

相互連携可能な情報提供システムの 研究

— 用語体系および検索システムへの適用 —

高久雅生

図書館情報メディア研究科

筑波大学

2004年9月

A Study of Cooperative Information Service Systems

**— Applications to Terminological Systems and
Information Retrieval Systems —**

Masao Takaku

Graduate School of Library, Information and Media
Studies

University of Tsukuba

September 2004

要旨

本研究では、相互連携の考え方に基づく情報提供システムについて、用語体系と検索システムを対象として研究を行った。本研究は 3 つの研究に大別される。第 1 は検索システム間の相互連携であり、第 2 は用語体系間の相互連携、第 3 はそれらを発展させた異種情報提供システム間の相互連携である。これらそれぞれについて相互連携可能な方式を提案し、システム構築を通じてその有効性を示した。

World Wide Web の出現によりインターネット上の情報空間に存在する情報量が急激に増大し、利用者が増えていった。利用者が多様化するのにもとまない、多様な情報アクセス要求が存在している。例えば、あるキーワードに関連した情報が欲しい、特定の分野・カテゴリに属する情報が欲しい、または、特定の Web サイトや Web ページにある情報が欲しいなど、多様なアクセス要求がある。一方、情報提供を行う側にも、このような多様なアクセス要求に応えるために、自らが提供している情報について多様なアクセスを可能にしたいというニーズがある。

このような多様なアクセス要求に応えるためには、それぞれのアクセス要求を満たすような情報提供システムがそれぞれ必要となるが、それは容易ではない。情報提供システムにはそれぞれ長所と短所があり、多様なアクセス要求に応えるには、それぞれの要求に応えねばならず不足する面が出てくる。つまり、単一の情報提供システムで全ての要求に対応することは現実的ではない。既存システムの不足する点を補うためには、新たな情報提供システムを構築することが有効である。しかし、新たな情報提供システムを一から構築することは容易ではなく、いかにそのコストを低減するかが問題となってくる。

そこで本研究では、多様なアクセス要求に応えることができるようにすることを目指し、情報提供システムの相互連携の考え方を取り入れて、この問題を解決する。情報提供システムの相互連携性を高めることにより可能となるサービスとしては、例えば用語体系であれば、自らの慣れた分類構造を使って、別用途で提供されている情報提供システムの情報を眺めたり検索すること、また、情報検索であれば、自らの慣れた検索クライアントやインタフェースを通じて、普段は利用しない検索システムに対して、普段と同様のやり方で検索すること、などが考えられ、いずれも有用である。

本研究では、相互連携性の向上というアプローチのもと、用語体系と検索システムという 2 種類

の情報提供システムを対象とした研究を行った。この 2 種類の情報提供システムを取り上げた理由は、これらがいずれも情報提供システムの基盤となる重要な役割を担っている点、また、その構築・提供・保守の過程が様々な場所や組織により独立に行われている点、さらに、用語体系と検索システムはそれぞれが完全に独立の存在であるわけではなくこれら同士を連携した情報提供が行われることが多いという点の 3 点からである。

以下では、本研究で行った 3 つの相互連携のそれぞれについて述べる。

第 1 の検索システムの相互連携については、情報検索用の通信プロトコル Z39.50 に着目した。利用者にとっては、ユーザ自らの慣れた検索インタフェースから様々な検索システムをそのまま検索したいというニーズがあり、一方、検索システムの提供を行う側から見れば、検索サービスを実行するサーバを一つ提供することによって、世界中のどのようなクライアントからでも検索してほしいというニーズもある。これらのニーズを満たすために、Z39.50 プロトコルが国際標準として策定されたが、1999 年当時、日本語書誌データを対象としたサーバ・クライアント双方を備えた Z39.50 システム構築の報告例は、筆者の属する研究グループによるものを除き見当たらなかった。そこで、Z39.50 に基づく検索システム間の相互連携を考慮したシステムとして、日本語書誌データを対象とした Z39.50 検索サーバ（以下、Z39.50-JP）を構築した。大規模な日本語書誌データに対する実用的な検索システムとして検索サービスが行えることを示した。また、Z39.50 における検索スキーマと検索対象データとの間の実装依存の問題を解決するために、Z39.50 サーバが持つ検索対象データと検索用スキーマの両者に Dublin Core メタデータを用いる手法を採用した Z39.50 サーバ（以下、Z39.50-DC）を構築した。

第 2 の用語体系の相互連携については、様々な用途により作成されてきた全く異なる用語体系を、相互に交換可能として共通に扱うという考え方を採用した。これにより、様々な用語体系を共通の相互連携可能な「用語体系」ととらえ、情報資源として活用できるようになる。ここでいう用語体系とは、用語概念間で構造、特に階層構造を持つものを指し、具体的には、分類体系やシソーラス、電子化辞書、Web ディレクトリ、Web サイト上のサイトマップなどを含む。用語体系に必要な要素としては、同定可能な用語（見出し語）と階層関係を取り上げた。階層関係は、様々な分野・用途で用いられる基本的な関係であり、それぞれの用語体系毎にそれぞれの意味付けがされている。本研究では、これらの関係をどれか一つの見方から限定することは行わず、単純化して階層関係として統一し、このように単純化した階層関係を基本とすることにした。この考え方により構築した複数用語体系ブラウジングシステム（以下、MTermBrowser）は、複数の用語体系を自由に組み合わせながら重ね合わせることができる特徴を持ち、重ね合わせた結果をグラフィカルな表現としてブラウジングできる。そのため利用者は、用語体系の構造を容易に把握でき、かつ、複数用語体系間での観点の違いも発見できる。

第 3 の異種情報提供システム間の相互連携については、Web サービス（Web Services）の枠組みに着目した。Web サービスは、XML により定義、記述されたメッセージを Web 上でやり取りすることにより、連携して動作可能なソフトウェアシステムを指す。用語体系や検索システムなど

情報提供システムの基盤となるシステムは、単体で使うだけではなく、他システムのコンテンツやサービスと連携することにより大きな意味を持つようになる。そのため、情報提供システムを他システムの一構成要素として扱うための枠組みが必要になってくる。そこで本研究では、Web サービスによる用語体系データの提供および検索システムを含む異種情報提供システムとの相互連携手法について研究を行った。各用語体系を Web サービスを通じてネットワーク上から自由にアクセスでき、それらを共通に利用できる枠組みを考案し、用語体系データを提供する Web サービスシステム（以下、Term-WS）を構築した。さらに、既存の検索システムなどと Term-WS を連携させた 2 種類の応用システムを構築することにより、Term-WS の相互連携可能性を示した。Term-WS を使うことにより、各用語体系のデータを共通の方法で容易に取得できるため、各用語体系を組み合わせ使用する応用システムを様々な形で構築でき、用語体系の提供、利用、再利用が柔軟に行えるようになることを示した。

本研究全体を通じたテーマは、情報提供システムの相互連携性である。本研究では相互連携性の要素として、共有、連携、独立という 3 つの要素を重視した。

共有とは、情報提供システムを共通に使えることにより、情報資源の共有が可能となることを指し、別々の情報提供システムを切り替えて使うなどの応用も可能となる。例えば検索システムにおいては、検索システムとのやりとりや検索式の構成要素を共通化する必要がある。Z39.50-JP、Z39.50-DC においては、Z39.50 プロトコルに基づく検索サービスを提供することにより、世界中のどのようなクライアントからでも Z39.50 による共通したアクセスを行えることを示した。また、用語体系を扱う際には、複数体系を同時に利用するためお互いの持つ用語とその関係を共有できるものとする必要がある。MTermBrowser、Term-WS では、用語体系における見出し語と階層関係に的を絞って、これを共通化してとらえ、様々な用途で作成される用語体系を共通に扱えるようにした。

連携とは、異なる情報提供システム同士を自由に組み合わせて動作させることであり、システムの相互連携性を強化することにより、そのシステムが従来もっていなかった機能を容易に追加していくことが可能になる。また、新たな情報提供システムを構築する際にも、一からシステムを構築するだけでなく、既存のシステムを組み合わせ新たなサービスを提供することが可能となる。Z39.50-JP、Z39.50-DC では、Z39.50 プロトコルによってクライアント・サーバ間のやりとりが共通化されていることを活かして、Z39.50 プロトコルを用いる他システムと連携させることができる。MTermBrowser では、複数の用語体系を自由な組み合わせで重ね合わせることで、体系間の連携を行う手法を開発した。さらに Term-WS では、Web サービスをバックエンドとした応用システムにより、様々なサービスと用語体系とを連携させるシステムが容易に構築できることを示した。

独立とは、情報提供システムをネットワーク上の分散環境のもとで独立して管理、運用できることであり、様々な目的を持った多様な情報提供システムが作られる可能性が広がる。つまり、個々の組織や個人によって作成された情報が、その組織・個人のもとからそのまま発信できるように

なっていることが、情報の効率的な提供、発信において重要となる。Z39.50-JP、Z39.50-DC は、検索システムをネットワーク上で分散的にそれぞれ独立に提供することを前提としたシステムである。用語体系はそもそも、それぞれの用途、目的に沿って作られており、独立した性質を持っているため、MTermBrowser、Term-WS はその独立性に対応した。とりわけ Term-WS では、ネットワーク上に分散して提供される用語体系を想定し、これに対応する Web サービスによる提供手法を考案し、その提供手法に基づくシステムを構築した。

このように本研究では、既存システムでは不十分であった問題を解決するために、多様な情報提供システムの相互連携の考え方に基づいたシステムの研究を行った。対象とした用語体系と検索システム以外の情報提供システムにおいても相互連携の考え方に基づくシステムを提供することにより、それぞれのシステムの長所をさらに活かした情報提供システムができるようになり、ユーザの多様な情報アクセス要求に応えることができるようになる。

Abstract

I studied an approach of cooperation among information service systems including terminological systems and information retrieval systems. This work consists of three studies: the cooperation among information retrieval systems, the cooperation among terminological systems, and the cooperation among heterogeneous information service systems. I suggested cooperative methods in these fields and showed their effectiveness by building systems that implement them.

The amount of information on the Internet and the number of users have been increasing exponentially with the World Wide Web. Various new information needs have been appearing with the increasing users. For example, some users need information related to a certain keyword, or others need information on a certain field/category, or yet others just retrieve information from a certain Web site/page. Meanwhile, people who provide information through the Web have been increasing too, and they need to allow their users to access their information in various ways.

In order to satisfy various end-users' needs, it is necessary to build every information service system to satisfy all the access need. But it is difficult to build the systems to satisfy various needs. Every system has each advantage and disadvantage, the system cannot satisfy the various needs. Therefore it is not realistic to build one single system to satisfy all users' needs. It is effective to build new information service systems to make up for the disadvantages of existing systems. However it is difficult to build a new system from scratch because of the cost to build it.

I tackled this problem with an approach of cooperation among information service systems aiming at satisfying users' various needs. Improving the cooperative capability of a system expands its services. In the case of the terminological systems, a user can use unfamiliar service system navigating with his/her familiar terminological system. In the case of the information retrieval systems, a user can search unfamiliar systems using their familiar search client or interface.

My approach to improve the cooperative capability of a system was applied to two information service systems: terminological systems and information retrieval systems. These two systems were chosen for the following reasons. Both systems provide the basis for the information service systems and play an important role in the service systems. And, both systems involve many different organiza-

tions and/or individuals in building, providing and maintaining each system. Some of both system is not independently provided, since information services are often provided with cooperation of them.

Next, I describe details of my three main studies.

First, the cooperation among information retrieval systems was achieved by the Z39.50 standard. Z39.50 is a communication protocol for information retrieval service. Users have a need to use various retrieval systems with their familiar search client. Providers of a retrieval system have a need to allow users to search their system by providing just a single server. In order to satisfy these needs, Z39.50 was developed as an international standard. Early in 1999, we were one of the first group that developed a retrieval system based on Z39.50 for Japanese bibliographic data. I developed a Z39.50 retrieval system (Z39.50-JP) supporting for the cooperation among retrieval systems. I showed that Z39.50-JP was suitable for practical use for large Japanese bibliographic data set. I developed another Z39.50 system (Z39.50-DC) for solution of the mapping problem of the attribute set in Z39.50 that had been pointed out and was pointed out in this study also. Z39.50-DC uses the Dublin Core Metadata Element Set as a common schema. Z39.50-DC stores bibliographic data in a format of the Dublin Core metadata and allows a user to search the system using the Dublin Core access points.

Second, the cooperation among terminological systems was achieved by an approach that heterogeneous terminological systems can be seen as a common hierarchical structure. By using this approach, various terminological systems can be used to create cooperative systems that have the common structure, and can be utilized as information resources. A terminological system has a structure, particularly a hierarchical structure, among the terms. Specifically, it includes taxonomies, thesauri, dictionaries, Web directories, and Web sitemaps, and comprises terms and hierarchical relationships which the terms have. The exact meaning of hierarchical relationships differs from one terminological system to another. I unified hierarchical relationships and used the resulting relationships as a basic structure. I developed a browsing system (MTermBrowser) for various terminological systems based on the approach of the unified relationships. A user of MTermBrowser can integrate one or more terminological systems in arbitrary combination and browse the integrated result as a graphical representation, and understand the hierarchical relationships among selected terms intuitively.

Third, the Web services provide a infrastructure of cooperative software systems that communicate messages based on XML through the WWW. My approach to cooperative heterogeneous information service systems can be used with a infrastructure of the Web services. To achieve the cooperation among some basic information service systems, including terminological systems and retrieval systems, is often needed rather than a stand-alone system, a framework for handling a basic system as a sub-system of another system is needed. So, I applied the framework based on the Web services to the basic information service systems: terminological systems and retrieval systems. I suggested a cooperative method for providing data of terminological systems through the Web services and cooperating

terminological systems with heterogeneous information service systems including retrieval systems. My method is a framework that allows each terminological system to be commonly accessible through the Web services from the Internet. I developed an information service system (TermWS) that provides data of a terminological system based on the framework. And I developed two applications that used TermWS in cooperation with other information service systems. Since TermWS allows a user easily access data of each terminological system in a common way, it is possible to build various cooperative applications using TermWS and to provide, use and reuse a terminological system more flexibly.

In this paper, I described the cooperative information service systems. I think that three concepts, unification, cooperation and independence, are important for the cooperation among information service systems.

First, unification of a system means that information resources can be shared by using a system in common way. Unification of a system enables more flexible use of the system; e.g. a user can switch easily from one system to another. In case of a retrieval system, it is necessary that the system can communicate with others by using common protocol and structure of queries. In Z39.50-JP and Z39.50-DC, I showed that a retrieval service based on Z39.50 protocol allowed a user to use the server with any Z39.50 clients from around the world. In case of a terminological system, it is necessary that the system have the structure of terms and relationships in common with the others. In MTermBrowser and TermWS, I focused on terms and hierarchical relationships of the systems and standardized them. As a result, various terminological systems, which are created for various purposes, can be used in common way.

Second, the cooperation of a system means that various different service systems can be used cooperatively with each other. By improving cooperative capabilities of a system, it is possible to add new capabilities to the system easily. Additionally, it is possible to provide a new service by combining existing systems. In the case of Z39.50-JP and Z39.50-DC, the system can cooperate with other systems based on Z39.50. In the case of MTermBrowser, I developed a method that cooperates with multiple terminological systems by integrating terms and relationships of the systems in arbitrary combination. In the case of TermWS, I successfully developed applications that provide cooperation between terminological systems and various service systems based on the Web services.

Third, the independence of a system means that the system can be independently operated and managed under the present distributed network environment. By providing a system independently, it is possible to provide many service systems for various purposes because it is important for individual person or organization to create and provide effectively information from them. In the case of Z39.50-JP and Z39.50-DC, each system is independently provided through Z39.50 protocol under the distributed network environment. The terminological systems have the nature of independence due to their purposes. So, MTermBrowser and TermWS support it. Especially in TermWS, I suggested a

method to provide data of terminological systems under distributed network environment and developed the system based on the method.

Thus, I studied information service systems based on the cooperation of various systems aiming to overcome disadvantages of existing systems. I believe that the cooperative approach like this study can be applied to different systems other than retrieval system and terminological system, and an application of the cooperative approach to the system makes more effective service and satisfies more users' needs.

目次

第 1 章	序論	1
1.1	現状と問題点	1
1.2	本研究のアプローチ	2
1.2.1	概要	2
1.2.2	検索システム間の相互連携	4
1.2.3	用語体系間の相互連携	4
1.2.4	異種情報提供システム間の相互連携	5
第 2 章	データの再利用が可能な複数用語体系ブラウジングシステムの構築	7
2.1	背景と目的	7
2.2	用語体系の重ね合わせ	8
2.3	対象データ	9
2.4	ブラウジングシステム	11
2.4.1	概要	11
2.4.2	インデックスの作成	13
2.4.3	可視化・表示プログラム	14
2.4.4	開発環境	17
2.4.5	実行例	17
2.5	考察	21
2.5.1	関連研究	21
2.5.2	用語体系の拡張	22
2.5.3	用語体系構築への重ね合わせ手法の活用例	22
2.6	まとめ	24
第 3 章	Web サービスによる用語体系データの提供とその応用システム	25
3.1	背景と目的	25
3.2	Web サービス	26

3.3	用語体系データ	27
3.4	Web サービスを用いた用語体系データ提供	27
3.4.1	Web サービスシステム	28
	用語体系データ Web サービス	29
	GraphViz Web サービス	32
3.4.2	フロントエンドシステム	32
	用語体系ブラウジングシステム	33
	Google Web API との連携システム	34
3.4.3	開発環境	35
3.5	考察	36
3.6	まとめ	39
第 4 章	Z39.50 による日本語書誌データ検索システム	40
4.1	背景と目的	40
4.2	Z39.50 検索システム	41
4.2.1	システムの概要	41
4.2.2	日本語書誌データの処理	42
4.2.3	Z39.50-ULIS3 サーバ	44
4.2.4	Z39.50-ULIS3 クライアント	45
4.2.5	システムの実行例	48
4.2.6	開発システムの変遷	50
4.3	考察	52
4.3.1	システムの意義	52
4.3.2	Z39.50 検索システムの将来	53
4.4	まとめ	55
第 5 章	Dublin Core を共通スキーマとした Z39.50 検索システム	56
5.1	背景と目的	56
5.2	システムの構成	57
5.3	検索例	61
5.4	考察	62
5.4.1	関連研究	62
5.4.2	JAPAN/MARC データの Dublin Core への変換	63
5.5	まとめ	63
第 6 章	結論	65

参照文献	69
謝辞	75
発表文献一覧	77

目次

1.1	多様な情報アクセス要求と情報提供システム	2
2.1	用語体系における階層関係の記述	9
2.2	重ね合わせ例: 重ね合わせ結果	9
2.3	対象とした用語体系のデータ例: ODP	12
2.4	システムの構成	13
2.5	階層関係インデックスの Perl での表現例	14
2.6	可視化・表示プログラムのデータの流れ	15
2.7	動的に生成した XML (NDC): 「図書館教育」	15
2.8	動的に生成した XML (重ね合わせ結果): 「アニメーション」	16
2.9	dot 形式に変換する XSLT スタイルシート例	16
2.10	動的に生成した画像の例	17
2.11	HTML による一覧表示: 「図書館」(NDC)	18
2.12	HTML による一覧表示: 「図書館活動」(NDC)	18
2.13	グラフ構造のクリックابلマップ表示: 「図書館」(NDC)	19
2.14	グラフ構造のクリックابلマップ表示: 「図書館活動」(NDC)	19
2.15	グラフ構造のクリックابلマップ表示 (複数階層同時表示): 「コンピュータ」 (ODP)	20
2.16	複数用語体系の重ね合わせ結果: 「アニメーション」	20
2.17	見出し語検索機能 (検索結果): 「エネルギー」	21
2.18	重ね合わせ結果 (EIC+ODP): 「エネルギー」	23
3.1	用語体系 Web サービス・概念図	27
3.2	開発したシステムの概要	28
3.3	用語体系 Web サービスのインタフェース	29
3.4	doWordSearch におけるリクエスト例	29
3.5	doWordSearch におけるレスポンス例 (抜粋)	30

3.6	getWordList におけるリクエスト例	31
3.7	getWordList におけるレスポンス例	31
3.8	Graphviz Web サービスのインタフェース	32
3.9	doGraphViz におけるリクエスト例	32
3.10	doGraphViz におけるレスポンス例	33
3.11	用語体系ブラウジングシステム (1)	34
3.12	用語体系ブラウジングシステム (2)	35
3.13	用語体系ブラウジングシステムにおけるデータの流れ (見出し語検索時)	35
3.14	用語体系ブラウジングシステムにおけるデータの流れ (用語関係表示時)	36
3.15	Google Web API との連携システム	37
3.16	Google Web API との連携システムにおけるデータの流れ (1)	38
3.17	Google Web API との連携システムにおけるデータの流れ (2)	38
4.1	システム全体図	42
4.2	Japan/MARC の変換	43
4.3	検索処理	45
4.4	返戻処理	46
4.5	接続ウィンドウの機能	46
4.6	検索ウィンドウの機能	47
4.7	検索ウィンドウ: Z39.50-ULIS3 サーバ	48
4.8	検索ウィンドウ: 図書館情報大学デジタル図書館システム Z39.50 サーバ	49
4.9	検索ウィンドウ: University of Wisconsin, Madison	50
5.1	システムの構成	57
5.2	Dublin Core を共通スキーマとした Z39.50 検索システムの検索例	62

表目次

2.1	各用語体系の用語数および階層関係数	10
2.2	任意の複数体系を重ね合わせた場合の重なり語数	11
3.1	対象とした各用語体系の概要	27
4.1	Bib-1 と Japan/MARC の対応表	44
4.2	開発システムの相違点	51
5.1	Dublin Core と JAPAN/MARC の対応関係	58
5.2	Bib-1 に追加された Dublin Core エlement	61
6.1	情報提供システム間の連携可能性	66

第 1 章

序論

1.1 現状と問題点

以下に述べるように、情報提供システムとその利用環境は大きく変化・発展してきたが、その過程の中から新たな問題点・解決すべき課題の存在もまた明らかになってきた。

まず、情報提供システムの重要な要素である情報検索システムにおいては、1960 年代以降、大型計算機と専用端末に基づく中央集中型のオンライン情報検索システムが提供されてきた。この状況は、1990 年代初頭に至ってネットワーク技術の発達と安価な計算機資源の普及が進むまで続いた。この検索環境では、検索システムは商用データベースが主流で高額な使用料を要するため、ノイズの少ない的確な情報検索を短時間で行う必要がある。そのため、実際の検索は熟練したサーチャやオペレータなどの専門家を介することが多く、一般の利用者が日常的に検索システムを使用することは少なかった。1990 年代に入り、インターネットが普及してくると、この状況は変わり始め、一般の利用者がそのままネットワークを介して利用できる検索システムが出現するようになった。

とりわけ、WWW (World Wide Web; 以下 Web という) [1][2] の出現によりこの状況は一変した。インターネット上の情報空間に存在する情報量が急激に増大し、利用者を増やしていった。また、Web は情報発信を容易にし、行政情報や企業、団体、さらには多くの個人利用者の手によるコンテンツも作られていった。この膨大な情報から必要な情報を入手するために、サーチエンジンと呼ばれる検索システムが開発され、一般利用者に提供されるようになった [3]。まず、初期のサーチエンジンとして Yahoo! [4] が出現した。これは、図書館で使われる分類表などにみられる階層的な分類体系を Web 上のコンテンツに適用し、Web 上のコンテンツを人手により組織化することを試みたものであり、成功を収めた。この Yahoo! のような形態のシステムは「ディレクトリ型サーチエンジン」と呼ばれる。次いで、Altavista [5] などの Web 上のコンテンツを自動的に収集し、収集したページ群全てを対象に全文検索を行うタイプのサーチエンジンが現われた。これは「ロボット型サーチエンジン」と呼ばれる。

一方、Web の特性として、簡易な検索手法であるナビゲーションに基づくブラウジングも行われ

ている。Web サイトで情報提供を行う際に各サイト内のコンテンツを階層的に整理した「サイトマップ」の手法も一般的になっている。

サーチエンジンにおける問題点としては、以下の2つがある。ディレクトリ型サーチエンジンを使う場合には、利用者はその階層のどこに目的とする情報があるかを探す必要がある。しかし、サーチエンジンの対象は全分野にまたがるため、全分野の階層は巨大なものになりがちで、これを把握することは困難である。また、ロボット型サーチエンジンにおいては、利用者が欲しい情報を得るためには適切なキーワードを入力し、その語が含まれる Web ページを検索するという手順を取る。その際、自らの検索要求に合った適切なキーワードを選択することも困難な問題である。

一方、検索システムの相互運用性の確保も重要な課題である [6]。その理由は、様々な検索システムが存在し、個々のシステムに依ってその使用法・使い勝手が異なるために、エンドユーザが使いなれた検索インタフェースで検索が行えないからである。Z39.50 [7] はこの課題に応えるべく開発された標準規格である。Z39.50 は、様々な検索システム間で相互運用を可能とする通信プロトコルであり、欧米を中心に広く普及している。しかし本著者がこの研究を始めた当時、Z39.50 における日本語を始めとする非欧米言語の処理については未発達の状態であり、日本語書誌を対象とした Z39.50 検索システムの構築や運用の際の問題点の検証があまりなされていなかった。

1.2 本研究のアプローチ

1.2.1 概要

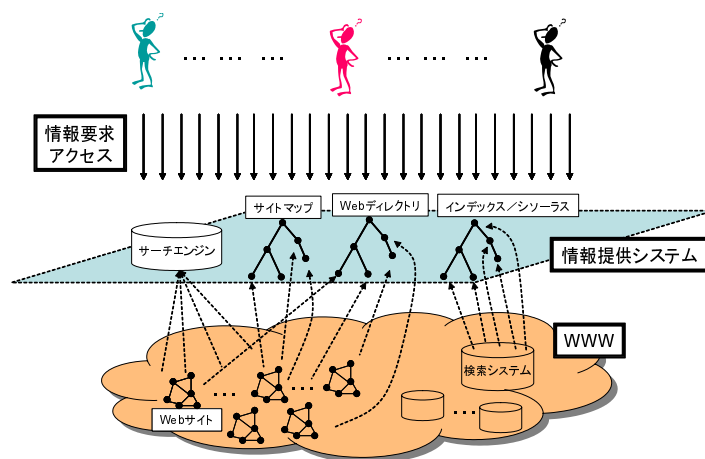


図 1.1 多様な情報アクセス要求と情報提供システム

今日、利用者の多様化にともない、多様な情報アクセス要求が存在している。例えば、あるキーワードに関連した情報が欲しい、特定の分野・カテゴリに属する情報が欲しい、または、Web サイトや Web ページを特定した情報要求など、多様なアクセス要求がある（図 1.1）。

一方で、情報提供を行う側にも、このような多様なアクセス要求に応えるために、提供している情報について多様なアクセスを可能にしたいというニーズがある。このような多様なアクセス要求に応えるためには、それぞれのアクセス要求を満たすような情報提供システムがそれぞれ必要となる。

1.1 節で述べたように多様な情報提供システムが構築されているが、既存の情報提供システムだけで全てのアクセス要求に応えられているわけではなく、既存システムでは不十分である。

情報提供システムにはそれぞれ長所と短所があり、多様なアクセス要求に応えるには、それぞれの要求に応えねばならず不足する面が出てくる。つまり、単一の情報提供システムで全ての要求に対応することは現実的ではない。例えば、ロボット型サーチエンジンの場合、適切なキーワードを選択できれば、Web 上の情報を検索するのに有用だが、キーワードの選択が難しいという問題がある。またディレクトリ型サーチエンジンの場合には、利用者が要求する分野・カテゴリを適切に示すカテゴリがあれば有用だが、適切な階層を探したり、把握したりするのが難しいという問題がある。

既存システムの不十分な点を補うためには、新たな情報提供システムを構築することが有効である。しかし、新たな情報提供システムを一から構築することは、コストの面からも、困難な問題である。例えば、新たな電子図書館サービスを行う場合、情報の提供や整理などで利用するカテゴリやボキャブラリを一から構築し維持していくのは大きなコストがかかるため、利用者の情報要求に沿うサービスを提供できない場合も多く、いかにそのコストを低減するかが問題となっている。

そこで本研究では、情報提供システム同士を組み合わせるという手法により、それぞれの長所を生かしたまま短所を補いあえるのではないかと考えた。つまり、既存の情報提供システムを組み合わせることによって、それぞれの短所が補われ、多様なアクセス要求に応えることができるようになる。しかしながら、それぞれに全く異なる目的で構築された情報提供システム同士を組み合わせることは容易ではなく、どのように連携させ整合性を保つのか、どのようにそれぞれを効率的に管理するのかなど、困難な問題を含んでいる。

そこで本研究では、情報提供システムの相互連携の考え方を取り入れて、この問題を解決する。情報提供システムの相互連携性を増すことにより、情報提供サービスを独立して作成・提供しながらも、複数の提供サービス間をまたがって利用でき、かつ、自由な組み合わせで他サービスと連携可能であるような、ユーザの多様なアクセス要求に応えるシステムが作成できる。

情報提供システムの相互連携性を高めることにより可能となるサービスとしては例えば、用語体系であれば、自らの慣れた分類構造を使って、別用途で提供されている情報提供システムの情報を眺めたり検索すること、また、情報検索であれば、自らの慣れた検索クライアントやインタフェースを通じて、普段は利用しない検索システムに対して、普段と同様のやり方で検索すること、などが考えられ、いずれも有用である。

本研究では、相互連携の考え方に基づく情報提供システムについて、用語体系と検索システムを対象として研究を行った。この2種類の情報提供システムを取り上げたのは、以下の理由からであ

る。まず、これらはいずれも基盤であって、重要な役割を担っている。この2種類の情報提供システムの構築・提供・保守は、様々な場所や組織により独立に行われている。さらに、用語体系と検索システムはそれぞれが独立の存在であるわけではなく、これら同士を連携した情報提供が行われることが多い。そのため、本研究のアプローチが有効と考えた。

本研究は3つの研究に大別される。第1は検索システム間の相互連携であり、第2は用語体系間の相互連携、第3はそれらを発展させた異種情報提供システム間の相互連携である。これらそれぞれについて相互連携可能な方式を提案し、システム構築を通じてその有効性を示した。以下の節では、それぞれの研究の概要を述べる。

1.2.2 検索システム間の相互連携

検索システムの相互連携については、様々な検索システムがサービスされている現状から、利用者にとっては、相互に違うインタフェースを使い分けなければならない煩雑さがあり、ユーザ自らの慣れた検索インタフェースから様々な検索システムをそのまま検索したいというニーズがある。一方、検索システムの提供を行う側から見れば、検索サービスを実行するサーバを一つ提供することによって、世界中のどのようなクライアントからでも検索してほしいというニーズもある。これらのニーズを満たすために、Z39.50が標準化されたが、1999年当時、日本語書誌データを対象としたサーバ・クライアント双方を備えたZ39.50システム構築の報告例は、筆者の属する研究グループによるものを除き見当たらなかった。そこで本研究では、Z39.50に基づく日本語書誌データ検索システムが必要であると考え、その有効性を確認するためのシステム構築を行い、情報知識学会誌に論文[8]として発表した。なお、上記論文[8]における筆者の担当箇所は、Z39.50サーバの設計と開発にある。このシステムについては、4章で述べる。

さらに、Z39.50システムにおいてはBib-1アトリビュートセットによるアクセスポイント指定が一般的であるが、異なるZ39.50サーバ間では同じフィールドが別のマッピングがされている場合があるなど、相互運用性が確保されていない問題もある。そこで本研究では、Dublin Coreメタデータ[9][10]を共通スキーマとしたZ39.50サーバの提供により、この問題の解決が図れると考え、その有効性を確認するためのシステム構築を行い、デジタル図書館ワークショップにおいて口頭発表した[11]。このシステムについては、5章で述べる。

1.2.3 用語体系間の相互連携

用語体系の相互連携については、様々な用途により作成されてきた全く異なる用語体系を、相互に交換可能として共通に扱うという考え方を採用した。これにより、様々な用語体系を共通の相互連携可能な「用語体系」ととらえ、情報資源として活用できるようになる。

ここでいう用語体系とは、用語概念間で構造、特に階層構造を持つものを指し、具体的には、分

類体系やシソーラス、電子化辞書、Web ディレクトリ、Web サイト上のサイトマップなどを含む。この際、用語体系に必要な要素としては、同定可能な用語（見出し語）と階層関係を取り上げた。階層関係は、十進分類や組織図に見られるように様々な場所で用いられる基本的な関係である。これらの階層関係は、それぞれの体系内でそれぞれの意味付けがされているが、本研究では、これらの関係をどれか一つの見方から限定することは行わず、単純化して階層関係として統一して扱うこととした。このように単純化した階層関係を基本とすることによって、たとえ別の分野で相互に関連の少ない用語体系間であっても一定の利用が可能となる。また、類似の用途で作成された体系間であれば、それぞれの体系を相互に利用することはより有用となる。

例えば、異なる分野では使われる用語も異なることから、分野毎に用語体系が存在し、それぞれの用語体系は独立して構築・保守されている。情報提供システムを作成する側から見た場合、新たに用語体系を構築する際には、既存の似た分野の構造の一部を取り込んで再利用できれば、構築のコストを減らすことができ有用である。一方、情報提供システムのエンドユーザの側から見た場合、自分の専門外の分野の文献を検索する際に、自分の分野の用語体系を使って検索できると便利である。

本研究では、用語体系の構築を効率的に行うためには、以下のニーズを満たすことが重要になると考えた。

- 用語体系をブラウジングする。
- 用語体系の構造を把握する。
- 複数の用語体系の間の相違点・共通点を発見する。
- 用語体系データの再利用を行う。

そこで、これらのニーズを満たすために、データの再利用が可能な複数用語体系ブラウジングシステムが必要であると考え、その有効性を確認するためのシステムを構築し、情報知識学会誌に論文[12]として発表した。このシステムについては、2章で詳しく述べる。

1.2.4 異種情報提供システム間の相互連携

異種情報提供システム間の相互連携については、Web サービス（Web Services）[13][14]の枠組みに着目した。Web サービスは、XMLにより定義、記述されたメッセージをWeb上でやり取りすることにより、連携して動作可能なソフトウェアシステムを指す。

用語体系や検索システムなど情報提供システムの基盤となるシステムは、単体で使うだけではなく、他システムのコンテンツやサービスと連携することにより大きな意味を持つようになるものも多い。特に、他システムと連携したり、他システム内の一モジュールとして扱いたいというような場合では、その情報提供システムを他システムの一構成要素として扱うための枠組みが必要になってくる。例えば用語体系においては、用語辞書や分類体系、シソーラスなど用語体系を単体で使う

だけでなく、検索システムなど他の情報提供システムにおけるコンテンツやサービスと連携することが必要となることも多く、用語体系を扱う際にはその再利用性や他システムとの連携可能性が重要となる。

そこで本研究では、異種情報提供システム間の相互連携手法に Web サービスの枠組みを採用することにより、これらの問題を解決できると考えた。つまり、Web サービスの枠組みに基づいて、用語体系データを提供し、検索システムを含む異種情報提供システムとの相互連携を行う応用システムの構築により、異種情報提供システム間の相互連携性が高まると考えた。そこで、その有効性を確認するためのシステム構築を行い、情報知識学会誌に論文 [15] として発表した。このシステムについての詳細は、3 章で述べる。

第 2 章

データの再利用が可能な複数用語体系 ブラウジングシステムの構築

2.1 背景と目的

概念間の関係や用語の意味を記述した用語体系は、情報検索や自然言語処理などの分野で幅広く用いられてきた。近年では、シソーラス、分類表、Web Directory などのさまざまな用途で用語体系が作成されている。Web 上では、全分野を網羅する約 44 万件からなる ODP (Open Directory Project) [16]、生物や化学、行政など幅広い分野から構成される環境用語を対象とした EIC ネット [17] における用語集など、多種多様な用語体系が存在する。また、Web 上の様々なサービスが独自の用語体系を提供することによって、利用者の利便をはかろうとしている。また、次世代の Web とも言われる Semantic Web [18][19] でも、その基盤となるオントロジーの構築の重要性がうたわれている。このように用語体系は、情報提供システムに欠かせない存在になりつつある。

これらの用語体系を作成し維持していくには、以下のようなプロセスを踏む必要がある。まず、体系の設計者は体系全体の設計を行い、その語彙を収集し、分析を行なう。そして、それらの用語間の関係を一貫性をもって把握し、管理しなければならない。しかし、用語体系は一般に大量の用語と用語間の関係を含むため、これらのプロセスには多くのコストがかかっている。

筆者は、上記のような用語体系の作成、維持を支援するには、用語体系のブラウジングと複数用語体系の重ね合わせおよび再利用が有効であると考えた。これらの利点を以下に挙げる。

- **ブラウジング**

多くの用語とそれら同士の関係が記述された用語体系の構造を把握し分析するためには、体系を文字として記述したものを眺めるよりも、これらの構造を概観できるように、わかりやすく可視化し、ブラウジングできるようにすることが有効である。

- **複数の用語体系の重ね合わせ**

用語体系の作成の際には、複数の用語体系がもっている用語とそれらの関係を重ね合わせて表示することが、体系の把握および語彙の収集、用語の持つ観点の把握に有効である。本論文では、これを重ね合わせと呼び、複数の用語体系がもっている用語とそれらの関係を重ね合わせることを指す語として用いる。

- 用語体系の再利用

用語体系の重ね合わせを行うことにより、似たような目的を持った既存の用語体系の持つ構造、全く別の目的を持った体系の構造、特定の分野の体系の構造などの一部を再利用することができ、それが有効な場合もある。例えば、網羅的な体系を作成する際に、特定分野の体系を参照し、その構造を再利用したり、逆に特定分野の体系を作成する際には、汎用的な体系の構造を参照し、その構造を周縁領域の構造に再利用したりする場合である。

本章では用語体系のブラウジング、重ね合わせ、再利用が可能な用語体系ブラウジングシステムの構築について述べる。このシステムでは、任意の見出し語とそれに関係する語を複数の体系間で自由に重ね合わせ、その結果をブラウジングできるようにした。このような重ね合わせの結果を元にして利用者（体系の設計者）が用語の取舍選択を行うことが、複数用語体系の再利用に有効であると考えられる。さらに、システムが扱う複数の用語体系とその重ね合わせ結果を XML (Extensible Markup Language) 形式 [20] で提供することにより、複数用語体系間での再利用やデータ交換が可能になるものと考えられる。

2.2 用語体系の重ね合わせ

本研究では、用語体系を重ね合わせてその結果をブラウジングすることが用語体系の構築に有効であると考え、その機能を設けた。

別々の目的で作られた用語体系間では、体系が持っている見出し語に関する情報の量や詳細度、さらには、何を見出し語とするかについてのとらえ方が大きく異なる。このため、それらの体系同士を重ね合わせるためには、それぞれの体系に共通の情報、つまり、どの体系も共通に持っている「見出し語」と「上位・下位関係」の情報のみを基礎として重ね合わせを行う必要がある。

筆者らはこれまで、それぞれの用語体系が共通に持っている情報だけを用いて重ね合わせを行う手法について実験を行ってきた [21]。この実験手法を元にして、本研究では、どの体系にも共通して存在する見出し語とその語の持つ階層関係を対象として、見出し語の照合のみにより用語体系の重ね合わせを行う手法を採用した。

なお、本研究で構築したブラウジングシステムの目的は、利用者が重ね合わせ結果をブラウジングしながら、用語の取舍選択を判断するための判断材料を与えることにある。つまり、利用者は重ね合わせ結果を見ながら、新たな用語を収集したり、観点の違いを発見できる。これにより、用語体系の把握をすることができる。そのため、本手法による重ね合わせ結果の妥当性について、シス

テムの側では判断しないこととした。

本研究では、用語体系を、見出し語をノードとし、見出し語間の階層関係をエッジとする有向グラフ構造としてとらえる。以下では、「上位語 下位語」とノード間の関係を矢印として示す。

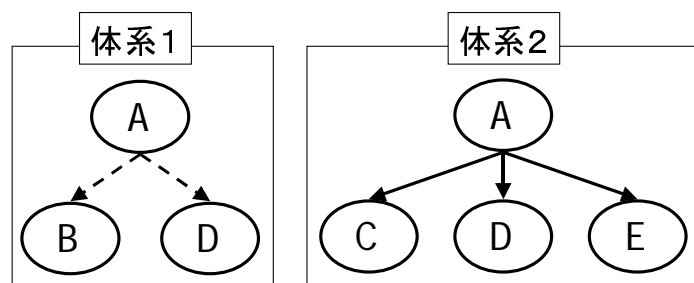


図 2.1 用語体系における階層関係の記述

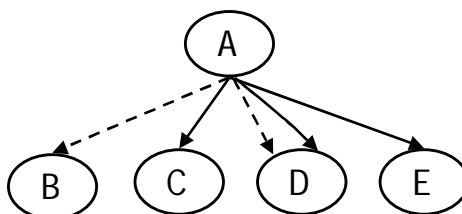


図 2.2 重ね合わせ例: 重ね合わせ結果

用語体系の重ね合わせは、複数の用語体系の部分グラフ同士を、同一の見出し語からなるノードを同一のノードとみなして結合し、各ノードの持つ関係、つまりエッジについてはそのまま保持することで行った。ここで、用語 A, B, C, D, E について階層関係が記述されている体系 1、体系 2 がある場合（図 2.1）を例にとって説明する。まず、用語 A および D が両体系に共通しているためこれらを結合する。さらに、用語 A の持つ階層関係 A → B（体系 1）、A → D（体系 1, 2）、A → C（体系 2）、A → E（体系 2）をそのまま保持して、重ね合わせ結果とする（図 2.2）。

2.3 対象データ

本システムが対象とした用語体系は、以下の 5 つである。

1. EDR 専門用語単語辞書（情報処理）中の名詞・概念体系辞書（以下 EDR）[22]
2. 日本十進分類表 新訂 8 版機械可読データファイル（以下 NDC）[23]
3. 国立環境研究所による環境情報ポータルサイト EIC ネット [17] 上の環境用語集（以下 EIC）
4. Open Directory Project [16] による Web ディレクトリの階層構造データ（以下 ODP）

表 2.1 各用語体系の用語数および階層関係数

用語体系	用語数	異なり語数	階層関係数	特徴
EDR	118,782	118,134	184,958	情報処理分野の専門用語を対象とした大規模な体系
NDC	30,659	25,410	102,639	全学問分野を対象とした大規模な体系
EIC	391	361	380	生物・化学・気象などの複合領域から構成される環境分野を対象とした体系
ODP	446,926	150,387	446,928	Web 上に存在する情報全てを対象とした大規模な体系
ODP-kt	2,059	1,617	2,058	Web 上に存在する情報のうち、子ども向けの情報に限定した部分を対象とした体系
計	–	292,479	–	

5. ODP のデータのうち、「Kids and Teens」階層に限ったもの（以下 ODP-kt）

これらの用語体系が持つ用語数、階層関係数、その特徴を表 2.1 に示す。また、この 5 つの用語体系データをそれぞれ任意の組み合わせで重ね合わせた場合に、重ね合わされる見出し語数を表 2.2 に示す。表中の「EIC + ODP」という記述は、体系 EIC と体系 ODP との組み合わせで重ね合わせを行うことを示す。なお、重ね合わせ手法の実際の活用例については 2.5.3 節を参照のこと。

実際の用語体系のデータの一例として、用語体系 ODP における「Top/World/Japanese/コンピュータ/ソフトウェア/インターネット/コミュニケーション」階層を中心とした階層構造の XML データの抜粋を図 2.3 に示す。これは ODP プロジェクトが作成、配布しているデータ形式である。この例では、<Topic>要素に一階層についての情報が記述されており、<narrow>要素にはその階層の持つ下位関係が記述されている。また、<d:Title>要素にその語の見出し語が記述され、<catid>要素にはその見出し語を一意に識別する ID が記述されている。<narrow>要素の r:resource 属性で示した ID が<Topic>要素の r:id 属性を示すことで、上位・下位関係の対応も知ることができる。

表 2.1 の用語体系を選択した理由は、様々な分野や規模の用語体系に本システムが適用できることを示すためである。本システムは、図 2.3 に示したような見出し語と上位・下位関係が記述されたデータであれば対象としうる。

表 2.2 任意の複数体系を重ね合わせた場合の重なり語数

組み合わせ	重なり語数
EIC + ODP	4
EIC + EDR	11
EIC + NDC	45
EIC + ODP-kt	0
ODP + EDR	142
ODP + NDC	489
ODP + ODP-kt	1,617
EDR + NDC	1,166
EDR + ODP-kt	0
NDC + ODP-kt	2
EIC + ODP + EDR	1
EIC + ODP + NDC	3
EIC + ODP + ODP-kt	0
EIC + EDR + NDC	2
EIC + EDR + ODP-kt	0
EIC + NDC + ODP-kt	0
ODP + EDR + NDC	39
ODP + EDR + ODP-kt	0
ODP + NDC + ODP-kt	2
EDR + NDC + ODP-kt	0
EIC + ODP + EDR + NDC	1
EIC + ODP + EDR + ODP-kt	0
EIC + ODP + NDC + ODP-kt	0
EIC + EDR + NDC + ODP-kt	0
ODP + EDR + NDC + ODP-kt	0

2.4 ブラウジングシステム

2.4.1 概要

本システムは、Web 上からのユーザの要求に応じて動的に用語とその用語の持つ階層関係を可視化し、提示する。本システムは、各用語体系から作成されるインデックス、可視化・表示プログラ

```

...
<Topic r:id="Top/World/Japanese/ コンピュータ/ソフトウェア/インターネット">
  <catid>499398</catid>
  <d:Title>インターネット</d:Title>
  <narrow r:resource="Top/World/Japanese/ コンピュータ/ソフトウェア/インターネット/メー
ラー"/>
  <narrow r:resource="Top/World/Japanese/ コンピュータ/ソフトウェア/インターネット/ブラウ
ザ"/>
  <narrow r:resource="Top/World/Japanese/ コンピュータ/ソフトウェア/インターネット/FTP"/>
  <narrow r:resource="Top/World/Japanese/ コンピュータ/ソフトウェア/インターネット/ホーム
ページ作成・管理"/>
  <narrow r:resource="Top/World/Japanese/ コンピュータ/ソフトウェア/インターネット/コミュニ
ケーション"/>
</Topic>
<Topic r:id="Top/World/Japanese/ コンピュータ/ソフトウェア/インターネット/コミュニケーション">
  <catid>510586</catid>
  <d:Title>コミュニケーション</d:Title>
  <narrow r:resource="Top/World/Japanese/ コンピュータ/ソフトウェア/インターネット/コミュニ
ケーション/Odigo"/>
  <narrow r:resource="Top/World/Japanese/ コンピュータ/ソフトウェア/インターネット/コミュニ
ケーション/AOL インスタント・メッセンジャー"/>
  <narrow r:resource="Top/World/Japanese/ コンピュータ/ソフトウェア/インターネット/コミュニ
ケーション/ICQ"/>
  <narrow r:resource="Top/World/Japanese/ コンピュータ/ソフトウェア/インターネット/コミュニ
ケーション/MSN_Messenger"/>
</Topic>
<Topic r:id="Top/World/Japanese/ コンピュータ/ソフトウェア/インターネット/コミュニケーション
/Odigo">
  <catid>550024</catid>
  <d:Title>Odigo</d:Title>
</Topic>
<Topic r:id="Top/World/Japanese/ コンピュータ/ソフトウェア/インターネット/コミュニケーション
/AOL インスタント・メッセンジャー">
  <catid>550027</catid>
  <d:Title>AOL インスタント・メッセンジャー</d:Title>
</Topic>
<Topic r:id="Top/World/Japanese/ コンピュータ/ソフトウェア/インターネット/コミュニケーション
/ICQ">
  <catid>550138</catid>
  <d:Title>ICQ</d:Title>
</Topic>
<Topic r:id="Top/World/Japanese/ コンピュータ/ソフトウェア/インターネット/コミュニケーション
/MSN_Messenger">
  <catid>550286</catid>
  <d:Title>MSN_Messenger</d:Title>
</Topic>
...

```

図 2.3 対象とした用語体系のデータ例：ODP

ムの 2 つから構成される（図 2.4 参照）。可視化の表示形式としては、グラフ構造のクリックブルマップ表示や HTML による一覧表示がある。さらに、表示された見出し語はリンクとなっており、次々とブラウザ上で各用語の持つ関係を開覧していくことが可能である。これらの表示は、各用語体系および複数の用語体系の重ね合わせ結果のどちらにも適用可能である。

また、各用語とそれらの持つ関係を効果的にブラウジングするためには、対象データ全てを一度

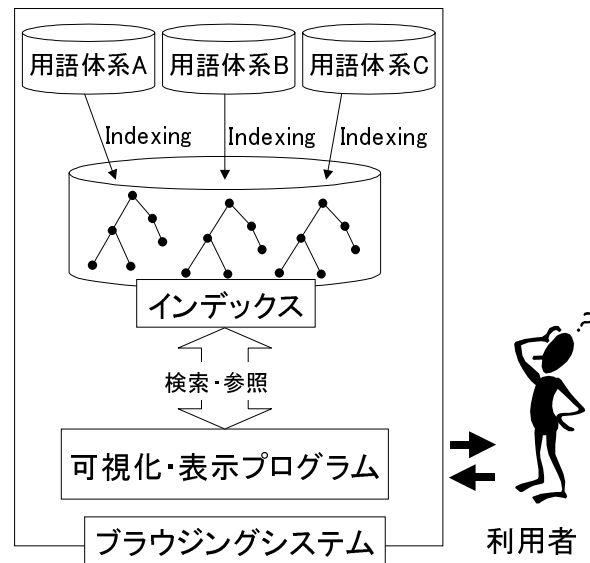


図 2.4 システムの構成

に表示するよりも、任意の見出し語を中心として、一定の範囲に表示を限定する必要があると考えた。そこで、利用者が指定した見出し語と、その語の上位語・下位語から数階層たどれる範囲までの用語を表示することとした。このうち、上位語・下位語についてそれぞれ、どの範囲の階層まで表示するかを利用者が指定できるようにした。

また、複数用語体系の重ね合わせに関しても重ね合わせたい体系の組み合わせをユーザが自由に指定できるようにした。例えば、 n 個の体系がある場合には、

$$\sum_{i=1}^n {}_nC_i$$

通りの組み合わせを指定できる。

また、各用語体系もしくは重ね合わせ結果を再利用するための XML データを取得することもできる。

また、表示したい見出し語を選択するための機能には、見出し語検索（完全一致、部分一致）と最上位階層の見出し語一覧を表示する機能がある。

なお、ユーザが本システムを利用するためには、Web ブラウザさえあればよく、その他の特別なソフトウェアは必要としない。

2.4.2 インデックスの作成

本システムでは、複数の用語体系間での重ね合わせの組み合わせを任意に指定できるようにした。そのため、体系間の重ね合わせについては、結果をあらかじめ保持しておくのではなく、重ね

合わせの対象となる用語体系の組み合わせを指定したブラウジングを利用者が要求するたびに動的に行うこととした。この動的な重ね合わせ処理を高速化するために、1) 階層関係インデックス、2) 見出し語検索用インデックスの2つのインデックスを作成した。

```
%WORDS      = ("510586" => 'コミュニケーション');  
%CHILDREN   = ("510586" => '550024|550027|550138|550286');  
%PARENTS    = ("510586" => '510586');
```

図 2.5 階層関係インデックスの Perl での表現例

階層関係インデックスは、見出し語 ID をキーとし、1) 見出し語、2) 上位語 ID の集合、3) 下位語 ID の集合をそれぞれ値に持つ3つの Perl のハッシュファイルとして作成した。図 2.3 のデータに対して作成されるハッシュ表現を図 2.5 に例示する。

見出し語検索用インデックスは、見出し語を対象とした Suffix Array インデックスとして作成した。

Intel PentiumIII 550MHz × 4, メモリ 2 GB のマシンでインデックスの作成に要する時間 (10 回の試行の平均) をそれぞれ以下に示す。

- EDR: 358 秒
- NDC: 33 秒
- EIC: 0.4 秒
- ODP: 954 秒
- ODP-kt: 2.1 秒

どの用語体系においても、十分に実用的な時間内でインデックスが作成できる。

これらのインデックスを作成することによって、利用者からのブラウジングのリクエストがあるたびに、高々 1 秒程度の実時間で、任意の見出し語の持つ階層関係を動的に生成することが可能になった。

2.4.3 可視化・表示プログラム

可視化・表示プログラムにおけるデータの流れを図 2.6 に示す。可視化を行う CGI プログラムでは、まず、2.4.2 節で述べたインデックスを参照して動的にある語を表わす XML データを生成する。次に、利用者が要求した表示形式にしたがって、その XML を元に XSLT スタイルシート [24] による変換を行い、提供するようにした。また、データの再利用のために、XSLT による変換をせず、XML データそのものを取得することもできる。

図 2.7 に用語体系 NDC の見出し語「図書館教育」について動的に生成した XML 表現を示す。

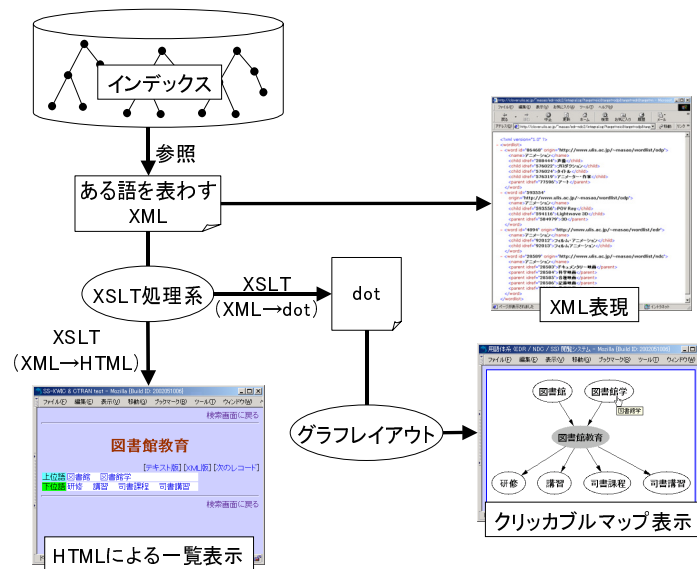


図 2.6 可視化・表示プログラムのデータの流れ

```
<?xml version="1.0"?>
<wordlist>
  <word id="134">
    <name>図書館教育</name>
    <child idref="135">研修</child>
    <child idref="136">講習</child>
    <child idref="137">司書課程</child>
    <child idref="138">司書講習</child>
    <parent idref="127">図書館</parent>
    <parent idref="128">図書館学</parent>
  </word>
</wordlist>
```

図 2.7 動的に生成した XML (NDC) : 「図書館教育」

XML 表現では、一用語の情報を<word>要素として表現し、id 属性にその見出し語を示す ID を記録し、<word>要素中の<name>要素で見出し語を表現した。<word>要素中の<parent>、<child>要素では、その見出し語のそれぞれ上位語、下位語を表現し、idref 属性にその上位語・下位語の ID を表現した。

さらに、図 2.8 に EDR、NDC、ODP の 3 つの用語体系に存在する見出し語「アニメーション」について重ね合わせた結果として動的に生成した XML を示す。XML の構造自体は図 2.7 の一体系での XML 表現とほぼ同一であるが、重ね合わせ結果における XML 表現では、<word>要素に origin 属性を加えて、その見出し語がどの用語体系に由来するものであるかを表現した。

表示形式に HTML を指定した場合は、この XML に対して HTML 生成用の XSLT スタイルシー

```

<?xml version="1.0"?>
<wordlist>
<word id="86468" origin="http://www.ulis.ac.jp/~masao/wordlist/odp">
  <name>アニメーション</name>
  <child idref="288444">声優</child>
  <child idref="576022">プロダクション</child>
  <child idref="576024">タイトル</child>
  <child idref="576319">アニメーター・作家</child>
  <parent idref="77596">アート</parent>
</word>
<word id="593554" origin="http://www.ulis.ac.jp/~masao/wordlist/odp">
  <name>アニメーション</name>
  <child idref="593556">POV Ray</child>
  <child idref="594116">Lightwave 3D</child>
  <parent idref="584979">3D</parent>
</word>
<word id="4894" origin="http://www.ulis.ac.jp/~masao/wordlist/edr">
  <name>アニメーション</name>
  <child idref="92012">フィルム・アニメーション</child>
  <child idref="92013">フィルムアニメーション</child>
</word>
<word id="28589" origin="http://www.ulis.ac.jp/~masao/wordlist/ndc">
  <name>アニメーション</name>
  <parent idref="28583">ドキュメンタリー映画</parent>
  <parent idref="28584">科学映画</parent>
  <parent idref="28585">各種映画</parent>
  <parent idref="28586">記録映画</parent>
  <parent idref="28587">教育映画</parent>
</word>
</wordlist>

```

図 2.8 動的に生成した XML (重ね合わせ結果): 「アニメーション」

```

<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:output method="text"/>
  <xsl:template match="/">
    digraph G {
      <xsl:apply-templates/>
    }
  </xsl:template>
  <xsl:template match="word">
    "<xsl:value-of select="@id"/>"
    [label="<xsl:value-of select="name"/>", style="filled", color="gray"];
    <xsl:for-each select="child">
      "<xsl:value-of select="@idref"/>" [label="<xsl:value-of select="."/>"];
      "<xsl:value-of select="..@id"/>" -&gt; "<xsl:value-of select="@idref"/>";
    </xsl:for-each>
    <xsl:for-each select="parent">
      "<xsl:value-of select="@idref"/>" [label="<xsl:value-of select="."/>"];
      "<xsl:value-of select="@idref"/>" -&gt; "<xsl:value-of select="..@id"/>";
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>

```

図 2.9 dot 形式に変換する XSLT スタイルシート例

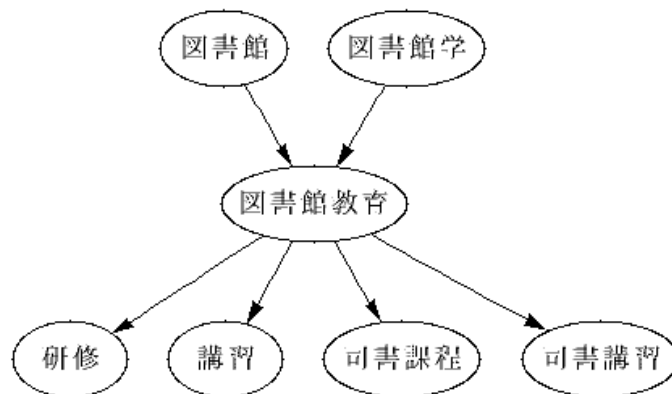


図 2.10 動的に生成した画像の例

トで変換した結果を返す。また、表示形式にクリッカブルマップや GIF 画像を指定した場合は、クリッカブルマップ生成用の XSLT スタイルシートで、グラフ表現の dot 形式 [25] に変換し、その結果を dot コマンドでレイアウトした結果を返す。図 2.9 に XML 表現を dot 形式に変換する XSLT スタイルシートの例を示し、図 2.10 には Graphviz [25] でグラフィレイアウトされた画像例を示す。

なお、動的に生成する XML の形式は、表示形式には依存していないため、新たな表示形式に容易に対応可能である。さらに、これらの XML 形式をそのまま提供することで、用語体系データの交換や再利用が可能となる。

2.4.4 開発環境

本システムは、CGI の枠組みを利用して Web 上でユーザの要求に応じて動的に用語体系を可視化、閲覧できるシステムとして構築した。開発に利用した環境は以下の通りである。Web サーバには Apache [26] を用い、CGI プログラムは Perl5 [27] で作成した。用語体系の検索などを高速に行うために Berkeley DB ライブラリ [28] および Suffix Array ライブラリ Sary [29] を利用したインデックスを作成した。グラフ構造のレイアウトおよび画像出力は、グラフ描画ソフトウェア Graphviz [25] を用いた。また、XML から各種表示形式への変換を行う XSLT 処理系には Libxslt [30] を利用した。

2.4.5 実行例

本節では、各機能を例とともに説明する。

HTML による一覧表示機能は、各見出し語の上位語と下位語を表示する機能である。表示された上位語と下位語はハイパーリンクになっており、ブラウザ上で語をクリックすることで、指定し

た見出し語の表示を次々で行うことができる。図 2.11 は NDC における用語「図書館」の持つ関係を表示しており、この中から「図書館活動」を選んでいところである。クリックすると図 2.12 のように「図書館活動」の上位語と下位語を見ることができる。

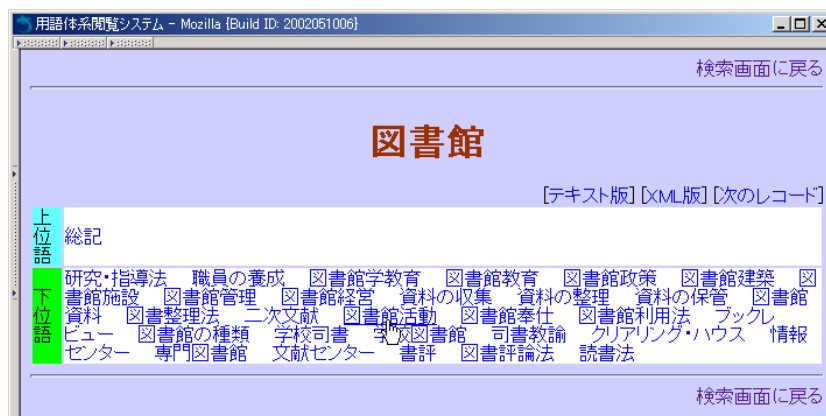


図 2.11 HTML による一覧表示: 「図書館」(NDC)

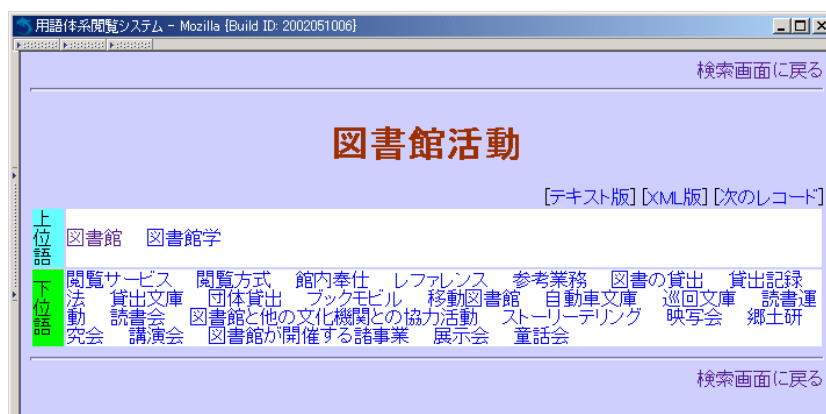


図 2.12 HTML による一覧表示: 「図書館活動」(NDC)

各用語体系のグラフ構造のクリッカブルマップ表示機能は、各用語体系における見出し語をノードとし、上位階層の語を上、下位階層の語を下に配置したグラフ表現を表示する機能である。表示された上位語と下位語のノードはハイパーリンクになっており、クリックすることで、次々と指定した見出し語の表示を行うことができる。図 2.13 は「図書館」を表示しており、この中から「図書館活動」を選んでいところである。このノードをクリックすると図 2.14 のように「図書館活動」についての上位語と下位語を見ることができる。

また、指定した語から上下に何階層まで表示するかも指定できる。図 2.15 は、ODP における見出し語「コンピュータ」からたどることができる 3 階層分の見出し語を同時に表示した例である。

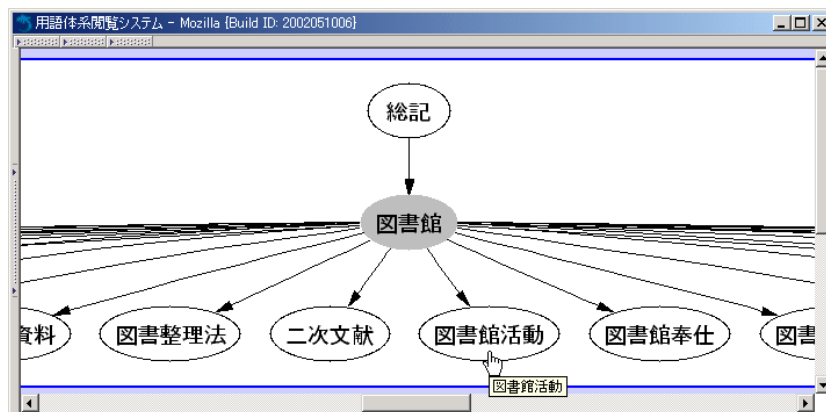


図 2.13 グラフ構造のクリックابلマップ表示: 「図書館」(NDC)

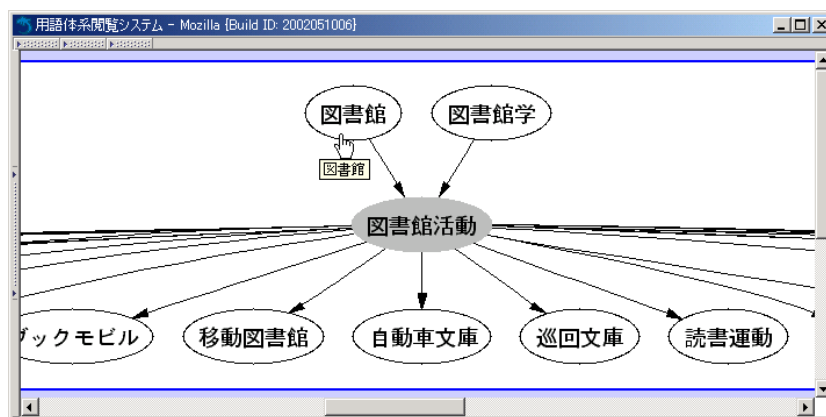


図 2.14 グラフ構造のクリックابلマップ表示: 「図書館活動」(NDC)

この複数階層の同時表示機能により、表示する範囲を広げて眺めることが可能になり、全体像の把握がより容易となる。

複数用語体系の重ね合わせ結果の表示では、異なる用語体系の関係が一目でわかるように、EDR は青、NDC は赤、ODP はオレンジ色というように、各用語体系での関係を表すエッジはそれぞれ別の色として表示している。また、一体系中で別の ID を振られている見出し語も、重ね合わされて表示される。図 2.16 は、複数の用語体系の重ね合わせ表示の例であり、図 2.8 と同じく EDR、NDC、ODP の 3 つの用語体系に存在する見出し語「アニメーション」の重ね合わせ結果に対して、グラフ構造のクリックابلマップ表示を行っている。ここで、見出し語「アニメーション」は、

- EDR における「フィルムアニメーション」の上位語
- NDC における各種映画の下位語
- ODP における「アート」の下位語

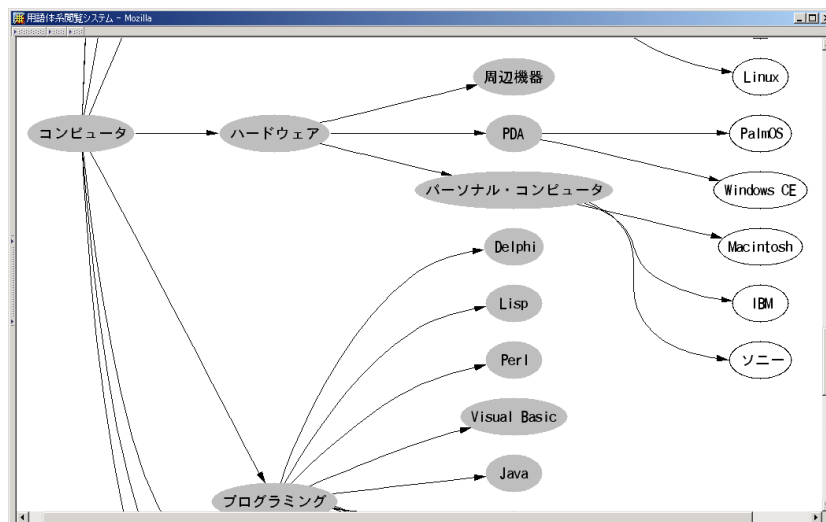


図 2.15 グラフ構造のクリックابلマップ表示（複数階層同時表示）：「コンピュータ」(ODP)

- ODP における「3D グラフィックス・ソフトウェア」の下位語

という、複数の体系における階層構造上の見出し語を重ね合わせたものである。このように重ね合わせを行うことで、その見出し語が持つ関係をより広い観点から一望できる。グラフ構造を表示する際には、その見出し語同士を結ぶエッジの色を変えることで、それぞれがどの用語体系に由来する関係であるかが直観的に判別できる。

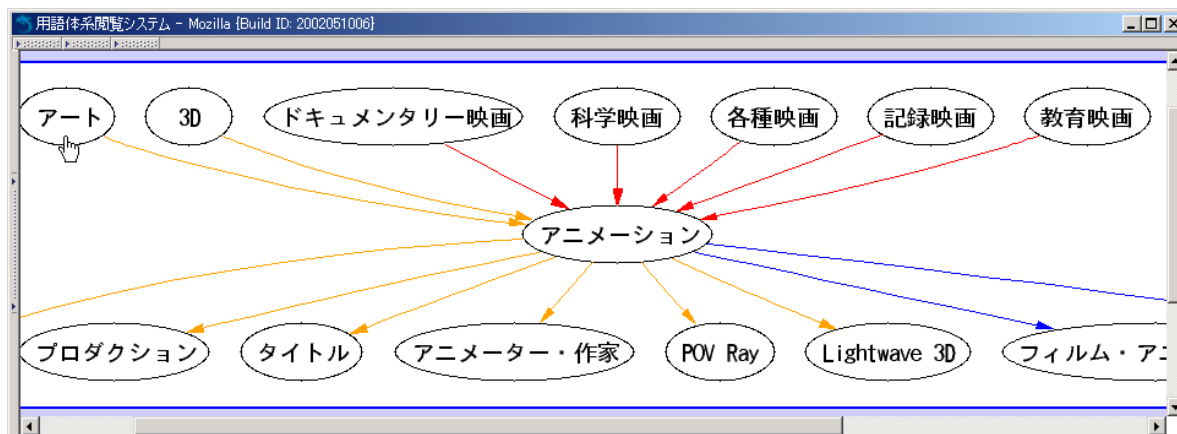


図 2.16 複数用語体系の重ね合わせ結果：「アニメーション」

また、見出し語検索機能では、各用語体系の見出し語を検索できる。見出し語の検索は、検索したい用語体系を複数選択することで、複数の用語体系を同時に横断的に検索できる。この見出し語検索では、入力された単語と完全一致する見出し語を表示するだけでなく、部分一致した見出し語

も表示する。図 2.17 は「エネルギー」に対する検索結果である。検索結果は、完全一致した語、部分一致した語の順で表示し、ヒットした見出し語数と各見出し語毎に、その語の属する用語体系、見出し語 ID、見出し語、見出し語の階層関係の表示へのハイパーリンクを表示する。ここでは、EDR、NDC、EIC、ODP の 4 つの用語体系に一致する見出し語があることがわかる。また、部分一致した検索語については、一致した部分をわかりやすいように赤で強調表示している。この中から見出し語を選択してクリックすると、グラフ構造のクリッカブルマップ表示機能へ飛ぶことができる。



The screenshot shows a web browser window with the title bar '「エネルギー」にヒットした見出し語リスト - Mozilla [Build ID: 2002061006]'. The main content area has a light blue background and is titled '「エネルギー」にヒットした見出し語リスト'. It is divided into two sections: '完全一致:' and '部分一致:'. The '完全一致:' section lists 7 hits, and the '部分一致:' section lists 127 hits. Each section contains a table with columns for '用語体系', 'ID', '見出し語', '統合表示', and 'その他の表示形式'.

用語体系	ID	見出し語	統合表示	その他の表示形式
EDR	10838:	エネルギー	*	[Text] [XML] [HTML] [GIF] [Dot]
NDC	17011:	エネルギー	*	[Text] [XML] [HTML] [GIF] [Dot]
ODP	216534:	エネルギー	*	[Text] [XML] [HTML] [GIF] [Dot]
EICnet	18:	エネルギー	*	[Text] [XML] [HTML] [GIF] [Dot]
NDC	13010:	エネルギー	*	[Text] [XML] [HTML] [GIF] [Dot]
NDC	787:	エネルギー	*	[Text] [XML] [HTML] [GIF] [Dot]
NDC	11321:	エネルギー	*	[Text] [XML] [HTML] [GIF] [Dot]

用語体系	ID	見出し語	統合表示	その他の表示形式
EDR	2712:	アークエネルギー	*	[Text] [XML] [HTML] [GIF] [Dot]
EDR	5985:	イオン化エネルギー	*	[Text] [XML] [HTML] [GIF] [Dot]
EDR	10839:	エネルギー材料-情報相互依存性	*	[Text] [XML] [HTML] [GIF] [Dot]
EDR	10841:	エネルギーアナライザ	*	[Text] [XML] [HTML] [GIF] [Dot]
EDR	10843:	エネルギーアナライザ	*	[Text] [XML] [HTML] [GIF] [Dot]
EDR	10854:	エネルギーギャップ	*	[Text] [XML] [HTML] [GIF] [Dot]
EDR	10884:	エネルギーデータベース	*	[Text] [XML] [HTML] [GIF] [Dot]
EDR	10904:	エネルギーデータベース	*	[Text] [XML] [HTML] [GIF] [Dot]

図 2.17 見出し語検索機能（検索結果）：「エネルギー」

2.5 考察

本研究では、用語体系の作成、維持を支援するブラウジングシステムを構築した。本節では、本研究における用語体系の重ね合わせとブラウジングという 2 つの観点から、既存の関連研究との相違点と用語体系の拡張について述べる。

2.5.1 関連研究

用語体系閲覧システムの既存の研究には、後藤ら [31] や市丸 [32] によるシソーラスや用語辞書の概念体系をグラフィカルに表示し、ブラウジングするシステムがある。RDFViz [33] は、RDF データのグラフ構造を表示するシステムで、Web 上から利用できる。また、情報可視化に関する研究では、Pirolli ら [34] による階層構造を Focus+Context 手法を用いて可視化するシステムなどが

開発されている。これらのシステムはいずれも、個別の体系を表示して閲覧する機能を持つが、本研究のように、複数の体系を重ね合わせて表示する機能はない。

また、複数のオントロジーを自動的に統合する試み [35][36][37] や複数言語の体系を統合する試み [38][39] も行われている。しかし、これらのアプローチでは、統一された一つの体系を構築することに主眼が置かれていて、複数体系を重ね合わせてその結果をブラウジングしながら取捨選択したり、再利用したりするという本研究のような観点からの言及はされていない。

一方、Noy ら [40] は利用者との対話によりオントロジーのマージを半自動化する手法を提案している。この手法に基づくシステムは、オントロジー構築用エディタ Protégé [41] 上で、PROMPT プラグイン [42] として実現されている。しかし、PROMPT は 2 つのオントロジーのマージを扱うもので、本研究のように同時に 3 つ以上の体系を重ね合わせることはできない。

2.5.2 用語体系の拡張

本研究で構築したブラウジングシステムの枠組みを使えば、複数の重ね合わせ手法を切り替えることも原理的には可能であると考えられる。例えば、用語の照合に依らず、異なる言語の体系同士を重ね合わせできる手法などが考えられる。Sowa は文献 [43] において、このような手法をアラインメント (alignment) と呼び、「オントロジー間での概念および関係のマッピング」と定義としている。本システムにおいて様々な体系同士を重ね合わせる際に、その重ね合わせ手法を拡張したり、このアラインメントの考え方にもとづく手法を切り替えて使うことで、より一層、複数体系間の相互利用が強まるものとする。

また、ブラウジングシステムが動的に生成する XML の形式は、表示形式には依存していないため、新たな表示形式に容易に対応可能である。例えば、VRML (Virtual Reality Modeling Language) による 3 次元表示やグラフ表現のカスタマイズなどができると考える。

さらに、この XML 形式をそのまま提供することで、用語体系データの交換や再利用が可能となる。例えば、一部の用語の構造をそのまま取り出して自分の体系に組み込むことなどが可能になるものとする。

2.5.3 用語体系構築への重ね合わせ手法の活用例

重ね合わせ結果を用語の構築に利用する方法にはいくつかの方法がある。以下では 2 つの例を挙げ、その具体例について考察する。

まず第 1 の例では特定分野に特化した用語体系に対して、汎用の用語体系の関係を重ね合わせることの意義について述べる。図 2.18 は、EIC と ODP の 2 つの用語体系にある見出し語「エネルギー」を重ね合わせた結果である。このうち、エネルギー { 省エネルギー, 自然エネルギー, 燃料電池, バイオマス, 原子力, その他 } の階層関係は EIC に由来する関係であり、ビジネス エネル

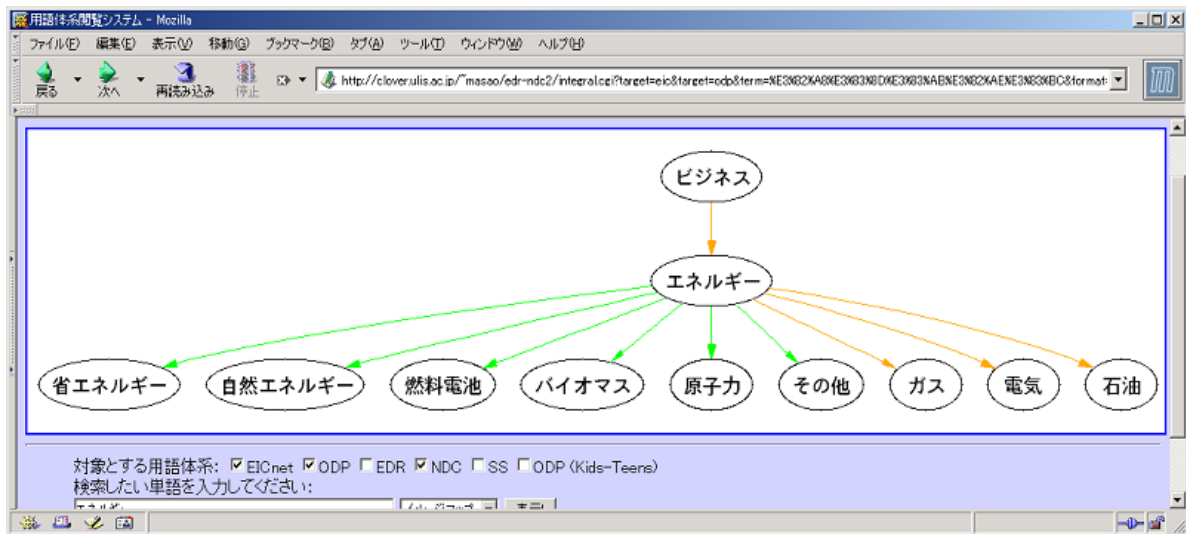


図 2.18 重ね合わせ結果 (EIC+ODP) : 「エネルギー」

ギーおよびエネルギー { ガス, 電気, 石油 } は ODP に由来する関係である。EIC は環境という特定分野に特化した用語体系であり、その専門的なトピックに焦点を当てて体系を構築している。一方で、EIC ネットは環境分野におけるポータルサイトであり、その情報提供の対象者は専門家だけでなく広く一般をも対象としているため、一般利用者を対象とした情報提供も重要となる。そのため、全分野を対象とし汎用の用語体系である ODP が持っているビジネス エネルギー { ガス, 電気, 石油 } という関係を、環境に特化した用語体系 EIC に取り込むことにより、不足している観点の補完を行うことは有効である。

次に第 2 の例では、新しく用語体系を構築する際の重ね合わせ手法の利用例について述べる。例えば、図 2.16 は、3 つの用語体系にある「アニメーション」という見出し語についての重ね合わせ結果である。図中にある主な関係としては、映画関連分野としての「アニメーション」、「アニメーション」ソフトウェア、「アニメーション」作品 (タイトル、作家など) という見出し語に関連する属性が見てとれる。このようにいくつかの体系にまたがって重ね合わせを行うことにより、その見出し語やその見出し語に関連する分野におけるいくつかの観点を眺めることができる。これにより語彙を収集したり分析することができ、新しく情報提供システムを立ち上げる際に必要な観点を集められる。例えば、アニメーション分野のポータルサイトを新たに構築しようとした情報提供者が、図 2.16 にある関係を元に、情報提供に必要な観点を収集したり整理することができる。例えば、アニメーション作品の情報を提供する場合にも、その作品情報に必要な属性を集めたり、また作品情報だけでなく、アニメーション作成に関わるソフトウェア情報といった項目を設けることもできる。

2.6 まとめ

用語体系の再利用による作成支援を目指して複数用語体系のブラウジングシステムを構築した。分野や規模の異なる 5 つの用語体系を対象として、ブラウジングシステムがこれらに適用可能なことを示した。また、用語間の関係を分かりやすく表示し、その体系の構造を直観的に把握できるようになった。また、複数用語体系を重ね合わせて表示することで、用語の持つ様々な関係をとらえることができるようになり、それらの関係を新たな体系の構築に再利用することができる。

このシステムをさらに発展させることにより、様々な用語体系間での再利用が可能となり、様々な分野の用語体系構築に寄与するものと考えている。

第 3 章

Web サービスによる用語体系データの提供とその応用システム

3.1 背景と目的

概念間の関係や用語の意味を記述した用語体系は、情報検索や自然言語処理などの分野で広く利用されている。近年では、シソーラスや辞書、分類表、Web ディレクトリなどの様々な用途で用語体系が作成され、利用に供されている。例えば、自然言語処理用辞書として開発された EDR [22]、Web 上の情報を網羅する分類体系 ODP (Open Directory Project) [16]、図書の分類を目的とした日本十進分類表 (NDC) [44] などがある。また、大規模に作成されたもの以外にも、専門機関や研究者・個人などによる用語集が規模・分野を問わず公開されている。Web 上では、たとえば、環境関係の分野の用語集 [17]、建築工事標準分類 [45] など、多種多様な用語体系データが数多く公開されており、これら用語体系を利用できる機会が増している。Web サイトで情報を提供する際にも、分類やキーワードなど一種の用語体系を用いて情報を組織化して提供することが広く行われており、用語体系は Web 上における情報提供には欠かせない存在になりつつある。また、用語体系は元々、それぞれの組織や分野ごとに独立して作成されており、Web 上で提供される際にも、作成者それぞれが独自の形式で提供している。

また、用語体系を作成、保守する際には、他の用語体系を参照したり、再利用したりすることが有効である。例えば、網羅的な用語体系を作成したい場合には、既存の特定分野向けに作られた構造の一部を参照し、その構造を再利用したり、その逆に、特定分野向けの体系を構築する場合には、既存の汎用的な用語体系を参照し、その構造を周縁領域の構造として再利用したりすることができる。しかし、このような再利用を効率良く行うには、複数の用語体系データ間を連携するための枠組みが必要となる。

さらに、用語体系データは基礎的な学術情報資源であり、検索システムなど他のシステムとの連携も重要な観点である。用語体系と他のシステムとを組み合わせる場合には、用語体系自体にアク

セスするための統一的なインタフェースが必要になってくる。

これらの要求に対応しうるものとしては、分散的な情報検索プロトコルである Z39.50 のシソーラス・ナビゲーションに関するプロファイル Zthes [46] があるが、他システムとの連携、親和性の点で十分とは言い難い。

一方、他のシステムとの連携の枠組みとして、最近 Web サービス (Web Services) [13][14] が注目されている。これは、XML により定義、記述されたメッセージをインターネット (特に Web) 上でやり取りすることにより、連携して動作可能なソフトウェアシステムを指す。Web サービスは、元来、電子商取引、特に B2B (Business to Business) などにおける企業データなどでの活用が有望視され、標準化が進められた。我々は、用語体系データのような基礎的な学術情報を提供するシステムでも、Web サービスの枠組みを利用できるものと考え、本研究においてそのシステムを構築した。

筆者は、これまで、複数用語体系をブラウジングするシステムを構築し、用語体系データを Web 上で提供する研究 [12][21] を行ってきた (2 章参照)。これらの研究では、用語体系をサーバサイドで一元的に管理し、ユーザに提供するシステムを構築した。しかし、この手法では、独立して構築される性質を持つ用語体系を柔軟に提供することが難しく、他のシステムとの連携も容易には行えないという問題があった。

そこで、筆者は、用語体系データの提供において、ネットワーク上で独立した用語体系提供システムを構築し、それらの用語体系に対して統一的なアクセス手法を提供するため、Web サービスの枠組みを利用した用語体系の提供システムを考案し、2 つの異なる用語体系を対象としたシステムをそれぞれ構築した。加えて、これらの用語体系提供 Web サービスシステムの応用例として、Web サービスをバックエンドとして利用する 2 種類のフロントエンドシステムを構築した。

3.2 Web サービス

Web サービス (Web Services) とは、一般に「XML により定義、記述されたインタフェースを持ち、インターネット上で XML メッセージをやり取りする、URI により識別可能なソフトウェア部品」を指し、Web 上で分散オブジェクト技術である RPC (Remote Procedure Call) を実現する手法として注目されている。

Web サービスでは、サーバとクライアントとの間のやりとりは、定められた XML メッセージを通じて行われる。サービス側では、クライアントからの要求である XML メッセージを受け取ると、そのメッセージに従った処理を適宜行い、XML メッセージでの返答を行う。これらの XML メッセージで用いられる、データの型やインタフェースの定義は WSDL 仕様 [47] に基づいて定義する。また、ネットワーク上を XML メッセージが送受信される際のやりとりは、SOAP 仕様 [48] に従って行われる。

3.3 用語体系データ

用語体系とは、見出し語とその語の持つ関係を記述したものである。用語の持つ関係には様々なものがあるが、本論文では、用語間の階層関係、つまり上位語・下位語の関係のみを扱うこととし、用語体系を、見出し語をノードとし、見出し語間の階層関係をエッジとする有向グラフ構造と見る。

また、用語体系における見出し語には、同一の表記であっても意味の異なる用語が存在するものもあるため、用語体系中で一意となる見出し語 ID が振られているものとする。

本研究で用いたデータは以下の 2 つである（表 3.1 参照）。

1. EDR 専門用語辞書（情報処理）中の名詞・概念体系辞書（以下 EDR）[22]
2. Open Directory Project [16] による Web ディレクトリの階層構造データ（以下 ODP）

表 3.1 対象とした各用語体系の概要

用語体系	用語数	階層関係数	特徴
EDR	117,242	184,958	情報処理分野の専門用語を対象とした大規模な体系
ODP	446,926	446,928	Web 上に存在する情報全てを対象とした大規模な体系

これらは、我々が以前に作成したデータ [12] であり、XML で記述してある。

3.4 Web サービスを用いた用語体系データ提供

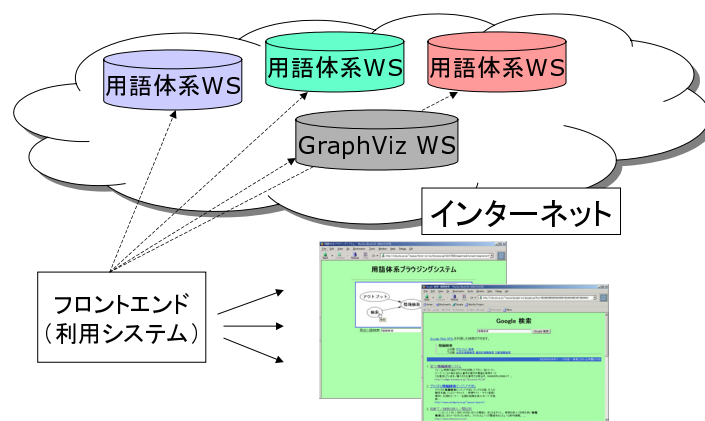


図 3.1 用語体系 Web サービス・概念図

図 3.1 に Web サービスを用いた用語体系データ提供の概念図を示す。図中の「用語体系 WS」と

は、用語体系データを提供する Web サービスを指し、「GraphViz WS」はグラフィケアウト用の Web サービスを指す。

これらの用語体系 Web サービスは、それぞれ一つのサービスが一つの用語体系に対応することとし、見出し語検索や見出し語が持つ構造の取得を行うインタフェースを持つ。これらは独立した別個のシステムとして構築されるため、それぞれの組織や分野ごとに各用語体系の作成者がインターネット上で独立して構築し、維持・管理できる。フロントエンド（利用システム）では、その用途に合わせて一つもしくは複数の用語体系 Web サービスを利用して処理を行う。フロントエンドの環境は問わないことが、Web サービスの特性であるため、幅広い環境で用語体系データを利用できるようになる。

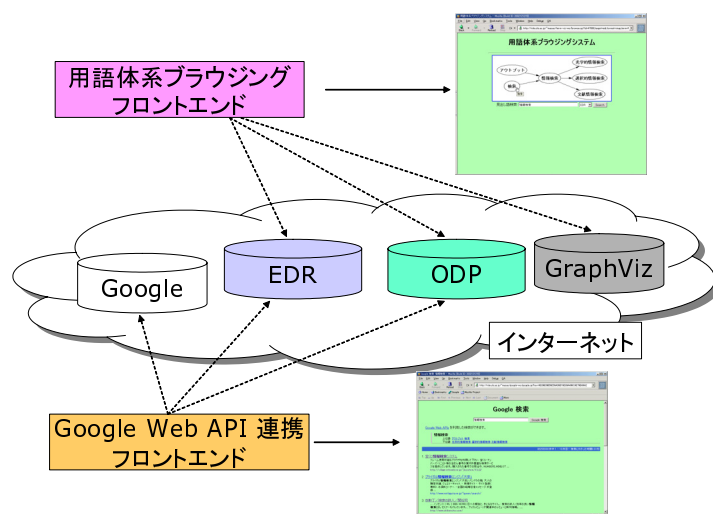


図 3.2 開発したシステムの概要

本研究で構築したシステムの概要を図 3.2 に示す。用語体系提供 Web サービスの一例として EDR、ODP を対象としたシステムを構築し、グラフィケアウト用の Web サービス（GraphViz Web サービス）を構築した。そして、これら 3 つの Web サービスと連携して動作する用語体系ブラウジングフロントエンドを構築した。また、Google 社が提供する Web サービス Google Web API と EDR、ODP を対象とした用語体系 Web サービスとを連携させた、Google Web API 連携フロントエンドを構築した。各システムの詳細については、以下の節でそれぞれ述べる。

3.4.1 Web サービスシステム

本研究では、用語体系データ Web サービスと GraphViz Web サービスの 2 種類の Web サービスを構築した。

用語体系データ Web サービス

3.3 節で述べた、2 つの用語体系を対象とした Web サービスによる用語体系データ提供システムを構築した。

本研究では、複数の用語体系データを連携したシステムを提供できるようにするため、一つの用語体系を一つの Web サービスに対応させることによって、複数の用語体系 Web サービスを独立して提供できるようにした。

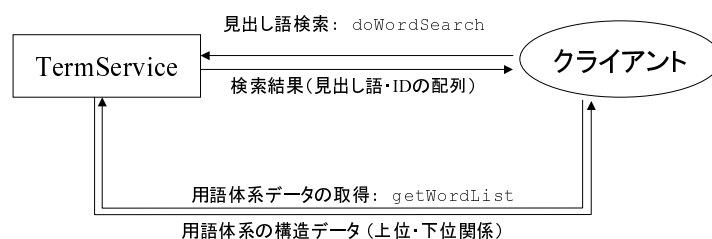


図 3.3 用語体系 Web サービスのインタフェース

用語体系データ提供 Web サービスにおける呼出し可能なサービスには、以下の 2 つがある（図 3.3）。

- doWordSearch: 見出し語検索を行う
- getWordList: 特定の見出し語の構造を取得する

doWordSearch は見出し語検索を行うサービスである。クライアント側から、検索語となる文字列を引数として渡すと、サービス側では、見出し語に対して文字列検索を行って、検索結果として、完全一致および部分一致した見出し語についてそれぞれ、見出し語と見出し語 ID から成る配列を返す。例えば、「情報検索」を検索語として見出し語検索を行う場合の XML メッセージを図 3.4 に示す。検索語は<term>要素の内容として指定されて送られる。

```
<?xml version="1.0" encoding="utf-8" ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <n1:doWordSearch xmlns:n1="urn:Term">
      <term>情報検索</term>
    </n1:doWordSearch>
  </env:Body>
</env:Envelope>
```

図 3.4 doWordSearch におけるリクエスト例

用語体系 Web サービス側では、予め構築しておいた用語インデックスから見出し語を検索し、完全一致および部分一致する見出し語の一覧をクライアントに返す。図 3.5 に、用語体系 EDR に

```

<?xml version="1.0" encoding="utf-8" ?>
<env:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <n2:doWordSearchResponse
      xmlns:n1="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:n2="urn:Term"
      env:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <return xsi:type="n2:WordSearchResult">
        <exactMatchElements xsi:type="n1:Array" n1:arrayType="n2:Node[1]">
          <item>
            <name xsi:type="xsd:string">情報検索</name>
            <idref xsi:type="xsd:string">47888</idref>
          </item>
        </exactMatchElements>
        <substrMatchElements xsi:type="n1:Array" n1:arrayType="n2:Node[44]">
          <item>
            <name xsi:type="xsd:string">光学の情報検索</name>
            <idref xsi:type="xsd:string">30645</idref>
          </item>
          <item>
            <name xsi:type="xsd:string">選択の情報検索</name>
            <idref xsi:type="xsd:string">55142</idref>
          </item>
          (...中略...)
          <item>
            <name xsi:type="xsd:string">文献情報検索言語</name>
            <idref xsi:type="xsd:string">95639</idref>
          </item>
        </substrMatchElements>
      </return>
    </n2:doWordSearchResponse>
  </env:Body>
</env:Envelope>

```

図 3.5 doWordSearch におけるレスポンス例（抜粋）

おける、図 3.4 のリクエスト（検索語は「情報検索」）に対応する応答の抜粋を示す。完全一致する見出し語の一覧と部分一致する見出し語の一覧を、それぞれ<substrMatchElements>要素、<exactMatchElements>要素として表現する。これらの要素は、配列構造として規定されており、配列の要素となる個々の見出し語に関する情報は、<item>要素中の<name>要素には見出し語表記が、同じく<item>要素中の<idref>要素には見出し語 ID が表現されている。

getWordList は特定の見出し語の構造を取得するためのサービスである。クライアント側から、ある見出し語の ID を引数として渡すと、サービス側では、その ID の用語が持つ階層関係データをクライアントに返す。例えば、見出し語 ID 「47888」の用語についての階層関係を要求する場合は、クライアント側からの XML メッセージとして、図 3.6 に示すリクエストが送信される。見出し語 ID は<id>要素の内容として指定される。

クライアント側から見出し語 ID が引数として渡されると、サービス側では、その ID の用語が持つ階層関係データを用語インデックスから取得し、その構造をクライアントに返す（図 3.7）。一用語の持つ構造を表現する XML 構造は、以下の各要素からなる。

- <origin>要素: どの用語体系における見出し語かを示す ID（URI 形式）
- <idref>要素: 見出し語 ID
- <name>要素: 見出し語

```

<?xml version="1.0" encoding="utf-8" ?>
<env:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <n2:getWordList xmlns:n1="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:n2="urn:Term"
      env:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <id xsi:type="xsd:string">47888</id>
    </n2:getWordList>
  </env:Body>
</env:Envelope>

```

図 3.6 getWordList におけるリクエスト例

```

<?xml version="1.0" encoding="utf-8" ?>
<env:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <n2:getWordListResponse xmlns:n1="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:n2="urn:Term"
      env:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <return xsi:type="n1:Array" n1:arrayType="n2:Word[1]">
        <item>
          <origin xsi:type="xsd:string"></origin>
          <parent xsi:type="n1:Array" n1:arrayType="n2:Node[2]">
            <item>
              <name xsi:type="xsd:string">アウトプット</name>
              <idref xsi:type="xsd:string">2913</idref>
            </item>
            <item>
              <name xsi:type="xsd:string">検索</name>
              <idref xsi:type="xsd:string">29007</idref>
            </item>
          </parent>
          <name xsi:type="xsd:string">情報検索</name>
          <child xsi:type="n1:Array" n1:arrayType="n2:Node[3]">
            <item>
              <name xsi:type="xsd:string">光学の情報検索</name>
              <idref xsi:type="xsd:string">30645</idref>
            </item>
            <item>
              <name xsi:type="xsd:string">選択の情報検索</name>
              <idref xsi:type="xsd:string">55142</idref>
            </item>
            <item>
              <name xsi:type="xsd:string">文献情報検索</name>
              <idref xsi:type="xsd:string">95638</idref>
            </item>
          </child>
          <id xsi:type="xsd:string">47888</id>
        </item>
      </return>
    </n2:getWordListResponse>
  </env:Body>
</env:Envelope>

```

図 3.7 getWordList におけるレスポンス例

- <parent>要素: 上位関係にある用語の一覧
- <child>要素: 下位関係にある用語の一覧

<parent>、<child>要素の内容は、doWordSearch における<exactMatchElements>要素と同様の構造を持っており、<name>、<idref>要素を要素とする配列である。

GraphViz Web サービス

グラフィケアウト・ツールである GraphViz [25] を使い、グラフ表現を可視化するための Web サービスを構築した。このサービスを用語体系 Web サービスと連携させたシステムを構築することによって、用語体系の階層関係を分かりやすく表示することなどが可能となり、ユーザの用語体系構造把握を助けることができる。

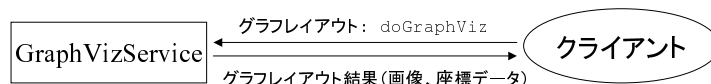


図 3.8 Graphviz Web サービスのインタフェース

GraphViz Web サービスの持つインタフェースは doGraphViz である (図 3.8)。doGraphViz では、渡される dot 形式の文字列を dot コマンドで処理して、グラフィケアウトを行い、指定された形式への変換を行い、クライアントに返す。

```
<?xml version="1.0" encoding="utf-8" ?>
<env:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <n2:doGraphViz xmlns:n1="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:n2="urn:GraphViz"
      env:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <dot xsi:type="n1:base64">ZGlnbmFwaCBHIsKICBncmFwaCBbcmFua2Rpcj0iVEIiXTsKICA0Nzg4OCBb
bGFpZWw9IuaDheWgseaknOe0oiJdOwogIDMwNjQ1IFtsYWJlbD0i5YWJ5a2m
55qE5o0F5aCx5qSc57SiI107CiAgNDc4ODggLT4gMzA2NDU7CiAgNTUxNDIq
W2xhYmVsPSLpgbjmip7nmoTmg4X1oLHmpJzntKIiXTsKICA0Nzg4OCAtPiA1
NTE0MjsKICA5NTYzOCBbbGFpZWw9IuaWh+eMruaDheWgseaknOe0oiJdOwog
IDQ3ODg4IC0+IDk1NjM4OwogIDI5MTMgW2xhYmVsPSLjggLjggbjg4jgg5fj
g4Pjg4giXTsKICAyOTZlIC0+IDQ3ODg4OwogIDI5MDA3IFtsYWJlbD0i5qSc
57SiI107CiAgMjkwMDcgLT4gNDc4ODg7Cn0K
</dot>
      <format xsi:type="xsd:string">png</format>
    </n2:doGraphViz>
  </env:Body>
</env:Envelope>
```

図 3.9 doGraphViz におけるリクエスト例

doGraphViz におけるリクエスト例を図 3.9 に示す。doGraphViz では、<dot>要素の内容として dot ファイル形式を、<format>要素の内容として出力形式を渡す。このリクエストに対応する応答メッセージを図 3.10 に示す。応答では、Base64 符号化された出力 (<return>要素) がそのまま返ってくる (図 3.10)。

3.4.2 フロントエンドシステム

本節では、ここまで述べてきた用語体系 Web サービスをバックエンドとして利用した 2 種類の応用システムについて述べる。


```
<?xml version="1.0" encoding="utf-8" ?>
<env:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <env:Body>
    <n2:doGraphVizResponse xmlns:n1="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:n2="urn:GraphViz"
      env:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <return xsi:type="n1:base64">>Z3JhcGggMS4wMDAgNS4zNjEgMj41M3MjIENk9kZSA0Nzg0CAgMi42NjcgMS4z
NjEgMS4yMTIwMj41MDAg5oF5aCx5qSc57SiIHNvbG1kIGVsbG1wc2UgYmxx
Y2sgbGlnaHRncmV5Cm5vZGUGmYkxMj41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpgbjmip7nmoTmg4Xl0LHmpJzntKIgc29saWQgZWxsaXBzZSBibGFjayBs
aWdoOdG4yZkxKbm9kZSA5NTYzOACAgNl41NDIuODAwIDAuODAwIDAuMzYxIDBUbnJlIDAU
NTAwIOWFieWtpueahOaDheWgseaknOeo0iBzb2xpZCB1bGxpcHNlIGJseYWNr
IGxpZ2h0Z3JleQpub2RlIDU1MTQyYjIcAYLjY2NyAwLjM2MSAxLjYxNSAwLjUw
MCGzpg
```

図 3.10 doGraphViz におけるレスポンス例

用語体系ブラウジングシステム

複数の用語体系 Web サービスに対応した、用語体系ブラウジングシステムを構築した。本システムは、文献 [12] において構築した複数用語体系ブラウジングシステムを、用語体系 Web サービスをバックエンドとして動作するように変更したものであり、用語体系からの見出し語の検索（図 3.11）や各見出し語の持つ構造の表示（図 3.12）などが可能である。

バックエンドとなる用語体系 Web サービスと、フロントエンドである用語体系ブラウジングシステムの間のデータの流れを図 3.13 と図 3.14 に示す。

ユーザが見出し語を指定した検索を行う場合は、以下のようなやりとりが行われる。まず、ユーザの検索要求をフロントエンドであるブラウジングシステムが受けとり、バックエンドである用語体系 Web サービスに doWordSearch メッセージとして要求を送る。Web サービスでは、doWordSearch メッセージに対応した検索を行い、検索結果をフロントエンドに返し、フロントエンドはこの結果を検索結果一覧としてユーザに示す。

また、検索結果一覧からユーザが一用語を選択し、その用語の持つ構造を可視化する際には、次のようなやり取りが行われる。まず、ユーザの選択した用語をブラウジングシステムが受け取り、用語体系 Web サービスに対してその用語の持つ関係を含む構造を取得するための `getWordList` メッセージを送る。それに対して、用語体系 Web サービスからは、`getWordList` で指定された見出し語の構造をブラウジングシステムに対して送り返す。ブラウジングシステムでは、この見出し語の構造を Graphviz Web サービスでのグラフィケイアウトに使われる `dot` 形式に変換し、`doGraphViz`



図 3.11 用語体系ブラウジングシステム (1)

メッセージとして Graphviz Web サービスに送る。Graphviz Web サービスでは、この要求を元にグラフィックレイアウトを行い、結果をブラウジングシステムに返す。ブラウジングシステムでは、この結果をユーザに提供する。

Google Web API との連携システム

用語体系 Web サービスを他の Web サービスと連携させた応用の一つとして、Google Web API [49] との連携システムを構築した。Google Web API は、Google 社が提供する Web 検索エンジンを Web サービスとして利用できるようにしたものである。

本システムでは、Google が提供している検索サービスと用語体系データの検索を同時に行い、用語体系に見出し語があった検索語については、その上位語・下位語を表示する (図 3.15)。さらに、表示された上位語・下位語をクリックすると、検索フォームの検索語にその見出し語が追加される。

図 3.16、3.17 に、Google 連携システムにおけるデータの流れを示す。

まず、ユーザが検索語を入力すると、複数の用語体系について、それぞれ以下の処理を行い、各用語体系でヒットした各見出し語ごとに、整形して表示する (図 3.16)。

1. 用語体系 Web サービスに対して見出し語を検索する (doWordSearch)
2. 完全一致した語があった場合には、
3. 完全一致したそれぞれの見出し語の構造を取得する (getWordList)

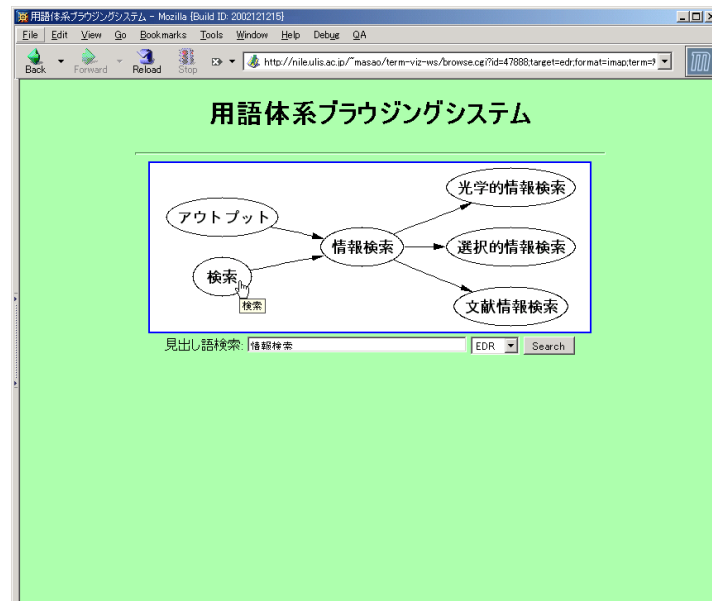


図 3.12 用語体系ブラウジングシステム (2)

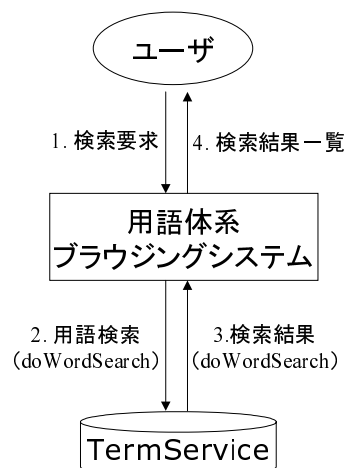


図 3.13 用語体系ブラウジングシステムにおけるデータの流れ (見出し語検索時)

同時に、Google Web サービスに対して、ユーザの入力した検索語で検索を行い (doGoogleSearch)、検索結果を取得し、表示する (図 3.17)。

3.4.3 開発環境

本研究で開発したシステムの構築に利用した環境は以下の通りである。Web サーバには Apache [26] を使い、Web サービス実装としては、プログラミング言語 Ruby [50] による

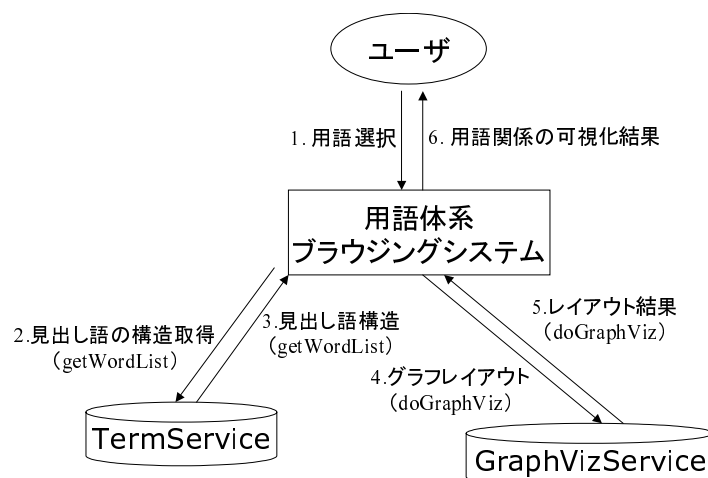


図 3.14 用語体系ブラウジングシステムにおけるデータの流れ（用語関係表示時）

SOAP4R [51] を利用した。また、用語体系の検索などを高速に行うために Berkeley DB ライブラリ [28][52] および Suffix Array ライブラリ Sary [29] を利用したインデックスを作成した。グラフ構造のレイアウトおよび画像出力は、グラフ描画ソフトウェア Graphviz [25] を用いた。

3.5 考察

本節では、Web サービスで用語体系データを提供する際の設計方針とその効果について述べる。

本研究では、用語体系 Web サービスをデータの提供システムとして捉え、一つの利用語体系が一つの URI でサービスを行う方針で、システムを設計し構築した。つまり、別の用語体系 Web サービスを提供する場合は、それぞれを独立した Web サービスとして構築する。本研究では、これらの Web サービス間で共有しうるインタフェース定義を行った。

3.1 節で述べたように、様々な観点から作られた多くの用語体系が存在している。これらの用語体系はそれぞれの方針で作成されており、これらを効率的に再利用するためには、アクセス方式を統一することが重要となる。

用語体系の提供において、Web サービスの枠組みを採用し、それらの Web サービスにおけるインタフェースを統一することが、アクセス方式の統一につながるものと考えられる。つまり、用語体系の提供側においては、この統一されたアクセス方式に従ったサーバを作成することで、他の体系用に構築されたフロントエンドをそのまま利用することが可能となる。一方、用語体系を利用する側においても同様に、一つのフロントエンドだけで、複数の用語体系を同時に利用したり、切り替えて利用したりすることが容易になる。

また、3.4.2 節では、Google Web API という全く別の Web サービスとの連携について述べた。このような検索システムなど、他の Web サービスと連携する応用システムにより、用語体系の利

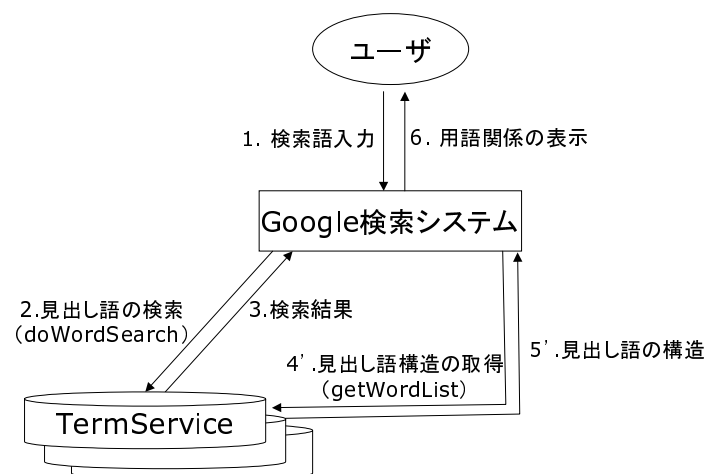


図 3.16 Google Web API との連携システムにおけるデータの流れ (1)

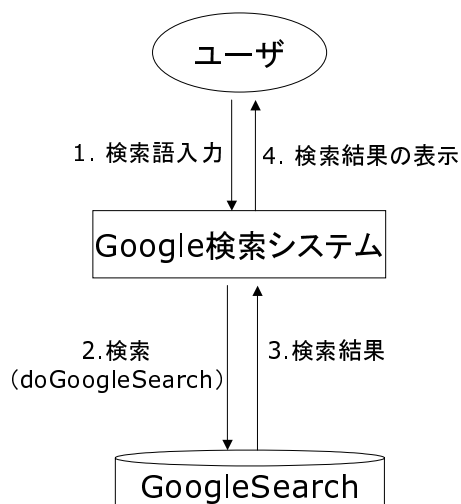


図 3.17 Google Web API との連携システムにおけるデータの流れ (2)

提供する場合には、これらのアクセスポリシーを柔軟に運用することは困難である。そこで本研究では、Web サービスを使ってネットワーク上で分散的に提供の方針を採用した。これにより、上述のようなアクセスポリシーの違いを越えた、用語体系データの提供が可能になると考える。

Web サービスにおけるパフォーマンスや負荷などの問題も考慮すべき問題である。本研究で構築したシステムは用語体系データ提供システムであり、データの取得要求がどのような経路でアクセスされるかに依るため、そのアクセス性能一般については一概には測れないが、Web サービスとフロントエンドシステムとが同一の LAN 内に存在する場合には、用語体系アクセスにかかる時間は、数秒程度と実時間で処理可能であった。

3.6 まとめ

本研究では、用語体系の再利用性の向上とその効率的な提供を目指して Web サービスに基づく用語体系の提供システムを構築した。さらに、これらのネットワーク上に独立して存在する提供システムを利用した複数の応用システムを構築し、その有効性を示した。本章で述べた Web サービスの枠組みを利用することにより、本章で述べたフロントエンド応用システム以外の他システムとも容易に連携可能であり、用語体系の提供の可能性はより広がるものとする。

第 4 章

Z39.50 による日本語書誌データ検索システム

4.1 背景と目的

Z39.50[7] はクライアント・サーバ環境において、クライアントとサーバが通信する際のデータ構造やデータ変換の処理方式を規定したプロトコルである。すなわち、システム内部でのデータ構造や検索処理がシステムごとに異なっても、通信する際に同じ構造と処理方式にしておけば、それぞれのシステム構成を維持したまま互いのシステムを利用できる。

1960 年代後半からオンライン検索システムの開発が始まり、1972 年には DIALOG や ORBIT などがデータベースベンタとして多くの商用データベースを抱え、オンラインサービスを開始した。時同じくしてアメリカ議会図書館、OCLC、RLIN などが機械可読目録 (MARC) のデータベースを構築し、書誌ユーティリティとしての機能を果たし始めた。これらのシステムは公衆回線を通じて端末からホストコンピュータにアクセスする中央集中型システムであり、この時代はこの形態のシステムが情報を得る上で最も効率がよかった。

しかしながら、中央集中型システムと言えども全ての情報が集中しているわけではないため、様々な情報要求を満たすためには複数のシステムを利用せざるを得なかった。ところが、ここで問題となったのはコストと使い勝手であった。すなわち、これらのデータベースを使用するにはそれぞれ高額の使用料を支払わなければならない、そのうえ、複数のデータベースの使用料に見合う情報は必ずしも得られなかった。特に書誌ユーティリティにおいては重複データが数多くあり、情報を得る上で効率が悪く、使い勝手の面でもシステムごとに接続手順や検索コマンドに違いがあり、利用者の検索要求を十分満たせないという状況であった。

こういう状況の下、1979 年に主要な書誌ユーティリティ間で目録データを交換し、互いのデータを透過的に検索できることを目指したプロジェクトが開始され、その後、OSI の規格と平行して ANSI の構成団体である NISO で議論され、1988 年に標準情報検索プロトコル Z39.50 として規格

が制定された [6][54]。

1990 年代に入り、TCP/IP をベースとするインターネットの普及にあわせ、Z39.50 の規格は OSI ベースの 1992 年の Version2 から 1995 年の Version3 に発展した。この時期に、アメリカ議会図書館、OCLC をはじめ、アメリカ、ヨーロッパで図書館システムを中心に数多くの Z39.50 検索システムが実装されるようになり [55][56][57]、1999 年 6 月現在、Index Data 社が作成しているサーバリスト [58] に登録されているだけでも 228 にのぼる。

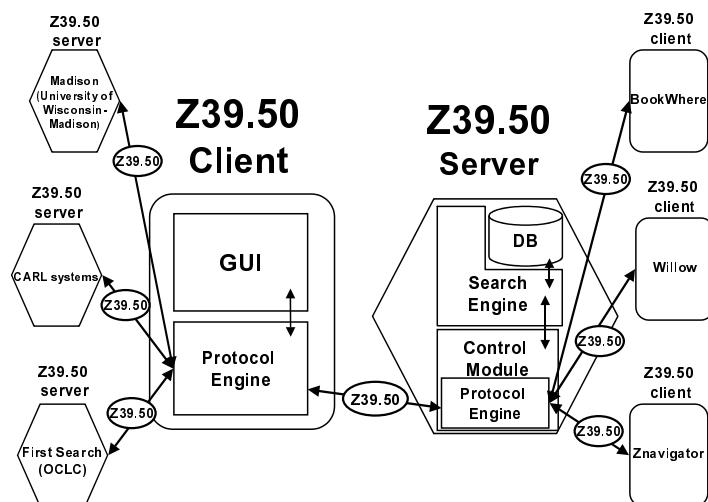
日本でも欧米の動きに呼応するように、Z39.50 に関連したシステムが構築されるようになってきた。既存のシステムを使ったゲートウェイサービスとしては、OCLC の SiteSearch を使った紀伊国屋のサービス [59]、海外の Z39.50 サーバへのゲートウェイ機能を持つシステムとしては、丸善の MARUZEN Z39.50 Gateway-Cat[60]、科学技術振興事業団 (JST) の JOIS-withSTN[61] がある。図書館情報大学デジタル図書館 [62]、早稲田大学図書館の WINE[63]、東京工業大学の電子図書館システムには Z39.50 の検索システムが含まれており [64][65]、学術情報センターの電子図書館システム NACSIS-ELS は、クライアントとサーバの間のアクセスに Z39.50 を使用している [66][67]。また、地球観測衛星データシステムに Z39.50 を応用した例 [68] や、英語の書誌データを対象にした Z39.50 サーバの構築例もある [69]。また、石田による Japan/MARC を対象とした日本語検索のための仕様策定への提案も行なわれている [70]。

図書館情報大学 (ULIS) では 1995 年以来、日本語書誌データを対象に Z39.50 サーバのシステム構築実験を続け、1996 年に安斎によって Web ブラウザから検索できるシステムが完成した [71]。このシステムを Z39.50-ULIS1 と呼ぶ。その後、システム構成が見直され、1998 年にクライアントシステムも含む Z39.50 の基本機能を完全に備えた Z39.50-ULIS2 の開発が行われた [72]。引き続き、クライアント機能を改良するとともに、大規模データに耐えられるシステムにするため、Z39.50-ULIS2 のサーバ部分を全面的に作り直した。本章ではこの新しく開発した Z39.50-ULIS3 について述べる。Z39.50-ULIS3 はより実験システムとしての性格を強め、拡張性と移植性を高めた。

4.2 Z39.50 検索システム

4.2.1 システムの概要

本研究では日本語書誌データベースを検索するために Z39.50 に準拠したサーバとクライアントを UNIX ワークステーション Sun Enterprise 3000 上で開発した (図 4.1)。サーバは Japan/MARC をデータとして持ち、Z39.50 に基づいたクライアントからの検索要求に応える。クライアントは検索語の日本語入力、検索結果の日本語表示が可能であり、Z39.50 に準拠した他のサーバにもアクセスできる。Z39.50 のクライアントとサーバは APDU(Application Protocol Data Unit) と呼ばれる情報フレームをそれぞれのプロトコルモジュール間で通信することで検索セッションを維持する



[73].

サーバはコントロールモジュール、プロトコルエンジン、検索エンジン、データベースから構成され、データベースはさらにインデックス部と Japan/MARC レコード部から構成される。要求は全てクライアントから APDU 形式で出され、サーバは受け取った APDU を解析しコントロールモジュールに渡す。コントロールモジュールはクライアントからの要求が検索要求であれば検索エンジン呼び出す。検索エンジンはデータベースを検索し、検索結果をコントロールモジュールに返す。コントロールモジュールは検索結果を履歴集合として保存し、ヒット件数をプロトコルエンジンに渡す。プロトコルエンジンはヒット件数をクライアントに返す。また、コントロールモジュールへのクライアントからの要求が返戻要求の場合、履歴集合として保存してあった検索結果を SUTRS(Simple Unstructured Text Record Syntax) 形式でクライアントに返戻する。

本システムは Index Data 社が開発した Z39.50 システム構築用ツールキット YAZ(Yet Another Z39.50 Toolkit)[74] を利用した。YAZ は C 言語で書かれたプログラム群からなり、Z39.50 の定義するデータ構造を忠実に再現している。また、サーバの検索エンジンには全文検索エンジン Namazu[75][76] を用いた。クライアントのプロトコルエンジンは YAZ を用いて開発し、GUI 部分は Tcl/Tk を用いて開発した。

4.2.2 日本語書誌データの処理

本システムは日本語書誌データ Japan/MARC の検索を可能にするため、検索エンジンおよびデータベースで日本語処理を行うとともに、クライアントで日本語入力および表示が行えるようにした。

この節では、Japan/MARC データを全文検索エンジン Namazu で利用できるようにするための切り出し処理およびインデックス処理について述べる。なお全文検索エンジン Namazu は、フリーで利用可能な Web 上の検索エンジンとして広く使われており、本システムでは、これを Japan/MARC のタグフィールドに対応するように改造し、Z39.50 サーバで利用できるようにした。

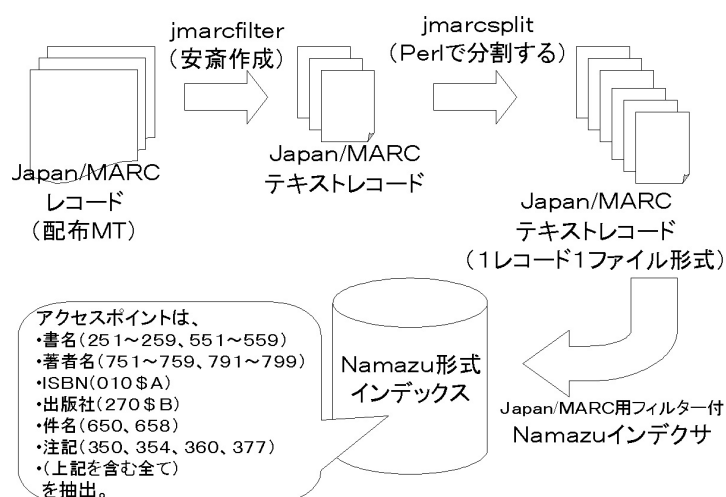


図 4.2 Japan/MARC の変換

本システムは 1983 年 ~ 1998 年 5 月までの配布分 1,019,696 レコードをデータとして持つ。国立国会図書館から配布された Japan/MARC レコードのデータはあらかじめ安斎作成のデコーダ [77] でプレインテキストにしたものをもとに 1 レコード・1 ファイル形式に変換する。このレコードファイルから Japan/MARC のタグフィールドの情報 [78][79] をもとに、ISBN、書名、著者名、件名、出版者、注記ごとのインデックスと全アクセスポイントを対象にしたインデックスを作成した (図 4.2 参照)。この際、日本語のテキストデータは日本語形態素解析ツール・茶筌 [80] でわかち書きを行なったものを Namazu のインデックスに登録しておく。インデクシングの所要時間は 14 時間 25 分 16 秒で、茶筌で切り出したキーワード数は 4,414,062 語であった。

Z39.50 以前の検索システムではシステム毎に検索質問の構造が異なり、複数のデータベースを利用する際の障害となっていた。Z39.50 ではこの問題に対して、検索質問の包括的なスキーマを定義し、実装の際にこのスキーマを実際のアクセスポイントにマッピングすることで対処している。書誌情報に対するスキーマは 2 つの数字の組み合わせで表される Bib-1 アトリビュートセットで定義されている。例えば、Bib-1 を使えば、書名・著者名・前方一致・後方一致・フレーズなどの検索質問の特性を特定できる。本システムではこのアトリビュートセット Bib-1 を用いた。Bib-1 と Japan/MARC でのタグ・フィールドとの対応関係を表 4.1 に示す。

表 4.1 Bib-1 と Japan/MARC の対応表

項目	Bib-1(USE Attribute)	Japan/MARC タグ	Japan/MARC 説明
ISBN	ISBN(1=7)	010\$A	: 国際標準図書番号 (ISBN)
書名	Title(1=4)	251 ~ 259 \$A \$B \$D 551 ~ 559 \$A \$X \$D	: 記述フィールド : 書名 : 副書名 : 巻次等 : 書名アクセスポイント : カタカナ形 : ローマ字形 : 巻次の読み
著者名	Author(1=1003)	751 ~ 759 \$A \$X \$B 791 ~ 799 \$A \$X \$B	: 著者名アクセスポイント : カタカナ形 : ローマ字形 : 漢字形 : 多巻ものの各巻著者標目 : カタカナ形 : ローマ字形 : 漢字形
件名	Subject Headings (1=21)	650 658	: 個人名件名標目 : 一般件名標目
出版社	Publisher (1=1018)	270\$B	: 出版者、頒布者等
注記	Note (1=63)	350 354 360 377	: 一般注記 : 原タイトル注記 : 装丁と定価に関する事項 : 内容注記
全て	Any (1=1016)	数字 3 桁\$.	: タグ・フィールドを除いた全て

4.2.3 Z39.50-ULIS3 サーバ

Z39.50-ULIS3 サーバは接続機能、検索機能、返戻機能、終了機能の基本機能を持つ。接続機能は複数の Z39.50 クライアントによる接続要求を受け付け、検索・返戻時に用いる推奨メッセージサイズやサーバが提供している機能、Z39.50 のバージョン情報などのパラメータの折衝を行う機

能である。

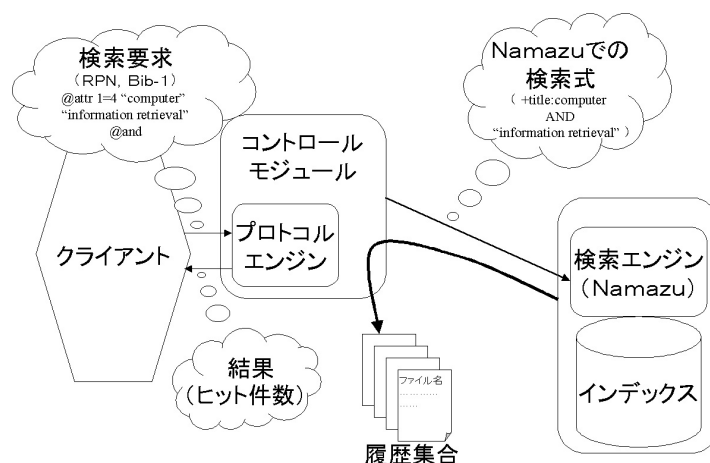


図 4.3 検索処理

検索はコントロールモジュールがプロトコルエンジンから検索要求を受けとり、コントロールモジュールが逆ポーランド記法の検索式を検索エンジンに渡すことによって行われる。検索エンジンはインデックスを検索して、コントロールモジュールに検索結果を返す。コントロールモジュールはヒットしたレコードのファイル名を履歴集合ファイルに出力した後、ヒット件数をプロトコルエンジンを通してクライアントに返す（図 4.3 参照）。サーバは検索式として Type1-Query と Type101-Query を受けつける。検索質問は AND, OR, AND-NOT の論理演算、Bib-1 のアクセスポイント指定、検索語の属性指定、履歴集合指定が可能である。履歴検索は履歴集合同士や、検索語と履歴集合の論理演算を行える。履歴集合同士の検索は履歴ファイルのみで行うため短時間で処理できる。

返戻はまず履歴集合名、レコードシンタックス、返戻レコード開始番号、レコード件数を指定し、指定された履歴集合から実際の Japan/MARC レコードファイルを抜き出し、Japan/MARC レコードファイルを SUTRS 形式で返すことによって行われる（図 4.4 参照）。クライアントへ返戻するレコードのサイズは、接続要求時の推奨メッセージサイズのパラメータ指定による。

終了機能はクライアントとの接続を切り、そのセッションでの履歴集合ファイルの削除を行なう機能である。

4.2.4 Z39.50-ULIS3 クライアント

クライアントの役割は GUI を通したユーザからの接続要求、検索要求、返戻要求を解析・変換し、サーバに対して出すことである。それぞれの要求はプロトコルエンジンによって APDU に変

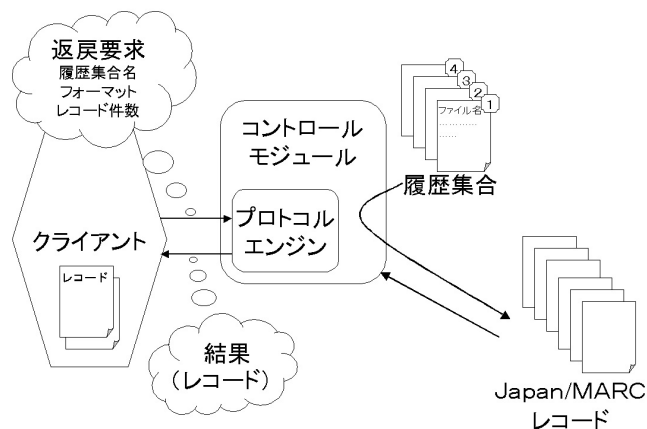


図 4.4 返戻処理

換されてサーバに渡され、サーバから返された APDU は出した要求に応じてクライアント内で処理される。ユーザは本検索クライアントを用いて著者や書名などのアクセスポイントを指定したり、前方一致や後方一致、履歴を利用して検索できる。また、レコード数、履歴集合、詳細度合を指定して検索結果を見ることができる。

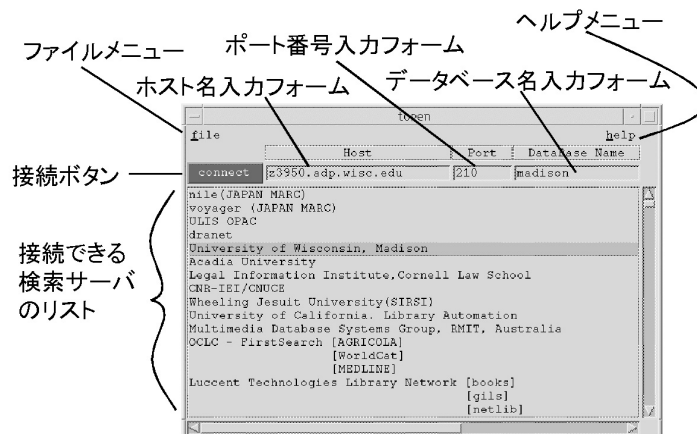


図 4.5 接続ウインドウの機能

図 4.5 は接続ウインドウの画面である。画面中央には OCLC First Search や University of California など約 200 のサーバのリストが表示されており、ユーザが接続したいサーバを選択して

などのレコード形式を選択できる。

ファイルメニューの「New Window」を選択すると、現在接続しているサーバに接続したまま、新しい接続ウインドウを起動させ、他のサーバに接続できる。接続メニューの「Disconnect」を選択すると、現在接続しているサーバと接続を切り、新しい接続ウインドウが起動し、他のサーバに接続できる。ファイルメニューの「Exit」を選択するとクライアントを終了する。ヘルプメニューを選択すると、検索式の例が表示される。

4.2.5 システムの実行例

これまで述べてきたように、Z39.50-ULIS3 システムは Z39.50-ULIS3 クライアントと、Z39.50-ULIS3 サーバの 2 つの独立したアプリケーションからなっている。この節では、Z39.50-ULIS3 クライアントを用いて Z39.50-ULIS3 サーバを含めたいくつかの Z39.50 サーバを検索した実行例をあげることで、Z39.50-ULIS3 サーバや Z39.50-ULIS3 クライアントが日本語を扱えることや、Z39.50-ULIS3 クライアントが Z39.50-ULIS3 サーバだけでなく他の Z39.50 に基づいたサーバを検索できることを示す。



図 4.7 検索ウインドウ: Z39.50-ULIS3 サーバ

まず、Z39.50-ULIS3 サーバを検索した例を図 4.7 に示す。検索結果表示欄を元に検索を順に追うと、

1. 「ガラス」を検索 ヒットした件数が 667 件
2. 「工芸」を検索 ヒットした件数が 1373 件
3. 1 番目と 2 番目の検索結果集合の AND 検索 ヒットした件数が 19 件

という検索要求が出されていることがわかる。3 番目の検索式の“set=1”は1 番目の検索集合を“set=2”は2 番目の検索集合を指している。“and”はその AND をとるという意味である。返戻結果表示部分は3 番目の検索結果集合を表示している。

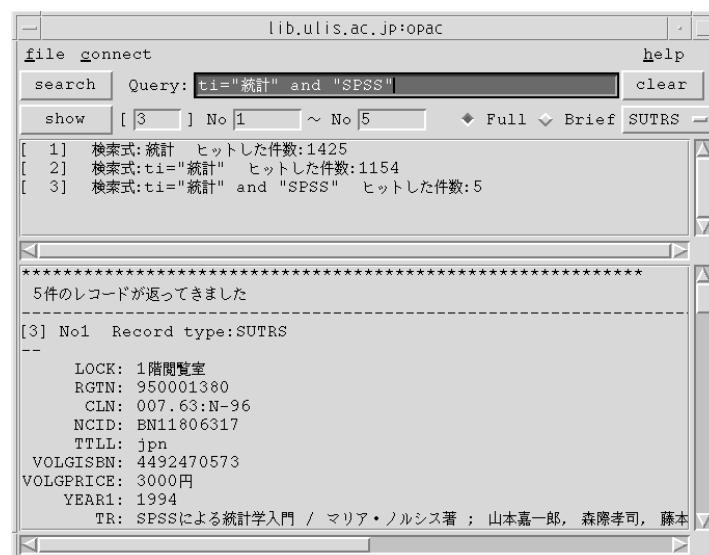


図 4.8 検索ウィンドウ：図書館情報大学デジタル図書館システム Z39.50 サーバ

次に、Z39.50-ULIS3 以外の日本語を用いたサーバを検索している例を図 4.8 に示す。図 4.8 は図書館情報大学デジタル図書館システム Z39.50 サーバを検索しているところである。検索結果表示欄を元に検索を順に追うと、

図 4.9 は University of Wisconsin, Madison のサーバを検索しているところである。検索結果表示欄をもとに検索を順に追うと、

1. 「統計」を検索 ヒットした件数が 1425 件
2. 「書名に統計」を検索 ヒットした件数が 1154 件
3. 「書名に統計」と「SPSS」の AND 検索 ヒットした件数が 5 件

という検索要求が出されていることがわかる。2 番目と3 番目の検索式の“ti=”はアクセスポイントに書名を使うことを表している。3 番目の検索式の“and”は「書名に統計」と「SPSS」の AND をとるという意味である。返戻結果表示部分は3 番目の検索結果集合を表示している。

最後に海外のサーバを検索しているところを図 9 に示す。図 4.9 は University of Wisconsin, Madison のサーバを検索しているところである。検索結果表示欄をもとに検索を順に追うと、

1. 「書名に library」を検索 ヒットした件数が 5948 件
2. 「著者名に Lancaster」を検索 ヒットした件数が 498 件

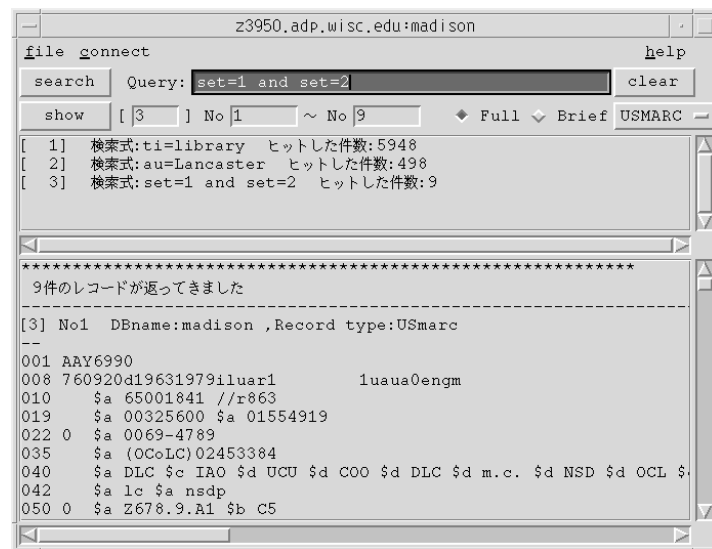


図 4.9 検索ウインドウ：University of Wisconsin, Madison

3. 1 番目と 2 番目の検索結果集合の AND をとる検索 ヒットした件数が 9 件

という検索要求が出されていることがわかる。1 番目の検索式の“ti=”はアクセスポイントに書名を使うことを表しており、2 番目の検索式の“au=”はアクセスポイントに著者名を使うことを表している。3 番目の検索式の“set=1”は 1 番目の検索集合を“set=2”は 2 番目の検索集合を指している。“and”はその AND をとるという意味である。返戻結果表示部分は 3 番目の検索結果集合を表示している。

4.2.6 開発システムの変遷

今回開発した Z39.50-ULIS3 システムと以前の Z39.50-ULIS1 システム [82] および Z39.50-ULIS2 システム [83] との相違点を表 4.2 にまとめた。

Z39.50-ULIS1 は本学で開発した最初の Z39.50 システムであり、Z39.50 システムを構築するための基本技術が得られた。しかしながら、この時点ではインターネット上に日本語書誌データを提供することが主要な目的であったため、利用者に馴染みのある Web ブラウザをクライアントとし、CGI を介して検索するシステムとして構築した。そのため、純粹のクライアントとサーバの両方を含む完全な Z39.50 システムにするには至らなかった。

Z39.50-ULIS2 は Z39.50-ULIS1 のシステム構築の基本技術を継承し、Z39.50-ULIS1 では実現しなかった Z39.50 の仕様に完全にそったシステム構築を目指し、純粹な Z39.50 クライアントを作成するとともにサーバも新たに構築し直すことにした。しかしながら、実装面でいくつかの不具合があり、検索システムとして見た場合、不満が残るシステムであった。

表 4.2 開発システムの相違点

	Z39.50-ULIS1 (1996 年 11 月)	Z39.50-ULIS2 (1998 年 2 月)	Z39.50-ULIS3 (1999 年 5 月)
構成	サーバ Web ブラウザから検索 (CGI 接続プログラム)	サーバ クライアント	サーバ クライアント
開発環境	Sun SPARC Station10 YAZ-1.2 DBMS: ADABAS C 言語	IBM RS6000 YAZ-1.3 DBMS: Oracle7 C 言語, SQL	Sun Enterprise 3000 YAZ-1.4pl2 検索エンジン: Namazu C 言語, Perl
データ	Japan/MARC 1 年分 (73,549 件) ULIS-OPAC (85,651 件)	Japan/MARC (1,601 件)	Japan/MARC 15 年 5 ヶ月分 (1,019,696 件)
検索機能	論理演算可 フィールド指定可 (書名、著者、ISBN、 件名、NDC、 NDLC, JP-No) 履歴検索可	論理演算一部可 フィールド指定可 (書名、著者、全指定) 履歴検索可 (履歴集合同士のみ)	論理演算可 フィールド指定可 (書名、著者、ISBN、 件名、出版者、 注記、全指定) 履歴検索可
レコード	ADABAS のレコード	Oracle のテーブル	1 レコード 1 ファイル
インデックス語 の切りだし	MHSA[84]	なし (SQL の LIKE)	茶釜

Z39.50-ULIS3 は Z39.50-ULIS2 と同じクライアント・サーバ型のシステム構成であるが、Z39.50-ULIS1 と Z39.50-ULIS2 で得た技術を統合し、サーバの実装面で多くの改良を加え、設計と実装の両面においてより充実したシステムとなった。

Z39.50-ULIS3 サーバでの主な改良点は以下の 2 点である。

1. 大規模システムの構築 (データ量の増大)

本システム (Z39.50-ULIS3) では約 100 万件のデータを Z39.50 で提供する基盤を構築するために検索レスポンスの面での大幅な改善を求めた。そのために、ULIS2 のシステムの見直

しを迫られ、Namazu を中心とする全文検索エンジンの導入に至った。検索実験も約 100 万件のデータに対して行い、支障のないレスポンスタイムで検索できた。

2. Namazu の改造による資源共有

フリーで入手可能な Namazu に対して Japan/MARC に対応させるための改造を行なったことにより、Web 上で広く利用されている Namazu の検索システムの資源（インデックス）をそのまま Z39.50 で提供できる。

4.3 考察

以上、本研究で開発した Z39.50 検索システムのクライアントとサーバの構成と機能について述べてきた。ここでは、本システムの意義と Z39.50 検索システムの将来について考察する。

4.3.1 システムの意義

本システムの意義は実験システムという観点から、システム構築をとおして様々な試みができるという点である。Z39.50 はクライアントとサーバ間のやりとりのみを規定しているので、クライアントとサーバ自身は自由に構築できる。しかし、自由に構築できる反面、自ら構築しなければならず、Z39.50 システムを必要としているが開発に不慣れな場合はかなりの負担となる。そこで、Z39.50 システムはどのような性格のシステムなのか、どのようなシステム構成にすればよいのか、どのくらい手間がかかるのか、必要な機能は何で、これからどのような機能が必要になるかなどを実験をとおして知る必要がある。

システム構築はシステムの性格、データの性質、開発の利便性などを考慮に入れて行った。Z39.50 検索システムは自由度が大きい部分と制限の強い部分を同時に合わせ持つシステムであり、機能的には相互接続性を保証すること、セッションを維持することが重要である。その他、検索レスポンス、日本語処理、検索式入力形式、履歴検索、横断検索なども考慮する必要がある。本システムで扱う書誌のデータは、量が多いこと、処理単位が基本的にレコードであること、ひとつのレコードが構造を持つこと、検索の際は複数のレコードにまたがってアクセスすること、データ更新のほとんどが追加のみであることなどが性質としてあげられる。また、開発の利便性については、どこまで既存の道具を使い、どこから独自にプログラムを書くのか、開発後の移植性などを考慮しなければならない。

以上のようなことを踏まえ、サーバについては、ファイル編成法やデータアクセス手法について実験を繰り返した結果、レコードファイルとインデックスを完全に分離し、レコードファイルを直接処理するのではなく、レコードファイルに対してアクセスポイントごとにインデックスをつけ、インデックスファイルを直接の処理対象にするという手法が最も効率的であることがわかった。関係データベースで直接書誌データを処理することも試してみたが、従来の書誌データ検索システム

で使われてきたオーソドックスな手法が最も有効であることが明らかになった。

Z39.50 の実装については、基本的な機能については実現したが、細かい部分については今後の課題として残っている。サーバに関していうならば、例えば、Bib-1 のアトリビュートタイプの内、アクセスポイント指定 (Use) 以外の Relation、Position、Structure、Truncation、Completeness の指定には対応しておらず、ElementSetName(詳細・簡略) の指定にも対応していない。Sort や Scan などの拡張機能についても未対応である。しかし、これらの機能は書誌データ検索としては必須の機能ではないと判断し、今後の課題とした。

文字コードについては、サーバとクライアントの内部では EUC で処理し、Japan/MARC は ISO-2022-JP で保存してある。そのため、サーバからクライアントへのレコード返戻時には ISO-2022-JP で転送する。ただし、クライアントは ISO-2022-JP だけでなく Shift-JIS と EUC のデータが来ても EUC に自動変換して文字化けせずに表示できるようにしてある。また、サーバはクライアントからの入力文字コードを内部文字コードである EUC に自動変換して処理している。このため、日本語文字については言語折衝 [85] がなくても、日本語データを持った Z39.50 サーバを検索できる。将来的には言語折衝の部分も実装したいと考えている。

クライアントのその他の機能実験としては、Sun のワークステーションのほか、IBM と Hewlett-Packard のワークステーションにも移植し、マルチプラットフォーム化を行った。また、サーバの自動登録機能についても開発した。

今後、サーバ情報を利用者同士や他システム間で交換することを想定して、接続時のサーバ情報は、`z39.50s://lib.ulis.ac.jp:210/opac` (図書館情報大学デジタル図書館) のように標準化された URL 形式 [86] で記述した。

このようにシステム構築を通して様々な試みを行っており、今後も引き続き実験を行う予定である。

4.3.2 Z39.50 検索システムの将来

本節では、Z39.50 検索システムの将来について論じ、今後の実験の方向性を探る。

検索システムに Z39.50 プロトコルを採用することのメリットは、使い慣れたクライアントで複数のサーバに接続できること、横断検索が可能になること、セッションを維持した履歴検索ができるという 3 点にまとめられる。

検索システムを相互に接続し、書誌データを交換しようという 1979 年からの計画は、1992 年の Z39.50 Version2 の規格が制定されたのを機に実装が進み、これから順調に発展していくように見えた。ところが、1994 年からの Web の爆発的な普及により、Z39.50 の存在意義を疑う意見が出てきた。しかし、Web はもともと研究者間の研究情報を共有するという目的で開発されたブラウジング指向のシステムであり、Web ブラウザ上のハイパーリンクをたどることによって、関連情報を次から次に見ることができる。ハイパーリンクは HTML と呼ばれるマークアップ言語によって記

述し、HTML ファイルは Web 用の転送プロトコル HTTP によって転送される。

Web が単なるブラウジングシステムに留まっていたならば、Z39.50 とは競合せず、比較されることはなかった。ところが、ブラウジングだけでは求める情報にたどり着くのが困難になり、検索エンジンがブラウジングシステムとは別に実装された。また、検索エンジンの他にもユーザとインタラクティブに動作するシステムが数多く提供された。こうなると、Web はもはや単体のブラウジングシステムというより、複数のシステムが提供される総合環境と位置づけられる。

その結果、本来 Z39.50 に適していた図書館 OPAC も Web ベースで提供されることとなり、利用者は使い慣れた Web ブラウザから OPAC を利用できるようになった。しかも、それぞれの OPAC のインタフェースは似通っており、使い勝手が阻害されるということもない。そのうえ、九州大学の Web ページ [87] では複数の OPAC に対して横断検索もできるようになっている。

このように、Z39.50 のメリットは大部分 Web で実現されてしまった。特に日本では、Z39.50 の普及より Web の普及のほうが先であったため、より Web の優位性が目立つ結果となった。しかしながら、これで Web があれば Z39.50 が不要ないということにはならない。上田も述べているように [54]、現実的観点からみると Web のほうが有利であるが、視点を変えることによって Z39.50 の重要性が浮かび上がってくる。

筆者が考える Z39.50 の重要性は次の 3 つである。

1. 既に数多く立ちあがっている海外サーバの利用および国内サーバの海外への提供
2. 複雑な検索要求には履歴検索が有効
3. 構造情報を含めたデータ交換
 - (a) Web とは違った検索に特化したネットワークの可能性
 - (b) 電子図書館システムにおける構造情報通信のための基盤技術

まず 1 点目についてであるが、欧米では Web が普及する前に Z39.50 の実装が進んだため、既に数多くの Z39.50 サーバが立ち上がっている。これらの資源を利用することはたいへん有用であり、前述のとおり、日本でも海外サーバに対してゲートウェイサービスが行われている。今後は海外のサーバを利用するだけでなく、日本でもサーバを立ちあげて世界に日本のデータを提供することが重要である。

次に 2 点目であるが、Web の検索エンジンにより数多くの人が自ら手軽に検索するようになり、このような検索エンジンがあれば検索はこれで全て間に合うという考えが増えてきた。しかし、検索要求や検索対象は質的にも量的にも多様性があり、Web コンテンツ、図書、雑誌記事、ファクト、数値データの検索を同列に扱うべきではない。確かに、Web コンテンツの検索は検索エンジンで用が済む場合が多いが、これはもともとの検索要求が比較的緩いことと、ブラウジング機能も併用しているので、検索がうまくいっているかのように見えるのだと思われる。この場合でも、適合度順付きの検索結果と実際の検索要求が一致しないなど Web 検索は多くの問題を抱えている。そして、より高い検索精度が要求される雑誌記事を対象にした場合はさらに深刻で、複雑な検索式を用いた

繰り返し検索が必要になってくる。そうなった場合、履歴検索は必須の機能と言えるだろう。

最後に 3 点目であるが、Web で使われている通信プロトコル HTTP と Z39.50 との本質的違いは、HTTP が基本的には HTML ファイルの単純転送であるのに対して、Z39.50 はデータ構造や処理方式を含んだ情報通信であるという点である。現在、Web コンテンツとは異なる形式の多量のデータを管理するため、独自の検索エンジンを持つ人が増えつつある。これまでは特定の検索エンジンを利用していたのが、独自の検索エンジン込みでそれぞれがデータを提供できるようになる可能性がある。そうすると、お互いのデータを交換し、再利用したいという要求がでてきてデータ構造を理解する共通のプロトコルが必要になる。この共通のプロトコルに Z39.50 を使えば、クライアント機能を内部に持ったサーバ同士が、Z39.50 を介して直接データを交換できる。こうなると、Z39.50 は検索プロトコルというよりデータ交換プロトコルであると言えるだろう。その結果、Z39.50 は構造データが対象の検索に特化したシステム、Web はコンテンツの提供を主目的にしたシステムというように、性格が異なるシステムとして共存することが可能であり、それぞれの性格を活かした関係も可能になる。

その他、構造を持ったデータ交換が可能ということで、電子図書館などのシステムで内部通信のプロトコルに使える。例えば、密結合が要求されるサブシステム間のデータ交換などシステム間の共通データ通信プロトコルとして利用することは大いに価値があると思われる。通信の部分は仕様が決まっている Z39.50 を使えばプログラム開発が楽にでき、その分、他の部分の開発に集中できる。

1999 年には、Z39.50-1995 (Version 3) の ISO 版である ISO 23950 が JIS 規格としても制定され [88]、日本でも今後ますます実装が進んでいくだろうと思われる。

4.4 まとめ

本研究では情報検索標準プロトコル Z39.50 に基づいた日本語書誌データ検索システムを構築した。日本でも 1999 年の東京工業大学や図書館情報大学の電子図書館システムなど、自館の OPAC データを Z39.50 でサービスするシステムが出てきたものの、日本語処理の問題、海外サーバを含む他のサーバとの相互接続性、レコードシンタックスおよびセマンティックス、メタデータと既存目録データの交換方式、フルテキストなど書誌データ以外の処理機能など解決すべき問題は数多く残っており、Z39.50 の実験システムとして本システムの意義は大きい。

第 5 章

Dublin Core を共通スキーマとした Z39.50 検索システム

5.1 背景と目的

Z39.50[7] は情報検索のための国際標準プロトコルであり、その特徴は多種多様な環境で構築された情報検索システムに対して透過的に検索できる点にある。4 章で述べたように、筆者はこれまでに JAPAN/MARC のタグ・フィールドと Z39.50 の代表的アトリビュートセットである Bib-1 とのマッピングを行なうことにより、JAPAN/MARC の書誌データを検索できる Z39.50 システムを構築した [8]。本章ではさらに JAPAN/MARC の書誌データを Dublin Core のスキーマを通して検索できる Z39.50 検索システムの構築について述べる。

Z39.50 ではデータベースの実装に依存しないスキーマを目指し、アトリビュートセットと呼ばれる論理的なスキーマを定義している。しかしながら、Z39.50 は複数のアトリビュートセットを使うと透過的な検索が妨げられるため、結果として大部分の Z39.50 サーバが Bib-1 というひとつのアトリビュートセットを使っている。ところが、実際の検索項目を Bib-1 にマッピングする際のスキーマが定められていないため、このマッピングがシステム依存になっており、異なる検索システムの間では同じフィールドが別のマッピングをしている可能性があるなど検索処理の一貫性が保たれていないという問題点が指摘されている [89]。このデータ間のスキーマの違いを吸収する枠組として Dublin Core の適用可能性が指摘されている [6]。

Dublin Core[9] は基本的な 15 エレメントを定義することによってインターネット上の多様な情報資源に対応し、情報資源発見の効率化をはかるために定義されたコアメタデータである。Dublin Core は単にシンプルなメタデータとして理解するのではなく、様々な分野に共通な要素、すなわち様々な分野にできるだけ共通の概念として認められた属性の要素の集まりとして定義されたコアメタデータとして理解すべきである [10]。Dublin Core が担う最も重要な役割は相互利用性 (Interoperability) であり、Z39.50 のアトリビュートセット、JAPAN/MARC や US-MARC などの

目録データ、Web コンテンツ、既存の各種メタデータなど多様な情報資源を Dublin Core をゲートウェイとして相互に結ぶことで、情報資源の発見が容易になる。

筆者は検索システム間の相互接続性を目指した Z39.50 と、データスキーマ間の相互利用性を目指した Dublin Core を結合させ、様々な情報資源を統一的に検索できるシステムの構築を目指している。本システムでは Z39.50 と Dublin Core を透過的に利用できる基盤整備という観点でシステム構築を行なった。そこで、JAPAN/MARC データを例にとり Dublin Core を共通スキーマとしてマッピングを行ない、Z39.50 の Bib-1 における Dublin Core 用のアクセスポイントを通して JAPAN/MARC を検索できるシステムを構築した。本システムにより JAPAN/MARC だけでなく、その他の書誌データについても Dublin Core を共通スキーマとした Z39.50 検索システムを容易に構築できるようになる。

5.2 システムの構成

本システムは JAPAN/MARC レコードを Dublin Core に基づいて記述し、利用者は Z39.50 の Bib-1 の Dublin Core 用アクセスポイントから検索できる。本システムはプロトコルエンジン、コントロールモジュール、データベース部、メタデータ変換部からなる（図 5.1）。

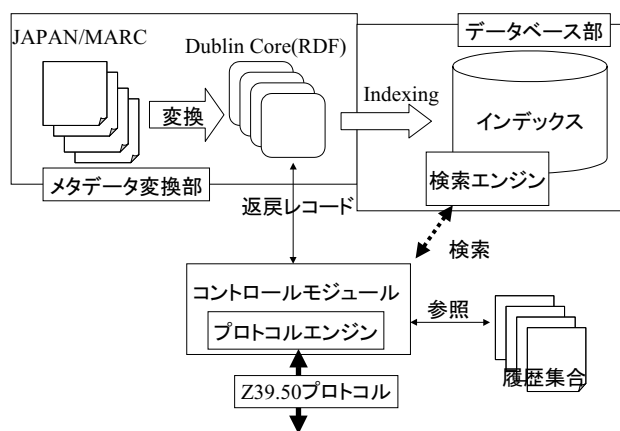


図 5.1 システムの構成

プロトコルエンジンは Z39.50 プロトコルを解釈し、コントロールモジュールに渡す。コントロールモジュールはプロトコルエンジンの解釈に基づき、検索エンジンへパラメータを渡し、セッション毎の情報と履歴集合を管理する。プロトコルエンジンには Index Data 社の YAZ[74] を利用した。

メタデータ変換部は既存の書誌データを Dublin Core に基づくメタデータに変換する。現在は

JAPAN/MARC から Dublin Core への変換機能を持っている。JAPAN/MARC のデータ [79] を元に Dublin Core の RDF 表現 [90] に変換した。RDF[91][92] は外部表現として XML (eXtensible Markup Language) [20] を用いるため、実際には Dublin Core は XML で表記される。データの変換は Perl スクリプトにより行なった。変換時に JAPAN/MARC のフィールド (サブフィールド) と Dublin Core の項目の間のマッピングを行なった。変換の際のマッピングの定義は、表 5.1 に示す。マッピングは Dublin Core に対して一対多となるように行ない、1 つの書誌項目が複数の Dublin Core エlement に含まれることはない。また、1 つの Dublin Core Element に対して、複数のデータ項目が存在した場合は、RDF におけるリスト要素を表す Bag を用いて表現した。なお、JAPAN/MARC で独自に定義されている外字については、適宜変換を行なった (なお、Â, Î, Û, Ê, ÔなどはA, I, U, E, Oに、その他の追加文字は空白文字に変換した。また、JAPAN/MARC では長音 (ー) の文字がマイナス (-) 記号になっているので、カタカナ・平仮名の直後に現れるマイナス記号は長音に変換した)。

表 5.1 Dublin Core と JAPAN/MARC の対応関係

Dublin Core	JAPAN/MARC	フィールドの意味
Title	251-9\$A	タイトルと責任表示に関する事項：本タイトル
	251-9\$B	タイトルと責任表示に関する事項：タイトル関連情報
	251-9\$D	タイトルと責任表示に関する事項：巻次、回次、年次等
	280\$A	叢書名に関する事項：叢書名
	280\$B	叢書名に関する事項：叢書番号
	280\$D	叢書名に関する事項：副叢書名
	280\$F	叢書名に関する事項：副叢書番号
	281-3\$A	シリーズに関する事項：本シリーズ名
	281-3\$B	シリーズに関する事項：シリーズ名関連情報
	281-3\$D	シリーズに関する事項：シリーズ番号
	281-3\$S	シリーズに関する事項：下位シリーズ名
	281-3\$T	シリーズに関する事項：下位シリーズ番号
	281-3\$X	シリーズに関する事項：シリーズの ISBN
	291-9\$A	多巻ものの各巻のタイトルと責任表示に関する事項：タイトル
	291-9\$B	多巻ものの各巻のタイトルと責任表示に関する事項：副書名
	291-9\$D	多巻ものの各巻のタイトルと責任表示に関する事項：巻次、回次、年次等
	354\$A	原タイトル注記：翻訳資料の原タイトル
	551-9\$A	タイトル標目 (タイトル関連情報の読み等を含む)：カタカナ形
	551-9\$X	タイトル標目 (タイトル関連情報の読み等を含む)：カタカナ形ローマ字形
	580\$A	叢書名標目：カタカナ形

(前ページの続き)

Dublin Core	JAPAN/MARC	フィールドの意味
	580\$X 581-3\$A 581-3\$X 591-9\$A 591-9\$X	叢書名標目：ローマ字形 シリーズのタイトル標目：カタカナ形 シリーズのタイトル標目：ローマ字形 多巻ものの各巻のタイトル標目：カタカナ形 多巻ものの各巻のタイトル標目：ローマ字形
Creator	251-9\$F 751-9\$A 751-9\$B 751-9\$X	タイトルと責任表示に関する事項：責任表示 著者標目：カタカナ形 著者標目：漢字形 著者標目：ローマ字形
Subject	650\$A 650\$B 650\$X 658\$A 658\$B 658\$X 677\$A 685\$A 685\$X	個人件名：カタカナ形 個人件名：漢字形 個人件名：ローマ字形 一般件名：カタカナ形 一般件名：漢字形 一般件名：ローマ字形 NDC：分類記号 NDLC 分類：分類記号（またはかな付） NDLC 分類：ローマ字付分類記号
Description	350\$A 377\$A	一般注記：一般注記 内容注記：内容に関する注記
Publisher	270\$B	出版・領布に関する事項：出版者、領布者等
Contributor	281-3\$F 291-9\$F 781-3\$A 781-3\$B 781-3\$X 791-9\$A 791-9\$B 791-9\$X	シリーズに関する事項：シリーズに関する責任情報 多巻ものの各巻のタイトルと責任表示に関する事項：著者表示 シリーズの著者名標目：ローマ字形 シリーズの著者標目：巻次の読み シリーズの著者標目：漢字形 多巻ものの各巻の著者標目：ローマ字形 多巻ものの各巻の著者標目：巻次の読み 多巻ものの各巻の著者標目：漢字形
Type	(該当無し)	
Date	270\$D	出版・領布に関する事項：出版、領布年月
Identifier	010\$A 020\$B 905\$A 906\$A	国際標準図書番号：ISBN 全国書誌番号：全国書誌番号（JP 番号） NDL の請求記号：請求記号 NDL の印刷カード番号：印刷カード番号
Format	(該当無し)	
Language	101\$A	著作の言語（翻訳物に適用）：テキストの言語
Source	(該当無し)	

(前ページの続き)

Dublin Core	JAPAN/MARC	フィールドの意味
Relation	(該当無し)	
Coverage	270\$A	出版・領布に関する事項：出版地、領布地等
Rights	(該当無し)	
(該当無し)	001	レコード識別番号：レコードコントロール番号
	020\$A	全国書誌番号：国名コード
	100\$A	一般的処理データ
	101\$C	著作の言語（翻訳物に適用）：原文の言語
	251-9\$W	タイトルと責任表示に関する事項：資料種別表示
	261\$A	並列タイトルに関する事項：並列タイトル
	265\$A	版に関する事項
	275\$A	形態に関する事項：特定資料種別と資料の数量
	275\$B	形態に関する事項：大きさ
	275\$E	形態に関する事項：付属資料
	360\$A	装丁と定価に関する事項：装丁
	360\$B	装丁と定価に関する事項：本体価格
	360\$C	装丁と定価に関する事項：税込価格
	386\$A	ファイル内容に関する注記（コンピュータファイル）：ファイル内容注記
	387\$A	システム要件に関する注記（コンピュータファイル）：システム要件注記
	551-9\$B	タイトル標目（タイトル関連情報の読み等を含む）：漢字形（所在フィールドの識別子）
	551-9\$D	タイトル標目（タイトル関連情報の読み等を含む）：巻次の読み
	580\$B	叢書名標目：漢字形（所在フィールドの識別子）
	580\$D	叢書名標目：叢書番号
	581-3\$B	シリーズのタイトル標目：漢字形（所在フィールドの識別子）
	581-3\$D	シリーズのタイトル標目：巻次等の読み
	591-9\$B	多巻ものの各巻のタイトル標目：漢字形（所在フィールドの識別子）
	591-9\$D	多巻ものの各巻のタイトル標目：巻次の読み
	677\$V	NDC：NDC 版次

データベース部はメタデータの検索を行なう。データベースの検索エンジンには、全文検索エンジン Namazu[75] を採用した。本システムでは、Dublin Core の RDF 表現を解析し、エレメントの内容を抽出するフィルタを Namazu のインデクサ (mknmz) のフィルタ機構に組み込んだ。このフィルタでは、XML パーサとして Perl モジュール XML::Parser を利用し、文字列がどのエレメントに含まれているかを判別して、Namazu のフィールド用インデクスに登録する。現在本システムに登録されているデータ件数は 1,593 件である。

本システムでは、Z39.50 の Bib-1 の Use アトリビュートに追加された Dublin Core 用の 15 項目

[93] に対応した検索ができる（表 5.2 参照）。この 15 項目のアクセスポイント指定は Dublin Core の項目と 1 対 1 の関係にあるため、アトリビュートの扱いがあいまいになることはない。Z39.50 プロトコルでの検索要求は、内部的に Namazu の検索式に変換され、検索エンジンに渡される。また、検索要求としては履歴検索も扱える。

表 5.2 Bib-1 に追加された Dublin Core エlement

Z39.50 Bib-1 Use Attribute		Dublin Core
Name	Value	
DC-Title	1097	Title
DC-Creator	1098	Creator
DC-Subject	1099	Subject
DC-Description	1100	Description
DC-Publisher	1101	Publisher
DC-Date	1102	Date
DC-ResourceType	1103	Type
DC-ResourceIdentifier	1104	Identifier
DC-Language	1105	Language
DC-OtherContributor	1106	Contributor
DC-Format	1107	Format
DC-Source	1108	Source
DC-Relation	1109	Relation
DC-Coverage	1110	Coverage
DC-RightsManagement	1111	Rights

5.3 検索例

実際に Z39.50 クライアントで本システムに接続し、Dublin Core アクセスポイントで検索した例を図 5.2 に示す。検索結果を返戻する際は、SUTRS 形式で返戻する。

この例では、

1. DC-Title で「河童」を検索: ヒット件数 1 件
2. DC-Creator で「赤川次郎」を検索: ヒット件数 2 件

という検索を行っており、表示されているレコードは 2 番目の検索によるものである。

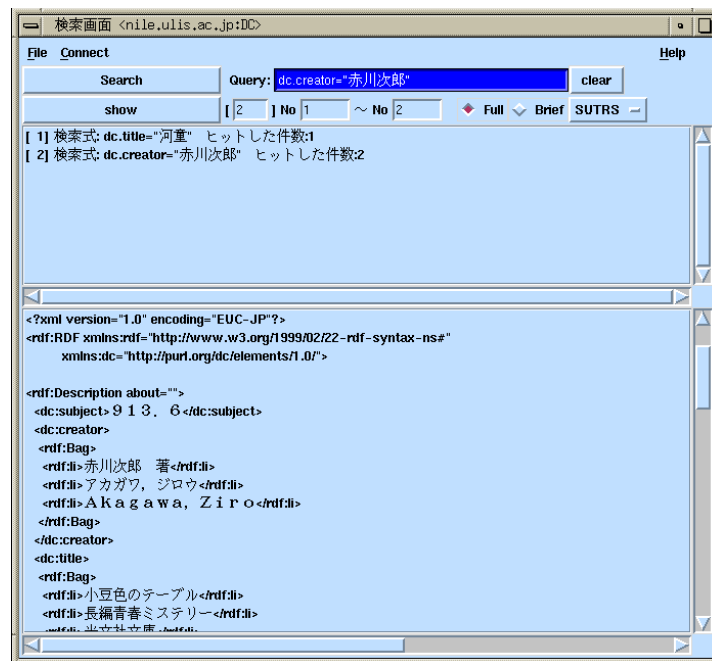


図 5.2 Dublin Core を共通スキーマとした Z39.50 検索システムの検索例

5.4 考察

5.4.1 関連研究

多様な情報資源を容易に利用できる仕組みを作りたいという要求は、どの分野でも共通の話題であり、それぞれの立場から様々な提案が行われている。この節では特に Z39.50 に関わる研究について考察する。

Z39.50 は ZIG (Z39.50 Implementors Group) によって規格の検討と合意がなされており、現在 ZIG におけるこのマッピング問題に対処することを目指した議論の中で、複数のアトリビュートセットを同時に扱うことを目的とした Attribute Architecture[94] の定義が進められている。この枠組の中心は Cross-Domain Attribute Set[95] であり、これは、Dublin Core を基本にした 13 項目のアクセスポイント指定からなるアトリビュートセットで、分野を横断した検索に利用することが期待されている。しかし、この方法は現行の Z39.50 システムに対して、Bib-1 から Cross-Domain Attribute Set に変更を求めることになり、アトリビュートセットの移行によって相互利用性が損なわれる可能性がある。

また、JAPAN/MARC と Z39.50 との関わりについて石田 [70][96] は、Bib-1 で流用するのではなく JAPAN/MARC 専用のアトリビュートセットを提案している。石田のシステムは JAPAN/MARC

に特化したアトリビュートセットを設計し、実際にシステムを公開して評価実験を行っている。しかし、この手法では JAPAN/MARC を検索する際には JAPAN/MARC の構造に従った検索ができるものの、JAPAN/MARC 以外のデータに対しては適用できず、Bib-1 との互換性もなくなってしまいう欠点がある。

Z39.50 と Dublin Core を結合した例としては図書館情報大学電子図書館 (ULIS-DL) [62] の Z39.50 サービスがある。ULIS-DL はサブジェクトゲートウェイを目指したメタデータの構築を行っており、ULIS Core と呼ばれる Dublin Core に基づいたエレメントセットを用いている。ULIS-DL はもともと Dublin Core を基本として作られたシステムであるので、標準検索プロトコル Z39.50 と組み合わせるのは自然であり、データそのものが元々 Dublin Core で作成されているため、システム構築時にアトリビュートセットのマッピングによる問題が生じることはない。しかし、今後 ULIS-DL がサブジェクトゲートウェイとしての能力を高め、他のデータを対象にする場合には本システムで研究しているスキーマ変換機能が必要になるだろう。

また、齋藤らは WoPEc と WAGILS という 2 つのメタデータを Dublin Core とマッピングし、Dublin Core のエレメントで検索するシステム [97] も構築している。

以上のように、Z39.50 の視点を重視するのか、JAPAN/MARC などある特定データの流通性を重視するのか、多様なメタデータの相互利用性を重視するのかによってアプローチが異なるが、本研究では Z39.50 の視点を踏まえた相互利用性を重視する立場からシステム構築を行なった。

5.4.2 JAPAN/MARC データの Dublin Core への変換

本論文は Z39.50 と Dublin Core のシステム基盤構築に焦点があるため、JAPAN/MARC と Dublin Core のマッピングについては深く立ち入らなかった。マッピングについては今後の問題として、今回のマッピング方針と将来展望について述べる。

Dublin Core には、Dublin Core Simple (DCS) と Dublin Core Qualifier (DCQ) という考え方の相違による 2 つのタイプがある。DCS は 15 項目のデータ要素をさらに細かく分けることはしない書き方で、DCQ はこの基本要素を細かく分けて記述する書き方である。今回は DCS を採用し、データは並列に並べた。JAPAN/MARC のタグに表されているデータのみを Dublin Core に変換し、JAPAN/MARC レコードのデータ部に含まれていない項目は Dublin Core への変換を行わず該当なしとした。データのマッピングについては現在検討中で、DCQ による記述も検討している。

5.5 まとめ

本章では JAPAN/MARC を例として Dublin Core を共通スキーマとしてマッピングを行ない、Z39.50 の Bib-1 の Dublin Core 用アクセスポイントを通して JAPAN/MARC を検索できるシステムの構築について述べた。この Dublin Core を共通スキーマとした Z39.50 検索システムにより

JAPAN/MARC だけでなく他のデータについても Dublin Core を共通スキーマにすれば本システムを利用して Z39.50 で検索できる。

第 6 章

結論

本研究では、相互連携の考え方に基づく情報提供システムについて、特に用語体系と検索システムを対象とした研究を行い、以下で述べる 4 つのシステムを構築することにより、相互連携の考え方の有効性を示した。

第 1 のシステムは、複数用語体系ブラウジングシステムである。このシステムは、用語間の階層構造を持つ複数の用語体系を単純化して、自由に組み合わせながら重ね合わせることができる特徴を持ち、重ね合わせた結果をグラフィカルな表現としてブラウジングできる。そのため利用者は、用語体系の構造を容易に把握でき、かつ、複数用語体系間での観点の違いも発見できる。

第 2 のシステムは、用語体系を提供する枠組みとして Web サービスを利用するシステムである。各用語体系を Web サービスを通じてネットワーク上から自由にアクセスでき、それらを共通に利用できる枠組みを考案し、実装した。この Web サービスを使うことにより、各用語体系のデータを共通の方法で容易に取得できるため、各用語体系を組み合わせ使用して使用する応用システムを様々な形で構築でき、用語体系の提供、利用、再利用が柔軟に行えるようになることを示した。

第 3 のシステムは、日本語書誌データを対象とした Z39.50 検索サーバである。情報検索の相互運用を考慮した通信プロトコルである Z39.50 プロトコルに対応し、検索エンジンとして全文検索エンジンを採用することにより、大規模データに対する実用的な検索システムとして検索サービスが行えることを示した。また、Web 検索で利用した検索システムをそのまま Z39.50 を通じて提供できるようになるため、検索資源の共有につながることを示した。

第 4 のシステムは、Dublin Core メタデータを共通スキーマとした Z39.50 サーバである。このシステムでは、Z39.50 サーバ一般が持つ検索スキーマと検索対象データとの間の実装依存の問題を解決するために、検索対象データと検索用スキーマの両者に Dublin Core メタデータを用いる手法を採用した。

本研究全体を通じたテーマは、情報提供システムの相互連携性である。多様な利用者と多様な情報提供システムが存在する現在の環境においては、その情報提供システムの構築と運用にはいくつもの問題がある。そこで、情報提供システム間の相互連携性を向上させることにより、これらの問

題が解決できることを、前述の 4 つのシステム構築を通じて示した。

本研究は 3 つの研究に大別される。第 1 は検索システム間の相互連携であり、第 2 は用語体系間の相互連携、第 3 はそれらを発展させた異種情報提供システム間の相互連携である。これらそれぞれについて相互連携可能な方式を提案し、その有効性を示した。前述の 4 つのシステムとの関連を言えば、検索システム間の相互連携は第 3・4 のシステムにおいて、用語体系間の相互連携は第 1 のシステムにおいて、異種情報提供システム間の相互連携は第 2 のシステムにおいて、それぞれ研究し、その相互連携手法を示した。

本研究で構築した情報提供システムについて、それぞれのシステム間の連携可能性を表 6.1 に示す。表中の「○」は本論文で既に述べたようにシステム構築を通じて連携可能であることを示した組み合わせであることを示す。また表中の「◎」は、システム構築は行っていないものの、原理上連携が可能である組み合わせであることを示す。また「◎」を付した Google WS は、Google 社が提供する Web サービスシステムである。

表 6.1 情報提供システム間の連携可能性

		用語体系WS		検索システム			Graphviz WS
		EDR (WS)	ODP (WS)	Z39.50-JP	Z39.50-DC	Google WS※	
用語体系WS	EDR (WS)	—	◎	○	○	◎	◎
	ODP (WS)	◎	—	○	○	◎	◎
検索システム	Z39.50-JP	○	○	—	◎	○	○
	Z39.50-DC	○	○	◎	—	○	○
	Google WS※	◎	◎	○	○	—	○
	Graphviz WS	◎	◎	○	○	○	—

本研究では相互連携性の要素として、以下の 3 つの要素を重視した。

- 共有

情報提供システムを共通に使えることにより、情報資源の共有が可能となり、別々の情報提供システムを切り替えて使うなどの応用も可能となる。

- 連携

異なる情報提供システム同士を連携させることによって、それぞれのシステムの長所を生かして、短所を補いあうことができるようになる。また、新たな情報提供システムを構築する際にも、一からシステムを構築するだけでなく、既存のシステムを組み合わせで新たなサービスを提供することが可能となる。

- 独立

情報提供システムを独立して管理、運用できることであり、様々な目的で多様な情報提供シ

システムが作られる可能性が広がる。

この 3 つの要素を前述の 4 つのシステムに当てはめると以下ようになる。

- 共有

第 1、第 2 のシステムでは、用語体系における見出し語と階層関係に的を絞って、これを共通化してとらえ、様々な用途で作成される用語体系を共通に扱えるようにした。また第 3、第 4 のシステムにおいては、Z39.50 プロトコルに基づく検索システムを構築し、検索サービスを提供することにより、世界中のどのようなクライアントからでも Z39.50 プロトコルに基づいて共通したアクセスを行えることを示した。

- 連携

第 1 のシステムの論文では、複数の用語体系を自由な組み合わせで重ね合わせることで、体系間の連携を行う手法について述べた。第 2 のシステムでは、Web サービスをバックエンドとした応用システムにより、様々なサービスと用語体系とを連携させるシステムが容易に構築できることを示した。さらに第 3、第 4 のシステムでは、Z39.50 プロトコルによってクライアント・サーバ間のやりとりが共通化されていることを活かして、Z39.50 プロトコルを用いる他システムと連携させることができる。

- 独立

用語体系はそもそも、それぞれの用途、目的に沿って作られており、独立した性質を持っているため、第 1、第 2 のシステムはその独立性に対応した。とりわけ第 2 のシステムでは、ネットワーク上に分散して提供される用語体系を想定し、これに対応する Web サービスによる提供手法を考案し、その提供手法に基づくシステムを構築した。また第 3、第 4 のシステムは、検索システムをネットワーク上で分散的に提供することを前提としたシステムである。

相互連携可能な情報提供システムの提供という観点から、まず第一に、共有および連携を可能にすることが重要である。これは、相互システム間のやりとりを共通化し、もしくは、相互システムを共有しうるものとするところである。さらに、このように共有し連携できるようにしたシステムは、それぞれの提供者が、ネットワーク上の分散環境の下で独立して提供することが必要となる。つまり、個々の組織や個人によって作成された情報は、その組織・個人の下からそのまま発信できるようにになっていることが、情報の効率的な提供、発信において重要となると考えるからである。

例えば、用語体系を相互連携させる際には、複数体系を同時に利用するためお互いの持つ用語とその用語が持つ関係を共有できるものとする必要がある。つまり、まず相互連携のうち共有のプロセスとして、その用語体系の持つ用語を見出し語として抽出し、さらに見出し語間の関係を他の体系と対応できるように取り出す必要がある。そうして共有できるようにした用語体系を実際に連携させる過程では、連携の目的に沿った複数用語体系の操作のやり取りを一種のプロトコルとして

きちんと定義する必要がある。この定義されたプロトコルは相互連携する体系間で共有できることが望ましい。本研究ではこの共有過程について、第 1 のシステムにおいて研究し、複数の様々な用語体系の見出し語とその階層関係の共有ができることを示した。また、連携過程については第 2 のシステムにおいて研究し、Web サービスに基づく相互連携の際のやり取りの共有および WSDL 仕様といった XML に基づくやり取りの定義を行う手法を示し、相互連携が行えることを示した。

また、検索システムを相互連携させる際には、検索式の構成要素や検索システム間のやり取りを共通化する必要がある。つまり、まず共有のプロセスとして、検索の際の要素として検索語やアクセスポイント指定などの検索式の構成要素や、検索結果の表現を取り上げ、それらを他システムとの間で共有できるよう定義し、他システムとの連携の際にはこの共有された定義を用いて相互連携を行う必要がある。本研究では、Z39.50 プロトコルを幅広く共有が進みつつある標準として取り上げ、そのプロトコルで規定され共有されている検索の際のデータ構造やアクセスポイント指定を用い、日本語書誌データや Dublin Core メタデータなどのデータを対象として実験システムを構築し、他の Z39.50 クライアントから検索が行え、相互連携に用いることができることを示した。

以上述べてきたように本研究では、既存システムでは不十分であった問題を解決するために、多様な情報提供システムの相互連携の考え方、すなわち、検索システム間の相互連携、用語体系間の相互連携、異種情報提供システム間の相互連携の考え方に基づいたシステムの研究を行った。その結果、用語体系と検索システム以外の情報提供システムにおいても相互連携の考え方に基づくシステムを提供することにより、それぞれのシステムの長所をさらに活かした情報提供システムができるようになり、ユーザの多様な情報アクセス要求に応えることができるようになると思う。

参考文献

- [1] ティム・バーナーズ-リー. 高橋徹 (訳). Web の創成: World Wide Web はいかにして生まれどこに向かうのか. 毎日コミュニケーションズ, 279p. (ISBN 4-8399-0287-9).
- [2] 益岡竜介, 木庭袋圭祐. World-Wide Web (WWW). 情報処理. Vol. 36, No. 12, 1995, pp. 1155–1165.
- [3] 原田昌紀. サーチエンジン徹底活用術. 東京, オーム社, 1997, 250p. (ISBN 4-274-06230-9).
- [4] Yahoo!. (参照 2002-08-03) <<http://www.yahoo.com/>>.
- [5] AltaVista. <<http://www.altavista.com/>>.
- [6] Clifford A. Lynch. “The Z39.50 Information Retrieval Standard: Part I: A Strategic View of Its Past, Present and Future”. D-Lib Magazine. 1997.
<<http://www.dlib.org/dlib/april97/04lynch.html>>.
- [7] ANSI/NISO Z39.50-1995. Information Retrieval (Z39.50): Application Service Definition and Protocol Specification. <<ftp://ftp.loc.gov/pub/z3950/official/>>.
- [8] 宇陀則彦, 江草由佳, 高久雅生, 石塚英弘. Z39.50 による日本語書誌データ検索システム. 情報知識学会誌. Vol. 9, No. 2, 1999, pp. 1–15.
- [9] Dublin Core Metadata Initiative. The Dublin Core Metadata Element Set Version 1.1. last update 1999-07-02. <<http://purl.org/dc/documents/rec-dces-19990702.htm>>.
- [10] 杉本重雄. Dublin Core Metadata Element Set : 現在の状況と利用例. デジタル図書館. No. 14, 1999, pp. 3–18.
- [11] 高久雅生, 江草由佳, 宇陀則彦, 石塚英弘. Z39.50 による書誌データ検索システムの構築: Dublin Core を共通スキーマとして. デジタル図書館. No. 16, 1999, pp. 97–106.
- [12] 高久雅生, 江草由佳, 伊東千夏, 石塚英弘. データの再利用が可能な複数用語体系ブラウジングシステムの構築. 情報知識学会誌. Vol. 13, No. 1, 2003, pp. 10–22.
- [13] W3C. Web Services. <<http://www.w3.org/2002/ws/>> (参照 2003-09-04).
- [14] 日本ユニテック Digital Xpress 編集部 (編). SOAP/UDDI/WSDL Web サービス技術基礎と実践 徹底解説. 東京, 技術評論社, 2002, 429p. (ISBN 4-7741-1616-5).
- [15] 高久雅生, 江草由佳, 石塚英弘. Web サービスによる用語体系データの提供とその応用システ

ム. 情報知識学会誌, Vol. 14, No. 1, 2004, p. 11–22.

- [16] ODP: Open Directory Project. <<http://dmoz.org/>> (参照 2002-12-04).
- [17] EIC ネット. <<http://www.eic.or.jp/>> (参照 2002-12-04).
- [18] Tim Berners-Lee, James Hendler, Ora Lassila. The Semantic Web. Scientific American, May 2001, pp. 34–43.
- [19] 特集: セマンティック Web. 情報処理. Vol. 43, No. 7, 2002, pp. 707–750.
- [20] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler. Extensible Markup Language (XML) 1.0 (Second Edition). World Wide Web Consortium, 2000, REC-xml-20001006, <<http://www.w3.org/TR/REC-xml>> (最終更新 2000-10-06).
- [21] 伊東千夏, 高久雅生, 江草由佳, 大懸晶子, 宇陀則彦, 石塚英弘. 用語体系の統合及び可視化システムの試作: 用語体系の補完及び相違点の発見を目的として. 情報処理学会 情報学基礎研究会 研究報告. Vol. 101, No. 65, 2001, pp. 1–8.
- [22] 日本電子化辞書研究所. EDR 電子化辞書 2.0 版仕様説明書. 東京, 日本電子化辞書研究所, 1999, TR-006, <<http://www.iiijnet.or.jp/edr/>>.
- [23] 日本図書館協会編. 日本十進分類表新訂 8 版機械可読データファイル (NDC・MRDF8). 東京, 日本図書館協会, 1989, (電子媒体).
- [24] James Clark. “XSL Transformations (XSLT) Version 1.0”. World Wide Web Consortium, 1999, REC-xslt-19991116, <<http://www.w3.org/TR/xslt>> (最終更新 1999-11-16).
- [25] AT&T: Graphviz, <<http://www.research.att.com/sw/tools/graphviz/>> (最終更新 2000-10-19).
- [26] Apache Software Foundation. Apache HTTPD Project. <<http://httpd.apache.org/>> (参照 2002-12-04).
- [27] Larry Wall, Tom Christiansen, Randal L. Schwartz. (近藤 嘉雪訳). プログラミング Perl 改訂版. 東京, オライリージャパン, 1997, 759p. (ISBN 4-900900-48-6).
- [28] The Sleepycat Software Homepage. <<http://www.sleepycat.com/>> (参照 2002-12-04).
- [29] 高林哲. Sary: Suffix Array のライブラリとツール. <<http://sary.namazu.org/>> (最終更新 2002-09-18).
- [30] Daniel Veillard. The XSLT C library for Gnome. <<http://xmlsoft.org/XSLT/>> (参照 2002-12-04).
- [31] 後藤高信, 鈴木俊介, 後藤智範. 階層構造を持つ用語データのための Browsing Tool. 情報知識学会 第 7 回研究報告会 論文集, 1999, pp. 57–60.
- [32] 市丸夏樹. シソーラスブラウザ xthes homepage. <<http://lang.is.kyushu-u.ac.jp/pub/xthes/>> (参照 2002-12-04).
- [33] Dan Brickley. RDFViz. <<http://www.ilrt.bris.ac.uk/discovery/rdf-dev/>>

- rudolf/rdfviz/> (参照 2002-12-04) .
- [34] Peter Pirolli, Stuart K. Card, Mija M. Van Der Wege. The Effect of Information Scent on Searching Information Visualizations of Large Tree Structures. Advanced Visual Interfaces 2000, Palermo, pp. 161–172, <http://www.inxight.com/pdfs/white_papers/info_scent.pdf> (参照 2002-12-18) .
 - [35] José Luis Ambite, Yigal Arens, Eduard Hovy, Andrew Philpot, Luis Gravano, Vasileios Hatzivasiloglou, Judith Klavans. Simplifying Data Access: The Energy Data Collection Project. Computer. Vol. 34, No. 2, 2001, pp. 47–54.
 - [36] Eduard Hovy, Andrew Philpot, José Luis Ambite, Yigal Arens, Judith Klavans et al. Data Acquisition and Integration in the DGRC's Energy Data Collection Project. Proceedings of the dg.o 2001 Conference, 2001. <<http://www.dgrc.org/dgrc/dgo2001/papers/session-2/hovy.pdf>> (参照 2002-12-04) .
 - [37] Ryutaro Ichise, Hideaki Takeda, Shinichi Honiden. Rule Induction for Concept Hierarchy Alignment. In Proceedings of the IJCAI-01 Workshop on Ontology Learning (OL-2001), pp. 26–29, 2001. <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-38/ichise_IJCAI-OL.pdf> (参照 2002-12-18) .
 - [38] 荻野孝野. 異なる言語の繋ぎ手としての多言語シソーラスへの試み. 人工知能学会誌. Vol. 15, No. 4, 2000, pp. 567–574.
 - [39] Naoki Asanoma. Alignment of Ontologies: WordNet and Goi-Taikei. Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources, Pittsburgh, 2001. <<http://www.seas.smu.edu/~rada/mwnw/papers/WNW-NAACL-233.pdf.gz>> (参照 2002-12-18) .
 - [40] N. F. Noy, M. A. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In the Proceedings of the 17th National Conference on Artificial Intelligence, 2000. <[http://smi-web.stanford.edu/pubs/SMI_AbSTRACTS/SMI-2000-0831.html](http://smi-web.stanford.edu/pubs/SMI_Abstracts/SMI-2000-0831.html)> .
 - [41] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, M. A. Musen. Creating Semantic Web Contents with Protégé-2000. IEEE Intelligent Systems. Vol. 16, No. 2, 2001, pp. 60–71.
 - [42] Natasha Noy. The PROMPT Tab. <<http://protege.stanford.edu/plugins/prompt/prompt.html>> (参照 2002-12-04) .
 - [43] John F. Sowa. Glossary. <<http://www.jfsowa.com/ontology/gloss.htm>> (最終更新 2001-08-25) .
 - [44] 日本図書館協会分類委員会改訂. 日本十進分類法：新訂 8 版. 東京, 日本図書館協会分類委員会, 1978, 635p.
 - [45] 建築工事標準分類. (参照 2001-10-12) <<http://dbnet.watanabe.arch.waseda>

- ac.jp/code.html>.
- [46] Zthes: a Z39.50 Profile for Thesaurus Navigation. <<http://zthes.z3950.org/>> (参照 2003-07-30) .
 - [47] Erik Christensen, Francisco Curbera, Greg Meredith, Sanjiva Weerawarana. “Web Services Description Language (WSDL) 1.1”. W3C Note 15 March 2001. <<http://www.w3.org/TR/wsdl>>.
 - [48] Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, Dave Winer. “Simple Object Access Protocol (SOAP) 1.1”. W3C Note 08 May 2000. <<http://www.w3.org/TR/soap/>>.
 - [49] Google Web APIs. <<http://www.google.com/apis/>>.
 - [50] オブジェクト指向スクリプト言語 Ruby. <<http://www.ruby-lang.org/ja/>>.
 - [51] Hiroshi Nakamura. SOAP4R. <<http://www.jin.gr.jp/~nahi/Ruby/SOAP4R/>>.
 - [52] BDB. <<http://moulon.inra.fr/ruby/bdb.html>> (参照 2003-07-16) .
 - [53] Rickard E. Faith, Bret Martin. RFC 2229: A Dictionary Server Protocol. 1997, <<http://www.ietf.org/rfc/rfc2229.txt>>.
 - [54] 上田修一. Z39.50 とその可能性. 情報の科学と技術. Vol. 48, No. 3, 1998, pp. 126–133.
 - [55] 牛崎進. Z39.50: IR サービスの通信プロトコル. カレントアウェアネス. No.175, 1994, pp. 4.
 - [56] 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) .
 - [57] 松林正己. 標準情報検索プロトコル Z39.50 の国際的展望. 情報の科学と技術. Vol. 48, No. 3, 1998, pp. 144–155.
 - [58] Index Data. Z39.50 Target Information. last update 1999-06-30, Continually updated. <<http://www.indexdata.dk/targettest/targetstat.shtml>>.
 - [59] 牧野順子. Z39.50 対応電子図書館パッケージ・システム：OCLC SiteSearch System. 情報の科学と技術. Vol. 48, No. 3, 1998, pp. 172–174.
 - [60] 佐藤康之. MARUZEN Z39.50 Gateway-CAT—Z39.50 商用サービス利用事情—. 情報の科学と技術. Vol. 48, No. 3, 1998, pp. 167–171.
 - [61] 堀江隆. JOISwithSTN システムの開発—Z39.50 によるゲートウェイ—. 情報管理. Vol. 40, No. 8, 1997, pp. 692–701.
 - [62] 図書館情報大学附属図書館. 図書館情報大学デジタル図書館(ULIS-DL)(参照 1999-05-08). <<http://lib.ulis.ac.jp/>>.
 - [63] 早稲田大学学術情報システム(WINE)(参照 1999-05-08). <<http://wine.wul.waseda.ac.jp/>>.
 - [64] Titech Digital Library (東京工業大学電子図書館) . last update 1999-03-25. <<http://tdl.libra.titech.ac.jp/>>.
 - [65] 小島 明, 田中 純, 篠原 正紀, 落合 崇道. 大学での取り組みが進む電子図書館. NTT 技術ジャー

- ナル. Vol. 11, No. 1, 1999, pp. 55–58.
- [66] 安達淳. 電子図書館成立の条件. デジタル図書館. No. 1, 1994, pp. 23–27. <http://www.dl.ulis.ac.jp/DLjournal/No_1/adachi/adachi.html>.
- [67] 学術情報センター事業部データベース課. 電子図書館サービスの実施状況. 学術情報センターニュース. No. 47, 1998, p. 9. <<http://www.nacsis.ac.jp/CNEWS/N47/09.html>>.
- [68] 祖父江 真一, 新井 康平, 落合 治. 地球観測衛星データ用の相互運用プロトコル. 情報処理. Vol. 39, No. 3, 1998, pp. 222–228 (1998.3).
- [69] 原修. 情報検索プロトコル ANSI/NISO Z39.50 に対応した基本サーバの構築. 学術情報センター・セミナー研究レポート. 文部省学術情報センター編. 東京, 文部省学術情報センター, 1997, pp. 1–40.
- [70] 石田茂. Z39.50 と日本語書誌目録の連携に関する考察. 情報処理学会データベースシステム/情報学基礎研究報告. Vol. 99, No. 39, 1999, pp. 81–88.
- [71] 安齋宏幸, 山本毅雄, 石塚英弘. Z39.50 を用いた日本語書誌情報サーバの試作. 情報処理学会情報学基礎研究会報告. Vol. 96, No. 116, 1996, pp. 9–16.
- [72] 江草由佳, 真野泰久, 宇陀則彦, 石塚英弘. Z39.50 プロトコルによる日本語書誌データ情報検索システム. 第 6 回研究報告会講演論文集. 情報知識学会. 東京, 1998-5. 情報知識学会, 1998, pp. 29–36.
- [73] 安齋宏幸. Z39.50 の技術解説. 情報の科学と技術. Vol. 48, No. 3, 1998, pp. 134–139.
- [74] Index Data. The YAZ Toolkit. last update 1999-3-22. <<http://www.indexdata.dk/yaz/>>.
- [75] 高林哲. 全文検索システム Namazu. last update 1999-03-15. <<http://openlab.ring.gr.jp/namazu/>>.
- [76] 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. (ISBN 4-7973-0691-2).
- [77] 安齋宏幸. WWW による JAPAN/MARC の提供実験. つくば, 図書館情報大学, 1994. 卒業論文.
- [78] 黒澤正彦, 西村徹編. マークをうまく使うには: 機械可読目録入門. 東京, 三洋出版貿易, 1986, 310p. (ISBN 4-87930-030-6).
- [79] 国立国会図書館. Japan/MARC マニュアル: 図書編. 第 2 版. 東京, 国立国会図書館, 1998, 186p. (ISBN 4-87582-527-7).
- [80] 奈良先端科学技術大学院大学自然言語処理学講座. 日本語形態素解析ツール・茶筌. (参照 1999-5-8). <<http://cactus.aist-nara.ac.jp/lab/nlt/chasen.html>>.
- [81] Index Data. YAZ User's Guide and Reference: Introduction(参照 1999-5-4). <<http://www.indexdata.dk/yaz/yaz.shtml>>.
- [82] 安齋宏幸. 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.

- [83] 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. 卒業論文.
- [84] Hasebe Kigen, Nakamoto Ken'iti, Yamamoto Takeo. An information retrieval system on internet for languages without obvious word delimiters. Proceedings of International Symposium on Digital Libraries 1995. Tsukuba, University of Library and Information Science, 1995-8. pp. 181-185.
- [85] Z39.50 Maintenance Agency. Character Set and Language Negotiation (2). 1998. <<http://lcweb.loc.gov/z3950/agency/defns/charsets.html>>.
- [86] R. Denenberg, J. Kunze, D. Lynch eds. RFC 2056 : Uniform Resource Locators for Z39.50. 1996. <<http://www.ietf.org/rfc/rfc2056.txt>>.
- [87] 九州大学. 多機関 OPAC 横断検索. (参照 1999-05-08). <<http://zeus.lib.kyushu-u.ac.jp/htdocs/Qindex.html>>.
- [88] JIS X 0806 (ISO 23950). 情報検索 (Z39.50) 応用サービス定義及びプロトコル仕様. 東京, 日本規格協会, 1999, 272p.
- [89] Cliford A. Lynch. "Building the Infrastructure of Resource Sharing: Union Catalogs, Distributed Search, and Cross-Database Linkage". LIBRARY TRENDS. Vol. 45, No. 3, 1997, pp. 448-461.
- [90] Eric Miller, Paul Miller, Dan Brickley. Guidance on expressing the Dublin Core within the Resource Description Framework (RDF). last update 1999-07-01. <<http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/>>.
- [91] Ora Lassila, Ralph R. Swick, ed. Resource Description Framework (RDF) Model and Syntax Specification. World Wide Web Consortium, 1999, REC-rdf-syntax -19990222. <<http://www.w3.org/TR/REC-rdf-syntax/>>.
- [92] Dan Brickley, R.V. Guha, ed. Resource Description Framework (RDF) Schema Specification. World Wide Web Consortium, 1999, REC-rdf-schema-19990303. <<http://www.w3.org/TR/PR-rdf-schema/>>.
- [93] Ralph LeVan. Dublin Core and Z39.50. Draft Version 1.2. last update 1998-02-02. <<http://purl.org/DC/documents/notes-levan-19980202.htm>>.
- [94] Z39.50 Attribute Architecture. Version 1.1. last update 1999-07-09. <<http://lcweb.loc.gov/z3950/agency/attrarch/arch.html>>.
- [95] The Z39.50 Cross-Domain Attribute Set. Version 1.4. last update 1999-08-04. <<http://www.oclc.org/~levan/docs/crossdomainattributeset.html>>.
- [96] 石田茂. Z39.50 JAPAN/MARC 対応関連仕様と評価用試作システムの説明. デジタル図書館. No. 15, 1999, pp. 40-56.
- [97] 齋藤ひとみ, 宇陀則彦, 石塚英弘. Dublin Core Metadata Element Set による複数メタデータの検索. デジタル図書館. No. 11, 1998, pp. 48-55.

謝辞

本研究では、研究の計画、実行、論文執筆などの全ての過程で多くの皆様にご支援頂きました。ここにお名前を記し、深く感謝の意を表したいと思います。

はじめに、本研究全般にわたり常に力強い励ましとご指導を頂きました主任指導教官の石塚英弘先生（筑波大学教授）に深く感謝致します。杉本重雄先生（筑波大学教授）には、ご多忙の中、副指導教官をお引き受け頂き、熱心かつ親身なご指導を賜りました。阪口哲男先生（筑波大学助教授）には、副指導教官をお引き受け頂くとともに、情報メディアシステム分野における身近な相談役としても、様々なご助言を頂きました。特に研究環境の整備の面では、鋭いご指摘に気付かされることも多く、様々な局面でご支援くださいました。感謝致します。

田畑孝一先生（筑波大学教授）、石川徹也先生（筑波大学教授）には、本学位論文の予備審査および本審査を通じて、研究手法の評価、標題、用語法など様々な角度から、詳細かつ的確なご指摘と多くのご助言を頂きました。感謝致します。

山本毅雄先生（国立情報学研究所教授）には、本学位論文の本審査を通じて、さらには、2000年度までの図書館情報大学在籍の折りは石塚研究室の大学院生ゼミにも参加頂き、鋭いご指摘、ご助言とともに温かい励ましの言葉を頂きました。感謝致します。

石塚研究室のゼミに参加頂いた皆様や情報メディアシステム分野所属の教官および学生の皆様には、私の研究について議論、指摘を頂きました。同時に、他の多くの方々の真摯な研究活動を間近で拝見し、議論に参加する機会を得ることができたことは、私にとってかけがえのない財産となりました。感謝致します。

特に、図書館情報大学の学部入学以来の同期、江草由佳さんとは、共同研究の中で様々な議論を戦わせました。特に、研究が行き詰まった際など苦しい時でも常にそばにあり、叱咤激励して頂き、私の研究を推進する大きな原動力となるとともに、心強い存在でした。深く感謝致します。

同じく図書館情報大学の学部入学以来の同期、伊東千夏さんは、私が用語体系を対象とした研究を行う重要な契機となる共同研究と一緒にさせて頂き、研究上のご助言を頂きました。また、学生生活の中でも公私両面にわたってお世話になりました。深く感謝致します。

宇陀則彦先生（筑波大学助教授）、長谷川秀彦先生（筑波大学助教授）には、研究にあたっての心構えから、私生活における注意、学内における事務手続きまで、様々なことを教えて頂くとともに

に、様々な面で支援してくださいました。感謝致します。

一方、システム開発においては、身近な研究室の先生方や仲間、先輩、後輩たちと同様、いやそれ以上にインターネット上で活躍するハッカーの皆様の活動とその成果である有用なソフトウェアに助けられる点が多々ありました。とりわけ、グラフィレイアウトソフトウェア Graphviz の開発者の皆様、SOAP4R 作者のなひさん、全文検索システム Namazu を開発する Namazu Project の皆様、スクリプト言語 Ruby の作者まつもとゆきひろさんなど、多くのオープンソース技術者の皆様に感謝致します。また、オープンソース技術者の皆様から頂いたものはソフトウェアそのものだけでなく、オープンソース方式における開発による様々なノウハウの公開・共有の精神、効率的なシステム開発手法や先進的なシステム機能に刺激され、大いに励みになり、研究を行っていく原動力を得ることもできました。感謝致します。

最後に、私の学生生活を常に温かく支えてくれた両親に深く感謝致します。

発表文献一覧

査読制度のある学術雑誌に掲載の論文

- 江草由佳*, 高久雅生, 石塚英弘. 分散環境におけるデータベースに関する情報の共有システム. 情報知識学会誌, vol.14, no.3, 2004, pp.1-17.
高久雅生*, 江草由佳, 石塚英弘. Web サービスによる用語体系データの提供とその応用システム. 情報知識学会誌, vol.14, no.1, 2004, pp.11-22.
高久雅生*, 江草由佳, 伊東千夏, 石塚英弘. データの再利用が可能な複数用語体系ブラウジングシステムの構築. 情報知識学会誌, vol.13, no.1, 2003, pp.10-22.
- 江草由佳*, 高久雅生, 宇陀則彦, 石塚英弘. Z39.50 データベース選択支援環境. 情報知識学会誌. vol.11, no.2, 2001, pp.1-10.
- 宇陀則彦*, 江草由佳, 高久雅生, 石塚英弘. Z39.50 による日本語書誌データ検索システム. 情報知識学会誌. vol.9, no.2, 1999, pp.1-15.

査読制度のある国際会議録に掲載の論文

- Yuka EGUSA*, Masao TAKAKU, Hidehiro ISHIZUKA. Network Community Oriented Information Sharing System for Databases. International Symposium on Digital Libraries and Knowledge Communities in Networked Information Society 2004 (DLKC'04), 2004-03, Tsukuba, pp.64-71.

その他

- 伊東千夏*, 高久雅生, 江草由佳, 大懸晶子, 宇陀則彦, 石塚英弘. 用語体系の統合及び可視化システムの試作：用語体系の補完及び相違点の発見を目的として. 情報処理学会 情報学基礎研究会 研究報告. vol.101, no.65, 2001, pp.1-8.
- 江草由佳*, 高久雅生, 宇陀則彦, 石塚英弘. 情報共有による Z39.50 データベース選択支援

環境. 第 9 回研究報告会講演論文集. 情報知識学会, 東京, 2001-05. 情報知識学会, 2001, pp.37-44.

- 高久雅生*, 江草由佳, 宇陀則彦, 石塚英弘. Z39.50 による書誌データ検索システムの構築 : Dublin Core を共通スキーマとして. 第 16 回 デジタル図書館ワークショップ. つくば, 1999-11. デジタル図書館. no.16, 1999, pp.97-106.
- 高久雅生*, 石塚英弘. VRML と Java を用いたオリエンテーリング教育用ゲームの作成. 第 6 回研究報告会講演論文集. 情報知識学会, 東京, 1998-05. 情報知識学会, 1998, pp.53-59.