

データ型別索引に基づく異種情報の検索システムと それを用いた地球観測情報レジストリの実現

小島 功[†] 木本 正裕^{†‡} 的野 晃整[†]

[†] 産業技術総合研究所 情報技術研究部門 〒305-8568 茨城県つくば市梅園 1-1-1
[‡] ビジネスサーチテクノロジー株式会社 〒101-0041 東京都千代田区神田須田町 1-2-1
 E-mail: [†] {isao.kojima,a.matono}@aist.go.jp, [‡] kimoto@bsearchtech.com

あらまし 本論文では、近年の全文検索エンジンで支援されている、数値や日付などテキスト型以外での検索索引を構築できる機能を利用した、タグ付きのデータ等に対する柔軟な情報検索システムを提案する。検索対象データの特定のフィールドやそのデータ型に対応して索引を定義・構築したり、ランキングなどのカスタマイズを可能にすることで柔軟な検索を実現するとともに、スキーマに依存しない全文検索エンジンの性質を生かして異種のスキーマのデータを統合的に検索したり、スキーマの変化にも柔軟に対応することができる。

また、その実例として、地球観測情報資源に対するレジストリを実装したので報告する。ここでは、異なる XML スキーマに基づくデータを、緯度経度やポリゴン、雲量などによって統合的に検索することができる。また大量の検索結果に対しても十分な応答時間を実現することができる。

キーワード 情報検索、全文検索エンジン、索引、異種データ統合、地球観測情報、OGC, レジストリ

Implementation of a Registry System for Earth Observation Metadata Using Type-wise Search Engine Indices

Isao KOJIMA[†] Masahiro KIMOTO^{†‡} and Akiyoshi MATONO[†]

[†] Information Technology Research Institute(ITRI), National Institute of Advanced Industrial Science and Technology(AIST) 1-1-1, Umezono Tsukuba-City, Ibaraki 305-8568, Japan

[‡] Business Search Technology Corporation 1-2-1 Kanda Suda-cho, Chiyoda-ku, Tokyo, 100-0041 Japan

E-mail: [†] {isao.kojima,a.matono}@aist.go.jp, [‡] kimoto@bsearchtech.com

Abstract In this paper, we propose an architecture of information retrieval system which is based on type-wise search engine indices. The indices are constructed for various data types and are customized and extensible. Based on this architecture, we implemented a service registry of earth observation metadata. Using this system, it is possible to search over a large amount of heterogeneous metadata with various data schema

Keyword Information Retrieval, Search Engines, Index, Heterogeneous Data Integration, Earth Observation, Open Geospatial Consortium(OGC), Registry, Metadata

1. まえがき

1.1. 背景

Lucene などに代表される全文検索エンジン[1-4]は、応答時間に優れたキーワード検索や、検索条件に対するランキング、膨大な検索結果に対するページング、辞書に基づいた類義語検索など、大量の情報に対する情報検索のための優れた機能を有し、多様な応用に利用されている。そのため、キーワードを用いたテキス

トの全文検索だけでなく、日付や数値による比較演算や範囲検索の支援など、全文検索エンジンに要求される機能は高機能化、多様化する一方である。

この要求に対して、全文検索エンジンそのものの機能拡張や、全文検索の機能を関係データベースの拡張プラグインとして組み入れるなど、データベース問い合わせ処理と情報検索機能の良い連携を実現するための研究開発が数多く進められている[5-6,9-11]。

著者らは、GEO Grid[7]と呼ばれる地球観測情報を基礎とした情報基盤を研究開発中であり、この中で地理情報や計算サービスの記述（メタデータ）に対する検索サービス（レジストリ）[8]を実装しつつある。メタデータに対する検索要求は全文検索を中心とするが、観測日付や雲量、緯度経度といった異なるデータ型に対応した検索条件も処理できる必要があり、しかもメタデータの数や種類が膨大なので、この実現にあたっては、情報検索機能とデータベース問い合わせ処理の効果的な連携が必要になっている。

1.2. アプローチと比較：

情報検索とデータベース処理の統合については、Senna[5]や Oracle の全文検索プラグイン[6]などのように、全文検索の機能をデータベースシステムに取り込む方法が知られている。PostgreSQL の GiST あるいは GIN[9]のように、全文検索を含め拡張可能で汎用の転置索引を提供するものや、データベースに対する横断的なキーワード検索[10]、関係代数に対して情報検索の機能を拡張するもの[11]など、多様な視点からの研究が数多く行われている。前述のレジストリサービスでも、関係データベースを基礎に実装するものが殆ど[12]である。

しかし、多くの Web 応用では単純なフォーム検索に基づく処理がほとんどで、複雑な結合などを含む SQL の高い記述力が必要とされない場合が多い。一方で、全文検索エンジンの有するページングやランキング、スキーマレスのデータ検索といった特徴が、そのような機能を持たない関係データベースの実装に組み入れることで失われやすい、といった課題もある。

そこで本論文では、多くの全文検索エンジンで実現されつつある、データ型ごとに異なる検索索引を構築する機能を基礎として、データ型ごとに索引そのものやランキングを定義・拡張できる枠組みを提供することで、異種のデータを柔軟に検索できるシステムを提案する。

これは例えば、地理情報のメタデータにおいて XML で記述されるポリゴン型に対して R 木[13]型の索引を定義することで空間検索を支援したり、スキーマが異なる 2 つの日付型に対して同一の日付型の索引を提供することで異なるスキーマの日付データに対する横断的な検索を可能にするようなシステムである。

このための基本機能として、特定のフィールドに特定の索引を定義し、それに基づいたデータの解析、索引への格納、索引を使った検索などの機能を提供する API を定義して実装しつつある。

他のシステム等と比較すると、関係データベースに全文検索機能や独自の索引機能を組み込んだシステム

に対して、ランキングやページング、スキーマに依存しない検索対象といった全文検索エンジンの利点をそのまま生かせるという特徴がある。全文検索エンジンに関しては、GeoLucene[14]、LocalLucene[15]など、全文検索を基礎に独自の索引機能を提供するシステムも多数知られているが、これらの殆どはソフトウェアの改造であり、索引やランキングをカスタマイズ、プラグイン化するものではなく柔軟性が十分ではないと考えられる。

2. アーキテクチャ：

2.1. 概要：

提案システムの概要構成を図 1 に示す。基本的には、各データ型に固有の検索索引と、それらの間の集合演算を行う機能より構成される。基本的な構成は従来の全文検索ソフトウェアと等しいが、任意の型のデータを任意の索引アルゴリズムで検索できる拡張性を有する点が異なり、利用者は独自のデータ型やそれを支援する独自の索引アルゴリズムを導入することで、索引を拡張することが可能となっている。

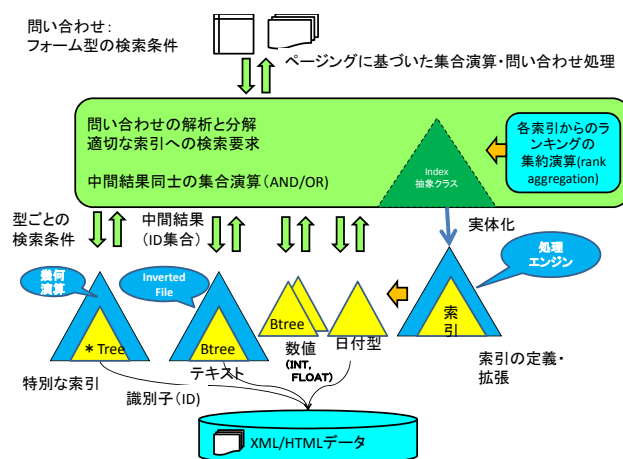


図 1 提案システムの基本構成

提案システムで支援する検索の機能は、フィールドに対する検索条件の組み合わせであり、HTML フォームや XForms[16]のような、フォーム形式を基礎とする検索条件式を扱う。検索条件はデータ型ごとに設けられた索引に対する部分検索条件に分解され、それぞれの索引からの検索結果がデータの識別子を用いて集合演算されて最終的な結果が返される。これは、基本的には関係データベースにおける 1 つの表の制約演算、あるいは複数の属性索引を用いた結合演算と類似の枠組みであるが、全文検索エンジンの特性に基づき、以下のような特色がある。

1. 演算の部分処理による、高い応答性能：全文検索エンジンは基本的にページングに基づいた検索結

果を戻す処理方式を提供している。たとえば検索結果が 100 万個であっても、1 ページ分のデータが得られた段階で制御を戻せるので、利用者は即時に結果を閲覧できる。次ページ以降のデータ処理を次の検索要求が来た段階でいちいち処理すれば、全部の処理の終了を待つ必要がない。各索引からの中間結果同士の演算も、最初に表示すべき 1 ページ分のデータが得られた段階で利用者に制御を戻すことができる。この方法は検索結果の全個数が途中の段階では分からないという欠点があるが、索引からの中間結果が膨大な場合でも、応答時間の大幅な改善が可能である。また、残りの処理は次ページの要求の段階で処理するだけでなく、バックグラウンドで実行するなど、最適化処理の余地が大幅に増える。

2. **結合が高速**：文書の識別子による結合は、識別子に例えば整数値などを用いることができるため、一般の結合演算に比べてデータ量も小さく、高速に処理できる。例えば、100 万個の中間結果同士の結合であっても、数メガバイト程度の領域しか使わないのでメモリ上で保持できるし、各索引の検索結果が識別子のソート順で出力できれば単なるマージで良い。順番が保持できなくても、高速なソートが可能であるなど、大規模なデータ同士の集合演算が十分実用的であると考えられる。
3. **ランキングの集約**：情報検索の場合、各索引は独自のランキングに基づく結果を返すので、最終的に集約・統合[17,18]する必要がある。これは既に多くの研究が知られており、本システムではこれらを索引同様に定義・組み込むことができる。このうち、識別子を用いて処理できる手法は結合と同様にメモリ上で高速に実現できる。

以上のように、本システムの構成は、特に膨大な検索結果が得られるような場合でも十分な応答時間を提供できる、という利点がある。

後述の地理情報の検索応用では、地図に基づく GUI により緯度経度で検索しようとするとき、検索結果の個数が利用者の意図とあわずに膨大な結果を返すことが多い。このような場合に対し、いったん全部の結果を得ないと利用者に制御が戻せない実装では、応答時間に問題が発生しやすく、本提案のようなシステム構成の利点は大きいと考えられる。

2.2. スキーマの支援

2.1 に示したように、本システムは全文検索を基礎とするので、対象となるデータはスキーマを持たず、構造に基づく検索の処理手法を考える必要がある。

提案システムは、後述のように API で提供されるた

め、これを用いて次のような方法で関係データベースに格納したのと同等の機能を支援できる。

1. 同じデータ型でも異なるフィールドに異なる索引を構築し、スキーマに基づいた検索はフィールドに対応する索引を用いる。データ型全体に対する横断的な検索は、各フィールドに対応する索引からの結果の OR で実現する。
2. データ型の索引で複数のフィールドを統合し、フィールドに依存した検索は、フィールドに基づくタグの含まれたデータを全文検索索引から取得し、それらの AND で処理をする。

なお、ポリゴンなど入力形式と格納形式が異なるものや、検索インターフェイスが異なるものも扱えるように、API の提供するクラスは抽象化される必要がある。

2.3. クラス構成

基本的にこのシステムは 3 章のように応用向けの検索システム実現に用いられるため、API で提供することが有益である。なお、従来の全文検索ソフトウェアで用いられるテキスト型データを転置ファイルに基づいて索引づける方法は、本システムの 1 応用としての実現と位置付けられる。

図 2 に提案システムのクラス図を示す。

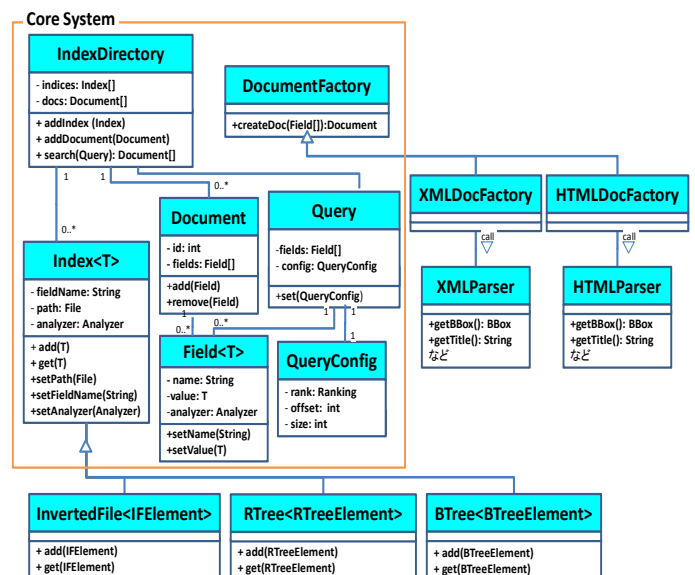


図 1 提案システムのクラス構成

オレンジ色の枠で囲まれたクラスが提案システムの中心的なクラスで、それ以外は利用者が定義することを前提としたクラスである。

まず、**Document** クラスから説明する。これは、検索対象のデータ(Document)を扱うもので、Document クラスは、Lucene と同様に、Field クラスのインスタン

ス集合をメンバに持ち、集合演算のための識別子 id を持つ。

Field クラスは、名前とそれに対応した値を持ち、検索対象データ内の特定のフィールドに対応する。Lucene の Field クラスは、文字列型しか扱えないが、提案システムの Field クラスでは任意の型のデータを扱うことが可能である。例えば、名前 "date" と、Date 型の値 "2009-01-01" のペアで構成される Field インスタンスと名前 "BBOX" と、Polygon 型の値 "(0,0)(0,1)(1,1)(1,0)" で構成される Field インスタンスを一つの Document インスタンス内に格納することが可能である。また Field クラスには、Analyzer クラスのインスタンスを定義することで、ストップワードや表記ゆれなどの問題を解消する。

Index クラスは抽象クラスで、これを継承したサブクラスとして、転置ファイルのためのクラスや、B 木索引のためのクラス、R 木索引のためのクラスなど、任意の索引アルゴリズムを実装する。Index クラスのメンバである、fieldName はこの索引が扱う Field の名前で、path は実際に索引データが格納される場所を示し、ranking は要素の順序を定義するクラスである。

Ranking クラスも同様に抽象クラスで、サブクラスとして、例えば、数字の大小に基づいた順序や、要素間の距離に基づいた順序などを定義したクラスを実装する。この Ranking クラスのサブクラスによる順序定義を各索引ごとに直接関連づける。

IndexDirectory クラスは、データの格納、および検索のインターフェイスとなるクラスである。IndexDirectory クラスは、複数の Index インスタンスをメンバとし、例えば、Date 型データを格納できる名前が "date" である B 木索引や、Polygon 型データを格納できる名前が "bbox" である R 木索引などをメンバとして追加する。また、文字列型やテキスト型データを格納できる匿名の転置ファイル索引は、デフォルトで追加されることを前提としている。IndexDirectory クラスに Index を追加した後にデータを格納する。データの格納は、与えられた文書を構文解析し、Field インスタンスの集合から構成される Document インスタンスを生成し、それを IndexDirectory クラスに追加する。この際、追加される Document 内のすべての Field の名前に対応して、存在するように索引を事前に定義しておく必要がある。例えば、Date 型の B 木索引 "date" や、Polygon 型の R 木索引 "BBOX"、文字列型の転置ファイル索引が定義されていた場合、Date 型で名前が "date" である Field は格納できるが、Date 型で名前が "作成日時" であるような Field は格納できない。

IndexDirectory クラスに対して検索を行なう場合、Query クラスのインスタンスを渡すと、解として

Document の列あるいは id の列が返される。Query クラスも Document クラスと同様に、Field インスタンスの集合をメンバに持ち、さらに解のサイズやオフセット、ソートのキーとなる Filed などを持つ QueryConfig をメンバに持つ。

後述のように実装では、id に基づいた Field 間の AND や OR 演算などを支援するが、簡単のため割愛する。この段階でランキングの集約を行うことがフレームワークとしては可能であるが、これも詳細は今後の課題となっている。

3. カタログサービスの実装：

3.1. 要求要件

次に、本提案システムを用いた地球観測情報資源のレジストリの実装について述べる。まず、OGC(Open Geospatial Consortium) の規定する、CSW(Catalog Service Web)という、メタデータの検索・登録のための規格[8]をサポートする必要がある、これには以下のような機能要件がある。

1. HTTP GET/POST に基づき、資源に対するフォーム検索をサポートすること。
2. 日付型、雲量などの数値型、BBOX(Bounding BOX)のような空間型、および全文検索の検索をサポートすること。
3. Transaction/Harvest といったメタデータ収集・登録の仕組みを有すること。

GEO Grid における要件：さらに、以下のような GEO Grid 固有の要求が考えられる。

- A) 異種・多様なプロファイル（スキーマ）をサポートすること。 衛星画像だけでなく、さまざまな地物のレジストリや、その上でのサービスのレジストリとしての機能が要求され、異種のメタデータスキーマの支援と、それに対する統合的な検索が必要である。スキーマの変更や追加や要素の拡張などにも柔軟に対応する必要がある。
- B) 大量データに対する検索を効果的にサポートすること。 提案システムでは数百万というオブジェクトを扱うが、既に述べたようにある緯度経度で検索をした場合、答えの個数が数万個になったり、あるいは逆に一つもないような場合がある。このような場合に対しても、十分な応答性能と必要な結果を効果的に検索・表示できること。
- C) オントロジに基づく検索をサポートすること。 地物に対する検索要求はたとえば検索用語間に階層関係や包含関係があるので、単にキーワード検索だけでは不十分である。

3.2. アプローチ

これらの要求に対して、従来の CSW の実装では、関係データベースを基礎とした実装がほとんど[12]であるが、我々は本論文で示したアーキテクチャに基づき、全文検索ソフトウェアを基礎に実現することとした。これは GEO Grid における上述の要件に対する利点を考慮したためである。

- A) 全文検索ソフトウェアに基づく実装は、基本的にスキーマレスであるため、多様なスキーマの扱いやスキーマの追加変更に対して極めて柔軟な対応が可能である。
- B) ランキング・ページングといった全文検索ソフトウェアが標準的に持つ機能は、結果の個数が膨大な場合のページングや、ランキングに基づく条件式の緩和や絞り込みなどの検索に有利である。
- C) オントロジによる検索は、検索語の階層関係を展開する程度でも十分な場合が多く、これは類語辞書など情報検索に基づく手法で対応できる。

また、CSW の支援においても、次のような利点が考えられる。

- 1. 多くの全文検索ソフトウェアは HTTP GET/POST による検索インターフェイスを有しているので、実装が容易である。
- 2. データ型ごとの検索は型ごとの索引を用いた本論文での手法により実装することができる。
- 3. 登録については、特に Harvest は全文検索ソフトウェアにおいて基本的な機能であるクロールに基づくデータ収集の機能であるため、追加的な実装が少なく済む。

以上のような特徴から、本提案システムに基づく実装が適していると考えられる。

3.3. 実装の詳細

本実装では、WISE[19]と呼ばれる全文検索ソフトウェアを基礎に実装を行った。これは、商用の全文検索ソフトウェアであり、1)データ型毎の索引構築の機能、2)HTTP に基づく API、3)類語辞書による検索の拡張などが可能で、本レジストリの実装に必要な基本的な機能を有している。

これに対して、提案システム実現のため、2 章で示したクラス構成の実装を行いながら、データ型索引の構築を行った。まず、対象データの XML スキーマの基本データ型に基づき索引を構築し、あわせて同じタイプのスキーマ要素は統合した。したがって、例えば日付型のデータはすべて同じ索引で管理され、作成日付、修正日付、観測日付といった特定属性に対する検索は、それぞれのタグを指定して行うことになる。次に、ポリゴンなど、XML スキーマの基本データ型では処理できないものや、特別な検索索引を構築したい場合などは、前述のインターフェイスに従って索引用のライブラリを登録することで独自の索引を構築した。本実装では、ポリゴンに対して範囲検索を行うため、R-Tree のライブラリを基礎とした索引を定義し、緯度経度で示されるポリゴンをこの索引に登録した。さら

にその上で、CSW の実装と GEO Grid で管理するためのデータの登録などを行った。

対象データは、GEO Grid における衛星画像や、地質図などのメタデータであり、XML のドキュメント数で約 130 万件である。スキーマは地理国際標準の ISO19115[18]、その国内標準の JMP2.0[21]、Web サービス記述の ebRim に基づくもの[22]など多岐にわたる。

さらに、問い合わせ側の機能として、類語辞書を利用したオントロジ検索を実装しつつある。これは全文検索によるキーワード検索において、類語辞書を利用してキーワード展開を行う機能である。これにより、上位概念を下位概念に展開したり、下位概念のキーワードに対して上位概念の用語を含んだ結果を返すなどの検索が可能となる。

3.4. 検索例

検索条件例を図 3 に示す。これは CSW に基づき、複数のデータ型に対する検索条件を論理演算で結合した問い合わせ条件である。

```
<?xml version="1.0" encoding="UTF-8" ?>
<csw:GetRecords xmlns:csw="http://www.opengis.net/csw" service="CSW"
  version="2.0.0" resultType="results" outputSchema="JMP"
  outputFormat="text/xml" startPosition="1" maxRecords="100">
  <csw:Query typeNames="csw:GetRecord">
    <csw:Constraint version="1.0.0">
      <Filter xmlns:ogc="http://www.opengis.net/ogc">
        <And>
          <!-- ISO|RIM|JMP -->
          <FileType>JMP</FileType>
          <!-- 検索開始日 -->
          <PropertyIsGreaterThanOrEqualTo>
            <PropertyName>createDate</PropertyName> <Literal>2000-12-01</Literal> </PropertyIsGreaterThanOrEqualTo>
          <!-- 検索終了日 -->
          <PropertyIsLessThanOrEqualTo>
            <PropertyName>createDate</PropertyName> <Literal>2008-12-01</Literal> </PropertyIsLessThanOrEqualTo>
          <!-- 雲量(JMPの場合はなし) -->
          <PropertyIsLessThanOrEqualTo>
            <PropertyName>cloudCoverage</PropertyName> <Literal>20</Literal> </PropertyIsLessThanOrEqualTo>
          <!-- 全文検索 -->
          <PropertyIsLike wildCard="%" singleChar="_" escape="\\">
            <PropertyName>AnyText</PropertyName> <Literal>鹿児島</Literal> </PropertyIsLike>
          <!-- Bounding Box -->
          <BBOX> <PropertyName>ows:BoundingBox</PropertyName>
            <gml:Envelope xmlns:gml="http://www.opengis.net/gml">
              <gml:lowerCorner>128.0287109375 30.370703125</gml:lowerCorner>
              <gml:upperCorner>146.793359375 47.0259765625</gml:upperCorner>
            </gml:Envelope>
          </BBOX>
        </And> </Filter> </csw:Constraint> </csw:Query> </csw:GetRecords>
```

図 3 検索条件例

図の XML は、createDate (作成日付) がある範囲で、(GreaterThanOrEqualTo と LessThanOrEqualTo で指定)、雲量がある量以下のレコードをある緯度経度範囲 (BBOX) の中から求めるものである。この検索は、日付、雲量、全文、ポリゴン (BBOX) それぞれのデータ型に対応した索引に対する検索に分割されて処理され、AND/OR の集合演算を行って結果を返すものである。図 4 に検索結果のメタデータの例を示す。

検索結果データのスキーマ、特に緯度経度に対する索引は独自に定義されているため、検索条件の与え方と結果のデータのスキーマとが一致していない点に注意されたい。


```
<?xml version="1.0" encoding="UTF-8" ?>
<getRecordsResponse xmlns="http://www.opengis.net/cat/csw/2.0.2" xmlns:ns2="http://www.opengis.net/og"
xmlns:ns3="http://www.opengis.net/gml" xmlns:ns4="http://www.w3.org/1999/xlink" xmlns:ns5="http://www.opengis.net/ows"
xmlns:ns6="http://purl.org/dc/elements/1.1/" xmlns:ns7="http://purl.org/dc/terms/" xmlns:ns8="http://www.isotc211.org/2005/gco"
xmlns:ns9="http://www.isotc211.org/2005/gmd" xmlns:ns10="http://www.opengis.net/gml/3.2"
xmlns:ns11="http://www.isotc211.org/2005/gts" xmlns:jmp20="http://gate.gsi.go.jp/ch/jmp/">
<SearchResults numberOfRecordsReturned="100" numberOfRecordsMatched="98">
<jmp20:MD_Metadata>
<jmp20:identificationInfo>
<jmp20:MD_DataIdentification>
<jmp20:citation>
<jmp20:title>20万分の1数値地質図幅「鹿児島」</jmp20:title>
<jmp20:date>
<jmp20:date>2004-12-01</jmp20:date>
<jmp20:dateType>002</jmp20:dateType>
</jmp20:date>
</jmp20:citation>
<jmp20:abstract>20万分の1地質図幅は、既存の地質資料に基づいて、国土地理院発行の20万分の1地勢図のそれぞれの区画毎に編纂した地質図です。この数値地質図は、原資料である地質図を数値化したものです。</jmp20:abstract>
<jmp20:pointOfContact>
<jmp20:individualName>CD-ROM編集：鹿野和彦・長谷川 功、表示システム作成：巖谷敏光・川畑 晶、データ編集：宮崎純一・巖谷敏光、ベクトルデータ作成：宮崎純一・中島和敏、ラスター画像データ作成：川畑 晶、メタデータ作成：渡辺和明、表訂：中島和敏
</jmp20:individualName>
<jmp20:organisationName>産業技術総合研究所地質調査総合センター</jmp20:organisationName>
<jmp20:contactInfo>
<jmp20:onlineResource>
<jmp20:linkage>http://www.gsj.jp/HomePageJP.html</jmp20:linkage>
</jmp20:onlineResource>
<jmp20:contactInfo>
<jmp20:role>009</jmp20:role>
</jmp20:contactInfo>
<jmp20:descriptiveKeywords>
<jmp20:MD_Keywords>
<jmp20:keyword>20万分の1、ベクトル、数値地質図、鹿児島、中国西部、九州及び南西諸島、日本</jmp20:keyword>
</jmp20:MD_Keywords>
</jmp20:descriptiveKeywords>
<jmp20:language>
<jmp20:isoCode>jpn</jmp20:isoCode>
</jmp20:language>
<jmp20:characterSet>023</jmp20:characterSet>
<jmp20:topicCategory>008</jmp20:topicCategory>
<jmp20:extent>
<jmp20:geographicElement>
<jmp20:EX_CoordinateBearingBox>
<jmp20:extentReferenceSystem>
<jmp20:authority>
<jmp20:title>測量法</jmp20:title>
<jmp20:date>
<jmp20:date>1960-07-01</jmp20:date>
<jmp20:dateType>003</jmp20:dateType>
</jmp20:date>
</jmp20:authority>
<jmp20:code>TD / B, L</jmp20:code>
<jmp20:extentReferenceSystem>
<jmp20:westBoundCoordinate>130</jmp20:westBoundCoordinate>
<jmp20:eastBoundCoordinate>131</jmp20:eastBoundCoordinate>
<jmp20:southBoundCoordinate>31.333333</jmp20:southBoundCoordinate>
<jmp20:northBoundCoordinate>32</jmp20:northBoundCoordinate>
</jmp20:EX_CoordinateBearingBox>
</jmp20:MD_Metadata>
</SearchResults>
</getRecordsResponse>
```

(以下略)

図 4 検索結果メタデータの例

3.5. 性能評価

本システムはなお実装中であるが、基本的な性能の評価を行った。現在格納しているデータとしては、ISO19115[20]、JMP[21]、ebRIM[22]に基づくメタデータであり、それぞれの個数は 1,208,718 個と 222 個、20 個である。主としてデータ数の多い ISO に基づくデータで評価を行った。平均レコードサイズはハードウェアのスペックは Xeon2.66Ghz、メモリ 8GB、ディスク 250GBx4 (RAID0)、OS は CentOS 64bit である。

基本検索性能：まず、検索結果の URL の列を返すまでの応答時間(ms、縦軸)を評価した、これは検索エンジンの提供する応答時間についての基本性能である。特に、ページング (1 度に取得する結果のレコード件数、横軸) についての影響を評価した。

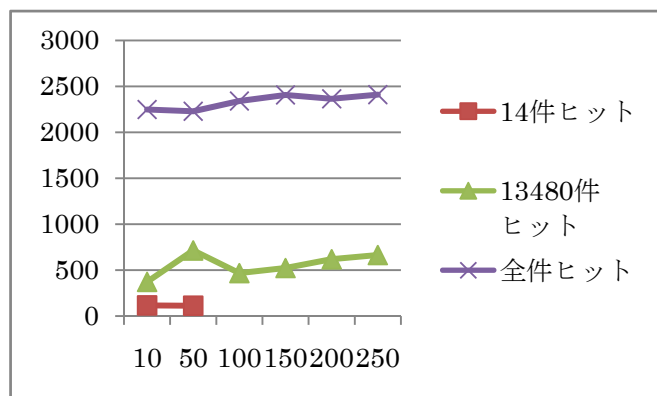


図 5.1 基本応答性能

図 5.1 のように、結果データの個数が支配的なため、

検索結果の個数に応じて応答時間が増えるが、100 万個程度の結果でもおおむね実用に耐える応答時間である。ページングの影響は小さい。

型ごとの索引とその論理演算：次に各データ型ごとの索引と、その論理演算について評価した。評価軸は同じで、検索結果としてその型の全件を返す。

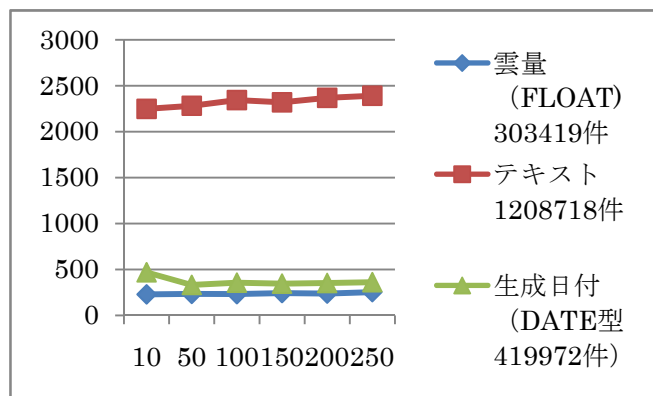


図 5.2 データ型ごとの応答性能

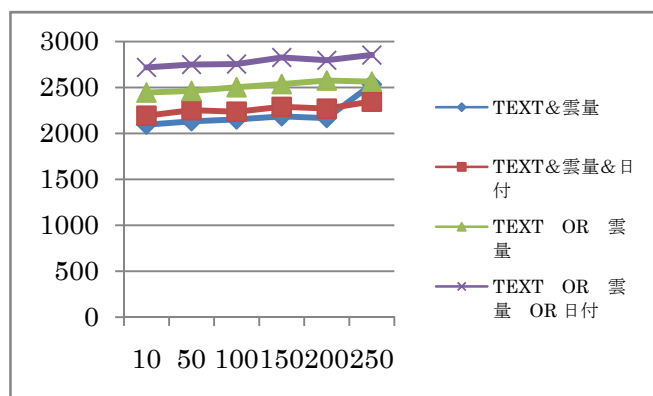


図 5.3 5.2 の結果同士の論理演算

図 5.2 および 5.3 のように、基本的にはデータ型に対する検索結果と、その論理演算の結果には大きな差がない。これは、各索引からの検索結果が識別子の順なため、単純なマージ処理で済むからである。図 5.3 のように、この方法は多数の索引を用いても性能低下が少なく、索引の数に対するスケーラビリティがあると考えられる。

利用者定義の索引を含む性能：次に、利用者定義の索引処理として、ポリゴン型とそのオーバーラップを処理できる索引とライブラリ[23]を導入した場合の評価を、大規模な取得件数の場合とあわせて行った。これは、R 木のライブラリでは点の包含関係しか判定できないので、BBOX のようなジオメトリ演算のためにはライブラリで処理をする必要があるからである。全文検索の結果と、BBOX 処理を結合し、処理するレコード数に対する応答時間を調べた。結果を図 6 に示す。

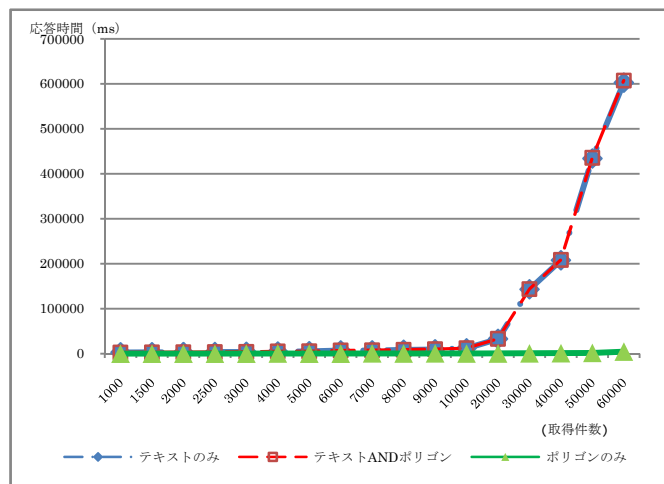


図 6 大規模な取得を行った場合。

図 6 ではテキストのみの場合とテキストとポリゴン処理の場合の 2 本の線が殆ど重なっており、索引の組み合わせによる処理のオーバーヘッドは小さく、本実装における利用者定義の索引の組み込みの問題点は少ないと考えられる。一方、取得件数がある一定以上の量になると急激に性能が悪化する。これは取得結果のテキストをメモリ上に保持できないという理由が考えられる。しかし、一度に数万件を取得する、という利用は現実的でなく、実用上の問題点としては無視できると考えられる。

4. 問題とまとめ

本論文では、データ型ごとの索引の定義機能に基づく拡張可能な情報検索システムとその基本機能、応用システムとしての地理情報レジストリの実装について述べた。本システムは、OpenLayers[24]を用いた UI に接続され、メタデータの検索環境を構築しつつあり、索引処理の結果が膨大な場合の最適化など改良を行う予定である。

次の課題としてはランキングの集約がある。ランキングの集約は様々な研究が知られているが、一般に結合演算以上の計算量がかかるものが多い[17]ので、その効果的な処理方法が課題となる。また、すでに述べた通り、ページ単位で結果を返す場合には、正確な結果の個数が最後に分かるといった欠点がある。

実装の方向性としては、格納対象データの定義ファイルを読み込んで自動的に索引付けしたり、そのデータ型に応じた Parser を結びつけて、データ収集時にデータを自動的に解析、登録できるようなユーティリティなどを提供することで、応用の構築を容易にする予定である。

5. 謝辞

日頃からご議論いただく情報技術研究部門 GEO Grid プロジェクト関係諸氏に深謝します。また、北川博之、石川博各教授をはじめとする筑波大学と静岡大学の各関係諸氏にも感謝いたします。なお、本研究の一部は、情報通信研究機構(NICT)の委託研究「ダイナミックネットワーク技術の研究開発」および科研費「不均質なペタバイト級時空間センサデータの統合利用基盤」により実施した。

参 考 文 献

- [1] Lucene, <http://lucene.apache.org/java/docs/>
- [2] Fast, <http://www.fastsearch.co.jp>
- [3] Rast, <http://projects.netlab.jp/rast>
- [4] HyperEstraier, <http://hyperestraier.sourceforge.net/>
- [5] Senna <http://qwik.jp/senna/>
- [6] Oracle Text, <http://www.oracle.com/technology/global/jp/products/text/>
- [7] S.Sekiguchi et al. "Design Principles and IT Overviews of the GEO Grid", *IEEE Systems Journal*, Vol2. Issue3, pp374-389, 2008.09
- [8] OGC Catalog Service Web, <http://www.opengeospatial.org/standards/cat>
- [9] PostgreSQL GiST and GIN index types, <http://www.postgresql.org/docs/8.3/static/textsearch-indexes.html>
- [10] S.Agrawal et al, "DBxplorer: A System for Keyword-Based Search over Relational Databases, *ICDE2002*, pp5-16.
- [11] W.Cohen, "Integration of Heterogeneous Databases without Common Domains using Queries based on Textual Similarity", *SIGMOD98*, pp201-212
- [12] OGC Implements by Specification (Catalogue Service) <http://www.opengeospatial.org/resource/products/byspec>
- [13] A.Guttman, "R-trees: a dynamic index structure for spatial searching" *SIGMOD84*, pp47-57, 1984
- [14] GeoLucene, <http://sourceforge.net/projects/geolucene>
- [15] LocalLucene, <http://sourceforge.net/projects/locallucene>
- [16] W3C XForms Working Group, <http://www.w3.org/MarkUp/Forms/>
- [17] C.Dwork et al. "Rank aggregation methods for the Web" *10th international conference on WWW*, 2001, pp613-622
- [18] J.A.Asalam and M.Montague, "Models for Metasearch", *24th SIGIR*, pp276-284, 2001
- [19] WISE, <http://www.bsearchtech.com/wise/>
- [20] ISO19115/ISO19119 Application Profile for CSW2.0, OGC, 04-038r2, 2004
- [21] JMP2.0 仕様書 www.gsi.go.jp/common/000043123.pdf
- [22] CSW-ebRIM Registry Service, Part1, OGC-07-110r2, 2007.
- [23] JTS Topology Suite: <http://www.vividsolutions.com/jts/jtshome.htm>
- [24] OpenLayers: <http://openlayers.org/>