

XML による仮想電子辞書の設計と構築

近藤 幹大[†] 野村 智彦[†] 今井 健一[†] 都司 達夫[‡] 寶珍 輝尚[‡] 樋口 健[‡]

[†] 福井大学大学院工学研究科 〒910-8507 福井市文京 3 丁目 9-1

[‡] 福井大学工学部 〒910-8507 福井市文京 3 丁目 9-1

E-mail: [†] [‡] { kondou, nomura, imai, tsuji, hochin, higuchi }@pear.fuis.fukui-u.ac.jp

あらまし 複数の電子辞書の構成要素をユーザが自由に組み合わせてできる仮想辞書を定義し、検索できるシステムについて述べる。個々の電子辞書は、実体辞書と呼び、XML 化された実体を持つものに対して、仮想辞書は実体辞書上に定義されるユーザー定義ビューであると言え、データの実体を持たない。本論文における仮想辞書システムはユーザからの仮想辞書に対する操作要求を受付けて、仮想辞書サーバに送信する GUI インターフェース、仮想辞書に対する検索や操作の要求を実体辞書への要求に変換する仮想辞書サーバ、および実体辞書サーバからなる。

キーワード XML, 問い合わせ処理, 情報検索

The design and construction of a virtual electronic dictionary by XML

Mikihiro KONDO[†] Tomohiko NOMURA[†] Ken'ichi IMAI[†]

Tatsuo TSUJI[‡] Teruhisa HOCHIN[‡] and Ken HIGUCHI[‡]

[†] Graduate School of Engineering, Fukui University 3-9-1 Bunkyo, Fukui, 910-8507 Japan

[‡] Faculty of Engineering, Fukui University 3-9-1 Bunkyo, Fukui, 910-8507 Japan

E-mail: [†] [‡] { kondou, nomura, imai, tsuji, hochin, higuchi }@pear.fuis.fukui-u.ac.jp

Abstract This paper describes a virtual electronic dictionary system. A virtual electronic dictionary can be defined by combining elements of more than one real XML dictionaries. It can be regarded as a user-defined view on real dictionaries. The system consists of the following three subsystems. (1) GUI that receives the requests from users and sends them to the virtual dictionary server, (2) the virtual dictionary server that converts the requests from GUI into the corresponding requests for real dictionary servers, (3) real dictionary servers that search the real XML dictionaries and send back the retrieved results.

Keyword XML, Query Processing, Information Retrieval

1. はじめに

現在、電子辞書の出版は記憶メディアの大容量化と低廉化により、その多様性を増してきている。特に、画像や写真、音声を取り込んだ「マルチメディア辞書」の発展は著しい。また、インターネットを介したオンライン辞書の検索サービスも充実してきている。

本来辞書には極めて豊富で多角的な情報が内包されており、これらの情報は「辞書の文法」に従って構造的に記述されている。しかし人手で辞書を引くとき、その構造や配列が固定であるために辞書が潜在的に有する豊富な情報を十分利用することが出来ない。そこで、辞書の持つ情報を有効に利用するためには、辞書の固定構造による強制から解放して、使用者のニーズにかなう情報を提示できる手段が必要である。

個々の辞書データの実体を持っている辞書(英和・和英辞書など)を実体辞書としたとき、仮想辞書は実体辞書の各項目に対するユーザ定義のビューであり、デー

タの実体を持たない。ビューは単に、(複数の)実体辞書のどの部分をどのような配列で再配置するかを指定するもので、ユーザの定義により実体辞書の一部を参照したり、複数辞書を特定の項目を介して、結合させたりすることができる。その際、複数辞書の切り替えを意識させることなく、論理的に閉じた単一の辞書として操作できることが最大の特徴である。

本研究において従来構築してきた仮想辞書システムは、辞書の検索をその辞書固有の検索関数に頼っているため、新規に辞書が追加されると、検索関数を新たに作成する必要があり、このことに多大な労力が費やされた。そこで本研究では、この点を解決すべく、辞書データの構文解析により、辞書の各項目を切り出して作成された構造化辞書を用いて辞書データの XML[3]化を行った。XML は構造記述力に優れ、その構造を解析するための汎用パーザの作成が容易であるので、検索関数の統一化が図れると考えたためである。このように辞書データのフォーマットを決めて XML

化を行うことにより、辞書の種類に依存しない汎用的な検索機能を提供することが可能であり、既存のアプリケーションに大幅な変更を加える必要のない仮想辞書システムを構築できる。さらに、このような複数のXML辞書を実体辞書として、定義される仮想辞書群を操作するための仮想辞書システムを設計構築中である。

電子辞書を含めた電子化出版物の構造化には、たとえばEPWING[4]などの規格が存在しており、それにしただった、電子辞書も出回っている。また、[2]では辞書のハイパーテキスト化の方式が提案されている。しかし、既存の電子辞書は複数の辞書を切り換えて使用できるものの、本研究におけるように辞書の論理構造を前面に押し出し、それを操作させることによって、論理的に単一の辞書を提供するものは見当たらない。辞書の論理構造を意識させることにより、言葉の持つ深層構造を学習することができ、また、辞書の開発者や編集者の作業を支援することもできると考えられる。

[4]によればEPWINGの規格は辞書中に表れる語句(キーワード)の検索のための索引付けが中心であるが、ユーザレベルに対する辞書構造の提示機能はなく、本仮想辞書システムが提供しているような検索パス式の指定による検索はできない。

2. 仮想辞書

個々の辞書データの実体を持っている辞書(英和・和英辞書など)を実体辞書としたとき、仮想辞書は実体辞書の各項目に対するユーザ定義のビューであると考えられ、データの実体を持たない。ビューは単に、(複数の)実体辞書のどの部分をどのような配列で再配置するかを指定するもので、ユーザの定義により実体辞書の一部を参照したり、複数辞書を特定の項目を介して、結合させたりすることができる。その際、複数辞書の切り替えを意識させることなく、論理的に閉じた単一の辞書として操作できることが最大の特徴である。ここでは構造記述力に優れ、柔軟な構造解析が行えるXMLを用いて実体辞書群を構造化した。

2.1. 仮想辞書に対する要求

仮想辞書の詳細な説明に入る前に、仮想辞書に対する要求を示す。例えば、

「ある英単語の同義語や関連語などについて、その日本語表現や使い方を知りたい。」

場合、英語ソーラス辞書を使って同義語や関連語を調べる。さらに、英和辞書を使って、調べた同義語や関連語を見出し語として、英和辞書を引く。

このように、1つ目の辞書から入手した情報に基づ

いて、2つ目の辞書を引くことを使用者は強いられる。これは、前述した『辞書の切り替えを意識すること』にほかならない。そして、この動作をすべての該当項目について繰り返すことは、非常に面倒である。複数の電子辞書を活用するとき、このように、冊子体の辞書と同じような手順を必要とすることは問題であり、複数の電子辞書を扱えることのメリットが生かし切れていないと考えられる。

また、単一の辞書でも、例えば、英和辞書について特定の特定の品詞(例えば名詞)についてのみ参照したいときがしばしばある。この場合、他の項目の出力を回避して、認知負荷を軽減できれば、より使いやすくなる。

2.2. 仮想辞書の定義

仮想辞書の結合例として「和訳付き英語ソーラス辞書」を紹介する。この辞書は英語ソーラス辞書で引いた類義語についてその和訳を英和辞書で付記するものである(図1)。

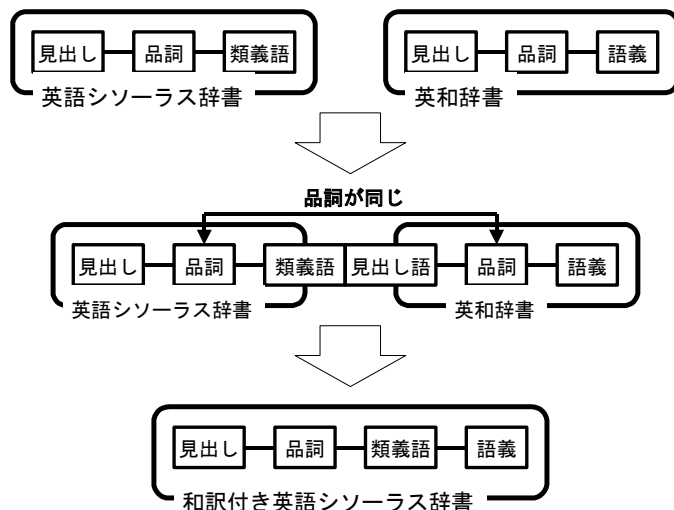


図1 和訳付き英語ソーラス辞書の定義

ここで、図1のような仮想辞書をユーザが理解する構文で記述する必要がある。仮想辞書の論理構造を定義するDEFINE句は、項目名と括弧の入れ子で表現される。この例では

和訳付き英語ソーラス辞書(見出し語,見出し語項目(品詞,品詞項目(類義語,類義語項目(語義))))

となる。母体辞書(この例では英語ソーラス辞書《ET》と英和辞書《EJ》)を指定するFROM句、仮想辞書の論理構造の各項目と該当する母体辞書の項目との対応付けを行うASSIGN句、この例では、

見出し語=ET.見出し語
 見出し語項目.品詞=ET.見出し語項目.品詞
 見出し語項目.品詞項目.類義語=ET.見出し語項目.品詞
 項目.語義項目.類義語
 見出し語項目.品詞項目.類義語項目.語義=EJ.見出し語
 項目.品詞項目.語義

となる。母体辞書の該当項目の結合情報(この例では英語シソーラス辞書の類義語と英和辞書の見出し語)を保持する *CENNECTED BY* 句は、

EJ.見出し語=ET.見出し語項目.品詞項目.語義項目.類義語

となる。そして、選択条件(この例では「品詞が同じ」)を表す *WHERE* 句

ET.見出し語項目.品詞=EJ.見出し語項目.品詞

が必要である。これらの句をまとめて *DEFINE* 文と呼ぶ。

2.3. 仮想辞書の検索

仮想辞書の検索には単に見出し語を指定するだけでなく、仮想辞書の論理構造を検索したい項目名と検索条件となる定数で記述する *FIND* 文を用いる。

和訳付き英語シソーラス辞書(見出し語["test"],
見出し語項目(品詞["名詞"],品詞項目()))

この例では和訳付き英語シソーラス辞書の見出し語が"test"、品詞が"名詞"であるものについて品詞項目以下をすべて検索することを示している。

3. 仮想辞書システム

現在、本研究において従来構築してきた仮想辞書システムは、辞書の検索をその辞書固有の検索関数に頼っているため、新規に辞書が追加されると、検索関数を新たに作る必要があり、このことに多大な労力が費やされた。そこで本研究では辞書データを XML 化して既存のアプリケーションに大幅な変更を加える必要のない仮想辞書システムの設計・構築を目指した。

3.1. 辞書データの XML 化

前述したように、辞書は「辞書の文法」に従った構造を持っている。この構造に従い、XML の文章構造定義(DTD)を記述して、辞書データの XML 化を行った。英語シソーラス辞書の構造を DTD にしたものを図 2 に示す。

この DTD を基にして既存電子辞書データを XML 化した。見出し語「test」に対して XML 化したものの一部分を以下に載せておく。

```
<ELEMENT thesaurus (見出し語,見出し語項目)*>
<ELEMENT 見出し語 (#PCDATA)>
<ELEMENT 見出し語項目 (品詞,品詞項目)*>
<ELEMENT 品詞 (#PCDATA)>
<ELEMENT 品詞項目 (語義,語義項目)*>
<ELEMENT 語義 (#PCDATA)>
<ELEMENT 語義項目 (用例*,類義語*,関連語*,成句*,反意語*,
  反対語*)>
<ELEMENT 用例 (#PCDATA)>
<ELEMENT 類義語 (#PCDATA)>
<ELEMENT 関連語 (#PCDATA)>
<ELEMENT 成句 (#PCDATA)>
<ELEMENT 反意語 (#PCDATA)>
<ELEMENT 反対語 (#PCDATA)>
```

図 2 英語シソーラス辞書の DTD

```
<見出し語>test</見出し語>
<見出し語項目>
<品詞>名詞</品詞>
<品詞項目>
  <語義 />
  <語義項目>
    <類義語>EXPERIMENT</類義語>
    <類義語>experimentation</類義語>
    ...
  </語義項目>
</品詞項目>
<品詞>動詞</品詞>
<品詞項目>
  ...
</品詞項目>
</見出し語項目>
```

図 3 において例えば、「見出し語」や「品詞」などは辞書のテキストとして現れる文字列であり、以下ではこれらの項目を具体項目という。また、「見出し語項目」や「品詞項目」はそれ以下の項目(の集合)に対して意味を与えるメタ項目であり、これを抽象項目と呼ぶ。ここで前提として実体辞書、仮想辞書に関わらず、抽象項目のインスタンスを一意に決定する具体項目のインスタンスが必ず存在するものとして、この具体項目を抽象項目に対するセレクトと呼ぶことにする。

辞書データの XML 化にあたっては、辞書データを XML 化した際の問題点として、記述が不統一でスキーマに沿った構造化ができない項目説明があったり、項目として切り出す時の終端子として決めたマークが必ずしも一定でなかったり、スキーマとしてこちらが想定していない項目が出現したりする例外が出現し、構造化できないものがあった。可能な限り手作業で対応したが、なお十分ではない。

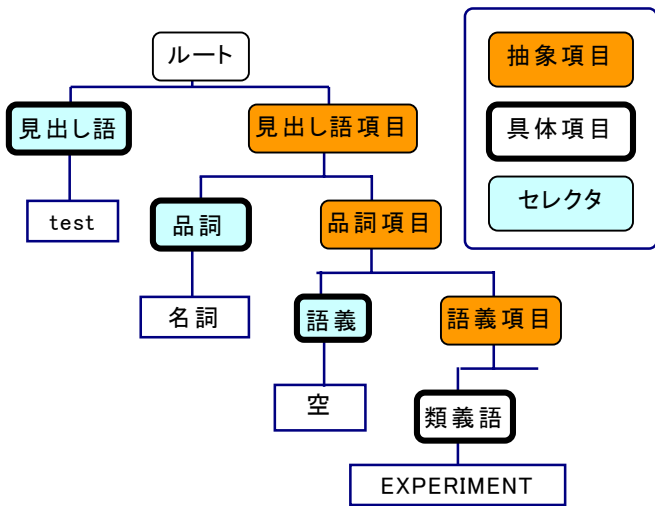


図3 抽象項目、具体項目とセレクトア

3.2. XML による仮想辞書システム

図4は本研究で設計した仮想辞書システムである。本システムは、ユーザインターフェース、仮想辞書サーバ、実体辞書サーバの3つから構成される。ユーザインターフェースからの要求に応じて仮想辞書サーバで処理を行い、必要に応じて実体辞書サーバに辞書データの問い合わせを行う。

仮想辞書サーバは、本仮想辞書システム中で唯一である。また、実体辞書サーバは、一般に複数のサイトに存在し、そのサイトに登録されている実体辞書群を操作する。

4. 仮想辞書サーバ

仮想辞書サーバは、ユーザの要求を受け取り、実体辞書サーバに対して辞書データの検索要求を出す。図4に示したように、以下の処理を行う。

- 実体辞書の登録・削除
- 仮想辞書の定義・削除
- 辞書検索

実体辞書の登録と削除は、実体辞書サーバに置かれている実体辞書の登録と登録の削除を行う。仮想辞書サーバが利用可能な実体辞書の情報を操作する。次に仮想辞書の定義と削除では、前述した『和訳付き英語シソーラス辞書』のような仮想辞書の定義の登録と抹消を行う。最後に辞書検索は、定義された仮想辞書を検索するものである。

4.1. 仮想辞書の定義

ここではユーザにより新規に定義された仮想辞書定義を仮想辞書サーバが保存する際の処理に関して述べる。前述したように仮想辞書は DEFINE 文によって定義される。DEFINE 文はユーザによってユーザインターフェースを介して生成される。生成された DEFINE 文は仮想辞書サーバに送られ格納される。

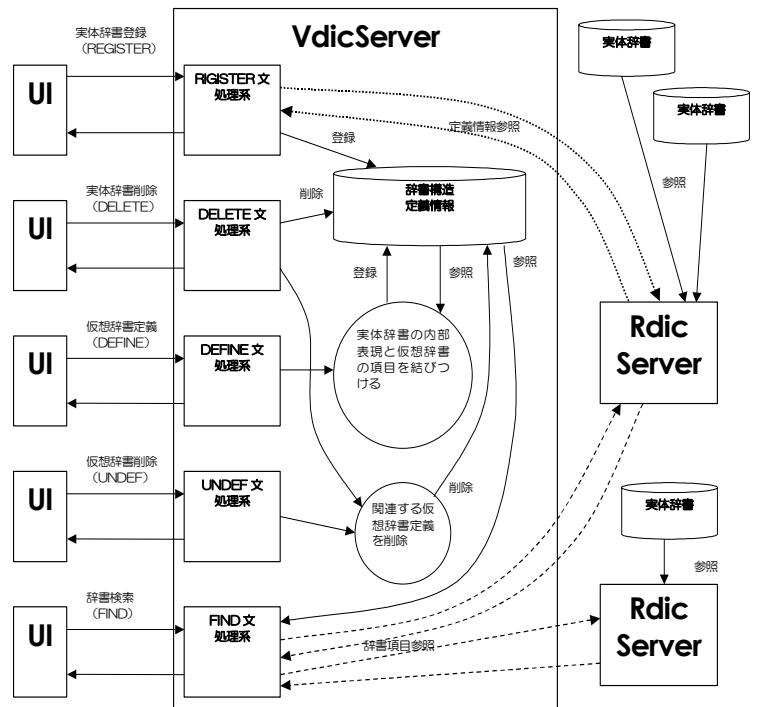


図4 仮想辞書システム

DEFINE 文における各句の各項目を取り出すには DOM の検索ライブラリを使用するため、DEFINE 文は XML 文に変換される。DOM とは XML 文書のタグとテキストを1つのノードとして構築されたオブジェクトのツリーを動的に操作するためのインターフェースのことである。前述した「和訳付き英語シソーラス辞書」の DEFINE 文を XML に変換したものを示す(図5)。仮想辞書の具体項目として参照される母体辞書のパス (ASSIGN 句の右辺) は名前空間を使用した XML で表現し、仮想辞書の具体項目の下位に展開している。FROM 句、CONNECTED BY 句、WHERE 句はルート要素(和訳付き英語シソーラス辞書)の直下に置く。

4.2. 仮想辞書の検索

母体辞書である実体辞書に対する検索文(FIND 文)の生成には、この XML による仮想辞書の DEFINE 文の DOM ツリーを使用して、逐次仮想辞書の項目定義を取り出すことによって行う。以下では FIND 文、

FIND 和訳付き英語シソーラス辞書(見出し語["test"], 見出し語項目(品詞["名詞"], 品詞項目()))

を使って処理の流れを追う。

まず、FIND 文の解析結果から仮想辞書名が得られる。仮想辞書サーバでは該当する辞書の DEFINE 文が読み込まれる。この場合は和訳付き英語シソーラス辞書である。このとき、この DEFINE 文が参照され DOM ツリーが作成される。さらに FROM 句が参照され、検索を行う母体辞書名が決定される。また、CONNECTED BY 句や WHERE 句も参照され、結合情報や結合条件が参照できるように保持する。

次に FIND 文の構造を順に辿っていくわけだが、

```

<和訳付き英語シソーラス辞書>
  <FROM xmlns:ET="./data/dtd/ET">
    英語シソーラス辞書
  </FROM>
  <FROM xmlns:EJ="./data/dtd/EJ">
    英和辞書
  </FROM>

  <CONNECTED BY>
    EJ.midashi_lv.part_lv.sense_lv.syn
  </CONNECTED BY >

  <WHERE>
    ET.midashi_lv.part=EJ.midashi_lv.part
  </WHERE>

  <見出し語>
    <ET:midashi/>
  </見出し語>
  <見出し語項目>
    <品詞>
      <ET:midashi_lv>
        <ET:part/>
      </ET:midashi_lv>
    </品詞>
    <品詞項目>
      <類義語>
        <ET:midashi_lv>
          <ET:part_lv>
            <ET:sense_lv>
              <ET:syn/>
            </ET:sense_lv>
          </ET:part_lv>
        </ET:midashi_lv>
      </類義語>
      <類義語項目>
        <語義>
          <EJ:midashi_lv>
            <EJ:part_lv>
              <EJ:sense/>
            </EJ:part_lv>
          </EJ:midashi_lv>
        </語義>
      </類義語項目>
    </品詞項目>
  </見出し語項目>
</和訳付き英語シソーラス辞書>

```

図5 仮想辞書の DEFINE 文

FIND 文の具体項目では母体辞書への問い合わせを行い、抽象項目ではその下位の構造を展開して処理を進める。また具体項目が定数により指定されている項目は、その値を保持する。検索結果は DOM ツリーとして仮想辞書の構造を持って作られていく。

FIND 文においてこの仮想辞書の見出し語が“test”と指定されている。これは同時に英語シソーラス辞書

の見出し語が“test”であることも示している。品詞に関しても同様である。これらから英語シソーラス辞書の見出し語が“test”で品詞が“名詞”であるものを検索することがわかる。次に FIND 文の品詞項目に処理が移る。ここで品詞項目は抽象項目であり、その下位に項目が指定されていないので、仮想辞書定義の品詞項目以下の全てが検索対象となる（すなわち類義語および類義語項目・語義）。検索項目となった類義語では、その下位の英語シソーラス辞書の syn へのパス部分が参照されて、実体辞書サーバへ問い合わせをするための FIND 文が生成される（図6）。

このようにして生成された母体辞書（実体辞書）

```

<find>
  <thesaurus>
    <midashi>test</midashi>
    <midashi_lv>
      <part>名詞</part>
      <part_lv>
        <sense>*</sense>
        <sense_lv>
          <syn target="true">*</syn>
        </sense_lv>
      </part_lv>
    </midashi_lv>
  </thesaurus>
</find>

```

図6 英語シソーラス辞書に対する FIND 文

thesaurus に対する FIND 文は thesaurus を管理する実体辞書サーバに送り出される。実体辞書サーバでは以下の類義語が検索される。

```

EXPERIMENT, experimentation, trial,
trial and error, trial run

```

続いて、仮想辞書サーバではこれらの類義語をセレクトタとして、抽象項目である類義語項目が参照され、語義に処理が及ぶ。ここで、語義は英和辞書の項目である。英和辞書の見出し語は CONNECTED BY 句により英語シソーラス辞書の類義語に結合されるように定義されているので、先に調べた英語シソーラス辞書の類義語を見出し語に指定して、母体辞書である英和辞書を引くように FIND 文が作成される。また、英和辞書の品詞では WHERE 句により品詞が同じものを選ぶように指定されているので、この例では名詞のものについての検索が行われる。よって、上記の5つの類義語ごとに作成される FIND 文の1つは図7のようになる。

最終的に例にあげた FIND 文に対する結果は図8のようになる。

```

<find>
  <eiwa>
    <midashi>trial</midashi>
    <midashi_lv>
      <part>名詞</part>
      <part_lv>
        <sense target="true"*</sense>
      </part_lv>
    </midashi_lv>
  </eiwa>
</find>

```

図7 英和辞書に対する FIND 文

```

<和訳付英語シソーラス辞書>
  <見出し語>test</見出し語>
  <見出し項目>
    <品詞>名詞</品詞>
    <品詞項目>
      <類義語>EXPERIMENT</類義語>
      <類義語項目>
        <語義/>
      </類義語項目>
      <類義語>experimentation</類義語>
      <類義語項目>
        <語義>実験すること, 実験作業.</語義>
      </類義語項目>
      <類義語>trial</類義語>
      <類義語項目>
        <語義>1 試み, 企て, (attempt). 2 《形容詞的に》 試験的な, 試すもの. 3 試練; 困難; 厄介なこと, 困りもの. 4 審判 《法律》 裁判, 審理; 公判.</語義>
      </類義語項目>
      <類義語>trial and error</類義語>
      <類義語項目>
        <語義/>
      </類義語項目>
      <類義語>trial run</類義語>
      <類義語項目>
        <語義/>
      </類義語項目>
    </品詞項目>
  </見出し項目>
</和訳付英語シソーラス辞書>

```

図8 検索結果

5. 実体辞書サーバ

実体辞書サーバは、仮想辞書サーバからの問い合わせに応じて処理を行う。仮想辞書サーバから発行される FIND 文を受け取り、所望のものを検索して、結果を返す。

実体辞書の検索には DOM を利用しているが、メモリの消費量が大きく 1 つの辞書をメモリ内に保持する

ことが出来ないので、見出し語によるインデックスをあらかじめ作成している。そして、検索対象となる見出し語の辞書項目のみを辞書から抽出して、DOM ツリーを作成し必要項目の取り出しを行っている。

図6は、前述したように英語シソーラス辞書から見出し語が“test”、品詞が“名詞”である類義語を抽出するために仮想辞書サーバで作られた実体辞書サーバへの問い合わせを行う FIND 文である。

実体辞書サーバへの FIND 文は対象辞書の DEFINE 句の XML 表現をテンプレートとして記述し、具体項目の値を検索条件として指定する。この値は任意の正規表現が使用可能である。また検索対象となる具体項目や抽象項目には target 属性を指定する。抽象項目に target 属性を指定したときには、その抽象項目のインスタンスが検索結果として返される。辞書の検索には検索の対象を辞書のスキーマに沿って記述するスキーマレベル検索と、辞書内のインスタンスによって検索対象を指定するインスタンスレベル検索がある。スキーマレベル検索では母体辞書の構造をテンプレートとしてそこに検索の条件を指定する。

図6の FIND 文を受け取った実体辞書サーバは、英語シソーラス辞書の見出し語“test”の項目を辞書データから抽出して、DOM ツリーに変換する。この DOM ツリーから検索を行うが、midashi_lv に対するセレクト項目 midashi、part_lv に対するセレクト項目 part により抽出すべき項目を絞っていき、前述したような 5 つの類義語を抽出し仮想辞書サーバに結果を返す(図9)。

```

<?xml version="1.0" encoding="Shift_JIS"?>
<results>
  <result dic="thesaurus"
    path="midashi[test]/part[NOUN]/sense[]/syn[*]">
    <syn>EXPERIMENT</syn>
    <syn>experimentation</syn>
    <syn>trial</syn>
    <syn>trial and error</syn>
    <syn>trial run</syn>
  </result>
</results>

```

図9 実体辞書サーバからの検索結果

図9を見てもわかるように、実体辞書サーバからの検索結果も XML 形式を取っている。これは仮想辞書サーバ側で、DOM などを利用して検索結果を扱うためである。result 要素に付加されている dic 属性、path 属性はそれぞれ検索対象の母体辞書名、その母体辞書内での位置を表しており、これらの情報により仮想辞書

サーバはこの検索結果がどの母体辞書のどの位置の情報なのかを判別できるようになっている。

6. GUI

以上述べてきた仮想辞書システムを Web 上で公開し利用するためには、辞書の構造を分かり易く操作できる GUI が必須となる。また、DEFINE 文にはルートノードからの多くのパス指定を含み入力に手間を要するので、良質の GUI を必要とする。

今回設計した GUI を用いてユーザは、実体辞書の登録と削除、仮想辞書の定義と削除、登録辞書のメタ情報の問い合わせ、辞書の検索などの操作を行うことができる。

ここでは本システムの GUI での操作について述べる。

6.1. 実体辞書の登録

実体辞書の登録には新規登録と仮想辞書の実体化が考えられる。

新規登録の場合、前提として XML 化された辞書が実体辞書サーバに置かれていることが必要である。

登録する辞書名とその辞書が置かれている URL を入力する。続いて、辞書の検索時にセクタの値をリストから選択するかテキストボックスに入力するかの指定を各具体項目に対して行う。この情報は仮想辞書の定義の際に母体辞書として選択されたときにも項目単位で引き継がれ、その仮想辞書の検索時に同じコントロールで入力することとなる。

GUI ではこの指定を、辞書の論理構造を表したテーブル中で、各具体項目に配置されたチェックボックスによって行う。この情報は実体辞書の登録情報と共に仮想辞書サーバへ送られる。

仮想辞書の実体化の場合は、仮想辞書のリストから実体化したい辞書を選択し、辞書を置く URL を指定するだけである。

6.2. 仮想辞書の定義

仮想辞書の定義では、辞書名、母体辞書、辞書の論理構造、仮想辞書の各具体項目についての母体辞書の項目との対応付け、結合項目、結合条件の 6 つを指定する。

最初に辞書名を入力した後、母体辞書を選択する。このとき、母体辞書は 2 つまで指定可能である。また、同一の辞書を選択することも可能である。

同一辞書を選択した場合は、仮想部分辞書を定義することになる。これは母体辞書からある具体項目だけを取り出し、その具体項目を見出し語として他の項目を引くという操作を備えた辞書である。

一つだけしか辞書を選択しない場合も仮想部分辞

書となる。これは母体辞書からある項目だけを取り出した辞書となる。例えば、和訳付き英語シソーラス辞書であれば、英語シソーラス辞書と英和辞書の二つの母体辞書を選択していることになる。

次に、辞書の論理構造の定義と、仮想辞書の各具体項目についての母体辞書の項目との対応付けの操作は同時に行う。辞書名とその横にラジオボタンの配置されたテーブル、および操作のリスト（項目の追加・削除、名前の変更、切り取りなど）が表示される。ラジオボタンで操作を反映する箇所を、リストから操作を選択する。項目の追加では新規ウインドウに選択済みの母体辞書一覧が表示される。対応付けたい辞書を選択すると辞書の論理構造が表示され、各項目に配置されたラジオボタンによって項目を対応付ける。以上を繰り返し仮想辞書の論理構造の定義と対応付けを実行する。母体辞書の選択時に 1 つだけ指定した場合は、仮想辞書の定義はここで終了である。

結合項目の指定では和訳付き英語シソーラス辞書の場合、英語シソーラス辞書で引いた類義語 (S) に英和辞書の見出し語 (K) が結合して日本語の語義を与える。(K,S) を結合項目と呼ぶ。(K) は見出し語で固定とする。GUI では最初にどちらの辞書を見出し語 (K) とするかを決定する。続いて表示されるもう一方の母体辞書の論理構造中の各ノードに配置されているラジオボタンにより (S) を決定する。

最後に結合条件の指定について、例えば英語シソーラス辞書で名詞として引いた類義語に対して英和辞書で動詞として引いては意味がなく、結合条件として品詞を一致させる必要がある。一般にこの結合条件は '=' による関係式の論理演算式であるが、入れ子状の複雑な論理演算式を入力することは面倒であるので、この論理演算式を簡単に入力するための GUI も用意する。

6.3. 辞書の検索

仮想辞書の検索指定は基本的にはセクタの値を条件付けしながら、対応する抽象項目中で検索項目を特定することによって行う。検索指定にはスキーマレベル検索指定とインスタンスレベル検索指定の二通りの方法がある。

スキーマレベル検索指定では、セクタの値に関わらず検索項目は同一である。例えば、ある見出し語の名詞の類義語、反対語を検索する場合などである。

これに対してインスタンスレベル検索指定は、セクタの値ごとに異なる検索項目が指定可能な検索方法である。例えば、ある見出し語について、動詞の類義語と、名詞の関連語を検索する場合がこれに該当する。

次に、和訳付き英語シソーラス辞書を例に具体的な操作を見ていく。まず、検索する仮想辞書を選択する

と、選択した仮想辞書の論理構造が表示される（図10）。この中に埋め込まれている各種コントロール（テキストボックス、チェックボックス、リストボックス etc）を使用することで、検索条件と検索項目を入力する。スキーマレベル検索、インスタンスレベル検索の切り替えはボタンで行う。

検索指定によって以降の入力方法が異なる。スキーマレベル検索を行う場合、品詞など値が少数で列挙できるセレクトタについては、リストボックスによる複数選択が可能である。また値の候補が多い、あるいはユーザが値についての認識を持ってないセレクトタの場合には、テキストボックスを使って値そのものまたは正規表現が入力可能である。この、リストボックスから指定するか、テキストボックスによる入力を行うかの設定は、実体辞書の登録時に行われたものを使用する。

検索条件を指定した後、セレクトタに対応する抽象項目の中で検索項目を指定する。インスタンスレベル検索をする場合、例えば名詞に対しては類義語を検索項目とし、動詞に対しては類義語と語義を検索項目としたいとする。このときはセレクトタ品詞の値に名詞を選択し、類義語にチェックを入れる。次に品詞の値に動詞を選択し、類義語と語義にチェックを入れるという操作を行う。途中で入力を確認したい場合や変更したい場合は、リストボックスで値を選択する。

このようにして検索条件、検索項目を指定し検索を開始すると、GUIによって FIND 文が生成され、仮想辞書サーバに送信される。仮想辞書サーバはこれを各母体辞書に対する FIND 文へ変換して、各実体辞書サーバへと送信する。実体辞書サーバは受信した FIND 文にもとづき、保有する実体辞書を検索し検索結果を仮想辞書サーバへ返す。仮想辞書サーバは実体辞書サーバからの検索結果を仮想辞書に対する検索結果に統合し、GUIに返す。

このとき、検索項目に対する検索結果を自分の好みの配置やフォント、背景色で出力したいという要求がある。このような出力編成 GUI を必要としているが、現在未設計である。



図 1 0 仮想辞書検索設定画面

7. おわりに

本論では、仮想辞書システムの設計について述べた。この設計に基づき現在構築中である。

初めに記したように、電子辞書の出版はその多様性を増してきている。しかし、本システムのように辞書の構造を前面に押し出し、それを操作することにより、きめの細かい検索を可能としている電子辞書は見当たらない。本仮想辞書システムにより、構造に沿った多彩な検索が可能になる。特に、本システムにより、言葉の持つ深層構造を理解して、学習することができ、また、辞書の開発者や編集者の作業を支援することもできると考えられる。本システムは複数の辞書を母体とする仮想辞書の操作を主眼としているが、単一の実体辞書を対象としても十分に利用効果があると考えられる。

3.1 ではスキーマに準拠せず、構造化の例外となる辞書データによる問題点が XML 化するときの問題点となったことを述べた。これらの問題点のいくつかは、言葉が持つ多様性を十分に説明するためには、ある程度不可避であると思われる。しかし、電子辞書の規格化がこのあたりも含めて十分に検討され、標準化されたならば、逆に辞書の編集段階から、規格準拠を考慮でき、統一のとれた電子辞書を構築できる。本研究における仮想電子辞書の場合には、このような標準化された実体辞書が母体辞書として使用できるような環境が望まれる。

現在は DOM を主に使っているが、今後の XML の発展次第では様々な XML データを扱う新たな技術を組み込むことが可能である。

謝辞 和英辞書および英和辞書の元データは三省堂（株）より提供していただいた。謝意を表す。

文 献

- [1] 吉田 将：辞書構築における諸問題、情報処理、27, 8, pp. 933-939.
- [2] 内藤、山下、松山、柵木：オンライン辞書のハイパーテキスト化手法、情報処理学会論文誌、34, 2, pp. 320-329, 1993.
- [3] W3C(XML), <http://www.w3c.org/>
- [4] EPWING コンソシアム, <http://www.epwing.or.jp/>
- [5] 今井、近藤、都司、宝珍、樋口：ユーザ定義可能な仮想電子辞書システム、情報処理学会研究報告「人文科学とコンピュータ」No54, 2002.