい。

このように、リンク部を Hub-Metadata にまとめるなど、Hub-Metadata のデータ量が多くなり、また記述の際の簡潔性は損なわれてしまう。実際の作業では、これは人が行うものではなく、リンクエンジンの仕事となる。

こここの linkPack では、アプリケーション側で SGML の id として URL や URN を直接扱えることを前提として DTD が書かれている。そうでない場合の DTD は、link の属性に sourcetype を加えてタイプを明記する必要がある。

#### 課題

本稿の段階では、dublin core の各 element と、それぞれの Metadata の各 element との関係、さらには他の情報へのリンクをどう関係付けるのかについての仕組みが、実現していない。現段階では、単に検索対象項目としての働きしかない。同じカタログ情報として Metadata 内にある関連項目との関連づけは、必要なことであり、今後の課題である。

また、データ間の新規の関係を、既存のデータにある Metadata にどのように記述していくのかが、本稿では規定されていない。すなわち、他のデータからリンクを張られたということをどのように知り、また Hub-Metadata に記述していくのかという仕組みである。これを記述書式のレベルでその仕様が規定できるかどうかの判断は、今後の課題である。

## 6 さいごに

情報管理において、Metadata は ID として働く重要なもので、木構造ではなく、リンクでの管理を視野に入れた際には、その中心的役割を果たす情報になる。電子図書館などで、文化資産と電子情報とを同時に扱う際にも、Hub-document としての Metadata は、重要な役割を果たすと思われる。

#### A selection of useful references

1. IFLA, 1997.02.20, Digital Libraries: Metadata Resources, URL=<http://www.nlc-bnc.ca/ifla/II/metadata.htm>
2. EAD, EAD DTD, <http://lcweb.loc.gov/loc/standards/ead/>
3. OCLC/NCSA, OCLC/NCSA Metadata Workshop Report, <http://www.oclc.org/5046/conferences/metadata/metadata.html>
4. TEI, <http://www.uic.edu:80/orgs/tei/>
5. Handls, <http://www.handl.net/>
6. GILS, <http://www.usgs.gov/public/gils/>

7. PICT, <http://www.w3.org/pub/WWW/PICT/>
8. RFC1736, 1995, Functional Recommendations for Internet Resource Locators
9. RFC1737, 1994, Functional Requirements for Uniform Resource Names
10. RFC1738, 1994, Uniform Resource Locators(URL)
11. RFC1807, A Format for Bibliographic Records
12. RFC1808, Relative Uniform Resource Locators
13. RFC1835, Architecture of the WHOIS++ service
14. RFC2056, Uniform Resource Locators for Z39.50
15. D-Lib, <http://www.dlib.org/>
16. Search Engine, [http://www.accesschicago.net/engines.html/](http://www.accesschicago.net/engines.html)